

В. В. Голенков, Н. А. Гулякина, Д. В. Шункевич

**Открытая технология
онтологического проектирования,
производства и эксплуатации
семантически совместимых гибридных
интеллектуальных компьютерных систем**

Минск
«Беспринт»
2021

Голенков, В. В. Открытая технология онтологического проектирования, производства и эксплуатации семантически совместимых гибридных интеллектуальных компьютерных систем / В. В. Голенков, Н. А. Гулякина, Д. В. Шункевич. – Минск : Бестпринт, 2021. – 690 с. – ISBN 978-985-7267-13-2.

В книге представлена документация текущей версии стандарта открытой технологии онтологического проектирования, производства и эксплуатации семантически совместимых гибридных интеллектуальных компьютерных систем (Технологии OSTIS). Описание Технологии OSTIS (Open Semantic Technology for Intelligent Systems) представлено в виде раздела базы знаний ostis-системы (системы, построенной по Технологии OSTIS) на внутреннем языке ostis-системы и обладает достаточной полнотой для использования этой технологии разработчиками интеллектуальных компьютерных систем. Предложена стандартизация интеллектуальных компьютерных систем, а также стандартизация методов и средств их проектирования, что является важнейшим фактором, обеспечивающим *семантическую совместимость* интеллектуальных компьютерных систем и их компонентов.

Книга предназначена для широкого круга читателей, интересующихся проблемами искусственного интеллекта, специалистами в области интеллектуальных систем и инженерии знаний. Также может быть использована студентами, магистрантами и аспирантами специальности «Искусственный интеллект».

Табл. 4. Ил. 444. Библиогр.: 82 назв.

Рекомендовано Советом БГУИР, протокол № 8 от 26.03.2021 г.

Рецензенты:

заведующий кафедрой технологий программирования факультета прикладной математики и информатики БГУ доктор техн. наук, проф. *А. Н. Курбацкий*;

заведующий лабораторией идентификации систем государственного научного учреждения «Объединенный институт проблем информатики НАН Беларуси» доктор техн. наук, проф. *А. А. Дудкин*

Предисловие

В настоящее время уровень требований, предъявляемых к комплексу технологий искусственного интеллекта существенно повысился – возникла необходимость разработки компьютерных систем, которые не только обладают высоким уровнем интеллекта, но и обладают семантической совместимостью, взаимопониманием, способностью координировать свою деятельность с другими системами при коллективном решении сложных "внештатных" задач. Очевидно, что эти требования предполагают существенное развитие стандартов интеллектуальных компьютерных систем. Важнейшая особенность стандарта интеллектуальной компьютерной системы, обеспечивающего их семантическую совместимость и взаимопонимание, заключается в том, что для этого интеллектуальные компьютерные системы должны использовать:

- один и тот же язык внутреннего представления знаний;
- один и тот же язык их коммуникации;
- одну и ту же систему понятий;
- обладать достаточно большим количеством общих (одинаковых) знаний.

Следовательно, разработка стандарта интеллектуальной компьютерной системы в той части этого стандарта, которая связана с выделением и формализацией общих (одинаковых) знаний, хранимых в памяти интеллектуальной компьютерной системы и необходимых для обеспечения их взаимопонимания, фактически осуществляет разработку достаточно большой одинаковой части всех интеллектуальных компьютерных систем (и не только прикладных), что существенно сокращает сроки их разработки.

К этому можно добавить возможность и целесообразность одинаковой для всех интеллектуальных компьютерных систем реализации целого ряда их способностей:

- способности понимать информацию, которой они обмениваются между собой,
- способности договариваться и координировать свои действия при коллективном решении сложных интеллектуальных задач в условиях нештатных (аномальных, нестандартных, непредусмотренных) ситуаций,
- способности принимать решения на основе их глубокого обоснования,
- способности обучаться и многие другие способности, обеспечивающие необходимый уровень интеллекта разрабатываемых интеллектуальных компьютерных систем.

Данная монография является первым этапом на пути комплексного решения указанных выше проблем. Предназначена она одновременно:

- для студентов, магистрантов и аспирантов, обучающихся по специальности "Искусственный интеллект";
- для разработчиков прикладных интеллектуальных компьютерных систем;
- для разработчиков технологий проектирования и производства интеллектуальных компьютерных систем;
- для научных работников, создающих новые модели и методы решения интеллектуальных задач.

Данная монография сочетает:

- строгую формализацию представляемой информации и её доступность (возможность первичного её понимания без предварительного изучения используемого формального языка);
- традиционную ("пассивную") форму представления материала (в "электронном" и "бумажном" виде) с "активной" формой в виде интеллектуальной справочной системы, когда компьютерная система не только обеспечивает оперативное редактирование информации, но и помогает пользователям быстрее и качественнее усваивать эту информацию (за счет возможности отвечать на широкий спектр вопросов и учитывать индивидуальные особенности, потребности и интересы пользователей).

Область искусственного интеллекта сочетает в себе как научно-исследовательский аспект и создание технологий разработки интеллектуальных компьютерных систем, а так и непосредственно разработку самих интеллектуальных компьютерных систем. Эта область развивается настолько быстрыми темпами, что за время обучения студентов и магистрантов ситуация в области искусственного интеллекта меняется существенно, поэтому подготовка специалистов в этой области требует особого подхода, учитывающего высокий уровень сложности этой научно-технической области, а также быстрые темпы развития теории интеллектуальных компьютерных систем, технологий их проектирования, непосредственно практика разработки конкретных прикладных интеллектуальных компьютерных систем.

Если специалист в области искусственного интеллекта не будет постоянно ориентироваться в тенденциях развития каждого из этих направлений развития работ в этой области, то он быстро перестанет быть конкурентноспособным. Это значит, что специалист в области искусственного интеллекта должен быть в достаточной степени и ученым, и создателем технологий следующего поколения, и разработчиком конкретных приложений.

Таким образом, подготовку специалистов в области искусственного интеллекта необходимо ориентировать не на конкретное состояние науки, технологии и практики в этой области, а на перманентный процесс эволюции всех этих направлений.

Сформировать у студентов и магистрантов реальные навыки в области искусственного интеллекта можно только путем поэтапного и непосредственного их включения в реальную деятельность в этой области (и в научно-исследовательскую деятельность, и в развитие технологий искусственного интеллекта, и в разработку прикладных интеллектуальных компьютерных систем на основе текущего состояния соответствующих технологий). Но для этого необходимо создать соответствующую научно-исследовательскую и инженерную инфраструктуру.

Научно-исследовательская деятельность в области искусственного интеллекта предполагает исследование феномена интеллекта и создание принципиально новых подходов (моделей и методов) к решению интеллектуальных задач и к разработке принципов организации соответствующих компьютерных систем.

Развитие технологий искусственного интеллекта включает в себя:

- разработку стандарта интеллектуальных компьютерных систем, соответствующего текущему состоянию технологий искусственного интеллекта;
- разработку методов, средств проектирования и реализации интеллектуальных компьютерных систем.

Разработка прикладных интеллектуальных компьютерных систем предполагает грамотное применение соответствующих технологий.

Методику обучения необходимо ориентировать не только на формирование навыков разработки прикладных интеллектуальных компьютерных систем по заданной технологии, но и на формирование навыков перманентного совершенствования и непосредственно самих прикладных интеллектуальных компьютерных систем, и технологий их разработки, и принципов (моделей и методов) решения интеллектуальных задач и организации интеллектуальных систем.

Данная монография рассматривается нами как первый выпуск целой серии коллективных монографий, которые будут представлять последующие версии *Стандарта Технологии OSTIS* (Open Semantic Technology for Intelligent Systems – Стандарта технологии, ориентированной на разработку семантически совместимых интеллектуальных компьютерных систем). При этом предполагается существенное расширение авторского коллектива и организация всей работы на развитие Стандарта Технологии OSTIS как открытого проекта, целью которого является коллективное совершенствование базы знаний, посвященной детальному описанию этого стандарта.

При этом при подготовке даже данного издания текущей версии Стандарта Технологии OSTIS мы приобрели хороший опыт организации коллективной деятельности такого рода, привлекая к этой работе целый ряд аспирантов, магистрантов и студентов, а также сотрудников других организаций.

Вклад некоторых из них в ряд разделов монографии позволил включить их в число соавторов этих разделов, что отражено непосредственно в тексте монографии.

Авторы выражают благодарность:

- Сотрудникам кафедры Интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники и кафедры Интеллектуальных информационных технологий Брестского государственного технического университета, а также сотрудникам ОАО «Савушкин продукт»;
- студентам кафедры Интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники Банцевич К.А., Бутрину С.В., Василевской А.П., Меньковой Е.А., Жмырко А.В., Григорьевой И.В., Загорскому А.Г., Марковцу В.С., Киневичу Т.О. за оказание технической помощи при подготовке текста к печати;
- ООО «Интелиджент семантик системс» и его генеральному директору Т. Грюневальду за финансовую поддержку работ по развитию *Технологии OSTIS*, а также финансовую поддержку издания *Стандарта OSTIS*;
- Рецензентам – д-ру техн. наук, профессору Александру Николаевичу Курбацкому и д-ру техн. наук, профессору Александру Арсентьевичу Дудкину;
- Коллегам из Советской (ныне Российской) Ассоциации Искусственного интеллекта и коллегам из Белорусского объединения специалистов в области искусственного интеллекта;
- Членам Программного Комитета ежегодных конференций OSTIS, а также всем участникам этих конференций за плодотворное и конструктивное обсуждение направлений развития семантических технологий и технологии OSTIS в частности.


```

/*****/
Стандарт OSTIS
⊃=
{
:= [Документация Технологии OSTIS]
:= [Документация Открытой технологии онтологического проектирования, производства и эксплуатации семантически совместимых гибридных интеллектуальных компьютерных систем]
:= [Описание Технологии OSTIS (Open Semantic Technology for Intelligent Systems), представленное в виде раздела базы знаний ostis-системы (системы, построенной по Технологии OSTIS) на внутреннем языке ostis-систем и обладающее достаточной полнотой для использования этой технологии разработчиками интеллектуальных компьютерных систем]
:= [Полное описание текущего состояния Технологии OSTIS, представленное в виде семейства разделов базы знаний, построенной по Технологии OSTIS]
:= [Семейство разделов базы знаний Метасистемы IMS.ostis, которое предназначено для комплексной поддержки онтологического проектирования семантически совместимых гибридных интеллектуальных компьютерных систем]
∈ семейство разделов базы знаний
:= [семейство разделов внутреннего представления базы знаний ostis-системы – интеллектуальной компьютерной системы, построенной по Технологии OSTIS]
⇒ финансовая поддержка*:
Intelligent Semantic Systems Ltd.
= {
• Анализ методологических проблем современного состояния работ в области Искусственного интеллекта
• Предметная область и онтология информационных конструкций и языков
• Описание общих принципов оформления внутреннего и внешнего представления информационных конструкций в ostis-системах
• Предметная область и онтология кибернетических систем
• Предметная область и онтология логико-семантических моделей компьютерных систем, основанных на смысловом представлении информации
• Предметная область и онтология знаний и баз знаний ostis-систем
• Предметная область и онтология решателей задач ostis-систем
• Предметная область и онтология интерфейсов ostis-систем
• Предметная область и онтология базовых интерпретаторов логико-семантических моделей ostis-систем
• Предметная область и онтология Экосистемы OSTIS
• Библиографический раздел Стандарта OSTIS
}
⇒ эпиграф*:
[From data science to knowledge science]
⇒ аннотация*:
[В настоящее время информатика преодолевает важнейший этап своего развития – переход от информатики данных (data science) к информатике знаний (knowledge science), где акцентируется внимание на семантических аспектах представления и обработки знаний.
Без фундаментального анализа такого перехода невозможно решить многие проблемы, связанные с управлением знаниями, экономикой знаний, с семантической совместимостью интеллектуальных компьютерных систем.
Основной особенностью Технологии OSTIS является ориентация на использование компьютеров нового поколения, специально предназначенных для реализации семантически совместимых гибридных интеллектуальных компьютерных систем. Предлагаемый Стандарт OSTIS оформлен в виде семейства разделов базы знаний специальной интеллектуальной компьютерной Метасистемы IMS.ostis (Intelligent MetaSystem for ostis-systems), которая построена по Технологии OSTIS и представляет собой постоянно совершенствуемый интеллектуальный портал научно-технических знаний, который поддерживает перманентную эволюцию Стандарта OSTIS, а также разработку различных ostis-систем (интеллектуальных компьютерных систем, построенных по Технологии OSTIS).]
:= [Процесс перманентной эволюции Стандарта OSTIS, совмещенного (интегрированного) с комплексными учебно-методическим обеспечением подготовки специалистов в области Искусственного интеллекта и представленного в виде специального раздела базы знаний]

```

- ⇒ *примечание**:
 [Подчеркнем, что *Стандарт OSTIS* – это не описание некоторого состояния *Технологии OSTIS*, а динамическая информационная модель процесса эволюции этой *технологии*]
- := [Стандарт Технологии OSTIS]
- := [Документация *Технологии OSTIS*, полностью отражающая текущее состояние *Технологии OSTIS* и представленная соответствующим *разделом базы знаний* специальной *ostis-системы*, которая ориентирована на поддержку проектирования, производства, эксплуатации и эволюции (реинжиниринга) *ostis-систем*, а также на поддержку эволюции самой *Технологии OSTIS* и которая названа нами *Метасистемой IMS.ostis*]
- := [Максимальный раздел *Стандарта OSTIS*, т.е. раздел, в состав которого входят все остальные *разделы* (подразделы) *Стандарта OSTIS*]
- := [Раздел *базы знаний*, текущее состояние которого отражает текущее состояние (текущую версию) перманентно эволюционируемого Стандарта Комплексной *Технологии OSTIS*]
- := [Представленное в форме раздела *базы знаний* специальной *ostis-системы* (*Метасистемы IMS.ostis*) полное описание (спецификация, документация) текущего состояния *Технологии OSTIS*]
- := [Мы рассматриваем *Стандарт OSTIS* (Документацию Стандарта *Технологии OSTIS*) как продукт *научно-технической деятельности*, к которому предъявляются высокие требования по полноте, согласованности, непротиворечивости, практической значимости разрабатываемой документации, описывающей текущее состояние *Технологии OSTIS*]

официальная версия Стандарта OSTIS

- := [официально издаваемая (публикуемая в бумажном и/или электронном виде) *версия Стандарта OSTIS*]
- ⊃ *Стандарт OSTIS-2021*
- := [Официальная *версия Стандарта OSTIS*, издаваемая непосредственно перед началом *Конференции OSTIS-2021*]

Стандарт OSTIS

- ← *формальная спецификация**:
Технология OSTIS
- := [Перманентно развиваемый в рамках открытого проекта комплекс моделей, методов и средств, ориентированных на онтологическое проектирование, производство, эксплуатацию и реинжиниринг семантически совместимых гибридных интеллектуальных компьютерных систем, способных самостоятельно взаимодействовать друг с другом]
- := [Технология разработки семантически совместимых и самостоятельно взаимодействующих интеллектуальных компьютерных систем]
- ⇒ *пояснение**:
 [Технология *OSTIS* – это технология принципиально нового уровня, это обусловлено:
- высоким качеством интеллектуальных компьютерных систем (*ostis-систем*), разрабатываемых на ее основе – их семантической совместимостью, способностью к самостоятельному взаимодействию, способностью адаптироваться к пользователям и способностью адаптировать (обучать) самих пользователей более эффективному взаимодействию с интеллектуальными компьютерными системами;
 - высоким качеством самой *технологии* – возможностью интегрировать самые различные *виды знаний* и самые различные *модели решения задач*, неразрывной связью процесса разработки интеллектуальных компьютерных систем и процесса повышения квалификации разработчиков.
-]

формальная спецификация*

- := [Бинарное ориентированное *отношение*, каждая *пара* которого связывает
- некий формальный текст, являющийся формальной спецификацией (формальной моделью, формальным описанием) с
 - некой сущностью, которая является объектом указанной спецификации (моделирования, описания)
-]
- ⊃ *sc-модель**
- := [быть формальной спецификацией (формальной моделью, формальным описанием) заданного объекта, представленной на внутреннем смысловом языке интеллектуальных компьютерных систем (в SC-коде)]

ostis-система

:= [интеллектуальная компьютерная система, построенная по Технологии OSTIS (по стандартам Технологии OSTIS), что обеспечивает:

- семантическую совместимость (взаимопонимание) всех этих систем между собой;
- их способность к самостоятельному взаимодействию, к координации своей деятельности при возникновении заранее непредсказуемых (нештатных) ситуаций.

]

⇒ следовательно*:

[Если интеллектуальные компьютерные системы не будут обладать указанными выше способностями, то ни о каких smart-предприятиях, smart-учреждениях, smart-городах, ни о каком smart-обществе и речи быть не может, т.к. все обстоятельства их деятельности заранее на этапе проектирования предусмотреть принципиально невозможно. Это означает, что интеллектуальные компьютерные системы должны научиться самостоятельно "отрабатывать" все заранее непредусмотренные обстоятельства.]

Стандарт OSTIS

⇒ основные цели*:

- [Дать старт открытому проекту перманентной эволюции Стандарта OSTIS]
- [Обеспечить подготовку молодых специалистов, квалификации которых достаточно для реализации быстрых темпов эволюции Стандарта OSTIS путем:
 - интеграции (совмещения) Стандарта OSTIS с комплексным учебно-методическим обеспечением подготовки и повышении квалификации соответствующих специалистов;
 - активного подключения студентов, магистрантов и аспирантов к эволюции Стандарта OSTIS в рамках учебного процесса.
-]
- [Существенно повысить качество и темпы эволюции Стандарта OSTIS путем создания работоспособного творческого коллектива авторов Стандарта OSTIS]
- [Существенно расширить число членов Авторского Коллектива Стандарта OSTIS – специалистов, непосредственно участвующих в реализации и эволюции Технологии OSTIS в рамках соответствующего общедоступного открытого проекта]

⇒ примечание*:

[Итак, наша цель – сформировать открытый творческий(!) коллектив, способный развивать Стандарт OSTIS. К сожалению, соответствующих прецедентов, мягко говоря, немного. Коллективно(!) творить человечество пока не научилось. Очень много откровенной имитации и неадекватности. Но научиться этому надо, ибо другого выхода из современного кризиса Искусственного интеллекта нет. Что касается инструментальной поддержки коллективного процесса, то это, не являясь сутью этого процесса, обеспечивает существенное повышение его темпов, что весьма важно. Особо подчеркнем то, что непосредственно коллективный творческий процесс эволюции Стандарта OSTIS, как и любой другой коллективный творческий процесс, принципиально(!) не может быть полностью автоматизирован. Автоматизировать согласование различных позиций (точек зрения) невозможно(!). Основной проблемой здесь является иллюзия человеческого взаимопонимания. Можно только фиксировать факт противоречивости точек зрения, фиксировать аргументы, фиксировать этапы процесса сближения этих точек зрения. Но не более – сам процесс сближения это "ручной" процесс, реализуемый исключительно авторами.]

⇒ направления эволюции*:

- [Конструктивное пополнение Стандарта OSTIS на уровне исходного текста базы знаний *Метасистемы IMS.ostis* (продумать при этом и технически поддерживать методики согласования авторских позиций
 - При четкой фиксации авторских прав и разных точек зрения
 - Через регулярные коллективные обсуждения - семинары.
-]
- [Совершенствование формы и стиля представления материала]
- [Совершенствование систематизации (структуризации, стратификации) материала, четкая фиксация наследования свойств]
- [Четкий формальный сравнительный анализ со всеми близкими подходами (Semantic Web, графовые базы данных и др.)]
- ["Наращивание" Стандарта OSTIS соответствующей учебно-методической информацией]

- [Интеграция в состав *Стандарта OSTIS всех(!)* развиваемых в настоящее время в области *Искусственного интеллекта* моделей представления знаний, моделей решения задач, моделей интерфейсов при обеспечении их *семантической совместимости*. Весь современный арсенал результатов работ в области *Искусственного интеллекта* в перспективе должен быть интегрирован в *Стандарт OSTIS*, но не как мозаика ("зоопарк") имеющихся трудносовместимых моделей, методов и средств построения *интеллектуальных компьютерных систем*, имеющихся сервисов и информационных ресурсов]
- [Загрузка исходного текста *Стандарта OSTIS* в состав *Базы знаний IMS.ostis* и организация дальнейшей работы по развитию *Стандарта OSTIS* как коллективной разработки *Базы знаний IMS.ostis*]
- [Все вопросы по подготовке специалистов в области *Искусственного интеллекта*, а также всю научно-техническую деятельность магистрантов и аспирантов организовать через *Базу знаний IMS.ostis*]
- [Существенно расширить библиотеки многократно используемых компонентов *ostis-систем*]

⇒ *примечание**:

[Фактически речь идет о поэтапном преобразовании современного представления всевозможных информационных ресурсов, различного вида стандартов, энциклопедий, википедий, толковых словарей, учебных пособий и др. в вид строгих формальных моделей, семантически совместимых друг с другом и полностью понятных, готовых к использованию интеллектуальными компьютерными системами. Одно дело – систематизировать некоторый материал, а другое дело – его постоянно совершенствовать и повышать эффективность его использования. Практическая ценность неразвиваемых текстов весьма низка из-за быстрого их морального старения.]

Стандарт OSTIS

⇒ *общие принципы организации эволюционных работ**:

- [Формирование работоспособного *Авторского коллектива Стандарта OSTIS*]
- [Формирование *Редакционной коллегии Стандарта OSTIS* для контроля целостности и качества *Стандарта OSTIS* в каждый момент времени]
- [Формирование *Консорциума OSTIS* для международного продвижения *Стандарта OSTIS*, для взаимодействия с международными структурами, занимающимися стандартизацией *интеллектуальных компьютерных систем и технологий Искусственного интеллекта*]
- [Повышение качества *подготовки специалистов в области Искусственного интеллекта* в вузах РФ (БГУИР, БрГТУ, БГУ, БНТУ, ГрГТУ, ПГУ) путём:
 - тесного сотрудничества и унификации *подготовки специалистов в области Искусственного интеллекта* в разных вузах;
 - интеграции учебно-методических материалов в состав *Стандарта OSTIS*;
 - непосредственного подключения студентов, магистрантов и аспирантов к реальному процессу эволюции *Стандарта OSTIS*, т.е. путём непосредственного включения студентов, магистрантов и аспирантов в состав *Авторского коллектива Стандарта OSTIS* со всеми вытекающими отсюда возможностями, правами и обязанностями.
- [Перед началом каждой (ежегодной) *конференции OSTIS* осуществлять издание очередной *официальной версии Стандарта OSTIS*, в которой отражаются основные изменения и дополнения *Стандарта OSTIS*, внесенные в *Стандарт OSTIS* за истёкший год после проведения предыдущей *конференции OSTIS*. При этом речь идет не только о содержательных (семантических) изменениях, но и об изменениях структуризации материала, изменениях в правилах и стиле оформления материала.]
- [Существенно повысить уровень конструктивности и полезности каждой *конференции OSTIS* для ускорения темпов *эволюции стандарта OSTIS*. Каждая *конференция OSTIS* должна быть посвящена:
 - подведению итогов *эволюции Стандарта OSTIS* за истёкший год;
 - анализу текущего состояния *Стандарта OSTIS*;
 - уточнению наиболее актуальных направлений *эволюции Стандарта OSTIS* (в первую очередь – на следующий год).

]

)

- ⇒ *примечание**:
 [Таким образом, *конференции OSTIS* должны стать ежегодной площадкой для согласования и координации деятельности в направлении *эволюции Стандарта OSTIS*, а также в направлении *подготовки специалистов в области Искусственного интеллекта*.
 Координация деятельности необходима
- не только между различными кафедрами различных *вузов*, осуществляющими *подготовку специалистов в области Искусственного интеллекта*,
 - но и между различными членами и группами *Авторского коллектива Стандарта OSTIS*,
 - а также между *Авторским коллективом Стандарта OSTIS*, *Редакционной коллегией Стандарта OSTIS* и *Консорциумом OSTIS*.
-]

⇒ *план издания официальных версий**:
 - { • *Стандарт OSTIS-2021*
 - := [Официальная версия *Стандарта OSTIS*, публикуемая (издаваемая) до начала проведения конференции OSTIS-2021 (16-18 сентября 2021 года)]
 - € *текст, построенный на основе русскоязычной терминологии*
 - ⇒ *примечание**:
 [При этом возможны некоторые англоязычные заимствования – SC-код, sc-текст и др.]
 - *Стандарта OSTIS-2022*
 - := [Официальная версия *Стандарта OSTIS*, публикуемая до Конференции OSTIS-2022 (апрель 2022 года)]
 - € *текст, построенный на основе русскоязычной терминологии*
 - *Стандарта OSTIS-2023*
 - := [Специальное англоязычное официальное издание версии Стандарта OSTIS, публикуемое до *Конференции OSTIS-2023* (апрель 2023 года) и ориентированное на широкий круг международной научно-технической общественности]
 - € *текст, построенный на основе англоязычной терминологии*
 - ⇒ *примечание**:
 [Данное англоязычное издание *Стандарта OSTIS* рассматривается нами как повод, визитная карточка, приглашение к переговорам с зарубежными коллегами и организациями, занимающимися стандартизацией *интеллектуальных компьютерных систем и технологий*]
 - ⇒ *примечание**:
 [В перспективе по мере возникновения необходимости мы будем переиздавать (в расширенном и дополнительном варианте) на английском языке некоторые версии *Стандарта OSTIS* для активизации различного рода международных переговоров.]
 - ⇒ *примечание**:
 [При ежегодном переиздании версий *Стандарта OSTIS* для "внутреннего пользования" – для работы *Редакционной коллегии Стандарта OSTIS* и *Авторского коллектива Стандарта OSTIS* мы будем ориентироваться на интеграцию использования как русскоязычной, так и англоязычной терминологии, что фактически означает включение в состав *Стандарта OSTIS* русско-английского и англо-русского словарей]
 - *Стандарт OSTIS-2023*
 - := [Официальная версия *Стандарта OSTIS*, публикуемая до *конференции OSTIS-2023* (апрель 2023 года) и использующая в равной степени как русскоязычную, так и англоязычную терминологию]
 - € *текст, построенный на основе интеграции русскоязычной и англоязычной терминологии*
 - *Стандарт OSTIS-2024*
 - ...

}

⇒ *авторский коллектив**:
Авторский коллектив Стандарта OSTIS
 - ⇒ *примечание**:
 [Работоспособность, квалификация и результативность *Авторского коллектива Стандарта OSTIS* определяют темпы и качество эволюции *Стандарта OSTIS*]
 - ⇒ *примечание**:
 [При подготовке к изданию каждой официально фиксируемой версии *Стандарта OSTIS*, как правило, будут формироваться специальные группы из общего *Авторского коллектива Стандарта OSTIS*, каждая из которых ориентируется на подготовку к изданию соответствующей версии *Стандарта OSTIS*]

автор Стандарта OSTIS

:= [соавтор Стандарта OSTIS]

:= [член Авторского коллектива Стандарта OSTIS]

⇒ *примечание**:

[соавтором *Стандарта OSTIS* может быть любой желающий, способный согласовывать свою персональную инициативную деятельность и свою точку зрения с другими соавторами *Стандарта OSTIS*]

Стандарт OSTIS

⇒ *редакционная коллегия**:

Редакционная коллегия Стандарта OSTIS

:= [редколлегия стандарта OSTIS]

:= [Редакционная коллегия, обеспечивающая развитие *Стандарта OSTIS*, совмещенного (интегрированного) с комплексным учебно-методическим обеспечением *подготовки специалистов в области Искусственного интеллекта.*]

⇒ *несёт ответственность за**:

[Корректность (непротиворечивость), системность, целостность, полноту всех разрабатываемых материалов *Стандарта OSTIS* и, в том числе, *учебно-методического обеспечения подготовки специалистов в области Искусственного интеллекта.*]

:= [Рабочий орган, обеспечивающий организацию коллективного творческого процесса по развитию *Стандарта OSTIS*, совмещенного с учебно-методическим обеспечением подготовки соответствующих специалистов.]

:= [Редакционная коллегия, несущая ответственность за качество перманентно эволюционируемого *Стандарта OSTIS.*]

:= [Редакционная коллегия, осуществляющая координацию деятельности авторов разработки очередной (следующей) версии *Стандарта OSTIS* к следующей *конференции OSTIS.*]

⇒ *примечание**:

[В частности, *Редколлегия Стандарта OSTIS* может осуществлять распределение работ по построению следующей версии *Стандарта OSTIS* с четкой привязкой ответственных авторов *Стандарта OSTIS* к соответствующим разделам *Стандарта OSTIS.*]

:= [Очень важная структура, определяющая научно-технический уровень, авторитет и репутацию всей *Технологии OSTIS* в глазах международной научно-технической общественности.]

⇒ *примечание**:

[Каждый член *Редакционной коллегии Стандарта OSTIS* должен быть достаточно активным членом *Авторского коллектива Стандарта OSTIS.* Все члены *Редакционной коллегии Стандарта OSTIS* должны иметь научную степень не ниже кандидата наук.]

- = {
- *Голенков В.В.*
 - *Головко В.А.*
 - *Гулякина Н.А.*
 - *Краснопрошин В.В.*
 - *Курбацкий А.Н.*
 - *Гордей А.Н.*
 - *Шункевич Д.В.*
 - *Азаров И.С.*
 - *Захарьев В.А.*
 - *Родченко В.Г.*
 - *Голубева О.В.*
- }

⇒ *консорциум**:

Консорциум OSTIS

⇒ *несёт ответственность за**:

- [Распространение, внедрение и продвижение комплекса стандартов *Технологии OSTIS* во все виды *человеческой деятельности* в рамках глобального комплексного прикладного проекта *Экосистемы OSTIS.*]

- [Взаимодействие с международными институтами и консорциумами, заинтересованными в стандартизации *интеллектуальных компьютерных систем* и соответствующих технологий их проектирования, производства, эксплуатации и реинжиниринга.]

консорциум*

- := [быть консорциумом для заданного инновационного продукта, которым, в частности, может быть стандарт некоторые перспективные технологии*]
- := [быть субъектом продвижение заданного инновационного продукта на международной арене*]
- € бинарное ориентированное отношение

Стандарт OSTIS

⇒ аналоги*:

[Аналогами (в широком смысле) *Стандарта OSTIS* можно считать:

- любую серьезную попытку систематизации результатов, полученных в области *Искусственного интеллекта* к текущему моменту:
 - учебник, достаточно полно отражающий текущее состояние *Искусственного интеллекта*;
 - справочник, содержащий достаточно полную информацию о текущем состоянии *Искусственного интеллекта*. Примером такого справочника является трехтомный справочник по Искусственному интеллекту (*ИскусИнтелСОиЭС-1990кн* , *ИскусИнтелМоделМетод-1990кн* , *ИскусИнтелПрограмАппаратСредства-1990кн*).
- любую попытку перехода от частных формальных моделей различных компонентов интеллектуальных компьютерных систем общей (объединённой, интегрированной) формальной модели *интеллектуальных компьютерных систем* в целом – к общей теории *интеллектуальных компьютерных систем*;
- любую попытку унификации технических решений, устранения "вредного" многообразия форм технических решений при разработке *интеллектуальных компьютерных систем*
- первые попытки разработки стандартов *интеллектуальных компьютерных систем* и технологий *Искусственного интеллекта*, которые чаще всего ограничиваются построением систем соответствующих понятий.

]

⇒ сравнение с аналогами*:

[Перечислим основные особенности и достоинства *Стандарта OSTIS* по сравнению с различного вида его аналогами:

- *Стандарт OSTIS* – это не просто систематизация современного состояния результатов в области *Искусственного интеллекта*, это систематизация, представленная в виде общей комплексной формальной модели интеллектуальных компьютерных систем и комплексной формальной модели технологии их разработки. Более того, текст *Стандарта OSTIS* представляет собой раздел *базы знаний* специальной *интеллектуальной метасистемы*, которая ориентирована:
 - на поддержку эволюция *Стандарта OSTIS*;
 - на поддержку разработки *интеллектуальных компьютерных систем* различного назначения;
 - на поддержку *подготовки специалистов в области Искусственного интеллекта*;
- *Стандарт OSTIS* – это динамический текст, перманентно отражающий все новые и новые научно-технические результаты, получаемые в области *Искусственного интеллекта* в рамках *Общей теории интеллектуальных компьютерных систем* и *Общей комплексной технологии разработки интеллектуальных компьютерных систем*. Здесь важной является оперативность фиксации новых научно-технических результатов, т.е. минимизация отрезка времени между моментом получения новых результатов и моментом интеграции описания этих результатов в состав *Стандарта OSTIS*. В перспективе авторы новых научно-технических результатов в области *Искусственного интеллекта* будут заинтересованы лично публиковать (интегрировать) свои результаты в состав *Стандарта OSTIS*, т.е. становиться соавторами *Стандарта OSTIS*, чтобы обеспечить необходимую оперативность такой публикации и отсутствие искажений своих результатов. Динамичность *Стандарта OSTIS* и достаточная оперативность интеграции в его состав новых научно-технических результатов в области *Искусственного интеллекта* делает *Стандарт OSTIS* всегда актуальным и никогда морально устаревшим;
- В рамках стандарта OSTIS нет противопоставления между научно-технической информацией, добываемой в области Искусственного интеллекта, и учебно-методической информацией, используемой для подготовки и самоподготовки специалистов в области Искусственного интеллекта. информация о том, чему учить, должна быть "переплетена", интегрирована с информацией о том, как учить.

]

⇒ ключевые знаки*:

- ⟨ • *Технология OSTIS*
 - := [Open Semantic Technology for Intelligent Systems]

- := [Открытая технология онтологического проектирования, производства и эксплуатации семантически совместимых гибридных интеллектуальных компьютерных систем]
- *технология*
 - := [комплекс моделей, методов и средств, обеспечивающих решение соответствующего класса задач, выполнение соответствующего класса действий, реализацию соответствующего вида деятельности]
 - ⊃ *технология разработки интеллектуальных компьютерных систем*
 - ⊃ *Технология OSTIS*
 - := [Технология разработки ostis-систем]
 - := [Технология разработки семантически совместимых интеллектуальных компьютерных систем, способных к самостоятельному взаимодействию]
 - ⊃ *технология проектирования интеллектуальных компьютерных систем*
 - ⊃ *технология производства интеллектуальных компьютерных систем*
 - := [технологии сборки и установки интеллектуальных компьютерных систем]
 - ⊃ *технология эксплуатации интеллектуальных компьютерных систем*
 - ⊃ *технология реинжинеринга интеллектуальных компьютерных систем*
 - := [технология совершенствования эволюции интеллектуальных компьютерных систем]
 - ⊃ *технология эволюции технологий*
 - := [метатехнология эволюции]
 - ⊃ *открытая технология*
 - := [технология, доступная не только для тех, кто желает ее использовать, но и для тех, кто желает участвовать в ее развитии – для того, чтобы технология быстро развивалась, она должна иметь широкий круг своих пользователей и разработчиков]
 - ⇒ *пояснение**:
 - [технология с достаточно свободным формированием её авторского коллектива и пользовательского коллектива]
 - ⊃ *свободно распространяемая технология*
 - ⊃ *свободно эволюционируемая технология*
 - := [технология эволюционируемая (совершенствуемая) с помощью свободно формируемого авторского коллектива и, соответственно, развиваемая в рамках открытого проекта]
 - ⊃ *технология проектирования*
 - ⊃ *технология производства*
 - ⊃ *технология эксплуатации*
 - ⊃ *технология эволюции*
 - ⊃ *онтологическая технология*
 - := [технология, в основе которой лежит формальное представление иерархической системы предметных областей и онтологий]
 - := [технология выполнения соответствующего вида деятельности, в основе которой лежит иерархическая система формальных онтологий, обеспечивающая четкую стратификацию указанной деятельности и наследование свойств между различными уровнями детализации этой деятельности]
 - ⊃ *технология онтологического проектирования*
 - ⊃ *технология онтологического производства*
 - ⊃ *технология онтологической эксплуатации*
- *интеллектуальная компьютерная система*
 - := [искусственная интеллектуальная система]
 - ⊃ *интеллектуальная система*
 - := [интеллектуальная кибернетическая система]
 - ⊃ *кибернетическая система*
 - ⊃ *гибридная интеллектуальная компьютерная система*
 - := [интеллектуальная компьютерная система, в которой глубоко интегрированы различные виды знаний и различные модели решения задач]
- *семантическая совместимость интеллектуальных компьютерных систем*[^]
 - := [параметр, определяющий степень (уровень) семантической совместимости каждой пары интеллектуальных компьютерных систем[^]]
- *семейство семантически совместимых интеллектуальных компьютерных систем*^{*}

- := [семейство интеллектуальных компьютерных систем, все пары которых имеют одинаковый уровень семантической совместимости (взаимопонимания)]
- *семантическая унификация интеллектуальных компьютерных систем*
 - := [процесс обеспечения высокого уровня семантической совместимости *интеллектуальных компьютерных систем* в ходе их проектирования, эксплуатации и реинжиниринга]
- *ostis-система*
 - := [интеллектуальная компьютерная система, разработанная по *Технологии OSTIS*]
 - С *интеллектуальная компьютерная система*
- *ostis-документация*
 - := [документация соответствующего объекта, представленная в виде *раздела базы знаний* некоторой *ostis-системы*]
 - Э *Стандарт OSTIS*
- *публикация ostis-документации*
 - := [внешнее представление *ostis-документации*, доступное широкому кругу читателей]
 - С *внешнее представление фрагмента базы знаний ostis-системы*
- *публикация ostis-документации**
 - := [быть публикацией заданной *ostis-документации**]
 - := *пояснение**:
[Бинарное ориентированное *отношение*, каждая *пара* которого связывает знак некоторого *раздела базы знаний* со знаком *файла*, который является внешним представлением указанного раздела, а также является либо копией электронной публикации материалов этого раздела, либо оригинал-макетом бумажной публикации указанных материалов*]
- *оглавление публикации ostis-документации**
 - := [быть оглавлением заданной публикации соответствующей *ostis-документации**]
 - := *пояснение**:
[Бинарное ориентированное *отношение*, каждая *пара* которого связывает знак некоторого *раздела базы знаний* либо знак *файла*, содержащего некоторый *документ*, с описанием иерархии *всех разделов*, входящих в состав указанного *раздела базы знаний* либо указанного *документа**]
- *база знаний ostis-системы*
- *раздел базы знаний ostis-системы*
- *Искусственный интеллект*
 - := [Научно-техническая дисциплина, направленная на изучение *интеллектуальных систем* для построения *искусственных интеллектуальных систем*]
 - ∈ *научно-техническая дисциплина*
- *интеллектуальная система*
 - := [кибернетическая система, имеющая достаточно высокий уровень *интеллекта*]
 - С *кибернетическая система*
 - := [система, в основе функционирования которой лежит обработка *информации*]
 - ⊃ *интеллектуальная компьютерная система*
 - := [искусственная *интеллектуальная система*]
 - := [интеллектуальная *компьютерная система*]
- *SC-код*
 - := [Semantic Computer Code]
 - := [Внутренний язык *ostis-систем*]
- *sc-конструкция*
 - := [информационная конструкция, принадлежащая *SC-коду*]
 - ⊃ *sc-текст*
 - := [синтаксически целостная *sc-конструкция*]
 - ⊃ *sc-знание*
 - := [семантически целостный *sc-текст*]
- *sc-элемент*
 - := [знак, входящий в в состав *sc-конструкции*]
- *sc-идентификатор*
 - := [внешний идентификатор *sc-элемента*]
 - := [внешний идентификатор знака, входящего в текст Внутреннего языка *ostis-системы*]
- *внешний язык ostis-систем*
 - := [язык коммуникации *ostis-систем* с их пользователями и другими *ostis-системами*]
 - Э *SCg-код*
 - := [Semantic Computer Code graphical]

- := [Язык графического представления знаний *ostis-систем*]
 - ⊃ *SCs-код*
 - := [Semantic Computer Code string]
 - := [Язык линейного представления знаний *ostis-систем*]
 - ⊃ *SCn-код*
 - := [Semantic Computer Code natural]
 - := [Язык структурированного представления знаний *ostis-систем*]
- *знание ostis-системы*
 - ⊃ *внутреннее представление знания ostis-системы*
 - ⊃ *внешнее представление знания ostis-системы*
- *предметная область*
 - := [предметная область, представленная в памяти *ostis-системы*]
 - := [*sc-модель* предметной области]
- *онтология*
 - := [формальная онтология, представленная в памяти *ostis-системы*]
 - := [*sc-модель* онтологии]
- *предметная область и онтология*
 - := [объединение *предметной области* и онтологии]
- *кибернетическая система*
 - ⊃ *интеллектуальная система*
 - ⊃ *компьютерная система*
 - ⊃ *интеллектуальная компьютерная система*
- *решатель задач кибернетической системы*
- *интерфейс кибернетической системы*
- *база знаний*
- *традиционная компьютерная технология*
 - := [современная информационная технология]
- *технология искусственного интеллекта*
- *логико-семантическая модель ostis-системы*
 - := [формальная логико-семантическая модель *ostis-системы*, представленная в *SC-коде*]
 - := [*sc-модель ostis-системы*]
 - := [результат проектирования *ostis-системы*]
 - := [стартовое состояние *базы знаний* проектируемой *ostis-системы*]
- *семантическая сеть*
- *семантический язык*
 - := [язык, построенный на основе *семантических сетей*]
- *семантическая модель базы знаний*
 - := [*база знаний*, представленная на *семантическом языке*]
 - ⊃ *sc-модель базы знаний*
- *семантическая модель решателя задач*
 - ⊃ *sc-модель решателя задач*
- *семантическая модель интерфейса кибернетической системы*
 - ⊃ *sc-модель интерфейса кибернетической системы*
- *семантическая окрестность*
 - := [семантическая спецификация]
 - := [*sc-окрестность*]
- *логическая формула*
 - := [*sc-модель* логической формулы]
- *логическая онтология*
 - ⊃ *онтология*
- *внешняя информационная конструкция ostis-системы*
 - := [информационная конструкция, записанная не в *SC-коде*]
- *файл ostis-системы*
- *параметр*
 - := [семейство множеств сущностей, эквивалентных в заданном семействе]
 - ⊃ *класс классов*
- *величина*
 - := [значение параметра]
- *шкала*

- *структура*
:= [фрагмент базы знаний *ostis-системы*]
- *теоретико-множественная онтология*
C *онтология*
- *материальная сущность*
- *пространственная сущность*
- *темпоральная сущность*
:= [временная сущность]
:= [временно существующая сущность]
- *темпоральная сущность базы знаний ostis-системы*
C *темпоральная сущность*
- *действие*
- *задача*
- *план*
- *протокол*
- *метод*
:= [метод решения задач заданного класса]
- *задачная онтология*
C *онтология*
:= [онтология методов решения задач в рамках заданной предметной области]
- *внутренний агент ostis-системы*
- *Язык SCP*
:= [Базовый язык программирования *ostis-систем*]
:= [Semantic Code Programming]
- *денотационная семантика языка программирования*
- *операционная семантика языка программирования*
- *информационно-поисковый агент ostis-системы*
- *логическое исчисление*
- *логический агент ostis-системы*
- *сообщение*
- *интерфейсное действие ostis-системы*
- *агент пользовательского интерфейса ostis-системы*
- *знаковая конструкция*
- *естественный язык*
- *методика разработки ostis-систем*
- *средство разработки ostis-систем*
- *базовый интерпретатор логико-семантических моделей ostis-систем*
- *семантический ассоциативный компьютер*
- *Библиотека многократно используемых компонентов ostis-систем*
:= [Библиотека *Технологии OSTIS*]
- *встраиваемая ostis-система*
- *понимание информации*
- *противоречие в базе знаний ostis-системы*
- *информационная дыра в базе знаний ostis-системы*
:= [неполнота в базе знаний *ostis-системы*]
- *разработчик баз знаний ostis-систем*
- *Экосистема OSTIS*
- *интеллектуальный портал научно-технических знаний*
- *интеллектуальная справочная система*
- *интеллектуальная help-система*
- *интеллектуальная корпоративная система*
- *интеллектуальная система в сфере образования*
⊃ *интеллектуальная обучающая система*
- *интеллектуальная система автоматизации проектирования*
- *интеллектуальная система управления проектированием*
- *интеллектуальная система управления производством*
- *Метасистема IMS.ostis*
- *ostis-система*
:= [интеллектуальная компьютерная система, построенная на *Технологии OSTIS*]
- *интеллектуальная компьютерная система*

```

:= [компьютерная система, основанная на знаниях]
• база знаний
:= [система знаний интеллектуальной компьютерной системы]
)

```

следует отличать*

```

⇒ { • Стандарт OSTIS
    ⇒ пояснение*:
      [как внутреннее представление Стандарта OSTIS в памяти ostis-системы]
  • Технология OSTIS
    ⇒ пояснение*:
      [как объект, специфицируемый (описываемый) стандартом OSTIS]
  • официальная версия Стандарта OSTIS
    ⇒ Стандарт OSTIS-2021
      ⇒ пояснение*:
        [как внешнее представление Стандарта OSTIS в виде исходного текста соответствующего раздела базы знаний]
}

```

/*****

Стандарт OSTIS-2021

← официальная версия*:

Стандарт OSTIS

⇒ примечание*:

[Данная официально изданная версия *Стандарта OSTIS*, которую Вы держите в руках, занимает особое место:

- Во-первых, это первый опыт издания (публикации) подобного документа, в рамках которого необходимо обеспечивать, с одной стороны, строгую формальность, а, с другой стороны, интуитивное и адекватное понимание формальных текстов со стороны читателей;
- Во-вторых, данный текст является описанием условно выделенной первой версии *Стандарта OSTIS* (*Стандарта OSTIS-2021*), в рамках которого представлены далеко не все разделы *Стандарта OSTIS*. Эти разделы будут представлены в последующих версиях *Стандарта OSTIS* (в *Стандарте OSTIS-2022*, в *Стандарте OSTIS-2023* и т.д.);
- Особенностью *публикации* (издания) Стандарта OSTIS версии OSTIS-2021, как, впрочем, и всех последующих версий, является то, что она оформлена в виде внешнего представления основной части *базы знаний* специальной *ostis-системы*, которая предназначена для комплексной поддержки проектирования семантически совместимых *ostis-систем*. Эту систему мы назвали *Метасистемой IMS.ostis* (Intelligent MetaSystem for ostis-systems). Последовательность изложения материала во внешнем представлении *базы знаний* не является единственно возможным маршрутом прочтения (просмотра) *базы знаний*. Каждый читатель, войдя в *Метасистему IMS.ostis*, может выбрать любой другой маршрут навигации по этой *базе знаний*, задавая указанной метасистеме те *вопросы*, которые в текущий момент его интересуют. Таким образом, читая предлагаемый вашему вниманию текст и одновременно работая с *Метасистемой IMS.ostis*, можно значительно быстрее усвоить детали *Технологии OSTIS* и значительно быстрее приступить к непосредственному использованию указанной технологии. Этому также способствует большое количество примеров семантических моделей различных фрагментов *интеллектуальных компьютерных систем*;
- Основной семантический вид *разделов баз знаний ostis-систем* – это формальное представление различных *предметных областей* вместе с соответствующими им *онтологиями*. При этом явно указываются связи между этими *предметными областями* и *онтологиями*. Таким образом, *база знаний Метасистемы IMS.ostis*, как и любых других *интеллектуальных компьютерных систем*, построенных по *Технологии OSTIS*, представляет собой иерархическую систему связанных между собой формальных моделей *предметных областей* и соответствующих им *онтологий*. Соответственно этому структурирован и текст *Стандарта OSTIS-2021*;
- В основе *Технологии OSTIS* лежит предлагаемая нами унификация *интеллектуальных компьютерных систем*, основанная, в свою очередь, на *смысловом представлении знаний в памяти интеллектуальных компьютерных систем*. Таким образом, данную *публикацию Стандарта OSTIS-2021* можно рассматривать как версию *стандарта* семантических моделей *интеллектуальных компьютерных систем*. Последующие *публикации*, посвящённые детальному описанию различных компонентов *Технологии OSTIS*, будут также оформляться как внешнее представление соответствующих *разделов базы знаний*

Метасистемы IMS.ostis и будут отражать следующие этапы развития *Технологии OSTIS*, следующие версии этой технологии, и, соответственно, следующие версии *Метасистемы IMS.ostis*;

- Все основные положения *Технологии OSTIS* рассматривались и обсуждались на ежегодных конференциях *OSTIS*, которые стали важным стимулирующим фактором становления и развития *Технологии OSTIS*. Мы благодарим всех активных участников этих конференций;
- Важной задачей *Стандарта OSTIS-2021* была выработка стилистики формализованного представления научно-технической информации, которая одновременно была бы понятна как человеку, так и интеллектуальной компьютерной системе. По сути это принципиально новый подход к оформлению научно-технических результатов, позволяющий:
 - существенно повысить уровень автоматизации анализа качества (корректности, целостности) научно-технической информации;
 - интеллектуальным компьютерным системам непосредственно (без какой-либо дополнительной "ручной" доработки) использовать информацию (знания), содержащуюся в разработанных специалистами документах;
 - существенно упростить согласование точек зрения различных специалистов, входящих в коллектив разработчиков той или иной научно-технической документации.
- Для выработки стилистики и формального представления научно-технической информации нам было важно привлечь к обсуждению и анализу материала *Стандарта OSTIS-2021* как можно больше коллег, участвующих в развитии и применении *Технологии OSTIS*. При этом некоторых коллег мы включили в число соавторов соответствующих разделов монографии. Основной целью написания *Стандарта OSTIS-2021* является создание технологических и организационных предпосылок к принципиально новому подходу к организации *научно-технической деятельности* в любой области и, в частности, в области создания и перманентного развития комплексной технологии проектирования и производства семантически совместимых интеллектуальных компьютерных систем (*Технологии OSTIS*). Суть указанного подхода заключается в глубокой конвергенции и интеграции результатов деятельности всех специалистов, участвующих в создании и развитии *Технологии OSTIS*, путем организации коллективной разработки *базы знаний*, являющейся формальным представлением полной *Документации Технологии OSTIS*, отражающей текущее состояние этой технологии.

]

⇒ соавторы*:

[На данном этапе к разработке и оформлению различных разделов текущей версии *Стандарта OSTIS-2021* кроме основных авторов были привлечены студенты, магистранты, аспиранты и преподаватели кафедры Интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники и кафедры Интеллектуальных информационных технологий Брестского государственного технического университета, а также сотрудники ОАО «Савушкин продукт» и ООО «Интелидженд семантик системс». Так, например,

- соавторами Раздела “*Предметная область и онтология искусственных нейронных сетей и соответствующая им предметная область и онтология действий по обработке искусственных нейронных сетей*” являются Головкин В.А., Ковалёв М.В., Крошечко А.А., Михно Е.В.;
- соавтором Разделов “*Предметная область и онтология логико-семантических моделей интерфейсов компьютерных систем, основанных на смысловом представлении информации*”, “*Предметная область и онтология интерфейсов ostis-систем*”, “*Предметная область и онтология интерфейсных действий пользователей ostis-системы*” является Садовский М.Е.;
- соавторами Раздела “*Предметная область и онтология естественных языков*” являются Гордей А.Н., Никифоров С.А., Бобёр Е.С., Святощик М.И.;
- соавторами Сегмента “*Предметная область и онтология субъектно-объектных спецификаций воздействий*” в Разделе “*Глобальная предметная область действий и задач и соответствующая ей онтология методов и технологий*” являются Гордей А.Н., Никифоров С.А., Бобёр Е.С., Святощик М.И.;
- соавтором Раздела “*Предметная область и онтология программных вариантов реализации базового интерпретатора логико-семантических моделей ostis-систем на современных компьютерах*” является Корончик Д.Н.;
- соавторами Раздела “*Предметная область и онтология семантически совместимых ostis-систем управления рецептурным производством*” являются Таберко В.В., Иванюк Д.С., Касьяник В.В., Пупена А.Н.

]

⇒ благодарности*:

[Благодарим студентов Банцевич К.А., Бутрина С.В., Василевскую А.П., Менькову Е.А., Жмырко А.В., Григорьеву И.В., Загорского А.Г., Марковца В.С., Киневича Т.О. и сотрудников кафедры Интеллектуальных

информационных технологий Белорусского государственного университета информатики и радиоэлектроники за оказание технической помощи при подготовке текста к печати, сотрудников кафедры Интеллектуальных информационных технологий Брестского государственного технического университета, сотрудников ОАО «Савушкин продукт», а также выражаем благодарность ООО «Интелиджент семантик системс» и его генеральному директору Т. Грюневальду за финансовую поддержку работ по развитию *Технологии OSTIS*, а также финансовую поддержку издания *Стандарта OSTIS*.]

:= [Издание Документации Технологии OSTIS-2021]

:= [Первое издание (публикация) Внешнего представления Документации Технологии OSTIS в виде книги]

€ публикация

:= [библиографический источник]

€ официальная версия Стандарта OSTIS

€ бумажное издание

€ научное издание

⇒ рекомендация издания*:

Совет БГУИР

⇒ рецензенты*:

{ • Курбацкий А.Н.

• Дудкин А.А.

}

⇒ издательство*:

Беспринт

€ {}

:= УДК*:

[004.8]

:= ISBN*:

[978-985-7267-13-2]

⇒ оглавление официальной версии Стандарта OSTIS*:

Оглавление Стандарта OSTIS-2021

Оглавление Стандарта OSTIS-2021

=
[

Принципы построения и структура Стандарта OSTIS	5
Принципы построения и структура официальной версии Стандарта OSTIS-2021	16
Анализ методологических проблем современного состояния работ в области Искусственного интеллекта	22
Предметная область и онтология информационных конструкций и языков	99
<i>Описание внутреннего языка ostis-систем</i>	116
<i>Предметная область и онтология внешних идентификаторов знаков, входящих в информационные конструкции внутреннего языка ostis-систем</i>	139
<i>Описание языка графического представления информационных конструкций в ostis-системах</i>	159
<i>Описание языка линейного представления информационных конструкций в ostis-системах</i>	205
<i>Описание языка структурированного представления информационных конструкций в ostis-системах</i>	238
Описание общих принципов оформления внутреннего и внешнего представления информационных конструкций в ostis-системах	244
Предметная область и онтология кибернетических систем	266
Предметная область и онтология логико-семантических моделей компьютерных систем, основанных на смысловом представлении информации	336
<i>Предметная область и онтология смыслового представления информации</i>	339
<i>Предметная область и онтология многоагентных моделей решения задач, основанных на смысловом представлении информации</i>	360
<i>Предметная область и онтология логико-семантических моделей интерфейсов компьютерных систем, основанных на смысловом представлении информации</i>	363
Предметная область и онтология знаний и баз знаний ostis-систем	366
<i>Предметная область и онтология множеств</i>	370
<i>Предметная область и онтология связок и отношений</i>	389
<i>Предметная область и онтология параметров, величин и шкал</i>	412
<i>Предметная область и онтология чисел и числовых структур</i>	422
<i>Предметная область и онтология структур</i>	428
<i>Предметная область и онтология темпоральных сущностей</i>	438

<i>Предметная область и онтология темпоральных сущностей баз знаний ostis-систем</i>	450
<i>Предметная область и онтология семантических окрестностей</i>	454
<i>Предметная область и онтология предметных областей</i>	458
<i>Предметная область и онтология онтологий</i>	469
<i>Предметная область и онтология логических формул, высказываний и формальных теорий .</i>	477
<i>Глобальная предметная область действий и задач и соответствующая ей онтология методов и технологий</i>	490
<i>Предметная область и онтология решателей задач ostis-систем</i>	539
<i>Предметная область и онтология действий, задач, планов, протоколов и методов, реализуемых ostis-системой, а также внутренних агентов, выполняющих эти действия</i>	542
<i>Предметная область и онтология Базового языка программирования ostis-систем</i>	563
<i> Предметная область и онтология денотационной семантики Базового языка программирования ostis-систем</i>	575
<i> Предметная область и онтология операционной семантики Базового языка программирования ostis-систем</i>	583
<i>Предметная область и онтология искусственных нейронных сетей и соответствующая им предметная область и онтология действий по обработке искусственных нейронных сетей . .</i>	585
<i>Предметная область и онтология интерфейсов ostis-систем</i>	604
<i>Предметная область и онтология интерфейсных действий пользователей ostis-системы . . .</i>	613
<i>Предметная область и онтология естественных языков</i>	617
<i>Предметная область и онтология базовых интерпретаторов логико-семантических моделей ostis-систем</i>	626
<i>Предметная область и онтология программных вариантов реализации базового интерпретатора логико-семантических моделей ostis-систем на современных компьютерах</i>	628
<i>Предметная область и онтология семантических ассоциативных компьютеров для ostis-систем</i>	650
<i>Предметная область и онтология Экосистемы OSTIS</i>	656
<i>Предметная область и онтология методов и средств реализации целенаправленного и персонафицированного процесса обучения пользователей для каждой ostis-системы, входящей в состав Экосистемы OSTIS</i>	661
<i>Предметная область и онтология ostis-систем, являющихся персональными ассистентами пользователей в рамках Экосистемы OSTIS</i>	667
<i>Предметная область и онтология семантически совместимых ostis-систем управления рецептурным производством</i>	668
<i>Предметная область и онтология семантически совместимых интеллектуальных порталов научно-технических знаний</i>	675
<i> Логико-семантическая модель Метасистемы IMS.ostis</i>	676

Библиографический раздел Стандарта OSTIS**683**

- | | |
|---|---|
|] | /* Завершили <i>Оглавление Стандарта OSTIS-2021</i> */ |
| } | /* Завершили <i>Титульную спецификацию Стандарта OSTIS</i> */ |

/* Раздел *****/

Анализ методологических проблем современного состояния работ в области Искусственного интеллекта

⊃=

{

←

конкатенация сегментов*:

- Структура деятельности в области Искусственного интеллекта
- Текущее состояние и проблемы дальнейшего развития деятельности в области Искусственного интеллекта
- Понятие Технологии OSTIS
- Использование Технологии OSTIS для повышения качества человеческой деятельности в области Искусственного интеллекта
- Понятие Экосистемы OSTIS

}

⇒

рассматриваемые вопросы*:

- [Каковы основные стратегические цели (сверхзадачи) научно-технической деятельности в области Искусственного интеллекта]
- [Какие проблемы являются на сегодняшний день актуальными для дальнейшего развития различных направлений Искусственного интеллекта и для развития Искусственного интеллекта в целом как общей (объединённой) научно-технической дисциплины, а также для развития различных форм деятельности в этой области (научно-исследовательской деятельности создания технологий разработки интеллектуальных компьютерных систем, образовательной деятельности, бизнеса)]
- [Какие проблемы являются на сегодняшний день актуальными для развития других научно-технических дисциплин и являются ли эти проблемы аналогичными тем, которые актуальны для развития Искусственного интеллекта]
- [Какие можно предложить подходы к решению указанных выше проблем и как для этого можно использовать создаваемый сейчас новый технологический уклад в области Искусственного интеллекта (следующий уровень технологий искусственного интеллекта)]
- [Как будет выглядеть на основе следующего уровня технологий Искусственного интеллекта комплексная автоматизация всех видов человеческой деятельности, а также взаимодействие различных видов человеческой деятельности, т.е. как будет выглядеть архитектура smart-общества]
- [Устраивает ли нас уровень семантической совместимости взаимопонимания между современными виртуальными компьютерными системами и что необходимо сделать для повышения этого уровня]
- [Устраивает ли нас уровень семантической совместимости взаимопонимания между современными интеллектуальными компьютерными системами их пользователями и что необходимо сделать для повышения этого уровня]

}

⇒

аннотация*:

[Предлагаемое вашему вниманию рассмотрение методологических проблем современного состояния работ в области Искусственного интеллекта состоит из следующих частей:

- Анализ актуальных проблем, препятствующих дальнейшему развитию Искусственного интеллекта как научно-технической дисциплины:
 - Проблемы развития научных исследований в области Искусственного интеллекта
 - Проблемы разработки технологий проектирования и реализации интеллектуальных компьютерных систем;
 - Проблемы формирования рынка интеллектуальных компьютерных систем;
 - Образовательные проблемы в области Искусственного интеллекта;
 - Проблемы развития бизнеса в области Искусственного интеллекта.
- Анализ проблем автоматизации сложных видов деятельности:
 - научно-исследовательской деятельности в рамках различных научных дисциплин;
 - создание технологий проектирования и производства (реализации) сложных технических систем;
 - инженерной деятельности по разработке сложных технических систем;
 - образовательной деятельности по наукоёмким техническим специальностям
- Формулировка принципов, лежащих в основе Технологии OSTIS, предназначенной для решения указанных выше проблем;

- Рассмотрение структуры *Экосистемы OSTIS*, построенной по *Технологии OSTIS* и обеспечивающей комплексную автоматизацию всех видов человеческой деятельности

- ⇒]
используемые знаки общих понятий и иных сущностей*:
- { • *деятельность*
 - := [область деятельности]
 - ⊃ *человеческая деятельность*
 - *вид деятельности*
 - ⊃ *проектирование*
 - := [проектная деятельность]
 - ⊃ *производство*
 - := [производственная деятельность]
 - ⊃ *наука*
 - := [научная деятельность]
 - *проект*
 - ⊃ *открытый проект*
 - *консорциум*
 - *технология*
 - ⊃ *информационная технология*
 - ⊃ *технология искусственного интеллекта*
 - *кибернетическая система*
 - ⊃ *интеллектуальная система*
 - ⊃ *интеллектуальная компьютерная система*
 - := [искусственная интеллектуальная система]
 - *конвергенция*[^]
 - := [уровень конвергенции (близости)]
 - ⊃ *конвергенция кибернетических систем*[^]
 - *интеграция*^{*}
 - ⊃ *интеграция кибернетических систем*^{*}
 - ⊃ *эkleктичная интеграция*^{*}
 - ⊃ *глубокая интеграция*^{*}
 - *интегрированная система*
 - ⊃ *эkleктичная система*
 - ⊃ *гибридная система*
 - *экосистема интеллектуальных компьютерных систем*
 - *рынок знаний*
 - := [рыночная организация порождения эволюции и применения знаний]
 - *smart-общество*
 - := [общество, в основе которого лежит экосистема интеллектуальных компьютерных систем и рынок знаний]
- ⇒ }
ключевые знаки*:
- { • *Искусственный интеллект*
 - ∈ *научно-техническая дисциплина*
 - ⊂ *научно-техническая деятельность*
 - *интеллектуальная система*
 - ⊃ *интеллектуальная компьютерная система*
 - *Общая теория интеллектуальных систем*
 - *Базовая комплексная технология проектирования интеллектуальных компьютерных систем*
 - *Технология производства спроектированных интеллектуальных компьютерных систем*
 - *Специализированная инженерия в области Искусственного интеллекта*
 - *Образовательная деятельность в области Искусственного интеллекта*
 - *Бизнес-деятельность в области Искусственного интеллекта*
 - ***Технология OSTIS***
 - ***ostis-система***
 - *смысловое представление информации*
 - *агентно-ориентированная модель обработки информации в памяти*
 - *стандартизация ostis-систем*
 - ***SC-код***

- абстрактная sc-машина
- конвергенция знаний в памяти
- ostis-систем
- конвергенция моделей решения задач в ostis-системе
- интеграция знаний в памяти ostis-системы
- интеграция моделей решения задач в ostis-системе
- ostis-сообщество
- ostis-технология
 - ⊃ ostis-технология проектирования
 - ⊃ ostis-технология производства
 - ⊃ технология эксплуатации ostis-систем
 - ⊃ технология реинжиниринга ostis-систем
- **Ядро Технологии OSTIS**
- OSTIS-портал научных знаний в области Искусственного интеллекта
- Проект IMS.ostis
- **Метасистема IMS.ostis**
- Проект Программной реализации универсальной абстрактной sc-машины
- Проект разработки Универсального sc-компьютера
- Специализированная инженерия, осуществляемая на основе Технологии OSTIS
- Образовательная деятельность в области Искусственного интеллекта, осуществляемая на основе технологии OSTIS
- **Консорциум OSTIS**
- **Экосистема OSTIS**
 - := [симбиоз семантически совместимых и координирующих свою деятельность *ostis-систем* и людей, направленный на существенное, качественное повышение уровня автоматизации всех видов человеческой деятельности]
 - ⇒ примечание*:

[Семантически совместимая (понятийно согласованная) формализация всех(!) видов человеческой деятельности является органической частью *Технологии OSTIS(!)*. То есть формализуемые отраслевые стандарты *всех видов человеческой деятельности* должны строго наследовать свойства всей системы базовых понятий и знаний, лежащих в основе Технологии OSTIS. Таким образом речь идет о строгой формальной модели *Экосистемы OSTIS* как единого целого и здесь есть место всем приложениям, но приведенным в комплексную систему. Если к построению такой комплексной формальной модели *всех видов человеческой деятельности* подходить системно, то все не так страшно, так как многие модели можно и нужно строить на основе аналогий, стратификации, наследования и свойств. Это придаст существенную динамику эволюции этих формальных моделей.

К сожалению, современная наука психологически ориентирована на поиск отличий, на выявление принципиальной (научной) новизны своих результатов (что является необходимым фактором оценки этих решений). В этом ничего плохого нет, но для решения системных проблем (в частности, для вывода *Искусственного интеллекта* из кризисного состояния) необходимо существенно активизировать поиск сходств, аналогий, реализацию конвергентных процессов по построению комплексных интегрированных моделей. Это не менее значимые научные результаты, чем выявление принципиально новых свойств и закономерностей.]
- *человеческая деятельность*
- *вид человеческой деятельности*
- *автоматизация человеческой деятельности*
- *качество человеческой деятельности*
- *субъект Экосистемы OSTIS*
- *Рынок знаний, реализованный в рамках Экосистемы OSTIS*
- *smart-общество*

}

Анализ методологических проблем современного состояния работ в области Искусственного интеллекта

⇒ *предисловие**:

[Анализ современного состояния работы в области *Искусственного интеллекта* показывает то, что указанная область *человеческой деятельности* находится в глубоком фундаментальном методологическом и трудновидимом кризисе. Поэтому основными целями данного раздела являются:

- выявление основных причин возникновения указанного кризиса;
- уточнение основных мер, направленных на устранение этого кризиса.

]

⇒ *основная цель**:

[Сформировать мотивацию и инфраструктуру для создания эволюции информационных технологий принципиально нового уровня, в основе которого лежат семантически совместимые *интеллектуальные компьютерные системы*, способные согласовывать свои действия в заранее непредсказуемых обстоятельствах]

⇒ *примечание**:

[Сейчас актуально не столько обсуждать различные вопросы *Искусственного интеллекта*, а обсуждать проблемы и пути решения этих проблем. Нельзя делать вид, что всё хорошо.]

Искусственный интеллект

⇒ *примечание**:

[Современное кризисное состояние *Искусственного интеллекта* вполне логично – это естественный этап эволюции любых сложных систем и технологий –

- Сначала накопление большого количества конкретных решений;
- Анализ полученного многообразия и превращение его в стройную систему качественно более высокого уровня.

Кризисов не надо бояться – их надо преодолевать. Диалектику и, в частности, переход количества в качество ещё никто не отменял.]

Современное состояние технологии Искусственного интеллекта

⇒ *пояснение**:

[К настоящему моменту мы научились разрабатывать *интеллектуальные компьютерные системы* самого различного назначения. Но для повышения уровня автоматизации всё более и более широких видов человеческой деятельности необходим качественный переход к разработке не отдельных *интеллектуальных компьютерных систем*, а целых комплексов самостоятельно взаимодействующих между собой *интеллектуальных компьютерных систем*. Это требует фундаментального переосмысления теории технологии проектирования *интеллектуальных компьютерных систем*. Эффективный переход количества в новое качество требует серьезных усилий]

⇒ *эпиграф**:

[Необходим переход от "зоопарка" локальных идей, сервисов и информационных ресурсов к их системе.]

⇒ *эпиграф**:

[From data science to knowledge science.]

⇒ *эпиграф**:

[Одно дело – создавать локальные шедевры и совсем другое дело – двигаться ко всеобщей гармонии.]

Современное состояние информационных технологий

⇒ *примечание**:

[Экспертам в процессе обсуждения инновационных вопросов приходится тратить много времени на формирование нового понятийного аппарата. <...> Мировому сообществу есть смысл задуматься над созданием нового искусственного языка, <...> чтобы с учётом возможностей современного мира сформулировать новую среду экспертного общения.]

⇒ *автор**:

Курбацкий А.Н.

будущие технологии Искусственного интеллекта

⇒ *примечание**:

[Смысл той информации, которой оперирует каждая(!) интеллектуальная компьютерная система, а также смысл того, что она делает (какие задачи она решает) должен быть четко формализован и является частью её базы знаний. Формализация этого смысла (в состав которой входит экспертное согласование системы используемых понятий) представляет собой первый этап проектирования каждой *интеллектуальной компьютерной системы*, обеспечивающий семантическую совместимость (взаимопонимание) *интеллектуальных компьютерных систем* и эффективное их взаимодействие (самостоятельную организацию коллективной деятельности). Таким образом, необходим универсальный формальный язык, который используется как экспертами, разработчиками, так и непосредственно самими *интеллектуальными компьютерными системами*.]

Технология OSTIS

⇒ *теоретический компонент**:

- { • [Комплексная семантическая теория интеллектуальных компьютерных систем (ostis-систем)]
 - [Комплексная семантическая теория человеческой деятельности как Экосистемы (симбиоза) с иерархическим комплексом интеллектуальных компьютерных систем]
 - [Теория перманентной эволюции (реинжиниринга) указанной Экосистемы (с минимизацией этапов локальной приостановки)]
- ⇒ *примечание**:
[Стандарты должны меняться быстро, а Экосистема должна быстро приводиться в соответствие с новыми стандартами.]
- }

Подготовка специалистов в области Искусственного интеллекта

⇒ *примечание**:

[Массовая подготовка высококвалифицированных *специалистов в области Искусственного интеллекта*, способных преодолеть современное кризисное состояние *Искусственного интеллекта*, фактически и является самым главным фактом преодоления указанного кризиса.

Необходимым условием и эпицентром вывода *Искусственного интеллекта* из кризисного состояния и повышения темпов эволюции технологий *Искусственного интеллекта* является организация *подготовки специалистов в области Искусственного интеллекта* на основе активного привлечения студентов, магистрантов и аспирантов к перманентному процессу эволюции *Технологии OSTIS*.

Очевидно, этому должно способствовать объединение соответствующей учебно-методической базы для разных кафедр, осуществляющих такую подготовку.

На современном этапе развития *Искусственного интеллекта* требуется не просто подготовка специалистов в этой области – а подготовка специалистов принципиально новой формации, способных

- рассматривать область *Искусственного интеллекта* не просто как многообразие *интеллектуальных компьютерных систем*, а как постоянно эволюционируемую Экосистему таких систем
- эффективно участвовать в решении как фундаментальных, системных, технологических проблем, так и практических, прикладных проблем эволюции указанной *Экосистемы*

Все это требует существенного переосмысления организации учебного процесса и учебно-методического обеспечения.

«Часто, например, совершенствование программных систем сводится к программным "заплаткам". Через какое-то время мы имеем программу со множеством "заплаток", как правило уже громоздкую и малоэффективную. В итоге – иногда её проще выбросить и создать новую.» (*Курбацкий. А.Н. Новая СЯИТЭВНА-2020 ст*)

Современная разработка каждой сложной программной системы требует построения качественной формальной ("цифровой") модели объекта управления, объекта автоматизации, причем *семантически совместимой* с соответствующими моделями в смежных системах.

Здесь важна общая математическая культура и унификация такой формализации.

В настоящее время методологический подход к инженерной деятельности при разработке компьютерных систем часто выглядит следующим образом: «Поставьте мне четкую инженерную задачу и я ее выполню. Но ответственность за ее постановку я с себя снимаю и не хочу учитывать критерии качества постановки задачи на более высоком уровне».

Для наукоемких проектов, реализуемых в рамках развивающихся технологий, это недопустимо.

Каждый инженер должен понимать, что он делает и каковы истинные более глубокие критерии качества его результата.

Нужна принципиально новая психологическая установка.

Необходимо учитывать не только желание заказчика, но и общие принципы и стандарты разрабатываемых *интеллектуальных компьютерных систем*.

В основе организации образовательной деятельности на современном этапе развития *Искусственного интеллекта* лежит:

- четкое формальное описание того, чему мы учим (каким знаниям и навыкам) – в нашем случае это описание текущей версии *Стандарта OSTIS* и направлений эволюции этого стандарта;

- уточнение того, что должен делать студент, магистрант и любой специалист для быстрого и качественного приобретения этих знаний и навыков.

Нужна комплексная учебная программа по специальности *Искусственный интеллект*, а не мозаика отдельных учебных дисциплин. И, соответственно этому необходимо комплексное учебно-методическое пособие, достаточно полно отражающее текущее состояние теории и технологии проектирования *интеллектуальных компьютерных систем*.

Использование проектного метода при подготовке специалистов в области Искусственного интеллекта предполагает составление систематизированного сборника упражнений и задач, в частности, направленных на эволюцию Технологии OSTIS и посильных для студентов специальности *Искусственный интеллект*:

- представление конкретных фрагментов различных предметных областей и онтологий;
- представление конкретных специфицированных методов (пополнение библиотек используемых методов из разных предметных областей, например, из теории графов);
- спецификация библиографических источников (в контексте *Базы знаний IMS.ostis*);
- выявление синонимии, омонимии, противоречий;
- сравнительный анализ и обзор близких внешних публикаций.

Таким образом, фронт самостоятельных, весьма полезных и посильных для студентов работ весьма широк. Главное сформировать у студентов профессиональный интерес, познавательную активность, инициативность и самостоятельность.]

⇒ проектный метод*:

[Для того, чтобы научиться разрабатывать *интеллектуальные компьютерные системы*, необходимо приобрести достаточно большой опыт участия и завершения разработки реально востребованных *интеллектуальных компьютерных систем*]

⇒ проектный метод*:

[Для того, чтобы научиться разрабатывать и совершенствовать *технологии искусственного интеллекта*, необходимо приобрести достаточно большой опыт успешного (!) участия в создании различных компонентов комплексной *технологии Искусственного интеллекта*]

⇒ пояснение*:

[Итак, современного специалиста в области *Искусственного интеллекта* необходимо учить:

- не только тому, как следует разрабатывать *интеллектуальные компьютерные системы* с помощью имеющихся (существующих) *методов и средств*, т.е. с помощью имеющихся *технологий*;
- но и тому, как надо развивать (совершенствовать) имеющиеся *технологии*.

Технология OSTIS рассматривается нами не столько, как предлагаемая технология разработки *интеллектуальных компьютерных систем*, а как предлагаемые принципы построения технологии разработки *интеллектуальных компьютерных систем* следующего поколения. Т.е. фактически мы предлагаем не саму технологию (*Технологию OSTIS*), а участие в её создании и развитии, которое может привести даже к радикальным изменениям текущего состояния (текущей версии) этой *технологии*. Это психологически снимает ощущение навязывания предлагаемой технологии и заменяет его на атмосферу партнерства, направленного на перманентную эволюцию указанной технологии. Такой подход создаст также условия для существенного повышения качества *подготовки специалистов в области Искусственного интеллекта*, поскольку дает возможность осуществлять обучение путём непосредственного вовлечения студентов и магистрантов в реальные, практически значимые процессы разработки *интеллектуальных компьютерных систем*, а также в процессы совершенствования (эволюции) соответствующих *технологий*.]

Анализ методологических проблем современного состояния работ в области Искусственного интеллекта

⇒ примечания*:

- [Самые тяжелые кризисные ситуации в научно-технической сфере – это те, которые носят фундаментальный и не совсем очевидный характер. Развитие кибернетики, информатики и искусственного интеллекта подтверждает это. За впечатляющими практическими и теоретическими достижениями незаметно возрастает огромный вал накладных расходов при разработке сложных больших систем – возрастает дублирование, нестыковки, несогласованности.]
- [Нет ничего более грустного, чем созерцать активную творческую деятельность большого количества умных людей, которые по инерции, не отдавая себе отчета, накапливают проблемы, препятствующие дальнейшему качественному развитию этой деятельности. Вместо того, чтобы "разгрести" эти проблемы на благо всем.]
- [Как только мы начнем серьезно относиться к формальному уточнению и согласованию всего многообразия понятий, используемых в области Искусственного интеллекта и различных его приложений,

как только мы начнем реальную(!) совместную работу по общей комплексной формальной теории интеллектуальных компьютерных систем и по созданию комплексной технологии их проектирования, многие современные проблемы *Искусственного интеллекта* начнут решаться. Нет ничего практичнее хорошей теории.]

- [Основной лейтмотив развития технологий *Искусственного интеллекта* – это не только создание компьютерной технологии разработки сей совместной *интеллектуальной компьютерной системы*, но и создание Метатехнологии перманентной эволюции(!) такой технологии. Иначе – эклектика, усугубляющая современный кризис. Для создания эффективно и самостоятельно взаимодействующих *интеллектуальных компьютерных систем* несущественных мелочей не бывает – дьявол кроется в деталях и тонкостях. Важен не столько инжиниринг, сколько реинжиниринг интеллектуальных компьютерных систем и человеческой деятельности в целом.]
- [Решение рассматриваемых кризисных проблем требует:
 - Существенного фундаментального общесистемного переосмысления всего того, что мы творим;
 - Осознания того, что кибернетика, информатика и искусственный интеллект – это общая фундаментальная наука, требующая единого серьезного математического аппарата;
 - Осознания того, что сейчас требуется не расширяемость многообразия точек зрения, а учиться их согласовывать, совершенствуя соответствующие методы.
- [Нам необходимо переходить от автоматизации отдельных видов *человеческой деятельности* к интегрированной автоматизации всего комплекса человеческой деятельности, к созданию и постоянной эволюции всей общечеловеческой *экосистемы*, состоящей из самостоятельно взаимодействующих *интеллектуальных компьютерных систем* как между собой, так и между людьми, автоматизацию деятельности которой они осуществляют. При этом надо помнить, что основные накладные расходы, основные проблемы, возникают на "стыках" при интеграции различных технических решений. Разработчик каждой подсистемы должен гарантировать отсутствие указанных накладных расходов.]
- [Самое главное – надо ориентироваться не на создание идеальной информационной *экосистемы*, а на создание эффективной технологии, направленной на перманентную эволюцию(!) указанной экосистемы.]
- [Уникальность современного кризиса в области кибернетики, информатики и искусственного интеллекта заключается в том, что, несмотря на глобальность этого кризиса, абсолютно реально создать локальный эпицентр по разрешению этого кризиса – в частности, в Республике Беларусь. Для этого есть все предпосылки – специальность *Искусственный интеллект*, опыт разработки компьютерных систем, достаточный научный уровень.]

/* Сегмент *****/

Анализ структуры Деятельности в области Искусственного интеллекта

⊃=

{

⇒

аннотация*:

[Для того, чтобы рассмотреть проблемы дальнейшего развития *деятельности* в области *Искусственного интеллекта* как *научно-технической дисциплины* и, в частности, проблемы комплексной автоматизации этой *деятельности*, необходимо уточнить структуру указанной *деятельности*.]

Искусственный интеллект

∈ область человеческой деятельности

∈ научно-техническая дисциплина

∈ вид человеческой деятельности

:= пояснение*:

[Область человеческой деятельности, основными целями которой являются:

- построение теории интеллектуальных систем;
- создание технологии разработки интеллектуальных компьютерных систем (искусственных интеллектуальных систем);
- переход на принципиально новый уровень комплексной автоматизации всех *видов человеческой деятельности*, который основан на массовом применении *интеллектуальных компьютерных систем* и который предполагает:
 - не только наличие *интеллектуальных компьютерных систем*, способных понимать друг друга и согласовывать свою деятельность,

- но и построение *Общей теории человеческой деятельности*, осуществляемый в условиях нового уровня её автоматизации, (теории деятельности *smart-общества*), которая должна быть "понятна" используемым интеллектуальным компьютерным системам и которая потребует существенного переосмысления современной организации *человеческой деятельности*.

]

= [Научно-техническая деятельность, направленная на построение теории интеллектуальных систем, а также на создание технологии проектирования и производства искусственных интеллектуальных систем (*интеллектуальных компьютерных систем*)]

= [Научно-техническая деятельность в области *Искусственного интеллекта*]

= [Деятельность в области *Искусственного интеллекта*]

= [Научно-техническая деятельность, направленная на исследование феномена *интеллекта*, а также на создание искусственных интеллектуальных систем (*интеллектуальных компьютерных систем*) и включающая в себя соответствующую научно-исследовательскую деятельность, инженерно-технологическую, инженерно-прикладную, образовательную и организационную деятельность]

= пояснение*:

[Междисциплинарная (трансдисциплинарная) область *научно-технической деятельности*, направленная на разработку и эксплуатацию *интеллектуальных компьютерных систем*, обеспечивающих автоматизацию различных сфер *человеческой деятельности*.]

= [Научно-техническая дисциплина направленная на разработку теории индивидуальных *интеллектуальных компьютерных систем* и *интеллектуальных сообществ* (коллективов) таких систем, а также средств поддержки их проектирования и реализации]

= [Научно-техническая дисциплина, являющаяся частью *кибернетики* (теория кибернетических систем), объектом исследования которой являются *интеллектуальные компьютерные системы* (искусственные интеллектуальные системы) и целями которой являются (1) разработка *теории интеллектуальных компьютерных систем*, (2) разработка *технологии(методов и средств) проектирования и производства компьютерных систем*, а также, (3) разработка конкретных интеллектуальных компьютерных систем различного назначения.]

⇒ декомпозиция*:

Декомпозиция *Искусственного интеллекта* по формам деятельности

= { • *Научно-исследовательская деятельность в области Искусственного интеллекта*

⇒ продукт*:

Общая теория интеллектуальных систем

= [Теория, уточняющая структуру и принципы функционирования *интеллектуальных систем*, а также акцентирующая внимание на причинах (предпосылках) возникновения свойства интеллектуальности (феномены *интеллекта*)]

∈ *коллективная научно-техническая деятельность*

∈ *вид человеческой деятельности*

= [научно-исследовательская дисциплина или направление]

⇔ *следует отличать*:*

продукт научно-исследовательской деятельности

- *Разработка Базовой комплексной технологии проектирования интеллектуальных компьютерных систем*

⇒ продукт*:

Базовая комплексная технология проектирования интеллектуальных компьютерных систем

⇒ примечание*:

[Комплексность данной *технологии* заключается в том, что она ориентирована (1) на проектирование *интеллектуальных компьютерных систем* в целом, а не только отдельных их компонентов и (2) на создание объединённой *технологии*, объединяющей самые различные технологические подходы на основе их *конвергенции* и *глубокой интеграции*]

∈ *разработка технологии проектирования искусственных объектов заданного класса*

∈ *вид человеческой деятельности*

- *Разработка Технологий производства спроектированных интеллектуальных компьютерных систем*

⇒ продукт*:

Технология производства спроектированных интеллектуальных компьютерных систем

= [технология реализации (сборки и установки) спроектированных интеллектуальных компьютерных систем]

- ⇒ *примечание**:
 [Очевидно, что данная *технология* должна быть самым тесным образом связана с *Базовой комплексной технологией проектирования интеллектуальных компьютерных систем* (по крайней мере данная *технология* должна "знать" в какой форме ей на "вход" передаётся результат проектирования). Поэтому имеет смысл говорить об объединённой технологии проектирования и производства *интеллектуальных компьютерных систем*]
- Є *производства спроектированных искусственных объектов заданного класса*
 Є *вид человеческой деятельности*
- *Специализированная инженерия в области Искусственного интеллекта*
 := [Множество Процессов разработки (проектирования и производства) *интеллектуальных компьютерных систем* различного назначения, кроме *интеллектуальных компьютерных систем автоматизации проектирования* и автоматизации производства *интеллектуальных компьютерных систем*]
 ⇒ *продукт**:
 множество специализированных *интеллектуальных компьютерных систем*
 ⇒ *основной sc-идентификатор**:
прикладная интеллектуальная компьютерная система
 С *проектирование и производство искусственного объекта заданного класса на основе заданной технологии*
 Є *вид человеческой деятельности*
 - *Образовательная деятельность в области Искусственного интеллекта*
 := [Деятельность, направленная на подготовку молодых специалистов области *Искусственного интеллекта* на перманентное повышение квалификации действующих специалистов в этой области]
 ⇒ *примечание**:
 [Сложность и высокая степень наукоемкости задач, больших своего решения на текущем этапе развития *Искусственного интеллекта*, добавляют к специалистам, работающим в этой области высокие требования к уровню их:
 математической культуры (культуры формализации);
 системной культуры;
 технологической культуры;
 инженерная культура;
 умения работать в коллективных наукоемких проектах.
]
 С *образовательная деятельность*
 Є *вид человеческой деятельности*
 - *Бизнес-деятельность в области искусственного интеллекта*
 ⇒ *пояснение**:
 [Речь идет о бизнес-деятельности в широком смысле как деятельности, направленный на создание инфраструктурных условий для качественного выполнения всех *видов деятельности* в области *Искусственного интеллекта*, а именно:
 на разработку и реализацию грамотной научно-технической политики, связывающей как тактические, так и стратегические цели;
 на глубокую *конвергенцию* всех форм и *видов деятельности* в области *Искусственного интеллекта*;
 на организацию взаимовыгодного сотрудничества различных школ и коллективов, работающих в области *Искусственного интеллекта*;
 на финансовое обеспечение;
 на кадровое обеспечение;
 на материально-техническое обеспечение;
 на организацию проведения различных мероприятий (конференций, выставок, семинаров);
 на публикационную деятельность и защиту интеллектуальной собственности;
 на материально-техническое обеспечение;
]
 С *бизнес-деятельность научно-технической области*

∈ вид человеческой деятельности

}

Разработка базовой комплексной технологии проектирования интеллектуальных компьютерных систем

⇒ декомпозиция*:

- {
 - Разработка общей теории интеллектуальных компьютерных систем
 - ⇒ продукт*:
 - Общая теория интеллектуальных компьютерных систем
 - ∈ разработка теории искусственных объектов заданного класса
 - ∈ вид человеческой деятельности
 - Разработка общей теории интеллектуальных компьютерных систем
 - ⇒ продукт*:
 - Теория проектирования интеллектуальных компьютерных систем
 - ⇒ примечание*:
 - [В состав этой теории входят методы проектирования, библиотеки проектирования и спецификация используемых промышленных средств.]
 - := [Разработка Теории проектной деятельности по построению формальных моделей интеллектуальных компьютерных систем]
 - ∈ теории проектирования интеллектуальных объектов заданного класса
 - ∈ вид человеческой деятельности
 - Разработка комплекса средств автоматизации проектирования интеллектуальных компьютерных систем
 - ∈ разработка комплекса средств автоматизации проектирования искусственных объектов заданного класса
 - ∈ вид человеческой деятельности

}

Разработка Технологии производства спроектированных интеллектуальных компьютерных систем

⇒ декомпозиция*:

- {
 - Разработка Теории производства спроектированных интеллектуальных компьютерных систем
 - := [Разработка Теории производственной деятельности по реализации (сборке и установке) интеллектуальных компьютерных систем]
 - ⇒ продукт*:
 - Теория производства спроектированных интеллектуальных компьютерных систем
 - ⇒ примечание*:
 - [В состав этой теории входят методы производства (сборки и установки) интеллектуальных компьютерных систем, а также спецификация используемых инструментальных средств.]
 - Разработка комплекса средств автоматизации производства интеллектуальных компьютерных систем
 - ∈ разработка комплекса средств автоматизации производства искусственных объектов заданного класса
 - ∈ вид человеческой деятельности

}

Специализированная инженерия в области Искусственного интеллекта

:= [проектирование и производство конкретной интеллектуальной компьютерной системы по заданной технологии]

⇒ обобщенная декомпозиция*:

- {
 - проектирование конкретных интеллектуальных компьютерных систем
 - ⇒ обобщенное разбиение*:
 - {
 - проектирование интеллектуальной компьютерной системы автоматизации проектирования соответствующего класса интеллектуальных компьютерных систем
 - проектирование интеллектуальной компьютерной системы автоматизации проектирования соответствующего класса объектов, не являющихся интеллектуальными компьютерными системами
 - проектирование интеллектуальной компьютерной системы, не являющейся системой автоматизации проектирования

}

- *производство конкретной спроектированной интеллектуальной компьютерной системы*

производство конкретной спроектированной интеллектуальной компьютерной системы

⇒ *примечание**:

[Производственная деятельность, направленная на *производство* (реализацию) спроектированной *интеллектуальной компьютерной системы* значительно уступает по уровню сложности деятельности по проектированию этой *интеллектуальной компьютерной системы*, так как это производство сводится к сборке результата *проектирования* (формальной логико-семантической модели разрабатываемой *интеллектуальной компьютерной системы*) и загрузки этой модели в память компьютера или программного *универсального интерпретатора логико-семантических моделей интеллектуальных компьютерных систем*, в качестве которого может быть использован

- либо специально разработанный для этого *компьютер*, ориентированные на обработку баз знаний и интерпретацию различных *интеллектуальных моделей решения задач*,
- либо программная эмуляция такого *компьютера*, реализованная на современных *компьютерах* фон-неймановской архитектуры.

Простота производства спроектированных систем характерна для производства не только интеллектуальных, но и любых других *компьютерных систем*.

Мы выделяем "производственный" этап реализации *интеллектуальных компьютерных систем* для того, чтобы по аналогии рассматривать этап производства (массового, мелкосерийного, разового производства) спроектированных искусственных объектов любого другого вида (микросхем, автомобилей, зданий, компьютеров).

Очевидно, что массовое производство некоторых видов продукции может иметь весьма большой уровень сложности, но при этом суть *производственной деятельности* как процесса перехода от проекта (спецификации) некоторого объекта к его реализации остаётся одной и той же независимо от уровня сложности реализуемого объекта (производимой продукции).]

*следует отличать**

- Э {
 - *специализированная интеллектуальная компьютерная система*
 - *интеллектуальная компьютерная система автоматизации проектирования интеллектуальных компьютерных систем*
 - *интеллектуальная компьютерная система автоматизации производства спроектированных интеллектуальных компьютерных систем*
- Э {
 - *человеческая деятельность*
 - ⊃ *научно-исследовательская деятельность*
 - ⊃ *научно-техническая деятельность*
 - *продукт человеческой деятельности*
 - ⊃ *продукт научно-исследовательской деятельности*
 - ⊃ *продукты научно-технической деятельности стакан*

Искусственный интеллект

⇒ *декомпозиция**:

Декомпозиция Искусственного интеллекта по направлениям

- = {
 - *Разработка теории представления знаний и технологии проектирования баз знаний актуальных компьютерных систем*
 - *Разработка теории решения задач и технологии проектирования решателей задач интеллектуальных компьютерных систем*

⇒ *декомпозиция**:

- {
 - *Разработка теории решения интерфейсных задач и технологии проектирования соответствующих решателей*

⇒ *часть**:

Разработка теории естественно языковых интерфейсов интеллектуальных компьютерных систем и технологии их проектирования

- *Разработка теории решения информационных задач в базах знаний интеллектуальных компьютерных систем и технологии проектирования соответствующих решателей*

```

    • Разработка теории решения поведенческих задач во внешней среде интеллектуальных компьютерных систем и технологии проектирования соответствующих решателей
  }
⇒ декомпозиция*:
  {
    • Разработка логических моделей решения задач и технологии проектирования соответствующих решателей
    • Разработка нейросетевых моделей решение задач и технологии проектирования соответствующих решателей
  }
  • Разработка универсальных интерпретаторов базовых моделей обработки баз знаний интеллектуальных компьютерных систем
  • Разработка общей теории человеческой деятельности, автоматизируемой с помощью комплекса взаимодействующих интеллектуальных компьютерных систем
}

} /* Завершили Сегмент "Анализ структуры Деятельности в области Искусственного интеллекта" */

/* Сегмент *****/
Анализ текущего состояния и проблем дальнейшего развития деятельности в области Искусственного интеллекта
⊃=
{
⇒ аннотация*:
  [Рассмотрим в каких направлениях должна происходить эволюция повышенного качества деятельности в области Искусственного интеллекта, а также эволюция продуктов этой деятельности]

  /*****/

Научно-исследовательская деятельность в области Искусственного интеллекта
⇒ проблемы текущего состояния*:
  {
    • [Отсутствует согласованность систем понятий в разных направлениях Искусственного интеллекта и, как следствие, отсутствует семантическая совместимость и конвергенция этих направлений, в результате чего ни о каком движении в направлении построения общей теории интеллектуальных систем с высоким уровнем формализации и речи быть не может. Существование и продолжающееся увеличение "высоты барьеров" между различными направлениями исследований в области Искусственного интеллекта проявляется в том, что специалист, работающий в рамках какого-либо направления Искусственного интеллекта, посещая заседания "не своей" секции на конференции по Искусственному интеллекту, мало что там может понять и, соответственно, извлечь полезного для себя.]

    • [Отсутствует мотивация и осознание острой необходимости в указанной конвергенции между различными направлениями Искусственного интеллекта.]

    • [Отсутствует реальное движение в направлении построения Общей теории интеллектуальных систем, поскольку отсутствует соответствующая мотивация и осознание острой практической необходимости в этом.]
  }

  /*****/

Разработка базовой комплексной технологии проектирования интеллектуальных компьютерных систем
⇒ текущее состояние*:
  [Современная технология Искусственного интеллекта представляет собой целое семейство всевозможных частных технологий, ориентированных на разработку и сопровождение различного вида компонентов интеллектуальных компьютерных систем, реализующих самые различные модели представления и обработки информации, различные модели решения задач, ориентированных на разработку различных классов интеллектуальных компьютерных систем.]

```

⇒ *проблемы текущего состояния**:

- { • [высокая трудоемкость разработки интеллектуальных компьютерных систем]
 - [необходимая высокая квалификация разработчиков]
 - [современные технологии *Искусственного интеллекта* принципиально не обеспечивают разработки таких *интеллектуальных компьютерных систем*, в которых устраняются недостатки современных *интеллектуальных компьютерных систем*]
 - [совместимость частных технологий *Искусственного интеллекта* практически отсутствует и, как следствие, отсутствует *семантическая совместимость* разрабатываемых *интеллектуальных компьютерных систем*, поэтому их системная интеграция осуществляется вручную.]
 - [Разрабатываемые *интеллектуальные компьютерные системы* не способны самостоятельно координировать свою деятельность друг с другом следовательно
 - нет общей комплексной технологии проектирования интеллектуальных компьютерных систем;
 - не обеспечивается совместимость и взаимодействие разрабатываемых систем (синтаксическая и семантическая совместимость);
 - нет совместимости между существующими частными технологиями проектирования различных компонентов интеллектуальных компьютерных систем (базы знаний, нейросетевые модели, интеллектуальные интерфейсы и т.д.);
 - есть инструментальные средства по разработке компонентов, но "склеивать" (соединять, интегрировать) разработанные компоненты надо вручную, то есть нет комплексных инструментальных средств, позволяющих разрабатывать интеллектуальные системы в целом
-]
- }

/*****/

Разработка технологии производства спроектированных интеллектуальных компьютерных систем

⇒ *текущее состояние**:

[Был сделан целый ряд попыток разработки *компьютеров* нового поколения, ориентированных на использование в *интеллектуальных компьютерных системах*. Но все они оказались неудачными, так как не были ориентированы на всё многообразие моделей решения задач в *интеллектуальных компьютерных системах*. В этом смысле они не были универсальными компьютерами для *интеллектуальных компьютерных систем*.]

⇒ *проблемы текущего состояния**:

- { • [Разрабатываемые *интеллектуальные компьютерные системы* могут использовать самые различные комбинации *моделей решения интеллектуальных задач* (логических моделей, соответствующих различного вида логикам, нейросетевых моделей различного вида, моделей целеполагания, синтеза планов, моделей управления сложными объектами, моделей понимания и синтеза текстов естественного языка и т.д.). Современные (традиционные, фон-неймановские) *компьютеры* не в состоянии достаточно производительно интерпретировать всё многообразие указанных моделей решения задач. При этом разработка специализированных *компьютеров*, ориентированных на интерпретацию какой-либо одной модели решения задач (нейросетевой модели или какой-либо логической модели) проблему не решает, так как в *интеллектуальной компьютерной системе* необходимо использовать сразу несколько разных моделей решения задач, причём в различных сочетаниях.]
- }

/*****/

Специализированная инженерия в области Искусственного интеллекта

:= [Деятельность, направленная на разработку *интеллектуальных компьютерных систем* различного назначения с использованием имеющихся для этого моделей, методов и средств]

:= [Деятельность по проектированию и производству *интеллектуальных компьютерных систем*]

:= [Деятельность, направленная на формирование рынка *интеллектуальных компьютерных систем*]

⇒ *в перспективе**:

Специализированная инженерия в области Искусственного интеллекта, осуществляемая специальной частью Экосистемы OSTIS

⇒ *продукт**:

Экосистема OSTIS

⇒ *субъект действия**:
часть Экосистемы *OSTIS*, осуществляющая специализированную инженерию в области Искусственного интеллекта

⇒ *проблемы текущего состояния**:

- { • [Отсутствует четкая систематизация многообразия интеллектуальных компьютерных систем, соответствующая систематизации автоматизируемых видов человеческой деятельности.]
- [Отсутствует конвергенция интеллектуальных компьютерных систем, обеспечивающих автоматизацию областей человеческой деятельности, принадлежащих одному и тому же виду человеческой деятельности.]
- [Отсутствует семантическая совместимость (семантическая унификация, взаимопонимание) между интеллектуальными компьютерными системами, основной причиной чего является отсутствие согласованной системы общих используемых понятий.]
- [Семантическая недружественность пользовательского интерфейса и отсутствие встроенной справочной системы, позволяющей запрашивать информацию об элементах интерфейса и возможностях системы, приводят к низкой эффективности эксплуатации всех возможностей интеллектуальной компьютерной системы.]
- [Анализ проблем автоматизации всех видов человеческой деятельности убеждает в том, что дальнейшая автоматизация человеческой деятельности требует не только повышения уровня интеллекта соответствующих интеллектуальных компьютерных систем, но и реализации их способности

- устанавливать свою семантическую совместимость (взаимопонимание) как с другими компьютерными системами, так и со своими пользователями;
- поддерживать эту семантическую совместимость в процессе собственной эволюции, а также эволюции пользователей и других компьютерных систем;
- координировать свою деятельность с пользователями и другими компьютерными системами при коллективном решении различных задач;
- участвовать в распределении работ (подзадач) при коллективном решении различных задач.

Важно подчеркнуть то, что реализация вышеперечисленных способностей создаст возможность для существенной и даже полной автоматизации системной интеграции компьютерных систем в комплексы взаимодействующих систем и автоматизации реинжиниринга таких комплексов. Такая автоматизация системной интеграции и её реинжиниринга:

- даст возможность комплексам кибернетических систем самостоятельно адаптироваться к решению новых задач;
- существенно повысит эффективность эксплуатации таких комплексов компьютерных систем, так как реинжиниринг системной интеграции компьютерных систем, входящих в такой комплекс, часто востребован (например, при реконструкции предприятия);
- существенно сокращает число ошибок по сравнению с "ручным" (неавтоматизированным) выполнением системной интеграции и её реинжиниринга, которые, к тому же, требуют высокой квалификации.

Таким образом следующий этап повышения уровня автоматизации человеческой деятельности настоятельно требует создания таких интеллектуальных компьютерных систем, которые могли бы легко сами (без системного интегратора) объединяться для совместного решения сложных задач.]

}

/*****/

Образовательная деятельность в области искусственного интеллекта

⇒ *текущее состояние**:

[Целенаправленная подготовка специалистов в области Искусственного интеллекта имеет богатую историю и осуществляется во многих ведущих университетах (Stanford University, MIT, МГУ (Москва), НИУ МЭИ (Москва), РГГУ (Москва), СПбГУ (Санкт-Петербург), ДВФУ (Владивосток), НГТУ (Новосибирск), НТУУ КПИ (Киев), БГУИР (Минск), БГУ (Минск), БрГТУ (Брест) и других).]

⇒ *проблемы текущего состояния**:

- { • [Поскольку деятельность в области Искусственного интеллекта сочетает в себе и высокую степень наукоемкости и высокую степень сложности инженерных работ, подготовка специалистов в этой области требует одновременного формирования у них как научно-исследовательских навыков, культуры и стиля мышления, так и инженерно-практических навыков, культуры и стиля мышления. С точки зрения методики и психологии обучения сочетание фундаментальной научной и инженерно-

практической подготовки специалистов является весьма сложной образовательной педагогической задачей.]

- [Отсутствует *семантическая совместимость* между различными учебными дисциплинами, что приводит к "мозаичности" восприятия информации]
- [Отсутствует системный подход к подготовке молодых специалистов в области *Искусственного интеллекта*]
- [Нет персонификации обучения, а также установки на выявление, раскрытие и развитие индивидуальных способностей]
- [Отсутствует целенаправленное формирование мотивации к творчеству]
- [Нет формирования навыков работы в реальных коллективах разработчиков]
- [Отсутствует адаптация к реальной практической деятельности]
- [Любая современная технология (в том числе и Технология OSTIS) должна иметь высокие темпы своего развития, поскольку без этого невозможно поддерживать высокий уровень её конкурентоспособности. Но для быстро развиваемой технологии требуется:
 - не просто высокая квалификация кадров, использующих и развивающих технологию,
 - но и высокие темпы повышения уровня этой квалификации, так как без этого невозможно эффективно использовать и развивать быстро меняющуюся технологию.

Из этого следует, что образовательная деятельность в области *Искусственного интеллекта* и соответствующая ей технология должна быть не просто важной частью деятельности в области *Искусственного интеллекта*, а частью, глубоко интегрированной во все остальные виды деятельности в области *Искусственного интеллекта*. Так, например, каждая *интеллектуальная компьютерная система* должна быть ориентирована не только на обслуживание своих конечных пользователей, не только на организацию целенаправленного взаимодействия со своими разработчиками, которые постоянно совершенствуют эту систему, и не только на обеспечение минимального "порога вхождения" для новых конечных пользователей и разработчиков, но и на организацию постоянного и персонифицированного повышения квалификации каждого своего конечного пользователя и разработчика в условиях постоянных изменений, вносимых в указанную *интеллектуальную компьютерную систему*. Для этого эксплуатируемая *интеллектуальная компьютерная система* должна "знать", что в ней изменилось, на что она способна и как эти способности инициировать (содержание и форма, соответствующих пользовательских команд)]

}
}

/*****

Бизнес-деятельность в области *Искусственного интеллекта*

⇒ *текущее состояние**:

[Острая потребность в существенном повышении уровня автоматизации в самых различных областях человеческой деятельности (в промышленности, медицине, транспорте, образовании, строительстве и во многих других), а также современные результаты в развитии *технологий Искусственного интеллекта* привели к существенному расширению работ по созданию *прикладных интеллектуальных компьютерных систем* и к появлению большого количества коммерческих организаций, ориентированных на разработку таких приложений.]

⇒ *проблемы текущего состояния**:

- { • [Не так просто обеспечить баланс тактических и стратегических направлений развития всех форм деятельности в области *Искусственного интеллекта* (научно-исследовательской деятельности, разработки технологии проектирования и производства интеллектуальных компьютерных систем, разработки прикладных систем, образовательной деятельности), а также баланс между всеми перечисленными формами деятельности.]
- [В настоящее время отсутствует глубокая конвергенция различных форм деятельности в области *Искусственного интеллекта* (в первую очередь, конвергенция развития технологий *Искусственного интеллекта* и разработки различных прикладных интеллектуальных компьютерных систем), что существенно затрудняет развитие каждой из этих форм.]
- [Высокий уровень наукоемкости работ в области *Искусственного интеллекта* предъявляет особые требования к квалификации сотрудников и к их способности работать в составе творческих коллективов.]
- [Для повышения квалификации своих сотрудников и для обеспечения высокого уровня своих разработок необходимо активное сотрудничество с различными научными школами, с кафедрами,

осуществляющими подготовку молодых специалистов в области *Искусственного интеллекта*, активное участие в подготовке и проведении соответствующих конференций, семинаров, выставок.]

}

/*****

Искусственный интеллект

⇒ *сверхзадачи текущего состояния**:

- {
 - [Построение и перманентное развитие *общей формальной теории интеллектуальных систем*]
 - ⇒ *подзадачи**:
 - {
 - [Уточнение требований, предъявляемых к интеллектуальным компьютерным системам – уточнение свойств интеллектуальных компьютерных систем, определяющих высокий уровень их интеллекта.]
 - [Конвергенция и интеграция всевозможных видов знаний и всевозможных моделей решения задач в рамках каждой интеллектуальной компьютерной системы.]
 - [Ориентация на последующую разработку унифицированных семантически совместимых формальных моделей интеллектуальных систем.]
 - [Ориентация на разработку различного вида универсальных интерпретаторов формальных моделей интеллектуальных систем (и в том числе компьютеров нового поколения) и обеспечение четкой стратификации между формальными моделями интеллектуальных систем и различными вариантами построения их интерпретаторов, обеспечивающей высокую степень независимости эволюции формальных моделей интеллектуальных систем и эволюции их интерпретаторов. Это требует особой детализации формальных моделей интеллектуальных систем.]
 - [Обеспечение коммуникационной ("социальной") совместимости (договороспособности) интеллектуальных компьютерных систем, позволяющей им самостоятельно формировать коллективы интеллектуальных компьютерных систем и их пользователей, а также самостоятельно согласовывать (координировать) деятельность в рамках этих коллективов при решении сложных задач в непредсказуемых условиях. Без этого невозможна реализация таких проектов, как "умный" дом, "умный" город, "умное" предприятие, "умная" больница и т.д.]
- [Создание и перманентное развитие *общей комплексной технологии* проектирования и производства *семантически совместимых интеллектуальных компьютерных систем*, способных координировать свою деятельность с себе подобными]
 - ⇒ *подзадачи**:
 - {
 - [Четкое описание стандарта интеллектуальных компьютерных систем, обеспечивающего семантическую совместимость разрабатываемых систем]
 - [Разработка мощных библиотек семантически совместимых и многократно (повторно) используемых компонентов разрабатываемых интеллектуальных компьютерных систем]
 - [Обеспечение низкого порога вхождения в технологию проектирования интеллектуальных компьютерных систем как для пользователей технологии (т.е. разработчиков прикладных или специализированных интеллектуальных компьютерных систем), так и для разработчиков самой технологии]
 - [Обеспечение высоких темпов развития технологии за счет учета опыта разработки различных приложений путем активного привлечения авторов приложений к участию в развитии (совершенствовании) технологии]
- [Разработка компьютеров нового поколения, ориентированных на производство высокопроизводительных *интеллектуальных компьютерных систем* самого различного назначения и высокого качества]
- [Создание глобальной *экосистемы* взаимодействующих между собой *интеллектуальных компьютерных систем*, обеспечивающих комплексную автоматизацию всех видов *человеческой деятельности*]
 - ⇒ *подзадача**:
 - [Построение формальной модели человеческой деятельности в контексте теории smart-общества]

- [Создание и перманентное развитие глобальной *социотехнической экосистемы*, которая состоит из *интеллектуальных компьютерных систем*, а также всех пользователей этих систем, которая обеспечивает комплексную автоматизацию всех *видов человеческой деятельности*]
- [Необходим переход от эклектичного построения сложных *интеллектуальных компьютерных систем*, использующих различные виды *знаний* и различные виды *моделей решения задач*, к их глубокой *интеграции* и унификации, когда одинаковые модели представления и модели обработки знаний реализуется в разных системах и подсистемах одинаково]
- [Необходимо сократить дистанцию между современным уровнем *теории интеллектуальных компьютерных систем* и практики их разработки.]

}

:= [Деятельность в области Искусственного интеллекта (как совокупность всех форм и направлений этой деятельности)]

⇒ *проблема текущего состояния**:

[Эпицентром современных проблем развития деятельности в области *Искусственного интеллекта* является *конвергенция* и *глубокая интеграция* всех форм, направлений и результатов этой деятельности. Уровень взаимосвязи, взаимодействия и *конвергенции* между различными формами и направлениями деятельности в области *Искусственного интеллекта* явно недостаточен. Это приводит к тому, что каждая из них развивается обособленно, независимо от других. Речь идет о *конвергенции* между такими направлениями *Искусственного интеллекта*, как представление знаний, решение интеллектуальных задач, интеллектуальное поведение, понимание и др., а также между такими формами *человеческой деятельности в области Искусственного интеллекта*, как научные исследования, разработка технологий, разработка приложений, образование, бизнес. Почему на фоне уже достаточно длительного интенсивного развития научных исследований в области *Искусственного интеллекта* до сих пор не создан рынок интеллектуальных компьютерных систем и комплексная технология *Искусственного интеллекта*, обеспечивающая разработку широкого спектра *интеллектуальных компьютерных систем* самого различного назначения и доступной широкому контингенту инженеров. Потому что сочетание высокого уровня наукоемкости и прагматизма этой проблемы требует для ее решения принципиально нового подхода к организации взаимодействия *ученых*, работающих в области *Искусственного интеллекта*, *разработчиков* средств автоматизации проектирования *интеллектуальных компьютерных систем*, *разработчиков* средств реализации интеллектуальных компьютерных систем, включая средства аппаратной поддержки интеллектуальных компьютерных систем, *разработчиков* прикладных интеллектуальных компьютерных систем. Такое *целенаправленное* взаимодействие должно осуществляться как в рамках каждой из этих форм деятельности в области *Искусственного интеллекта*, так и между ними. Таким образом, основной тенденцией дальнейшего развития теоретических и практических работ в области *Искусственного интеллекта* является конвергенция как самых разных видов (форм и направлений) человеческой деятельности в области *Искусственного интеллекта*, так и самых разных продуктов (результатов) этой деятельности. Необходимо ликвидировать барьеры между различными видами и продуктами деятельности в области *Искусственного интеллекта* в целях обеспечения их совместимости и интегрируемости. Проблема создания быстро развивающегося рынка семантически совместимых интеллектуальных систем – это вызов, адресованный специалистам в области *Искусственного интеллекта*, требующий преодоления "вавилонского столпотворения" во всех его проявлениях, формирование высокой культуры договороспособности и унифицированной, согласованной формы представления коллективно накапливаемых, совершенствуемых и используемых знаний. Ученые, работающие в области *Искусственного интеллекта*, должны обеспечить конвергенцию результатов различных направлений *Искусственного интеллекта* и построить:

- общую теорию интеллектуальных компьютерных систем;
- общую технологию проектирования семантически совместимых интеллектуальных компьютерных систем, включающую соответствующие стандарты интеллектуальных компьютерных систем и их компонентов. Инженеры, разрабатывающие интеллектуальные компьютерные системы, должны сотрудничать с учеными и участвовать в развитии технологии проектирования интеллектуальных компьютерных систем.

]

конвергенция в области Искусственного интеллекта

⇒ *разбиение**:

Направления конвергенции в области Искусственного интеллекта

∃ *конвергенция Искусственного интеллекта со смежными научными дисциплинами*

⇒ *примечание**:

{

Искусственный интеллект

⇔ *смежная дисциплина**:

- *Логика*

- Психология человека
- Зоопсихология
- Нейропсихология
- Этология
- Кибернетика
- Общая теория систем
- Семиотика
- Лингвистика

}

- Э *конвергенция различных направлений Искусственного интеллекта*
- := [Конвергенция различных направлений исследований в области Искусственного интеллекта, результатом которой должна быть формализованная практически ориентированная общая теория интеллектуальных систем и, в частности, интеллектуальных компьютерных систем]
- := [Конвергенция между различными направлениями и продуктами научных исследований в области искусственного интеллекта, результатом (целевым продуктом) которой должна стать общая формальная теория интеллектуальных компьютерных систем]
- ⇒ *примечание**:
[Разобобщенность различных направлений исследований в области искусственного интеллекта является главным препятствием создания общей комплексной технологии проектирования интеллектуальных компьютерных систем]
- Э *конвергенция различного вида знаний в памяти интеллектуальной компьютерной системы*
- := [Конвергенция и интеграция внутреннего представления в памяти интеллектуальной компьютерной системы различного вида знаний]
- Э *конвергенция различных моделей решения задач в памяти интеллектуальной компьютерной системы*
- := [Конвергенция и интеграция различных моделей решения задач, которая включает логико-семантическую типологию задач и типологию моделей решения задач и требует уточнения семантики таких понятий как задача, класс задач, метод, класс методов, модель решения задач (иерархический метод интерпретации класса методов)]
- Э *конвергенция интеллектуальных компьютерных систем*
- := [Обеспечение семантической совместимости (взаимопонимания) интеллектуальных систем, согласование используемых онтологий]
- := [Конвергенция между различными прикладными компьютерными системами, результатом (целевым продуктом) которой должна стать экосистема, состоящая из перманентно эволюционирующих, семантически совместимых и взаимодействующих интеллектуальных компьютерных систем, а также их пользователей]
- ⇒ *пояснение**:
[Конвергенция (семантическая совместимость) всех разрабатываемых интеллектуальных компьютерных систем (в том числе прикладных), преобразующая набор индивидуальных (самостоятельных) интеллектуальных компьютерных систем различного назначения в коллектив активно взаимодействующих интеллектуальных компьютерных систем для совместного (коллективного) решения сложных (комплексных) задач и для перманентной поддержки семантической совместимости в ходе индивидуальной эволюции каждой интеллектуальной компьютерной системы.]
- Э *конвергенция средств автоматизации проектирования различного вида компонентов интеллектуальных компьютерных систем*
- := [Конвергенция (семантическая совместимость) средств автоматизации проектирования различного вида компонентов интеллектуальных компьютерных систем, результатом которой должен быть общий комплекс средств автоматизации проектирования всех компонентов интеллектуальных компьютерных систем]
- := [Конвергенция между инструментальными средствами, обеспечивающими автоматизацию проектирования различных компонентов или различных классов интеллектуальных компьютерных систем, результатом (целевым продуктом) которой должен стать единый комплекс методологических и инструментальных средств, ориентированный на поддержку комплексного проектирования любых интеллектуальных компьютерных систем]
- Э *конвергенция логико-семантических моделей интеллектуальных компьютерных систем*
- ⇒ *примечание**:
[логико-семантические модели интеллектуальных компьютерных систем являются результатом ("сухим" остатком) проектирования этих систем и представляют собой формальное

	представления исходного (начального) состояния <i>баз знаний</i> разрабатываемых <i>интеллектуальных компьютерных систем</i>]
Э	<p><i>конвергенция средств интерпретации логико-семантических моделей разрабатываемых интеллектуальных компьютерных систем</i></p> <p>⇒ <i>пояснение*</i>:</p> <p>[Конвергенция (совместимость) средств реализации (производства) интеллектуальных компьютерных систем на основе спроектированных формальных моделей создаваемых интеллектуальных компьютерных систем (средств интерпретации спроектированных моделей интеллектуальных компьютерных систем). Такая интерпретация может осуществляться либо программным путем на современных компьютерах, либо путем создания принципиально новых компьютеров, специально ориентированных на интерпретацию формальных моделей интеллектуальных компьютерных систем, помещаемых в память указанных компьютеров]</p>
Э	<p><i>конвергенция между информационно-программным и аппаратным обеспечением интеллектуальных компьютерных систем</i></p> <p>:= [Конвергенция между Software и Hardware интеллектуальных компьютерных систем]</p>
Э	<p><i>Конвергенция различных форм деятельности в области Искусственного интеллекта</i></p> <p>:= <i>пояснение*</i>:</p> <p>[Конвергенция между</p> <ul style="list-style-type: none"> □ научными исследованиями по созданию общей теории интеллектуальных компьютерных систем; □ разработкой средств автоматизации проектирования интеллектуальных компьютерных систем; □ разработкой средств интерпретации спроектированных формальных моделей интеллектуальных компьютерных систем; □ разработкой прикладных интеллектуальных компьютерных систем различного назначения; □ подготовкой и перманентным повышением квалификации кадров, способных эффективно участвовать во всех перечисленных направлениях деятельности. <p>]</p> <p>:= [Конвергенция между</p> <ul style="list-style-type: none"> □ научно-исследовательской деятельностью в области искусственного интеллекта; □ инженерно-технологической деятельностью, которая направлена на разработку комплексной технологии проектирования интеллектуальных компьютерных систем и которая имеет высокий уровень наукоемкости; □ инженерно-прикладной деятельностью, которая направлена на разработку прикладных интеллектуальных систем и которая также имеет высокий уровень наукоемкости, обусловленной необходимостью качественной формализации соответствующих предметных областей и, в частности, методов решения задач в этих областях; □ образованием (образовательной деятельностью) в области искусственного интеллекта, повышение эффективности которого настоятельно требует раннего и поэтапного вовлечения студентов в реальные, а не учебные проекты – сначала в инженерно-прикладные, потом в инженерно- исследовательские проекты; □ деятельностью, направленной на создание инфраструктуры, обеспечивающей поддержку открытого массового активного международного сотрудничества по консолидации усилий, направленных на решение современных проблем в области искусственного интеллекта; □ бизнесом в области искусственного интеллекта, который не просто должен обеспечить финансовую поддержку перечисленных видов деятельности, но и обеспечить грамотный баланс между ними, грамотное сочетание тактических и стратегических целей <p>]</p> <p>⇒ <i>примечание*</i>:</p> <p>[Глубокая конвергенция между всеми этими формами деятельности возможна только тогда, когда <u>каждый</u> участник создания комплексной технологии искусственного интеллекта является участником <u>каждой</u> из перечисленных форм деятельности.]</p>

Искусственный интеллект

⇒ **методологические проблемы текущего состояния***:

- { • [Далеко не всеми учеными, работающими в области искусственного интеллекта принимается прагматичность практической направленности этой науки]

- [Не всеми принимается необходимость конвергенции различных направлений искусственного интеллекта и необходимость их интеграции в целях построения общей теории интеллектуальных систем]
- [Нет движения к построению общей компьютерной технологии интеллектуальных компьютерных систем]
- [Нет движения к построению экосистем интеллектуальных компьютерных систем]
- [Не всеми принимается необходимость конвергенции различных форм деятельности в области Искусственного интеллекта]

⇒ примечание*:

[Современная трактовка целей и задач *Искусственного интеллекта* как научно-технической дисциплины требует переосмысления, так как, к сожалению, носит несогласованный, а часто и значительно более узкий характер, чем этого требует текущее положение]

/*****/

следует отличать*

- Э {
- *конвергенция*
:= [Процесс сближения структурных и/или функциональных характеристик нескольких (как минимум двух) заданных сущностей]
:= [Процесс конвергенции заданных сущностей в ходе их изменения, совершенствование, эволюции]
 - С *процесс*
 - *конвергенция*[^]
:= [Степень близости (сходство) заданных сущностей]
∈ *свойство*
- }

конвергенция

⇒ примечание*:

[*Конвергенция* пар конкретных искусственных сущностей (например, технических систем) есть стремление их унификацию (в частности, к стандартизации), т.е. стремление к минимизации многообразия форм решения аналогичных практических задач – стремление к тому, чтобы все, что можно сделать одинаково, сделалось одинаково, но без ущерба требуемого качества. Последнее очень важно, так как безграмотная стандартизация может привести к существенному торможению прогресса. Ограничение многообразия форм не должно приводить к ограничению содержания, возможностей. Образно говоря, "словам должно быть тесно, а мыслям – свободно".]

⇒ примечание*:

[Методологически конвергенция искусственно создаваемых сущностей (артефактов) сводится (1) к выявлению (обнаружению) принципиальных сходств между этими сущностями, которые часто весьма закамуфлированы и их трудно "увидеть и (2) к реализации обнаруженных сходств одинаковым образом (в одинаковой форме, в одинаковом "синтаксисе"). Образно говоря, от "семантической" (смысловой) эквивалентности требуется перейти и к "синтаксической" эквивалентности. Кстати, в этом как раз и заключается суть (идея) смыслового представления информации (знаний), целью которого является создание такой языковой среды (*смыслового пространства*), в рамках которого (1) семантически эквивалентные информационные конструкции полностью совпадали, а (2) конвергенция информационных конструкций сводилась бы к выявлению изоморфных фрагментов этих конструкций.]

⇒ примечание*:

[Очень важно уточнить, формализовать понятие конвергенции (конвергенции знаний, методов, модели решения задач, конвергенции интеллектуальных компьютерных систем в целом)]

⊃ *конвергенция информационных конструкций*

:= [конвергенция синтаксических и семантических свойств информационных конструкций]

⊃ *конвергенция языков*

⊃ *конвергенция научных дисциплин*

:= [конвергенция различных научных дисциплин или различных направлений одной и той же и дисциплины]

⊃ *конвергенция баз знаний*

⊃ *конвергенция моделей решения задач*

⊃ *конвергенция гибридных решателей задач*

⊃ *конвергенция кибернетических систем*

- ⊃ *конвергенция интеллектуальных систем*
- ⊃ *конвергенция интеллектуальных систем, направленная на обеспечение их семантической совместимости*

конвергенция результатов научно-технической деятельности

⇒ *примечание**:

[Важным препятствием для конвергенции результатов научно-технической деятельности является сформировавшийся в науке и технике акцент на выявлении не сходств, а отличий. Чтобы убедиться в этом достаточно обратить внимание на то, что уровень научных результатов оценивается научной новизной, которая может имитироваться новизной не по существу, а по форме представления (например, с помощью новых понятий или даже новых терминов). Результаты в технике, например, в патентах также оцениваются отличиями от предшествующих технических решений. Но для конвергенции нужны другие акценты – ни поиск отличий, а выявление неочевидных сходств и превращения их в очевидные сходства, представленные в одинаковой форме.]

совместимость[^]

:= [совместимость заданных двух или более сущностей[^]]

:= [простота интеграции заданной группы сущностей[^]]

:= [интегрируемость[^]]

⇒ *примечание**:

[Степень (уровень) совместимости заданных сущностей может рассматриваться как оценка результата их конвергенции. Чем качественнее (основательнее, глубже) проведена конвергенция заданных сущностей, тем выше уровень их совместимости и, собственно, тем легче их интегрировать.]

- ⊃ *совместимость информационных конструкций[^]*
- ⊃ *семантическая совместимость информационных конструкций[^]*
- ⊃ *совместимость языков[^]*
- ⊃ *семантическая совместимость языков[^]*
- ⊃ *семантическая совместимость научных дисциплин[^]*
- ⊃ *совместимость баз знаний[^]*
- ⊃ *совместимость моделей решения задач[^]*
- ⊃ *совместимость кибернетических систем[^]*
- ⊃ *семантическая совместимость кибернетических систем[^]*
- ⊃ *семантическая совместимость[^]*

интеграция*

:= [объединение нескольких разных сущностей, в результате чего возникает некоторая объединённая целостная сущность*]

⊃ *эkleктичная интеграция**

:= [Интеграция разнородных (гетерогенных) сущностей, которой не предшествует конвергенция (сближение) этих сущностей*]

⊃ *глубокая интеграция**

⇒ *примечание**:

[Понятие *интеграции** и особенно понятие *глубокой интеграции** имеет тесную связь с понятием *конвергенции[^]*. Чем выше степень конвергенции (степень сближения) интегрируемых объектов, тем выше качество результата интеграции. Особенно, если речь идёт о глубокой интеграции.]

глубокая интеграция*

:= ["бесшовная" интеграция*]

:= [интеграция однородных сущностей, предполагающая глубокую взаимную "диффузию" (сращивание) соединяемых сущностей, которая не обязательно должна осуществляться физически]

⇒ *примечание**:

[Примером виртуальной глубокой интеграции является формирование коллектива семантически совместимых индивидуальных кибернетических систем]

:= [бесшовная интеграция*]

:= [гибридизация*]

:= [интеграция, результатом которой являются гибридные объекты*]

:= [интеграция, которой предшествует высокий уровень конвергенции интегрируемых объектов*]

:= [(конвергенция + интеграция)*]

:= ["бесшовная" интеграция]

- := [интеграция, в результате которой возникает гибридная система*]
- := [интеграция, которой предшествует конвергенция (в частности, унификация) интегрируемых систем, приведение этих систем к максимально похожему виду (общему знаменателю)*]
- := [интеграция с "диффузией" , взаимопроникновением на основе унификации того, что можно сделать одинаковым*]

интеграция*

- ⊃ *интеграция информационных конструкций*
- ⊃ *интеграция языков*
- ⊃ *интеграция научных дисциплин*
- ⊃ *интеграция баз знаний*
- ⊃ *интеграция моделей решения задач*
- ⊃ *интеграции индивидуальных кибернетических систем*
 - ⊃ *слияние индивидуальных кибернетических систем*
 - := [преобразование нескольких искусственных индивидуальных кибернетических систем в интегрированную индивидуальную кибернетическую систему, которая способна решать все задачи, каждая из которых могла бы быть решена в рамках какой-либо из интегрируемых систем]
 - ⊃ *формирование коллектива индивидуальных кибернетических систем*
 - := [формирования многоагентной системы, состоящей из индивидуальных кибернетических систем]
- ⇒ *примечание**:
[Эффективность интеграции индивидуальных кибернетических систем определяется тем, насколько объем задач, решаемых коллективом индивидуальных кибернетических систем, превысит объединенные объемы задач, решаемых членами коллектива в отдельности.]

} /* Завершили Сегмент “Анализ текущего состояния и проблем дальнейшего развития деятельности в области Искусственного интеллекта” */

/* Сегмент *****/

Уточнение Понятия Технологии OSTIS

⊃=
{

Технология OSTIS

- := [Комплекс (семейство) технологий, обеспечивающих проектирование, производство, эксплуатацию и реинжиниринг интеллектуальных *компьютерных систем (ostis-систем)*, предназначенных для автоматизации самых различных видов человеческой деятельности и в основе которых лежит смысловое представление и онтологическая систематизация знаний, а также агентно-ориентированная обработка знаний]
- := [Open Semantic Technology for Intelligent Systems]
 - ⇒ *сокращение**:
[OSTIS]
- := [Семейство (комплекс) *ostis-технологий*]
- := [Комплексная открытая семантическая технология проектирования, производства, эксплуатации и реинжиниринга гибридных, семантически совместимых, активных и договороспособных *интеллектуальных компьютерных систем*]
- ⇒ *принципы, лежащие в основе**:
 - { • [Ориентация на разработку *интеллектуальных компьютерных систем*, имеющих высокий уровень интеллекта и, в частности, высокий уровень *социализации*. Указанные системы, разработанные по *Технологии OSTIS*, будем называть *ostis-системами*]
 - [Ориентация на комплексную автоматизацию всех видов и областей *человеческой деятельности* путем создания сети взаимодействующих и координирующих свою деятельность *ostis-систем*. Указанную сеть *ostis-систем* вместе с их пользователями будем называть *Экосистемой OSTIS*]
 - [Поддержка перманентной эволюции *ostis-систем* в ходе их эксплуатации.]
 - [*Технология OSTIS* реализуется в виде сети *ostis-систем*, которая является частью *Экосистемы OSTIS*. Ключевой *ostis-системой* указанной сети является *Мемасистема IMS.ostis* (Intelligent MetaSystem for ostis-systems), реализующая *Ядро Технологии OSTIS*, которое включает в себя базовые (предметно независимые) методы и средства проектирования и производства *ostis-систем* с

интеграцией в их состав типовых встроенных подсистем поддержки эксплуатации и реинжиниринга *ostis-систем*. Остальные *ostis-системы*, входящие в состав рассматриваемой сети, реализуют различные специализированные *ostis-технологии* проектирования различных классов *ostis-систем*, обеспечивающих автоматизацию любых областей и видов человеческой деятельности, кроме проектирования *ostis-систем*.]

- [Конвергенция и интеграция на основе *смыслового представления знаний* всевозможных научных направлений *Искусственного интеллекта* (в частности, всевозможных базовых знаний и навыков решения *интеллектуальных задач*) в рамках *Общей формальной семантической теории ostis-систем*.]
- [Ориентация на разработку компьютеров нового поколения, обеспечивающих эффективную (в том числе производительную) интерпретацию логико-семантических моделей *ostis-систем*, которые представлены *базами знаний* этих систем, имеющими *смысловое представление*.]

}

}

/*****/

Понятие *ostis-системы*

⊃=

{

ostis-система

:= [интеллектуальная компьютерная система, спроектированная и реализованная по требованиям и стандартам *Технологии OSTIS*, которые задокументированы в *Общей теории ostis-систем*]

:= [Множество *ostis-систем* различного назначения]

∈ имя собственное

:= [Множество всевозможных *интеллектуальных компьютерных систем*, построенных по *Технологии OSTIS*]

⊂ интеллектуальная компьютерная система

:= [интеллектуальная компьютерная система, которая построена в соответствии с требованиями и стандартами *Технологии OSTIS*, что обеспечивает существенное развитие целого ряда *свойств* (способностей) этой *компьютерной системы*, позволяющих значительно повысить *уровень интеллекта* этой системы (и, прежде всего, ее *уровень обучаемости* и *уровень социализации*)]

⇒ разбиение*:

{ • индивидуальная *ostis-система*

• коллективная *ostis-система*

⇒ разбиение*:

{ • простой коллектив *ostis-систем*

• иерархический коллектив *ostis-систем*

}

}

⇒ примечание*:

[Когда речь идет о таком компоненте *Технологии OSTIS*, как *Общая теория ostis-систем*, имеется в виду строгое формальное уточнение того, как устроена *ostis-система*, какова ее архитектура, принципы организации памяти, принципы организации представления и обработки информации, принципы организации интерфейса с внешней средой (в том числе, с пользователями)]

⇒ принципы, лежащие в основе*:

{ • [Хранение информации в памяти *ostis-системы* ориентируется на *смысловое представление информации* – без синонимии и омонимии знаков, без семантической эквивалентности информационных конструкций, т.е. без дублирования информации.]

⇒ ключевой знак*:

смысловое представление информации

:= [смысл представленной информационной конструкции]

⇒ принципы, лежащие в основе*:

{ • [В рамках смыслового представления информационной конструкции все *знаки*, входящие в эту *информационную конструкцию* уникальны, т.е. обозначают *разные* описываемые *сущности*. Другими словами, в рамках *смыслового представления информационной конструкции* запрещено присутствие *синонимичных знаков*.]

- [В рамках *смыслового представления информационной конструкции* все сущности, описываемые этой *информационной конструкцией*, должны быть явно представлены своим знаком.]
 - [Каждый знак, входящий в *смысловое представление информационной конструкции* является *синтаксически элементарным* (атомарным) *фрагментом* этой конструкции, "внутренняя" структура которого несущественна (существенен только алфавит таких фрагментов).]
 - [Поскольку любая описываемая *сущность* может быть связана неограниченным числом *связей* с другими *сущностями* (при этом указанные связи также считаются описываемыми сущностями), *смысловое представление информационной конструкции* является *графоподобной конструкцией*]
 - [Интеграция (объединение, соединение) *информационной конструкции*, представленных в смысловой форме сводится к *склеиванию* (отождествлению) *синонимичных знаков*.]
 - [Смысл представленной информации содержится не в самих *знаках*, а в конфигурации *связей* между ними, которая отражает (является информационной моделью) описываемой конфигурации связей между описываемыми *сущностями*. Суть смыслового представления информационной конструкции заключается в том, что конфигурация *связей* между *знаками*, входящими в эту *информационную конструкцию*, становится *изоморфной* конфигурации *связей* между описываемыми *сущностями*, которые обозначаются этими *знаками*.]
 - [Способ (язык) *смыслового представления информации*, должен быть универсальным, т.е. должно быть обеспечено описание (и, прежде всего, обозначение) любых *связей* между описываемыми сущностями. При этом, если описываемые *связи* считать одним из видов описываемых *сущностей*, то можно описывать *связи* между *связями*, *связи*, связывающие *связи* с описываемыми *сущностями* иных видов.]
- }
- [Абстрактная память *ostis-системы* является графодинамической (т.е. нелинейной (графовой) и структурно перестраиваемой). Переработка информации в памяти *ostis-системы* сводится не столько к изменению состояния элементов памяти (это происходит только при изменении синтаксического типа элементов и при изменении содержимого тех элементов, которые обозначают файлы), сколько к изменению конфигурации связей между ними.]
 - [Ориентация на компьютеры нового поколения.]
 - [В основе организации решения задач в памяти *ostis-системы* лежит *агентно-ориентированная модель обработки информации*, управляемая ситуациями и событиями, возникающими в обрабатываемой информации (точнее, в обрабатываемой *базе знаний*). С точки зрения архитектуры *ostis-система* представляет собой *иерархическую* многоагентную систему с общедоступной памятью (т.е. с памятью, общедоступной всем агентам *ostis-системы*).
Заметим при этом, что общая память большинства исследуемых в настоящее время *многоагентных систем* является не общедоступной, а распределенной, т.е. представляет собой абстрактное (виртуальное) объединение, в состав которого входит память каждого агента многоагентной системы. Координация деятельности агентов *ostis-системы* при выполнении сложных *действий в памяти ostis-системы* реализуется также через *память ostis-системы* с помощью хранимых в памяти *методов* решения различных *классов задач*, а также с помощью хранимых в памяти *планов* решения конкретных задач.
На основании этого можно строить неограниченную иерархическую систему *агентов ostis-системы* – от элементарных агентов, обеспечивающих выполнение базовых действий в памяти *ostis-системы*, до неэлементарных агентов, представляющих собой коллективы (группы) элементарных и/или неэлементарных агентов, обеспечивающих решение различных типовых задач с помощью соответствующих методов и планов.]
 - [Реализация децентрализованного ситуационного управления деятельностью *ostis-систем* не только на уровне внутренних информационных процессов, но также на уровне организации индивидуальной деятельности во внешней среде и даже на уровне участия в коллективной деятельности в рамках различных коллективов *ostis-систем*. Организация выполнения *ostis-системой действий во внешней среде* осуществляется следующим образом:
 - Выделяются классы *элементарных действий во внешней среде*, для реализации каждого из которых вводятся *эффекторные агенты ostis-системы*.

- Координация деятельности *эффекторных агентов ostis-системы* при выполнении *сложных действий во внешней среде* осуществляется через *память ostis-системы* с помощью хранимых в памяти *методов и планов* решения различных задач *во внешней среде*, а также с помощью *рецепторных агентов ostis-системы*, обеспечивающих обратную связь и, соответственно, сенсомоторную координацию.

]

- [Унификация базового набора (базовой системы) используемых *понятий*, что является основой обеспечения *семантической совместимости* всех *ostis-систем*.]
- [В основе структуризации информации (*базы знаний*), хранимой в памяти *ostis-системы*, лежит иерархическая система *предметных областей* и соответствующих им *формальных онтологий*.]
- [Переход от исследования обработки данных (data science) к исследованию обработки знаний (knowledge science), что предполагает при разработке различных классов задач учет семантики обрабатываемой информации. В этом смысле традиционное программирование "хромеет на одну ногу"]
- [Способность к пониманию (к семантическому погружению, к семантической интеграции) новых приобретаемых знаний (и, в том числе, новых навыков) в состав текущего состояния *базы знаний*.]
- [Способность к *семантической конвергенции* (к обнаружению сходств) новых приобретаемых знаний (и, в частности, навыков) со знаниями, входящими в состав текущего состояния *базы знаний ostis-системы*.]
- [Способность к интеграции различных видов *знаний*.]
- [Способность к интеграции различных *моделей решения задач*.]
- [Способность *ostis-систем* понимать друг друга, а также любого своего пользователя путем согласования системы используемых понятий (по терминам и по денотационной семантике). Способность *ostis-системы* обеспечивать и поддерживать высокий уровень своей *семантической совместимости* (высокий уровень взаимопонимания) с другими *ostis-системами* в процессе собственной эволюции, а также эволюции других *ostis-систем*, которая приводит к расширению и/или корректировке системы используемых *понятий*.]
- [Способность *ostis-системы* согласовывать, координировать свою деятельность с другими системами при решении задач, которые усилиями одной (индивидуальной) интеллектуальной компьютерной системы не могут быть решены либо принципиально, либо за разумное время.]
- [Высокая степень индивидуальной обучаемости *ostis-систем*, обеспечиваемая высокой степенью их гибкости, стратифицированности, рефлексивности, а также универсальностью средств представления и образования *знаний*.]
- [Высокая степень коллективной обучаемости *ostis-систем*, обеспечиваемая высокой степенью их *семантической совместимости*.]

}

⇒ следовательно*:

[Перечисленные свойства *ostis-систем* свидетельствуют о том, что они имеют существенно более высокий *уровень интеллекта* и, в частности, более высокий *уровень социализации* по сравнению с современными *интеллектуальными компьютерными системами*.]

{ • *память**

• *ostis-система*

}

⇒ сужение второго домена заданного отношения для заданного первого домена*:

память ostis-системы

С *смысловая память*

{ • *информация, хранимая в памяти кибернетической системы**

• *ostis-система*

}

⇒ сужение второго домена заданного отношения для заданного первого домена*:

база знаний ostis-системы

С *смысловое представление информации*

решатель задач *ostis-системы*

С *агентно-ориентированная модель обработки информации в памяти*

смысловое представление информации⇒ *принципы, лежащие в основе**:

- { • [Каждый синтаксически элементарный (атомарный) фрагмент представленной информации является обозначением некоторой сущности, которая может быть реальной или абстрактной, конкретной (фиксированной, константной) или произвольной (переменной), постоянной или временной, четкой (достоверной) или нечеткой (недостоверной с возможным дополнительным уточнением степени правдоподобности).]

⇒ *следовательно**:

[В состав смыслового представления информации не могут входить буквы (не являются обозначениями сущностей), слова, словосочетания (не являются элементарными фрагментами), разделители, ограничители (не являются обозначениями сущностей)]

- [В рамках смыслового представления информации отсутствует синонимия (пары синонимичных знаков), омонимия (омонимичные знаки), семантическая эквивалентность (пары семантически эквивалентных информационных конструкций), т.е. отсутствует любая форма дублирования информации, а также отсутствует неоднозначность соотношения между знаками и их денотатами.]

}

⇒ *следовательно**:

[Смысловое представление информации не может выглядеть как цепочка (строка, последовательность) синтаксически элементарных фрагментов, поскольку каждая описываемая сущность и взаимно однозначно соответствующий ей ее знак может быть связана не с двумя, а с любым количеством описываемых сущностей. Другими словами, смысловое представление информации является нелинейной (графовой) информационной конструкцией.]

⇒ *следовательно**:

[Если внутреннее представление информации в памяти компьютерной системы является смысловым представлением, то обработка информации в такой памяти носит графодинамический характер и сводится не к изменению состояния элементов памяти, а к изменению конфигурации связей между ними.]

⇒ *примечание**:

[Ключевая проблема современного этапа развития общей теории интеллектуальных компьютерных систем и технологии их разработки – это проблема обеспечения **семантической совместимости**

- различных видов знаний, входящих в состав баз знаний интеллектуальных компьютерных систем;
- различных видов моделей решателей задач;
- различных интеллектуальных компьютерных систем в целом;

Для решения этой проблемы очевидно необходима унификация (стандартизация) формы представления знаний в памяти интеллектуальных компьютерных систем. Предлагаемым нами подходом для такой унификации и является ориентация на **смысловое представление информации** (знаний) в памяти интеллектуальных компьютерных систем. Основой предполагаемого нами подхода к обеспечению высокого уровня обучаемости и семантической совместимости интеллектуальных компьютерных систем, а также к разработке стандарта интеллектуальных компьютерных систем является унификация **смыслового представления информации** (знаний) в памяти интеллектуальных компьютерных систем и построение глобального **смыслового пространства** знаний.]

⇒ *примечание**:

[Информация в знаковой конструкции в основном содержится не в самих знаках (в их структуре), а в связях между знаками. При этом существенно, чтобы эти связи (синтаксические связи) имели четкую смысловую (семантическую) интерпретацию. Если структура знаков содержит информацию об обозначаемой сущности всегда можно заменить на "бесструктурные" знаки, которые имеют семантическую окрестность]

семантическая сетьС *смысловое представление информации*⇒ *пояснение**:

[Семантическая сеть нами рассматривается не как красивая метафора сложноструктурированных знаковых конструкций, а как формальное уточнение понятия смыслового представления информации, как принцип представления информации, лежащей в основе нового поколения компьютерных языков и самих компьютерных систем – графовых языков и графовых компьютеров.]

С *знаковая конструкция*⇒ *пояснение**:

[Семантическая сеть – это знаковая конструкция, обладающая следующими свойствами:

- "внутреннюю" структуру (строение) знаков, входящих в семантическую сеть не требуется учитывать при ее семантическом анализе (понимании)
- Смысл семантической сети определяется денотационной семантикой всех входящих в нее знаков и конфигурацией связей инцидентности этих знаков
- Из двух инцидентных знаков, входящих в семантическую сеть, один является знаком связи
- Отсутствие синонимии, омонимии

]

⇒ предлагаемый подход*:

SC-код

:= [Предлагаемое в рамках *Технологии OSTIS* уточнение понятия *семантической сети*]

С *семантическая сеть*

:= [Semantic Computer Code]

⇒ *смотрите**:

Описание внутреннего языка ostis-систем

многоагентная система

С *кибернетическая система*

⇒ *пояснение**:

[Кибернетическая система, представляющая собой множество кибернетических систем, способных коммуницировать, т.е. обмениваться информацией друг с другом (причем не обязательно каждый с каждым)]

агент*

:= [агент многоагентной системы*]

внешняя среда*

:= [внешняя среда кибернетической системы]

память*

:= [внутренняя (информационная) среда кибернетической системы]

⇒ *примечание**:

[Не каждая кибернетическая система (в том числе многоагентная система) имеет явно выделенную память, являющуюся хранилищем накапливаемой информации, накапливаемого опыта.]

многоагентная система

⇒ *разбиение**:

- *многоагентная система без общей памяти*
- *многоагентная система с общей памятью*

⇒ *разбиение**:

- *многоагентная система, в которой управление агентами осуществляется только путем обмена сообщениями между ними*
- *многоагентная система, в которой управление агентами осуществляется через общую для них память*

⇒ *разбиение**:

- *многоагентная система с централизованным управлением агентами*
- *многоагентная система с децентрализованным управлением агентами*

⇒ *разбиение**:

- *многоагентная система, в которой областью деятельности всех ее агентов является только внешняя среда этой системы*
- *многоагентная система, в которой областью деятельности ее агентов является как внешняя среда, так и память этой системы*

⇒ *примечание**:

[некоторые агенты такой системы могут работать только в памяти]

}

агентно-ориентированная модель обработки информации в памяти

:= [агентно-ориентированная модель решения задач]

:= [агентно-ориентированная архитектура решателя задач, представляющая собой многоагентную систему, в которой управление ее агентами осуществляется общей для них памятью и областью деятельности агентов является та же самая общая для них память]

⇒ *следовательно**:

[условием инициирования каждого указанного агента является возникновение в указанной памяти соответствующего вида ситуации или события]

⇐ *пересечение**:

- { • *многоагентная система, в которой управление агентами осуществляется через общую для них память*
- *многоагентная система с децентрализованным управлением агентами*
- *многоагентная система, в которой областью деятельности ее агентов является как внешняя среда, так и память этой системы*

агентно-ориентированная модель обработки информации в памяти

⇒ *принципы, лежащие в основе**:

- { • [Распределение целенаправленной деятельности между агентами, выполняющими различные действия в памяти, осуществляется на основе генерируемой в *базе знаний* иерархической системы, описывающей связь (сведение) инициированных целей (задач) с подцелями (подзадачами).]
- [Условием инициирования агента является появление в базе знаний формулировки той цели (задачи), которая, во-первых, инициирована, а, во-вторых, либо может быть полностью достигнута (решена) этим агентом, либо может быть этим агентом достигнута (решена) частично.]
- [В результате частичного достижения (решения) некоторой цели (задачи) агент может сгенерировать новые подцели (подзадачи).]
- [Таким образом, условием инициирования агента обработки информации (базы знаний) является появление соответствующей этому агенту ситуации или соответствия.]

⇒ *предлагаемый подход**:

абстрактная sc-машина

:= [Предлагаемое в рамках *Технологии OSTIS* уточнение понятия агентно-ориентированной модели обработки информации в памяти]

⊃ *абстрактная sc-машина*

⇒ *примечание**:

[Децентрализованное (агентно-ориентированное) управление процессом решения задач в *ostis-системах* реализуется как на внутреннем уровне (на уровне решателя задач *ostis-системы*), так и на внешнем уровне (на уровне взаимодействия между *ostis-системами*)]

стандартизация *ostis-систем*

:= [унификация *ostis-систем*]

⇒ *пояснение**:

[Стандартизация *ostis-систем* включает в себя:

- стандартизацию языка внутреннего представления информации в памяти *ostis-систем*;
- стандартизацию принципов децентрализованного управления обработкой информации в памяти *ostis-систем*;
- стандартизацию языка описания ситуаций и событий (в памяти *ostis-систем*), которые являются условиями инициирования различных информационных процессов в памяти *ostis-систем*;
- стандартизацию базового языка спецификации (описания, программирования) агентов, выполняющих соответствующие информационные процессы в памяти *ostis-систем*;
- стандартизацию базовых языков ввода/вывода информации в/из памяти *ostis-систем*.

]

SC-код

:= [Стандарт *смыслового представления информации* в памяти *ostis-системы*, а, точнее, *стандарт семантических сетей*]

абстрактная sc-машина

:= [Стандарт *агентно-ориентированной модели обработки информации в памяти ostis-системы*]

стандартизация

:= [унификация]

⇒ *проблемы текущего состояния**:

- {
 - [Разработка и совершенствование стандартов происходит очень медленно]
 - [В разработке и совершенствовании стандартов принимает участие явно недостаточное число профессионалов – не учитываются все мнения]
 - [В разработке и совершенствовании стандарта отсутствует четкая методика формирования консенсуса]
 - [При введении новой версии стандарта отсутствует четкая методика перевода на новую версию стандарта всех систем, разработанных по предыдущей версии]

⇒ *предлагаемый подход**:

[Стандарт – это перманентно совершенствуемая *база знаний*, поддержку эволюции которой осуществляет соответствующий портал]

конвергенция знаний в памяти ostis-системы⇒ *принципы, лежащие в основе**:

- {
 - [Вводится универсальный базовый язык внутреннего смыслового представления знаний в памяти ostis-систем (*SC-код*), по строению к которому все внутренние языки, ориентированные на представление знаний различного вида (логические языки, языки представления методов решения задач (в частности, программ), язык формулировки задач, онтологические языки и многие другие) являются подязыками *SC-кода*, синтаксис которых полностью совпадает с синтаксисом *SC-кода*.]
 - [Конвергенция различных знаний сводится к согласованию систем понятий, используемых для представления знаний различного вида. Такое согласование направлено на увеличение числа общих понятий, используемых при представлении различных знаний.]

конвергенция моделей решения задач в ostis-системе⇒ *принципы, лежащие в основе**:

- {
 - [Синтаксис языка представления соответствующего класса методов решения задач в памяти – синтаксис *SC-кода*]
 - [Денотационная семантика описывается в виде соответствующей онтологии и представляется в виде текста *SC-кода*]
 - [Операционная семантика каждой модели решения задач – коллектив агентов. Он может быть иерархическим на основе различных моделей решателей, но есть базовая модель интерпретации любых методов –
 - Язык SCP
 - синтаксис совпадает с синтаксисом *SC-кода*
 - денотационная семантика – процедурный язык программирования в графодинамической памяти
 - операционная семантика реализуется на уровне программной или аппаратной платформы
 - *sc-агенты* работают в общей среде – (*sc-памяти*) параллельно, асинхронно на основе ряда правил, позволяющих им не "мешать" друг другу

интеграция знаний в памяти ostis-системы*⇒ *пояснение**:

[Интеграция знаний в памяти *ostis-систем* сводится к склеиванию (отождествлению) синонимичных знаков]

интеграция моделей решения задач в ostis-системе*⇒ *пояснение**:

[Поскольку модель решения задач, используемая *ostis-системой*, представлена в памяти *ostis-системы* как соответствующий вид знаний, интеграция различных моделей решения задач может происходить в *ostis-системе* точно так же, как и интеграция любых других видов знаний. Кроме того, когда речь идет об интеграции различных моделей решения задач, имеется в виду возможность одновременного использования различных моделей решения задач при обработке одних и тех же знаний и, в частности, при решении одной и той же задачи. Такая возможность в *ostis-системе* обеспечивается *агентно-ориентированной моделью*

обработки информации в памяти *ostis*-системы. Таким образом, такого рода интеграция различных моделей решения задач для *ostis*-систем является тривиальной.]

ostis-система

⇒ *достоинства**:

- { • [Высокий уровень способности *ostis*-системы осуществлять семантическую интеграцию знаний в своей памяти (в частности, при погружении новых знаний в текущее состояние базы знаний) обеспечивается смысловым характером внутреннего кодирования информации, хранимой в памяти *ostis*-системы и, в частности, тем, что во внутреннем коде базы знаний *ostis*-системы запрещены омонимичные знаки и пары синонимичных знаков.]
- [Высокий уровень способности интегрировать различные виды знаний в *ostis*-системах обеспечивается тем, что каждый язык, ориентированный на представление знаний соответствующего вида является подязыком одного и того же базового языка *SC*-кода.]

⇒ *примечание**:

[Кроме того можно говорить об иерархии *sc*-языков]

- [Высокий уровень способности интегрировать различные модели решения задач в *ostis*-системах обеспечивается:

- тем, что все эти модели ориентированы на обработку информации, представленной в *SC*-коде
- один и тот же фрагмент базы знаний *ostis*-системы (т.е. одна и та же конструкция *SC*-кода) может одновременно обрабатываться несколькими разными моделями решения задач
- все модели решения задач в *ostis*-системах интегрируются с помощью одной и той же базовой модели решения задач – *sc*-модели решения задач

]

- [Высокий уровень обучаемости *ostis*-систем обеспечивается:
 - высоким уровнем семантической гибкости информации, хранимой в памяти *ostis*-системы, поскольку каждое удаление или добавление синтаксически элементарного фрагмента хранимой информации, а также удаление или добавление каждой связи инцидентности между такими элементами имеет четкую семантическую интерпретацию;
 - высоким уровнем стратифицированности хранимой информации, что обеспечивается онтологически ориентированной структуризацией базы знаний *ostis*-системы;
 - высоким уровнем рефлексии *ostis*-системы, что обеспечивается мощными метаязыковыми возможностями языка внутреннего представления информации (знаний) в памяти *ostis*-систем.

]

- [Каждая *ostis*-система имеет высокий уровень обучаемости (способности к быстрому расширению своих знаний и навыков) и высокий уровень социализации (способности к эффективному участию в деятельности различных коллективов – коллективов, состоящих из *ostis*-систем, и сообществ, состоящих из *ostis*-систем и людей)]

⇒ *детализация достоинства**:

- { • [Существуют четкие формальные критерии, определяющие *уровень семантической совместимости* (уровень семантической конвергенции) различных знаний, навыков, целых *ostis*-систем (точнее, баз знаний этих систем). Очевидно, что *уровень семантической совместимости* прежде всего определяется количеством "точек соприкосновения" в сравниваемых знаниях, навыках и базах знаний – это знаки, присутствующие в разных сравниваемых объектах, но имеющие одинаковые денотаты (т.е. обозначающие одинаковые сущности). При этом среди таких знаков, обозначающих одинаковые сущности и присутствующих в разных сравниваемых объектах особенно важны знаки, обозначающие *понятия*. Количество таких общих понятий в сравниваемых знаниях, навыках, базах знаний определяет уровень семантической совместимости (уровень согласованности) систем используемых понятий в сравниваемых указанных объектах. Увеличение количества знаков, обозначающих одинаковые сущности и присутствующих в разных сравниваемых объектах, может привести к тому, что в разных указанных сравниваемых объектах будут присутствовать не только семантически эквивалентные знаки, но и семантически эквивалентные целые фрагменты (целые информационные конструкции). Существенно при этом подчеркнуть, что семантически эквивалентные знаковые конструкции, представленные на внутреннем языке *ostis*-систем (в *SC*-коде), в памяти разных *ostis*-систем всегда являются синтаксически изоморфными графовыми конструкциями, в которых соответствие изоморфизма связывает знаки, хранимые

в памяти разных *ostis*-систем, но обозначающие одинаковые сущности (точнее, одну и ту же сущность). Заметим также, что в рамках памяти каждой индивидуальной *ostis-системы* синонимия знаков и, соответственно, семантическая эквивалентность знаковых конструкций запрещены.]

- [Благодаря постоянно развиваемым семантическим стандартам *Технологии OSTIS*, которые представлены системой формальных онтологий для самых различных предметных областей, разрабатываемые *ostis-системы* изначально имеют достаточно высокий уровень семантической совместимости со всеми остальными *ostis-системами*. Более того, в *Технологии OSTIS* выделяется целое ядро всех *ostis-систем*, содержащее фундаментальные базовые знания и базовые навыки, одинаковые для всех *ostis-систем* и позволяющее каждой копии этого ядра развиваться (общаться, специализироваться) в любом направлении.]
- [Каждая *ostis-система*, взаимодействуя с людьми (пользователями) или с другими *ostis-системами*, обладает способностью повышать уровень семантической совместимости (взаимопонимания) с ними, а также поддерживать (сохранять) высокий уровень такой совместимости в условиях (1) собственной эволюции, (2) эволюции других *ostis-систем* и пользователей, (3) эволюции семантических стандартов *Технологии OSTIS*. Указанное взаимодействие, в основном, направлено на согласование изменений в системе используемых понятий, т.е. корректировки соответствующих фрагментов онтологий.]
- [Благодаря высокому уровню семантической совместимости *ostis-систем* и смысловому представлению знаний в памяти *ostis-систем* существенно снижается сложность и повышается качество семантического анализа и понимания информации, поступающей (сообщаемой, передаваемой) *ostis-системе* от других *ostis-систем* или пользователей.]
- [Каждая *ostis-система* способна:
 - самостоятельно или по приглашению войти в состав *ostis-коллектива* (коллектива *ostis-систем*) или в состав *ostis-сообщества*, состоящего из *ostis-систем* и людей. Такие коллективы и сообщества создаются на временной (разовой) или постоянной основе для коллективного решения сложных задач;
 - участвовать в распределении (в т.ч. в согласовании распределения) задач – как "разовых" задач, так и долгосрочных задач (обязанностей);
 - мониторить состояние всего процесса коллективной деятельности и координировать свою деятельность с деятельностью других членов коллектива при возможных непредсказуемых изменениях условий (состояния) соответствующей среды.

]

}

- [Высокий уровень интеллекта *ostis-систем* и, соответственно, высокий уровень их самостоятельности и целенаправленности позволяет *ostis-системам* быть полноправными членами самых различных сообществ, в рамках которых *ostis-системы* получают права самостоятельно инициировать (на основе детального анализа текущего положения дел и, в том числе, текущего состояния плана действий сообщества) широкий спектр действий (задач), выполняемых другими членами сообщества, и тем самым участвовать в согласовании и координации деятельности членов сообщества.]
- [Способность *ostis-системы* согласовывать свою деятельность с другими *ostis-системами*, а также корректировать деятельность всего коллектива *ostis-систем*, адаптируясь к различного вида изменениям среды (условий), в которой эта деятельность осуществляется, позволяет существенно автоматизировать деятельность системного интегратора как на этапе сборки коллектива *ostis-систем*, так и на этапе его обновления (реинжиниринга).]

}

⇒

примечание*:

[Достоинства *ostis-систем* обеспечиваются:

- достоинствами *SC-кода* – языка внутреннего кодирования информации, хранимой в памяти *ostis-систем*;
- достоинствами организации *sc-памяти* – памяти *ostis-систем*;
- достоинствами *sc-моделей баз знаний* *ostis-систем* – средствами структуризации таких баз знаний;
- достоинствами *sc-моделей решения задач* – агентно-ориентированных моделей решения задач, используемых в *ostis-системах*.

]

}

/* Завершили рассмотрение понятия *ostis-системы* */

/*-*/

Понятие *ostis-сообщества*

⊃=
{

ostis-сообщество

:= [Человеко-машинный симбиоз, представляющий собой коллектив, состоящий из людей и *ostis*-систем и обеспечивающий высокий уровень автоматизации определённого (соответствующего) вида человеческой деятельности.]

⇒ *примечание**:

[В состав каждого *ostis*-сообщества входит корпоративная *ostis*-система, которая в рамках этого *ostis*-сообщества выполняет:

- роль координатора деятельности членов данного *ostis*-сообщества;
- роль памяти *ostis*-сообщества, т.е. хранителя общих (обобществляемых, общедоступных) знаний для всех членов данного *ostis*-сообщества, которое несет ответственность за совершенствование этих знаний, а также для всех членов всех тех *ostis*-сообществ, в состав которых данное *ostis*-сообщество входит (указанные субъекты являются пользователями рассматриваемых общих знаний). Таким образом, корпоративная *ostis*-система некоторого *ostis*-сообщества является "официальным" представителем этого *ostis*-сообщества во всех *ostis*-сообществах, в состав которых входит, и, следовательно, является координатором деятельности данного *ostis*-сообщества (как единого целого) в рамках всех *ostis*-сообществ, в состав которых оно входит;

]

*есть сходства**

⊃ { • *ostis-сообщество*
• *решатель задач ostis-системы*
}

⇒ *пояснение**:

{
ostis-сообщество
⊃ *многоагентная система, в которой управление агентами осуществляется через общую для них память*
⊃ *многоагентная система, с децентрализованным управлением агентами*
⊃ *многоагентная система, в которой областью деятельности её агентов является как внешняя среда, так и память этой системы*

решатель задач ostis-системы

⊃ *многоагентная система, в которой управление агентами осуществляется через общую для них память*
⊃ *многоагентная система, с децентрализованным управлением агентами*
⊃ *многоагентная система, в которой областью деятельности её агентов является как внешняя среда, так и память этой системы*
⊃ *агентно-ориентированная модель обработки информации в памяти*
}

многоагентная система с децентрализованным управлением агентами

⇒ *включение*: пример**:

- *оркестр, играющий без дирижера или даже без композитора*
⇒ *необходимое требование**:
[каждый участник оркестра должен иметь квалификацию дирижера или композитора]
- *комплексная строительная бригада, работающая без прораба*
⇒ *необходимое требование**:
[каждый участник строительной бригады должен иметь квалификацию прораба]
- *научно-исследовательская лаборатория, работающая без заведующего и научного руководителя*
⇒ *необходимое требование**:
[каждый участник научно-исследовательской лаборатории должен иметь квалификацию заведующего или научного руководителя]
- *кафедра, работающая без заведующего и ученого секретаря*

```

⇒ необходимое требование*:
   [каждый участник кафедры должен иметь квалификацию заведующего и ученого секретаря]
} /* Завершили рассмотрение понятия ostis-сообщества */
/*****/

Понятие ostis-технологии
⊃=
{
ostis-технология
← объединение*:
  {• ostis-технология проектирования
   ⇒ разбиение*:
     {• ostis-технология проектирования ostis-систем соответствующего класса
      ⊃ Базовая ostis-технология проектирования ostis-систем
      • ostis-технология проектирования соответствующего класса компонентов ostis-систем
      ⊃ Базовая ostis-технология проектирования баз знаний ostis-систем
      ⊃ Базовая ostis-технология проектирования решателей задач ostis-систем
      ⊃ Базовая ostis-технология проектирования интерфейсов ostis-систем
      • ostis-технология проектирования объектов заданного класса, не являющихся ostis-системами
      }
     • ostis-технология производства
      ⊃ технология производства спроектированных ostis-систем
      ⊃ ostis-технология управления производством спроектированных продуктов заданного класса, не являющихся ostis-системами
     • технология эксплуатации ostis-систем
      ⊃ Базовая технология эксплуатации ostis-систем
      ⊃ технология эксплуатации ostis-систем соответствующего класса
      ⊃ ostis-технология управления производством спроектированных продуктов заданного класса, не являющихся ostis-системами
      := [технология эксплуатации ostis-систем управления производством спроектированных продуктов заданного класса, не являющихся ostis-системами]
     • технология реинжиниринга ostis-систем
      ⊃ Базовая технология реинжиниринга ostis-систем
      ⊃ технология реинжиниринга ostis-систем соответствующего класса
    }
}

ostis-технология
:= [компонент Технологии OSTIS]
⊃ Ядро Технологии OSTIS
:= [Базовая ostis-технология]
⊃ частная ostis-технология
  ⊃ ostis-технология проектирования соответствующего класса компонентов ostis-систем
    ⊃ Технология проектирования баз знаний ostis-систем
    ⊃ Технология проектирования решателей задач ostis-систем
    ⊃ Технология проектирования невербальных интерфейсов ostis-систем с внешней средой
    ⊃ Технология проектирования интерфейсов ostis-систем с другими техническими системами
    ⊃ Технология проектирования пользовательских интерфейсов ostis-систем
  ⊃ специализированная ostis-технология проектирования ostis-систем соответствующего класса
    ⊃ Технология проектирования ostis-систем управления предприятиями рецептурного производства
    ⊃ Технология проектирования ostis-систем управления предприятиями производства молочной продукции
    ⊃ Технология проектирования интеллектуальных обучающих ostis-систем
    ⊃ Технология проектирования интеллектуальных обучающих ostis-систем для школьников
    ⊃ Технология проектирования интеллектуальных обучающих ostis-систем для подготовки специалистов в области Математики

```

Э *Технология проектирования интеллектуальных обучающих ostis-систем для подготовки специалистов в области Искусственного интеллекта*

ostis-технология проектирования

⇒ *примечание**:

[Каждой ostis-технологии проектирования соответствует своя ostis-система автоматизации проектирования соответствующего класса объектов]

⇒ *соответствующее семейство средств автоматизации**:

ostis-система автоматизации проектирования

⇒ *соответствующее семейство классов проектируемых объектов**:

(ostis-система автоматизации проектирования ostis-систем \cup ostis-система автоматизации проектирования объектов, не являющихся ostis-системами)

⊃ *ostis-технология проектирования ostis-систем соответствующего класса*

ostis-технология проектирования ostis-систем соответствующего класса

:= [технология проектирования *ostis-систем* соответствующего (заданного) класса, который, в свою очередь, соответствует определенному *виду человеческой деятельности*, подвиды которого автоматизируются с помощью указанных выше проектируемых *ostis-систем*]

ostis-технология

⇒ *отношение, заданное на данном множестве**:

- *частная технология**
- *специализированная технология**
- *комплекс специализированных технологий**

⇒ *пояснение**:

[Базовая частная или специализированная технология, входящая в состав комплексной *Технологии OSTIS*, которая:

- направлена на автоматизацию конкретного вида человеческой деятельности;
- ориентирована на использование ostis-систем (как индивидуальных, так и коллективных) в качестве самостоятельных субъектов или активных интеллектуальных инструментов, либо на использование человеко-машинных ostis-сообществ при решении:
 - как задач, выполняемых в памяти ostis-систем (в т.ч. в памяти коллективов ostis-систем);
 - так и задач, выполняемых во внешней среде ostis-систем, в процессе решения которых субъектами соответствующих действий либо ostis-системы (индивидуальные или коллективные), либо конкретные персоны, либо ostis-сообщества.

]

:= [Множество всевозможных технологий, соответствующих стандартам технологии OSTIS и направленных на автоматизацию различных конкретных видов человеческой деятельности]

⇔ *следует отличать**:

Технология OSTIS

⇒ *примечание**:

[*Технология OSTIS* в отличие от понятия *ostis-технологии* представляет собой не множество технологий, а комплекс взаимосвязанных между собой самых различных технологий, превращающий указанное множество технологий в единую объединенную технологию, в сумму взаимосвязанных глубоко интегрированных технологий. В этом смысле *Технология OSTIS* является максимальной *ostis-технологией*, в состав которой входят все *ostis-технологии*.]

⇒ *включение**: *пример**:

- *ostis-технология проектирования и перепроектирования*
- *ostis-технология производства*
- *ostis-технология образования*

Э *пример'*:

- *Технология OSTIS*
- *OSTIS-технология публикации и согласования результатов научно-технической деятельности (в широком смысле)*
- *OSTIS-технология проектирования, реализации и реинжиниринга ostis-систем*
- *OSTIS-технология разработки стандартов Технологии OSTIS*

ostis-технология коллективной разработки информационных ресурсов

⊃ *ostis-технология коллективного проектирования*

- ⊃ *ostis-технология коллективной разработки планов*
- ⊃ *ostis-технология публикации и согласования результатов научно-технической деятельности*
- ⊃ *ostis-технология*

ostis-технология эксплуатации ostis-систем

:= [Общие методы и средства (языковые и интерфейсные) организации взаимодействия ostis-систем со своими конечными пользователями]

⊃ *ostis-технология*

⇒ *примечание**:

[Поскольку в рамках Экосистемы OSTIS каждому человеку придется взаимодействовать с большим числом ostis-систем разного назначения, принципы организации взаимодействия всех ostis-систем со своими пользователями должны быть абсолютно одинаковыми. Удобство (usability) пользовательских интерфейсов должно быть направлено не только на синтаксическую красоту, но и на простую семантическую интерпретацию (понятность).]

ostis-технология проектирования ostis-систем

:= [Технология построения (разработки) логико-семантических моделей (sc-моделей) ostis-систем]

⊃ *ostis-технология*

⇒ *примечание**:

[Продуктом каждого завершенного (целостного) коллективного проекта, реализованного в рамках этой технологии, является полная *логико-семантическая модель ostis-системы*.]

⇒ *класс продуктов**:

логико-семантическая модель ostis-системы

⇒ *средство**:

Метасистема IMS OSTIS

⇒ *класс субъектов**:

коллектив разработчиков ostis-системы

⇒ *класс исходных данных**:

исходная спецификация ostis-системы

ostis-технология производства ostis-систем

:= [Технология сборки и установки ostis-систем]

⊃ *ostis-технология*

⇒ *исходная информация**:

логико-семантическая модель ostis-системы

⇒ *комплектация**:

универсальный интерпретатор логико-семантических моделей ostis-систем

⇒ *примечание**:

[Это, своего рода, "мотор", "движок" ostis-систем]

⇒ *методы**:

Методика производства ostis-систем

⇒ *активный инструмент**:

Метасистема IMS OSTIS

⇒ *продукты**:

ostis-система

ostis-технология реинжиниринга ostis-систем

:= [Технология обновления (перепроектирования) ostis-систем в ходе их эксплуатации]

⊃ *ostis-технология*

следует отличать*

- ⊃ {
 - *Технология реинжиниринга ostis-систем*
 - *Технология проектирования ostis-систем*

⇒ *примечание**:

[Эти технологии сходны. Их методы и средства совпадают. Не совпадают только исходные данные и результаты, которыми в *Технологии обновления ostis-систем* являются предшествующие и последующие состояния ostis-систем. В *Технологии проектирования ostis-систем* исходными данными являются исходные спецификации (замыслы) проектируемых ostis-систем, и результатами – полные логико-семантические модели этих систем]

Технология OSTIS

:= [Совокупность (интеграция, объединение) всех *ostis-технологий*]

← *интеграция**:
ostis-технология

:= [Комплекс (множество) семантически совместимых *технологий*, в состав которого входит *Ядро Технологии OSTIS* и иерархическая система *ostis-технологий*, каждая из которых ориентирована на *проектирование, производство, эксплуатацию* или *реинжиниринг* соответствующего *класса ostis-систем*, обеспечивающих автоматизацию соответствующего *вида человеческой деятельности*. При этом каждая такая проектируемая *ostis-система* автоматизирует либо область, либо *вид человеческой деятельности*, которая (который) является соответственно либо экземпляром (элементом), либо подвидом (подклассом) указанного выше *вида человеческой деятельности*, соответствующего используемой *специализированной ostis-технологии*.]

Ядро Технологии OSTIS

:= [Универсальная базовая *ostis-технология*]

:= [Универсальный компонент Технологии OSTIS]

∈ *ключевой элемент'*:
ostis-технология

← *ядро**:
Технология OSTIS

⊃ *технология*

⇒ *вид деятельности, выполняемой с помощью технологии**:

проектирование, производство, эксплуатация и реинжиниринг ostis-системы

← *объединение**:

{ • *проектирование ostis-системы*

:= [построение логико-семантической модели *ostis-системы*]

• *производство ostis-системы*

:= [сборка логико-семантической модели *ostis-системы* и загрузка этой модели в память универсального интерпретатора таких моделей]

• *эксплуатация ostis-системы*

:= [базовый (предметно-независимый) уровень организации деятельности конечного пользователя *ostis-системы* с помощью соответствующих методов и средств]

• *реинжиниринг ostis-системы*

:= [совершенствование *ostis-системы* в процессе её эксплуатации]

}

⇒ *создаваемые продукты**:
ostis-система

:= [*интеллектуальная компьютерная система*, построенная в соответствии со стандартом *Технологии OSTIS*, предъявляемым к продуктам, создаваемым с помощью этой технологии]

⇒ *примечание**:

[Указанный стандарт продуктов, создаваемых с помощью технологии OSTIS есть не что иное, как *общая формальная семантическая теория интеллектуальных компьютерных систем*]

⇒ *частная технология**:

• *Базовая Технология Проектирования ostis-систем*

⇒ *частная технология**:

• *Технология проектирования баз знаний ostis-систем*

• *Технология проектирования решателей задач ostis-систем*

• *Технология проектирования интерфейсов ostis-систем*

⇒ *частная технология**:

• *Технология проектирования невербальных интерфейсов ostis-систем с внешней средой*

• *Технология проектирования интерфейсов ostis-систем с другими техническими системами*

• *Технология проектирования пользовательских интерфейсов ostis-систем*

⇒ *реализация**:

Метасистема IMS.ostis

:= [Intelligent MetaSystem for ostis-systems design]

:= [OSTIS-система автоматизации проектирования *ostis-систем*]

- *Технология производства ostis-систем*
 - ⇒ *пояснение**:
[Основным компонентом, точнее, инструментальным средством *технологии производства ostis-систем* является *универсальный интерпретатор логико-семантических моделей ostis-систем*. Указанные *логико-семантические модели ostis-систем* являются результатом *проектирования ostis-систем* и представляют собой начальные (исходные) состояния *баз знаний* разрабатываемых *ostis-систем*. В отличие от *инструмента производства ostis-систем*, методика их производства весьма проста и сводится к сборке разработанных логико-семантических моделей (начального состояния *баз знаний*) разрабатываемых *ostis-систем* и загрузке этих моделей в память *универсального интерпретатора логико-семантических моделей ostis-систем*.]
 - ⇒ *реализация**:
универсальный интерпретатор логико-семантических моделей ostis-систем
 - ⇒ *пояснение**:
[Такой интерпретатор логико-семантических моделей *ostis-систем* может быть реализован либо программно на *современных компьютерах*, либо аппаратно в виде компьютеров нового поколения, ориентированных на реализацию интеллектуальных компьютерных систем.]
 - ⇒ *пояснение**:
[С формальной точки зрения универсальный интерпретатор логико-семантических моделей *ostis-систем* является "пустой" *ostis-системой*, которая способна приобретать и записывать формализованную информацию в свою память.]
- *Базовая технология эксплуатации ostis-систем*
 - := [Общая технология эксплуатации *ostis-систем*, включающая в себя общие методы и средства, используемые в процессе эксплуатации любых *ostis-систем*]
 - ⇒ *реализация**:
встраиваемая ostis-система поддержки эксплуатации ostis-систем
 - ⇒ *пояснение**:
[Данная *ostis-система* входит (интегрирована) в состав каждой *ostis-системы*.]
- *Базовая технология реинжиниринга ostis-систем*
 - ⇒ *реализация**:
встраиваемая ostis-система поддержки реинжиниринга ostis-систем
 - ⇒ *пояснение**:
[Данная *ostis-система* входит (интегрирована) в состав каждой *ostis-системы* и обеспечивает внесение изменений "руками" инженеров, сопровождающих эксплуатацию *ostis-системы*, или авторов базы знаний этой *ostis-системы* в текущее состояние базы знаний *ostis-системы* в ходе её эксплуатации]
- ⇒ *специализированная технология**:
 - *Общая технология проектирования ostis-систем автоматизации проектирования*
 - ⇒ *специализированная технология**:
 - *Технология проектирования ostis-систем автоматизации проектирования строительных объектов*
 - *Технология проектирования ostis-систем автоматизации проектирования автомобилей*
 - *Технология проектирования ostis-систем автоматизации проектирования интегральных микросхем*
 - *Технология проектирования ostis-систем управления производством*
 - ⇒ *специализированная технология**:
 - *Технология проектирования ostis-систем управления строительством различных объектов*
 - *Технология проектирования ostis-систем управления производством автомобилей*
 - *Технология проектирования ostis-систем управления производством микросхем*
 - *Технология проектирования ostis-систем управления предприятиями рецептурного производства*
 - ⇒ *специализированная технология**:
Технология проектирования ostis-систем управления предприятиями производства молочной продукции
 - *Технология проектирования интеллектуальных обучающих ostis-систем*
 - ⇒ *комплекс специализированных технологий**:
 - *Технология проектирования интеллектуальных обучающих ostis-систем для школьников*

- *Технология проектирования интеллектуальных обучающих ostis-систем для студентов по общеобразовательным дисциплинам*
- *Технология проектирования интеллектуальных обучающих ostis-систем для студентов по профильным дисциплинам*
- *Технология проектирования интеллектуальных обучающих ostis-систем для магистрантов*

}
⇒ комплекс специализированных технологий*:

- *Технология проектирования интеллектуальных обучающих ostis-систем по Математике*
- *Технология проектирования интеллектуальных обучающих ostis-систем по Искусственному интеллекту*

специализированная ostis-технология

⇒ *примечание**:

[Приведённый нами перечень *специализированных ostis-технологий* охватывает только некоторые области (фрагменты) *человеческой деятельности*, подлежащие автоматизации с помощью *ostis-технологий* в рамках *Экосистемы OSTIS*.]

Ядро Технологии OSTIS

⇒ *примечание**:

[Форма реализации *Ядра Технологии OSTIS* (в виде *ostis-системы IMS.ostis*) позволяет:

- использовать достоинства *Технологии OSTIS* для повышения уровня автоматизации развития самой *Технологии OSTIS* и для существенного повышения темпов такого развития;
- приобрести очень важный опыт применения *Технологии OSTIS*;
- создать центрально ядро *Экосистемы OSTIS*, обеспечивающее поддержку семантической совместимости всех *ostis-систем* и *ostis-сообществ*, входящих в состав *Экосистемы OSTIS*.

]

} /* Завершили рассмотрение *понятия ostis-технологии* */

/*****/

Технология OSTIS

⇒ *пояснение**:

[*Технология OSTIS* рассматривается нами как один из вариантов комплексного решения всех перечисленных выше сверхзадач, которые направлены на развитие деятельности в области искусственного интеллекта и которые, очевидно, сильно связаны друг с другом. Таким образом, *Технология OSTIS* включает в себя:

- и постоянно развивающуюся общую формальную теорию интеллектуальных компьютерных систем, представленную в виде базы знаний соответствующего портала научно-технических знаний;
- и постоянно развивающийся комплекс моделей, методов и средств, используемых при проектировании интеллектуальных компьютерных систем и оформленных в виде интеллектуальной системы информационной и инструментальной поддержки (автоматизации) проектирования семантически совместимых интеллектуальных компьютерных систем;
- и постоянно развивающуюся глобальную экосистему, состоящую из семантически совместимых взаимодействующих интеллектуальных компьютерных систем, ориентированных на комплексную автоматизацию всевозможных видов человеческой деятельности;

]

⇒ *пояснение**:

[Целью создания *Технологии OSTIS* является не только построение методики, обеспечивающей четкую организацию коллективной *человеческой деятельности* по проектированию, производству, эксплуатации и реинжинирингу *интеллектуальных компьютерных систем*, но и построение мощных средств автоматизации (компьютерной поддержки) этой деятельности. Здесь важно подчеркнуть то, что *интеллектуальные компьютерные системы*, разрабатываемые с помощью *Технологии OSTIS (ostis-системы)* могут быть использованы для автоматизации любых видов *человеческой деятельности* и, в том числе, для автоматизации коллективной *человеческой деятельности* по проектированию, производству, эксплуатации и реинжинирингу *ostis-систем*. В рамках *Технологии OSTIS* так и происходит - автоматизация проектирования, производства, эксплуатации и реинжиниринга *ostis-систем* осуществляется с помощью специально предназначенных для этого *ostis-систем*, некоторые из которых (например, для поддержки

эксплуатации и реинжиниринга *ostis-систем*) являются *ostis-системами*, встроенными (интегрированными) в те *ostis-системы*, поддержку эксплуатации и реинжиниринга которых они осуществляют.]

:= [Комплексная технология, обеспечивающая автоматизацию самых различных действий (в том числе, и всевозможных видов человеческой деятельности) на основе семантически совместимых интеллектуальных компьютерных систем, способных координировать (согласовывать) свои действия как с себе подобными, так и с людьми]

:= [Сумма (интеграция) всевозможных *ostis-технологий*]

⇒ *достоинство**:

- [Технология *OSTIS* представляет собой принципиально новый уровень развития *информационных технологий*, в основе которого лежит переход от (from) data science к (to) knowledge science]
- [Открытый характер *Технологии OSTIS* как для тех, кто желает участвовать в её развитии, так и для пользователей *Технологии OSTIS* – для разработчиков прикладных *ostis-систем*]
- [Низкий порог вхождения для желающих развивать и желающих использовать имеющиеся в текущий момент методы и средства *Технологии OSTIS*, что обеспечивается поддержкой качественного состояния документации по текущей версии *Технологии OSTIS* с дополнительным описанием эволюции (развития) *Технологии OSTIS*, а также плана дальнейшего её развития]
- [Децентрализованный характер управления проектами разработки *ostis-систем*, основанный на четком согласовании коллективом разработчиков проектных задач]
- [Ориентация на новое поколение компьютеров, без появления которых дальнейшее развитие *технологий искусственного интеллекта* невозможно. При этом *Технология OSTIS* позволяет достаточно конструктивно сформулировать требования, предъявляемые к таким компьютерам]
- [*Технология OSTIS* не только обеспечивает автоматизацию широкого многообразия видов человеческой деятельности, но и существенно повышает уровень (качество) этой автоматизации, благодаря (1) широкому применению методов и средств *искусственного интеллекта* и (2) созданию условий для *конвергенции*, семантической совместимости и *глубокой интеграции* как автоматизируемых видов человеческой деятельности, так и продуктов этой деятельности. В частности, это касается и автоматизации человеческой деятельности в области *искусственного интеллекта*. *Технология OSTIS* рассматривается как предлагаемый подход к конвергенции и интеграции как различных видов деятельности в области искусственного интеллекта, так и результатов этой деятельности (частных теорий различных компонентов и различных видов интеллектуальных систем, частных методов и средств проектирования различных видов и различных компонентов интеллектуальных компьютерных систем)]
- [Ориентация на разработку компьютерных систем и коллективов таких систем, имеющих высокий уровень *интеллекта*; Ориентация на разработку глобальной сети *интеллектуальных компьютерных систем*, обеспечивающей комплексную автоматизацию всех видов и областей *человеческой деятельности*]
- [Создание условий для формирования *рынка знаний* на основе иерархической системы семантически совместимых *порталов знаний*, соответствующих самым различным областям и *видам человеческой деятельности*]
- [Создание условий для перехода от традиционной формы публикации статей, монографий, отчетов и прочих документов к их публикации как фрагментов *баз знаний* соответствующих *порталов знаний*, что полностью исключает дублирование информации в публикуемых документах и обеспечивает непосредственное использование этой информации в *интеллектуальных компьютерных системах*]

⇒ *ближайшая задача**:

[Обеспечить низкий порог входа в *Технологию OSTIS*:

- для желающих участвовать в развитии *Технологии OSTIS*, т.е. в совершенствовании *Метасистемы IMS.ostis* (системы информационной поддержки и автоматизации проектирования *ostis-систем*), которая сама также является *ostis-системой*;
- для разработчиков *интеллектуальных компьютерных систем*, желающих использовать для этого *Технологию OSTIS* (эти разработчики являются конечными пользователями *Метасистем IMS.ostis*);
- для конечных пользователей всевозможных иных *ostis-систем*, т.е. компьютерных систем, разработанных по *Технологии OSTIS* с непосредственным использованием в качестве инструмента *Метасистемы IMS.ostis* (подчеркнем при этом, что базовые принципы организации взаимодействия *Метасистемы IMS.ostis* с конечными пользователями полностью совпадают с базовыми принципами организации взаимодействия всех остальных *ostis-систем*, разработанных с помощью *Метасистемы IMS.ostis*, со своими конечными пользователями. Это обусловлено тем, что *Метасистема IMS.ostis* сама также является *ostis-системой* – материнской *ostis-системой*).

⇒ *примечание**:

[Для решения указанной задачи необходимо создать инфраструктуру коллективного перманентного обновления (совершенствования) комплексной документации по *Технологии OSTIS*, которая:

- обеспечила бы достаточную полноту и четкость фиксации текущего состояния *Технологии OSTIS* и удовлетворяла бы как разработчиков *Технологии OSTIS* (т.е. разработчиков *Метасистемы IMS.ostis*), так и разработчиков *ostis-систем*, не являющихся *Метасистемой IMS.ostis* (т.е. конечных пользователей *Метасистемы IMS.ostis*), и также конечных пользователей любых *ostis-систем*;
- обеспечила бы высокие темпы совершенствования данной документации на основании (1) четких правил согласования и утверждения различного рода предложений, (2) максимально возможной автоматизации процессов анализа, согласования и утверждения указанных предложений, (3) постоянного расширения числа авторов и (4) четких правил защиты авторских прав;
- обеспечила бы четкую фиксацию границ между текущим состоянием *Технологии OSTIS* и разрабатываемыми, тестируемыми фрагментами её будущих версий с обоснованием таких нововведений и с планом их включения в соответствующую версию *Технологии OSTIS*;
- обеспечила бы четкую семантическую интеграцию документации той части *Технологии OSTIS*, которая касается проектирования семантических моделей *ostis-систем* и которая фактически сводится к проектированию *баз знаний ostis-систем*, а также документации той части *Технологии OSTIS*, которая описывает различные варианты программной или аппаратной реализации универсального интерпретатора логико-семантических моделей *ostis-систем*. Подчеркнем при этом, что универсальность используемого в *Технологии OSTIS* языка представления знаний дает возможность описывать на нем все, что угодно, в том числе и интерпретаторы семантических моделей *ostis-систем*. Но делать это нужно с разумной степенью детализации.

]

⇒ *ближайшая задача**:

[Осуществить конвергенцию и интеграцию всевозможных частных технологий проектирования и реализации различных видов компонентов интеллектуальных компонентов систем (в частности, баз знаний, различного вида логических моделей, искусственных нейронных сетей и т.п.)]

⇒ *класс создаваемых продуктов**:

- *ostis-система*
:= [индивидуальная *ostis-система*]
⇒ *примечание**:
[Существенно подчеркнуть, что *Технология OSTIS* порождает не просто множество *ostis-систем*, а множество семантически совместимых и взаимодействующих *ostis-систем*, образующих экосистему, которую будем называть *Экосистемой OSTIS* (*Экосистемой ostis-систем* и их пользователей). Таким образом, можно считать что интегрированным продуктом *Технологии OSTIS* является не множество *ostis-систем*, а система (*экосистема*) *ostis-систем*.]
- *коллектив ostis-систем*
⊃ *простой коллектив ostis-систем*
:= [коллектив *ostis-систем*, членами которого являются только индивидуальные *ostis-системы*]
⊃ *иерархический коллектив ostis-систем*
:= [коллектив *ostis-систем*, по крайней мере одним членом которого является коллектив *ostis-систем*]
- *ostis-сообщество*
⊃ *простое ostis-сообщество*
⊃ *иерархическое ostis-сообщество*
⊃ *Экосистема OSTIS*

⇒ *основной продукт**:

- *Экосистема OSTIS*
:= [Максимальное *ostis-сообщество*, направленное на автоматизацию всех видов человеческой деятельности]
- *Консорциум OSTIS*
:= [*ostis-сообщество*, направленное на развитие *Технологии OSTIS*]
- *Метасистема IMS.ostis*
:= [Метасистема, являющаяся
 - *корпоративной ostis-системой*, обеспечивающей организацию (координацию) деятельности *Консорциума OSTIS*;
 - формой представления реализации и фиксации текущего состояния *Ядра Технологии OSTIS*;

- корпоративной *ostis-системой*, взаимодействующей со всеми корпоративными *ostis-системами*, каждая из которых координирует развитие соответствующей *специализированной ostis-технологии*.

]

Экосистема OSTIS

⇒ *примечание**:

[Важной особенностью и достоинством *Технологии OSTIS* является то, что все остальные продукты её использования (конкретные *ostis-системы*) объединяются в сеть, т.е. становятся единым целостным продуктом использования *Технологии OSTIS – Экосистема OSTIS*]

⇒ *пояснение**:

[Социально-техническая сеть, состоящая из людей и *ostis-систем*, которые являются

- семантически совместимыми;
- постоянно эволюционирующими индивидуально;
- постоянно поддерживающими свою совместимость с другими агентами в ходе своей индивидуальной эволюции;
- способными децентрализованно координировать свою деятельность.

]

Технология OSTIS

⇒ *решаемая проблема**:

{ • *обеспечение семантической совместимости разрабатываемых компьютерных систем*

⇒ *подход к решению**:

- { • *применение смыслового представления информации в памяти компьютерных систем*
- *согласование и унификация системы используемых понятий и соответствующей иерархической системы формальных онтологий*

}

- *обеспечение поддержки семантической совместимости компьютерных систем в ходе их эксплуатации и эволюции*

⇒ *подход к решению**:

создание самоорганизованной экосистемы компьютерных систем

}

} /* Завершили Сегмент “Уточнение Понятия Технологии OSTIS” */

/* Сегмент *****/

Перспективы использования Технологии OSTIS для повышения качества человеческой деятельности в области Искусственного интеллекта

⊃=

{

⇒ *рассматриваемые вопросы**:

- [Могут ли практические результаты работ в области *Искусственного интеллекта*, существенно повысить эффективность развития *Искусственного интеллекта* как научно-технической дисциплины, включая все формы деятельности в области *Искусственного интеллекта*]
- [Какова перспектива использования *Технологии OSTIS* для автоматизации других областей и видов *человеческой деятельности*]

/* *****/

Научно-исследовательская деятельность в области Искусственного интеллекта

⇒ *субъект деятельности**:

OSTIS-сообщество научно-исследовательской деятельности в области Искусственного интеллекта

⇒ *примечание**:

[Имеются ввиду специалисты разных стран и разных направлений *Искусственного интеллекта*]

⇒ *направления деятельности**:

{ • *Конвергенция и интеграция различных направлений Искусственного интеллекта*

- *Конвергенция Искусственного интеллекта как отдельной научно-технической дисциплины с другими смежными научными дисциплинами*
⇒ *примечание**:
[Конвергенция с математикой, кибернетикой, информатикой, общей теорией систем, психологией, семиотикой, лингвистикой, гносеологией, логикой, методологией и др.]
- *Разработка Общей теории интеллектуальных систем*
⇒ *примечание**:
[Речь идет как о естественных, так и об искусственных интеллектуальных системах.]

}

⇒ *средство автоматизации**:*OSTIS-портал научных знаний в области Искусственного интеллекта*⇒ *технология**:*OSTIS-технология организации коллективной научно-теоретической деятельности*← *частная технология**:*Технология реинжиниринга ostis-систем*:= [Технология коллективного **реинжиниринга** баз знаний *ostis-систем*, обеспечивающая конвергенцию, интеграцию и согласование различных точек зрения и реализуемая абсолютно одинаковыми *ostis-системами*, которые встраиваются (интегрируются) в состав каждой *ostis-системы*]⇒ *реализация**:*Встраиваемая ostis-система поддержки реинжиниринга ostis-систем*⇒ *продукт**:*Общая формальная теория интеллектуальных систем*⇒ *примечание**:[В рамках *Технологии OSTIS* *Общая формальная теория интеллектуальных систем* представляется в виде базы знаний *OSTIS-портала научных знаний в области Искусственного интеллекта*.]***Общая формальная теория интеллектуальных систем***⇒ *примечание**:[Зачем нужна *Общая теория интеллектуальных систем*?

Очевидно, что без этой теории невозможно построить набор методов и средств, обеспечивающий комплексную поддержку разработки *интеллектуальных компьютерных систем* различного назначения и с различным набором навыков (способностей, возможностей), которыми могут обладать *интеллектуальные компьютерные системы*, но необязательно каждая из них. При этом важно не просто построить *Общую теорию интеллектуальных систем* и довести ее до строгого формального уровня, но также довести представление такой формальной теории до уровня базы знаний соответствующего *портала научных знаний*.]

OSTIS-технология организации коллективной научно-теоретической деятельности⇒ *пояснение**:

[Подчеркнем, что *Технология OSTIS* создает достаточно удобные (конструктивные) условия для решения таких проблем, как

- согласование систем понятий разных научных дисциплин (в частности, разных направлений *Искусственного интеллекта*) и, как следствие, возможность реализации достаточно качественной семантической совместимости;
- конвергенция разных научных дисциплин, важным механизмом которой является увеличение числа общих понятий, используемых этими дисциплинами (в частности, этого можно добиться путем введения таких понятий, каждое из которых является обобщением, например, двух понятий, одно из которых относится к одной дисциплине, а другое - к другой
- интеграция научных дисциплин.

Удобство (конструктивность, формализованность) решения указанных проблем обусловлено тем, что каждая научная дисциплина представляется постоянно развивающейся базой знаний, которая в *Технологии OSTIS* представляется в виде специальной семантической сети (в виде текста *SC-кода*), которой соответствуют достаточно простые синтаксические и семантические правила. Важной проблемой организации научно-теоретической деятельности является реализация эффективной процедуры согласования различных точек зрения и обеспечения их конвергенции и глубокой (бесшовной) интеграции. В рамках коллективного развития базы знаний портала научных знаний можно обеспечить:

- существенное сокращение времени, затрачиваемого на согласование используемых понятий;
- существенное повышение эффективности рецензирования самых различных предложений;

- существенное сокращение времени, затрачиваемого на публикацию научных результатов, так как меняется форма публикаций-публикации. Эти результаты оформляются в "смысловом" виде как фрагменты соответствующей базы знаний, что предполагает отсутствие дублирования научных текстов, т.е. отсутствие возможности представления одного и того же результата во многих формах в разных статьях и монографиях;
- автоматизацию анализа качества новых знаний, предлагаемых в состав совершенствуемой базы знаний;
- автоматизацию мониторинга общего качества всей базы знаний.

Очевидно, что качество накапливаемых человечеством *научно-технических знаний* во многом определяет качество (уровень развития) всего человечества как коллектива интеллектуальных систем. Качество указанных знаний определяется

- трудоемкостью их накопления и систематизации
- уровнем конвергенции различных научно-технических дисциплин (уровнем целостности всего комплекса знаний)
- четкостью фиксации текущего (согласованного) состояния накопленных знаний
- четкостью фиксации истории эволюции накапливаемых знаний
- трудоемкостью согласования различных техн ..
- четкостью фиксации противоречий и разногласий

]

язык научно-технической информации

⇒ *примечание**:

[Современные научно-технические тексты не являются естественно-языковыми – это смесь формальных и естественно языковых текстов. Но именно в таком виде осуществляется накопление человеческих знаний в Internete. Необходим универсальный формальный язык, в который был бы в достаточной степени удобным (привычным) и для человека, и для интеллектуальных компьютерных систем. Перспективным подходом к решению этой проблемы является разработка универсального языка семантических сетей.

Переход к *интегрированному* семантическому пространству научно-технических знаний, в основе которой лежит интернациональная семантическая формализация знаний

Например, научно техническая статья должна быть *завершенной* нек матем знаний, возволяющей, например, полностью автоматизировать анализ (рецензирование, верификацию) поступившей статьи (например верификацию показательства теорем).

А научно-технический журнал превращается в портал знаний по заданной научно-технической области.

Перманентное развитие баз знаний такого портала становится открытым проектом, в рамках которого

- каждый желающий может "высказать" свое предложение
- каждый может "высказать" замечание по поводу какого-либо предложения (быть рецензентом)
- внести исправления в свое предложение (по замечаниям)
- признать исправления своих замечаний
- "проголосовать" за/против предложение или исправленное предложение
- признание каждого предложения осуществляется автоматически на основании высказанных мнений с учетом квалификации каждого высказывшегося по отношению к соответствующему разделу базы знаний.
- текущая квалификация каждого специалиста постоянно уточняется (повышается) на основании вклада этого специалиста в развитие базы знаний портала
- вклад любого сепциалиста персонифицируется (защита интеллектуальной собственности) и его ценность (значимость) автоматически оценивается (на основе анализа того, как используется знания, автором или рецензентом – соавтором которых специалист является, в коммерческих проектах!! При этом ссылка на использованные знания является обязательной)

]

OSTIS-портал научных знаний в области Искусственного интеллекта

⇒ *в перспективе**:

[В перспективе каждый *ostis-портал научных знаний* может преобразоваться в сеть семантически совместимых *ostis-порталов научных знаний*, соответствующих различным направлениям заданной *научной дисциплиной* (например, различным направлениям *Искусственного интеллекта*)]

Коллектив специалистов в области Искусственного интеллекта

⇒ *в перспективе**:

[OSTIS-сообщество субъектов научно-исследовательской деятельности в области Искусственного интеллекта]

⇒ *уточнение**:

[указанными субъектами являются объединенные в сеть специалисты в области *Искусственного интеллекта*, неформальные группы таких специалистов, организации или подразделения различных организаций, работающих в области *Искусственного интеллекта* и *ostis-порталы научных знаний* в области Искусственного интеллекта]

← *часть**:

Экосистема OSTIS

/*****/

Разработка Базовой Комплексной технологии проектирования интеллектуальных компьютерных систем

⇒ *субъект деятельности**:

Коллектив разработчиков Базовой Комплексной технологии проектирования интеллектуальных компьютерных систем

⇒ *примечание**:

[Речь идет об открытом проекте разработки указанной технологии и, соответственно, об открытом международном коллективе разработчиков, формируемом на добровольной основе]

⇒ *направления деятельности**:

{ • *Разработка Общей теории интеллектуальных компьютерных систем*

⇒ *примечание**:

[Речь идет об искусственных (компьютерных) интеллектуальных системах и о разработке стандарта таких технологических систем.]

• *Разработка Теории проектирования интеллектуальных компьютерных систем*

⇒ *примечание**:

[Имеются в виду интеллектуальные компьютерные системы, соответствующие стандарту, разработанному в виде общей теории таких систем, имеется в виду рассмотрение самого процесса проектирования таких систем, т.е. рассмотрение методов их проектирования и проектных библиотек.]

• *Разработка комплекса средств автоматизации проектирования интеллектуальных компьютерных систем*

⇒ *примечание**:

[Данные средства автоматизации проектирования (средства решения проектных задач) при их реализации с помощью *Технологии OSTIS* входят в состав решателя задач *Метасистемы IMS.ostis*.]

• *Конвергенция и интеграция различного вида знаний, хранимых в памяти проектируемых интеллектуальных компьютерных систем*

• *Конвергенция и интеграция различных моделей решения задач, используемых проектируемыми интеллектуальными компьютерными системами*

}

⇒ *предлагаемый подход**:

Проект IMS.ostis

:= [Разработка Базовой Комплексной ostis-технологии проектирования ostis-систем]

:= [Разработка Базовой Комплексной технологии проектирования ostis-систем с помощью специально предназначенной для этого *ostis-системы*, которая названа нами *Метасистемой IMS.ostis*]

:= [Проект разработки *Метасистемы IMS.ostis*]

⇒ *принципы, лежащие в основе**:

{ • [Речь идет о проектировании не просто интеллектуальных компьютерных систем, а *ostis-систем*, в виде которых можно построить любую интеллектуальную компьютерную систему. Соблюдение этого принципа является важнейшей целью эволюции *Технологии OSTIS*]

• [Система автоматизации проектирования *ostis-систем* реализуется также в виде *ostis-системы* – *Метасистемы IMS.ostis*]

• [Эволюция технологии проектирования *ostis-систем* сводится к эволюции (реинжинирингу) базы знаний *Метасистемы IMS.ostis*.]

}

- ⇒ *примечание**:
 [Если речь вести о *Технологии OSTIS*, то следует говорить не только о самой данной технологии, но и о проекте, направленном на создание и перманентное совершенствование этой технологии, так как важнейшей особенностью и достоинством *Технологии OSTIS* являются высокие темпы ее эволюции. Указанное достоинство обеспечивается прежде всего тем, что Технология OSTIS реализуется в виде *ostis-системы (Метасистемы IMS.ostis).*]
- ⇒ *средство автоматизации**:
Метасистема IMS.ostis
 := [OSTIS-система автоматизации комплексного проектирования *ostis-систем*]
 ⇒ *примечание**:
 [При *Разработке Базовой Комплексной технологии проектирования интеллектуальных компьютерных систем* (точнее *ostis-систем*) средством автоматизации этой деятельности является не вся *Метасистема IMS.ostis*, а только ее часть – входящая в состав *Метасистемы IMS.ostis* типовая *Встраиваемая ostis-система поддержки реинжиниринга ostis-систем*, которая поддерживает деятельность разработчиков базы знаний *Метасистемы IMS.ostis*. Это обусловлено тем, что вся деятельность по *Разработке Базовой Комплексной технологии проектирования интеллектуальных компьютерных систем (ostis-систем)* сводится к разработке (инжинирингу) и обновлению (совершенствованию, реинжинирингу) *Базы знаний Метасистемы IMS.ostis.*]
- ⇒ *технология**:
Технология реинжиниринга ostis-систем
 ⇒ *реализация**:
Встраиваемая ostis-система поддержки реинжиниринга ostis-систем
- ⇒ *продукт**:
Комплексная ostis-технология проектирования ostis-систем
 ⇒ *реализация**:
Метасистема IMS.ostis
- := [Человеко-машинная деятельность, осуществляемая в рамках *Экосистемы OSTIS* и направленная на разработку и перманентное совершенствование *Метасистемы IMS.ostis*, которая является формой представления (отображения) (1) текущего состояния *Технологии OSTIS*, как комплекса методов и средств автоматизации (поддержки) разработки *ostis-систем* и (2) текущего состояния самого *Проекта IMS.ostis.*]
- ⇒ *примечание**:
 [Принципы (правила) организации деятельности в рамках *Проекта IMS.ostis* полностью совпадают с принципами (правилами) организации деятельности в рамках любого другого проекта, направленного на разработку и совершенствование любой другой *ostis-системы.*]
- ← *ключевой подпроект**:
Проект Экосистемы OSTIS
 := [Совместная деятельность ученых, инженеров и *ostis-систем*, входящих в *Экосистему OSTIS*, направленная на перманентное совершенствование *Экосистемы OSTIS* – на совершенствовании (реинжиниринг) входящих в неё *ostis-систем* и на создании новых *ostis-систем* и их включение в состав *Экосистемы OSTIS.*]
- ⇒ *пояснение**:
 [*ostis-система*, являющаяся:
 ostis-порталом научно-технических знаний по *Технологии OSTIS*, база знаний которого включает в себя:
 формальную теорию *ostis-систем*
 формальную теорию (методику) проектирования *ostis-систем*
 формальную спецификацию средств автоматизации проектирования *ostis-систем*
 библиотеку проектирования *ostis-систем*
 формальную спецификацию средств производства спроектированных *ostis-систем*
 ostis-системой автоматизации (поддержки) проектирования *ostis-систем*
 ostis-системой поддержки производства (сборки, синтеза, генерации) спроектированных *ostis-систем*
 ostis-системой поддержки реинжиниринга *ostis-систем* в ходе их эксплуатации
]

Метасистема IMS.ostis

- := [Универсальная базовая (предметно-независимая) ostis-система автоматизации проектирования ostis-систем (любых ostis-систем)]
- ⇔ *следует отличать**:
специализированная ostis-система автоматизации проектирования ostis-систем
- ∈ *ostis-система*
- ⇐ *корпоративная ostis-система**:
Консорциум OSTIS
- := [IMS.ostis]
- := [Интеллектуальная метасистема, построенная по стандартам *технологии OSTIS* и предназначенная (1) для инженеров *ostis-систем* – для поддержки проектирования. Реализации и обновления (реинжиниринга) *ostis-систем* и для разработчиков *Технологии OSTIS* – для поддержки коллективной деятельности по развитию стандартов и библиотек *Технологии OSTIS*.]
- ⇐ *форма реализации**:
Технология OSTIS
- ⇐ *продукт**:
Проект IMS.ostis
- := [Интеллектуальная Метасистема, являющаяся формой (вариантом) реализации (представления, оформления) *Технологии OSTIS* в виде *ostis-системы*]
- ⇒ *примечание**:
[Тот факт, что *Технология OSTIS* реализуется в виде *ostis-системы*, является весьма важным для эволюции *Технологии OSTIS*, поскольку методы и средства эволюции (перманентного совершенствования) *Технологии OSTIS* становятся фактически совпадающими с методами и средствами разработки любой (!) *ostis-системы* на всех этапах их жизненного цикла.
Другими словами, эволюция *Технологии OSTIS* осуществляется методами и средствами самой этой технологии.]
- := [Система комплексной автоматизации (информационной и инструментальной поддержки) проектирования и реализации *ostis-систем*, которая сама реализована также в виде *ostis-системы*.]
- := [Портал знаний по *Технологии OSTIS*, интегрированный с САПром *ostis-систем* и реализованный в виде *ostis-системы*.]
- ∈ *портал научно-технических знаний*

**{
Метасистема IMS.ostis**

- ∈ *система автоматизации проектирования*
- := [CAD-система]
- ⇐ *аббревиатура**:
[Computer Aided Design system]
- ∈ *интеллектуальная обучающая система*
- }
- ⇔ *семантическая эквивалентность**:
[Метасистема IMS.ostis является одновременно и системой автоматизации проектирования *ostis-систем*, и интеллектуальной системой, обучающей методам и средствам проектирования *ostis-систем*.]
- ⇒ *следовательно**:
[Этот факт существенно повышает качество проектирования прикладных *ostis-систем*, расширяет контингент разработчиков *ostis-систем* и интегрирует проектную (инженерную) деятельность в области искусственного интеллекта с образовательной деятельностью в этой области.]

/*-----*/

Разработка технологии производства спроектированных интеллектуальных компьютерных систем

- ⇒ *предлагаемый подход**:
Проект разработки универсальных интерпретаторов логико-семантических моделей ostis-систем
- ⇒ *класс продуктов**:
универсальный интерпретатор логико-семантических моделей ostis-систем
- := ["пустая" *ostis-система* – *ostis-система*, на базе которой можно построить любую *ostis-систему*, если логико-семантическую модель этой системы, загрузить в память указанной выше "пустой" *ostis-системы*]
- := [Базовый интерпретатор логико-семантических моделей *ostis-систем*]

:= [Интерпретатор Универсальной абстрактной sc-машины]

Проект разработки универсальных интерпретаторов логико-семантических моделей ostis-систем.

:= [Проект реализации универсальной абстрактной sc-машины]

⇒ *альтернативный подпроект**:

Проект Программной реализации интерпретаторов Универсальной абстрактной sc-машины

⇒ *альтернативный подпроект**:

Проект разработки универсальных sc-компьютеров

⇒ *применение**:

[Подчеркнём, что разные альтернативные варианты реализации универсального интерпретатора логико-семантических моделей ostis-систем (универсальной абстрактной sc-машины) никоим образом не влияют на процесс и результат проектирования ostis-систем, то есть на процесс и результат построения логико-семантических моделей разрабатываемых ostis-систем.

Другими словами, принципы представления и структуризации логико-семантических моделей ostis-систем и архитектура универсального интерпретатора этих моделей чётко стратифицированы и, следовательно, могут эволюционировать в достаточной степени независимо друг от друга. Тем не менее некоторая зависимость всё же есть – согласованная трактовка понятия универсальной sc-машины и согласованная форма (язык) передача логико-семантической модели разрабатываемой ostis-системы из *Метасистемы IMS.ostis* в "пустую" ostis-систему.]

Универсальная абстрактная sc-машина

⇒ *пояснение**:

[Абстрактная машина, которая задается:

- *SC-кодом* – внутренним языком представления знаний в памяти *ostis-системы*;
- *абстрактной sc-памятью*, которая уточняет динамику обрабатываемых текстов *SC-кода*;
- универсальным набором (семейством) *sc-агентов*, осуществляющих обработку текстов *SC-кода*.

]

⇒ *примечание**:

[В основе Универсальной абстрактной sc-машины лежит интерпретатор Языка SCP – Базового языка программирования *ostis-систем*.]

Язык SCP

:= [Базовый язык программирования ostis-систем с его синтаксисом, денотационной семантикой и операционной семантикой.]

:= [Язык программирования SCP (Semantic Code Programming)]

⇒ *смотрите**:

Предметная область и онтология Базового языка программирования ostis-систем

Проект программной реализации интерпретаторов Универсальной абстрактной sc-машины.

:= [Проект разработки программной реализации Универсальной абстрактной sc-машины на современных компьютерах]

⇒ *класс продуктов**:

программно реализованный интерпретатор Универсальной абстрактной sc-машины

⇒ *смотрите**:

Предметная область и онтология программных вариантов реализации базового интерпретатора логико-семантических моделей ostis-систем на современных компьютерах

Проект разработки универсальных sc-компьютеров

:= [Проект разработки аппаратной реализации универсальной абстрактной sc-машины в виде компьютера нового поколения, ориентированного на использование в интеллектуальных компьютерных системах (в нашем случае – в ostis-системах)]

⇒ *класс продуктов**:

универсальный sc-компьютер

:= [универсальный ostis-компьютер]

:= [семантический ассоциативный компьютер для ostis-систем]

:= [аппаратно реализованный интерпретатор абстрактной sc-машины]

⇒ *смотрите**:

Предметная область и онтология семантических ассоциативных компьютеров для ostis-систем

⇒ *примечание**:

[Тот факт, что универсальный sc-компьютер, разрабатывается под конкретную технологию проектирование интеллектуальных компьютерных систем (Технологию OSTIS), которая развивается, накапливает опыт разработки и внедрения самых различных прикладных интеллектуальных систем независимо от наличия универсальных sc-компьютеров, имеет принципиальное значение. Опыт создания компьютеров, имеющих принципиально новую архитектуру, показывает, что разработка компьютеров нового поколения без серьезной подготовки технологий их применения, без подготовки соответствующей инфраструктуры приводит к неэффективному использованию результатов разработки и к их быстрому моральному старению.]

⇒ *примечание**:

[Разработка универсального sc-компьютера является важнейшим следующим этапом развития технологии OSTIS, который обеспечит существенное повышение производительности (быстродействия) ostis-систем.

Развитие технологий искусственного интеллекта неизбежно приведёт к необходимости создания компьютеров принципиально нового поколения, предназначенных для использования в интеллектуальных компьютерных системах. Поэтому изначально ориентация Технологии OSTIS на компьютеры нового поколения является принципиальной и весьма перспективной особенностью Технологии OSTIS, обеспечивающей её высокую конкурентоспособность.]

Проект разработки универсальных интерпретаторов логико-семантических моделей ostis-систем

⇒ *примечание**:

[При построении любого интерпретатора любой информационной машины (в нашем случае – абстрактной sc-машины) должны быть чётко полно, а самое главное на формальном языке (в нашем случае – SC-коде) описано следующее:

- синтаксис, денотационная семантика и операционная семантика интерпретируемой машины (в нашем случае для абстрактной sc-машины это синтаксис и денотационная семантика SC-кода и языка SCP, а также операционная семантика языка SCP);
- синтаксис, денотационная семантика и операционная семантика интерпретирующей информационной машины;
- соотношение между указанными формальными моделями интерпретируемой информационной машины и интерпретирующей информационной машины, определяющее семантическую и операционную (функциональную) эквивалентность.

Подчеркнем, что без построения указанной строгой формальной модели соответствия (эквивалентности) интегрируемой и интерпретирующей информационной машины организовать качественную коллективную разработку интерпретаторов сложной информационной машины (например, абстрактной sc-машины) невозможно, так как будет совершаться большое количество поздно обнаруживаемых ошибок.]

⇒ *примечание**:

[Разрабатываемые сейчас варианты реализации *универсального интерпретатора логико-семантических моделей ostis-систем* (программный и аппаратный вариант) являются в известной мере привычными объектами проектирования для современных технологий проектирования программных систем и технологий проектирования интегрированных микросхем и их комплексов.

Тем не менее, повышение уровня сложности указанных объектов проектирования и указанных характеристик проектирования требует существенного повышения уровня интеллекта у соответствующих систем автоматизации (поддержки) проектирования). *Технология OSTIS* уже имеет достаточный опыт разработки *ostis-систем автоматизации проектирования* (достаточно указать Метасистему IMS.ostis, обеспечивающую автоматизацию проектирования ostis-систем). Таким образом для повышения качества разработки *Программной реализации универсальной абстрактной sc-машины* и разработки *универсального sc-компьютера* целесообразно разработать, соответственно, *OSTIS-систему поддержки проектирования сложных программных систем*, а также *OSTIS-систему поддержки проектирования интегральных микросхем и их комплексов*. Здесь речь может идти об интеллектуальных надстройках над существующими средствами автоматизации проектирования и управления проектами.

При проектировании *Программной реализации универсальной абстрактной sc-машины*, а также *универсального sc-компьютера* такая интеллектуальная надстройка абсолютно необходима, поскольку при проектировании указанных объектов необходимо четко отслеживать соответствия между компонентами этих объектов и компонентами интерпретируемой ими *универсальной абстрактной sc-машины*. Актуальность указанной интеллектуальной надстройки обусловлена также тем, что *универсальная абстрактная sc-машина* может корректироваться.

Следует отметить возможную связь между процессом проектирования *Программной реализации универсальной абстрактной sc-машины* и проектированием *универсального sc-компьютера*. Дело в том, что

Программную реализацию универсальной абстрактной sc-машины можно и нужно рассматривать как программную модель не только интегрируемой универсальной абстрактной sc-машины, но и проектируемого универсального sc-компьютера. Таким образом, реализацию универсальной sc-машины можно развивать в двух направлениях:

- в направлении повышения её производительности;
- в направлении более детальной эмуляции универсального sc-компьютера на уровне взаимодействия всё более "мелких" компонентов этого компьютера.

]

/*****

Специализированная инженерия в области Искусственного интеллекта

⇒ предлагаемый подход*:

Специализированная инженерия, осуществляемая на основе Технологии OSTIS

⇒ декомпозиция*:

- { • *Разработка ostis-систем автоматизации проектирования различных классов ostis-систем*
:= [Разработка специализированных ostis-технологий]
⇒ часть*:
- *Разработка ostis-систем автоматизации проектирования ostis-систем автоматизации проектирования*
:= [Разработка ostis-технологий проектирования]
- *Разработка ostis-систем автоматизации проектирования ostis-систем автоматизации производства*
:= [Разработка ostis-технологий управления производством]
- *Разработка ostis-систем автоматизации проектирования ostis-систем управления транспортными системами*
:= [Разработка ostis-технологий управления транспортными системами]
- *Разработка ostis-систем автоматизации проектирования диагностических ostis-систем*
:= [Разработка ostis-технологий диагностики (технической, медицинской)]
- *Разработка ostis-систем автоматизации проектирования обучающих ostis-систем*
:= [Разработка ostis-технологий обучения людей]
- *Разработка ostis-систем автоматизации проектирования ostis-систем управления "умными" домами*
:= [Разработка ostis-технологий управления "умными" домами]
- *Разработка ostis-систем автоматизации проектирования ostis-систем управления "умными" больницами*
:= [Разработка ostis-технологий управления "умными" больницами]
- *Разработка ostis-систем автоматизации проектирования ostis-систем управления "умными" поликлиниками*
:= [Разработка ostis-технологий управления "умными" поликлиниками]
- *Разработка ostis-систем автоматизации проектирования ostis-систем управления "умными" городскими районами*
:= [Разработка ostis-технологий управления "умными" городскими районами]
- *Разработка ostis-систем автоматизации проектирования ostis-систем управления "умными" городами*
:= [Разработка ostis-технологий управления "умными" городами]
- *Разработка (на основе соответствующих ostis-технологий проектирования) ostis-систем автоматизации проектирования различных классов объектов, не являющихся ostis-системами*
⇒ часть*:
- *Разработка семейства ostis-систем автоматизации проектирования различных видов интегральных микросхем*
- *Разработка семейства ostis-систем автоматизации проектирования различных видов автомобилей*
- *Разработка семейства ostis-систем автоматизации проектирования различных видов строительных объектов*

- *Разработка ostis-систем автоматизации производства*
:= [Разработка интеллектуальных систем управления производственными предприятиями]
- *Разработка ostis-систем управления транспортными средствами*
- *Разработка диагностических ostis-систем*
- *Разработка обучающих ostis-систем*
- *Разработка ostis-систем управления "умными" домами*
- *Разработка ostis-систем управления "умными" больницами*
- *Разработка ostis-систем управления "умными" поликлиниками*
- *Разработка ostis-систем управления "умными" городскими районами*
- *Разработка ostis-систем управления "умными" городами*

/*****

Образовательная деятельность в области Искусственного интеллекта

⇒ *предлагаемый подход**:

Образовательная деятельность в области Искусственного интеллекта, осуществляемая на основе Технологии OSTIS

∈ *образовательная деятельность*

∈ *человеческая деятельность, осуществляемая на основе Технологии OSTIS*

:= [человеческая деятельность, комплексная автоматизация которой осуществляется либо индивидуальной *ostis-системой*, либо *коллективом ostis-систем* (сетью *ostis-систем*)]

⇒ *субъект**:

OSTIS-сообщество Образовательной деятельности в области Искусственного интеллекта

:= [глобальное (максимальное) *OSTIS-сообщество*, осуществляющее Образовательную деятельность в области Искусственного интеллекта и обеспечивающее активное и взаимовыгодное сотрудничество между всеми заинтересованными в этом субъектами и, в первую очередь, с соответствующими кафедрами различных вузов]

⇐ *часть**:

Экосистема OSTIS

:= [глобальная сеть *ostis-систем* вместе с их пользователями]

:= [глобальное *ostis-сообщество*]

∈ *ostis-сообщество*

:= [локальная сеть *ostis-систем* вместе с их пользователями]

⇒ *пояснение**:

[Данное *ostis-сообщество* включает в себя:

- все кафедры, которые готовят молодых специалистов в области *Искусственного интеллекта* и которые могут входить в состав самых различных вузов;
- все те организации, которые разрабатывают или эксплуатируют интеллектуальные компьютерные системы и которые готовы сотрудничать с вузами для повышения квалификации поступающих к ним молодых специалистов в области *Искусственного интеллекта*;
- студентов, магистрантов и аспирантов, обучающихся в области *Искусственного интеллекта* в разных вузах;
- их преподавателей;
- семейство интеллектуальных обучающих *ostis-систем* по различным дисциплинам (направлениям) *Искусственного интеллекта*, которые семантически совместимы и тесно связаны с *OSTIS-порталом научных знаний по Искусственному интеллекту* и с *Метасистемой IMS.ostis*;
- *OSTIS-портал научных знаний по Искусственному интеллекту*, осуществляющий поддержку развития *Общей теории интеллектуальных систем* как естественного, так и искусственного происхождения;
- *Метасистема IMS.ostis*, осуществляющая поддержку развития *Общей теории интеллектуальных компьютерных систем* (искусственных интеллектуальных систем) и поддержку развития *Базовой универсальной комплексной технологии проектирования интеллектуальных компьютерных систем*;
- семейство персональных *ostis-ассистентов* студентов, магистрантов и аспирантов, обучающихся в области *Искусственного интеллекта*;
- семейство персональных *ostis-ассистентов* преподавателей, осуществляющих подготовку молодых специалистов в области *Искусственного интеллекта*;

- семейство кафедральных корпоративных *ostis-систем*, осуществляющих управление учебным процессом на уровне кафедр, обеспечивающих подготовку молодых специалистов в области Искусственного интеллекта. В рамках таких корпоративных *ostis-систем* осуществляется:
 - составление кафедрального расписания занятий на следующий семестр и его согласование с расписанием других кафедр этого же вуза;
 - распределение учебной нагрузки на очередной семестр и учебный год;
 - мониторинг проведения различного вида занятий (лекций, консультаций, семинаров, практических занятий, зачетов/экзаменов);
 - мониторинг самостоятельной деятельности обучаемых (курсовых и дипломных проектов, рефератов, диссертаций, тестов и др.);
 - фиксация текущего соответствия между учебными дисциплинами и разделами *Общей теории интеллектуальных систем* и *Базовой универсальной комплексной технологии проектирования интеллектуальных компьютерных систем* (речь идет не только о дисциплинах, непосредственно относящихся к *Искусственному интеллекту*, но и о различных общеобразовательных и смежных дисциплинах, таких, как теория познания, методология, иностранные языки, современные компьютерные системы и сети, компьютеры нового поколения, теория алгоритмов и программ, ориентированных на современные компьютеры, семантическая теория алгоритмов и программ, ориентированных на обработку баз знаний и др.). Принципиально важно сформировать у студентов, магистрантов и аспирантов целостную картину проблематики *Искусственного интеллекта* и место *Искусственного интеллекта* в общей Картине Мира. Барьеров между учебными дисциплинами быть не должно.
- Корпоративная *ostis-система* OSTIS-сообщества, являющегося субъектом *Образовательной деятельности в области Искусственного интеллекта*. Через эту корпоративную *ostis-систему* осуществляется взаимодействие между всеми членами указанного *ostis-сообщества* и, прежде всего между кафедрами, осуществляющими подготовку молодых специалистов в области *Искусственного интеллекта*.

1

⇒ принципы, лежащие в основе*:

- [Подготовка молодых специалистов в области *Искусственного интеллекта* должна осуществляться путем поэтапного и непосредственного их подключения к реальным коллективным проектам:
 - к развитию базы знаний по *Общей теории интеллектуальных систем*, хранимой в памяти соответствующего интеллектуального портала знаний;
 - к развитию базы знаний по *Общей теории интеллектуальных компьютерных систем*, хранимой в памяти соответствующего интеллектуального портала знаний (в памяти *Метасистемы IMS.ostis*);
 - к развитию базы знаний по *Базовой комплексной технологии проектирования интеллектуальных компьютерных систем*, хранимой в памяти интеллектуальной компьютерной системы автоматизации проектирования интеллектуальных компьютерных систем (в памяти *Метасистемы IMS.ostis*);
 - к развитию различных методов и средств проектирования различных компонентов *интеллектуальных компьютерных систем*;
 - к развитию различных специализированных технологий проектирования различных классов интеллектуальных компьютерных систем;
 - к разработке различных прикладных интеллектуальных компьютерных систем на основе развиваемой Базовой (универсальной) комплексной технологии проектирования интеллектуальных компьютерных систем.
- 1
- [Каждый студент и магистрант в процессе обучения привлекается к нескольким разным формам деятельности в области *Искусственного интеллекта* и, в частности, обязательно и к разработке приложений, и к развитию технологий. Специалист, занимающийся автоматизацией какой-либо деятельности должен на себе прочувствовать проблемы и трудности этой автоматизируемой деятельности]
- [Все студенты, магистранты и преподаватели должны активно участвовать в анализе эффективности своей образовательной деятельности и активно способствовать повышению эффективности и повышению уровня автоматизации этой деятельности с помощью развиваемой технологии проектирования и производства интеллектуальных компьютерных систем. Данный принцип можно условно назвать устранением синдрома "сапожника без сапог".]

- [Результаты самостоятельной работы студентов и магистрантов (лабораторных работ, практических занятий, рефератов, курсовых работ и проектов, дипломных работ и проектов, магистерских диссертаций) должны быть востребованы в тех проектах, к которым они подключены и должны быть доведены до уровня внедрения в эти проекты, т. е. должны быть по соответствующей процедуре согласованы и одобрены. При этом приветствуется и соответствующим образом поощряется любая такого рода инициатива студентов и магистрантов. Указанная востребованность (полезность) результатов самостоятельной работы студентов и магистрантов предполагает то, что отчеты по этим результатам оформляются в формализованном виде – в виде исходных текстов соответствующих фрагментов баз знаний. При этом указанные результаты могут требовать как весьма высокой квалификации, так и не очень высокой (например, квалификации первокурсника). К таким несложным, но весьма полезным работам относятся:
 - введение в *базы знаний* полезных библиографических ссылок и цитат;
 - сравнительный анализ различных положений, представленных в некоторой разрабатываемой базе знаний;
 - различные пояснения, примечания и комментарии, вводимые в *базу знаний*;
 - спецификация выявленных в разрабатываемой базе знаний ошибок, противоречий, информационных дыр и информационного мусора;
 - примеры, иллюстрирующие различные понятия;
 - упражнения к различным разделам разрабатываемых *баз знаний*, которые особенно актуальны для интеллектуальных компьютерных систем, используемых в учебном процессе (это не только интеллектуальные обучающие системы).
-]
- [Вклад каждого студента и магистранта в развитие всех проектов, в которых он принимает участие, фиксируется и при подведении итогов по каждому семестру соответствующим образом оценивается. Это своего рода предтеча будущего рынка знаний.]
- [Учебным пособием по каждой учебной дисциплине должна быть база знаний или некоторый раздел базы знаний некоторой интеллектуальной компьютерной системы. Такой может быть либо интеллектуальная обучающая система, либо, например, *Метасистема IMS.ostis*. Условием максимально эффективного проведения лекционного занятия является предварительное прочтение студентами или магистрантами материала предстоящей лекции (соответствующего раздела базы знаний). Тогда на лекции можно акцентировать внимание не на изложение материала, опубликованного в виде базы знаний, а на обсуждение непонятных фрагментов этого материала, на обсуждение проблем, касающихся содержания (принципиальных положений) этого материала. Все это формирует культуру взаимопонимания и согласования различных точек зрения, а также способствует повышению качества базы знаний, представляющей материал соответствующей учебной дисциплины.]
- [Важнейшей задачей подготовки молодых специалистов является формирование у них:
 - высокой математической культуры (культуры формализации);
 - высокой системной культуры (понимания того, что количество далеко не всегда переходит в ожидаемое качество);
 - высокого уровня технологической культуры, технологической дисциплины, четкого соблюдения текущих стандартов и способности участвовать в эволюции стандартов;
 - способности работать в наукоемких проектах в составе творческих коллективов с децентрализованным управлением;
 - способности к достижению семантической совместимости (взаимопонимания) со своими коллегами;
 - договороспособности (способности к согласованию различных точек зрения).
-]
- [Подготовку молодых специалистов в области *Искусственного интеллекта* можно осуществлять с ориентацией на следующие условно выделенные уровни их квалификации:
 - инженерия прикладных *интеллектуальных компьютерных систем* по заданной технологии;
 - инженерия специализированных технологий проектирования различных классов прикладных интеллектуальных компьютерных систем (на основе базовой универсальной комплексной технологии проектирования интеллектуальных компьютерных систем);
 - инженерия базовой универсальной комплексной технологии проектирования интеллектуальных компьютерных систем;

- инженерия программных и аппаратных средств, интерпретации логико-семантических моделей интеллектуальных компьютерных систем;
- инженерия комплексов интеллектуальных компьютерных систем;
- научно-исследовательская деятельность по развитию *Общей формальной теории интеллектуальных компьютерных систем*.

]]

)

/*****

Бизнес-деятельность в области Искусственного интеллекта

⇒ предлагаемый подход*:

Бизнес-деятельность в области Искусственного интеллекта, осуществляемая на основе Технологии OSTIS

⇒ субъект*:

OSTIS-сообщество Бизнес-деятельности в области Искусственного интеллекта, осуществляемой на основе Технологии OSTIS

:= [Глобальное (максимальное) OSTIS-сообщество, осуществляющее Бизнес-деятельность в области Искусственного интеллекта]

← часть*:

Экосистема OSTIS

∈ *ostis-сообщество*

⇒ пояснение*:

[Речь идет об ostis-сообществе, которое включает в себя все компании и лаборатории, работающие в области Искусственного интеллекта и желающие на взаимовыгодных условиях сотрудничать в направлении совместного, перманентного и интенсивного развития стандартов, методов и средств комплексного проектирования и производства семантически совместимых и договороспособных интеллектуальных компьютерных систем, способных самостоятельно и целенаправленно взаимодействовать друг с другом. Кроме указанных компаний и лабораторий в состав рассматриваемого ostis-сообщества входят:

- семейство корпоративных ostis-систем, которые "представляют интересы" указанных компаний и лабораторий в рамках рассматриваемого ostis-сообщества и которые обеспечивают автоматизацию "внутренней" деятельности (бизнес-процессов) этих компаний и лабораторий, включая делопроизводство, юридический мониторинг, бухгалтерскую деятельность, административно-хозяйственную деятельность, управление персоналом, управление выполняемыми проектами и т.д.;
- Корпоративная ostis-система OSTIS-сообщества, являющегося субъектом Бизнес-деятельности в области Искусственного интеллекта. Через эту корпоративную ostis-систему осуществляется взаимодействие между членами рассматриваемого ostis-сообщества – между компаниями и лабораториями, работающими в области искусственного интеллекта.

]]

Консорциум OSTIS

∈ *ostis-сообщество*

⇒ пояснение*:

[Весь комплекс деятельности в области *Искусственного интеллекта* мы декомпозировали на шесть форм (частей). Для каждой из этих форм деятельности создается свое *ostis-сообщество*, каждому из которых, в свою очередь, соответствует своя *корпоративная ostis-система*. *Консорциум OSTIS* объединяет все указанные *ostis-сообщества*, включая в свой состав (в состав *Консорциума OSTIS*) прежде всего все *корпоративные ostis-системы* указанных *ostis-сообществ*. Кроме того, для координации деятельности членов самого *Консорциума OSTIS* создается *Корпоративная ostis-система Консорциума OSTIS*. Напомним, что для каждого *ostis-сообщества*, создается соответствующая ему *корпоративная ostis-система*, являющаяся ключевым членом этого *ostis-сообщества* и осуществляющая координацию всех остальных его членов.]

⇒ член *ostis-сообщества**:

- *Корпоративная система Консорциума OSTIS*

← *корпоративная ostis-система**:

Консорциум OSTIS

← субъект*:

Деятельность в области Искусственного интеллекта, осуществляемая на основе Технологии OSTIS

- *OSTIS-портал научных знаний в области Искусственного интеллекта*
 ⇐ *корпоративная ostis-система*:*
OSTIS-сообщество научно-исследовательской деятельности в области Искусственного интеллекта
 ⇐ *субъект*:*
научно-исследовательская деятельность в области Искусственного интеллекта, осуществляемая на основе Технологии OSTIS
- *Метасистема IMS.ostis*
 ⇐ *корпоративная ostis-система*:*
OSTIS-сообщество Проекта IMS.ostis
 ⇐ *субъект*:*
Проект IMS.ostis
- *Корпоративная система OSTIS-сообщества Проекта разработки универсального интерпретатора логико-семантических моделей ostis-систем*
 ⇐ *корпоративная ostis-система*:*
Корпоративная ostis-система OSTIS-сообщества Проекта разработки универсального интерпретатора логико-семантических моделей ostis-систем
 ⇐ *субъект*:*
Проект разработки универсального интерпретатора логико-семантических моделей ostis-систем
- *Корпоративная система OSTIS-сообщества специализированной инженерии в области Искусственного интеллекта, осуществляемой на основе Технологии OSTIS*
 ⇐ *корпоративная ostis-система*:*
OSTIS-сообщество Специализированной инженерии в области Искусственного интеллекта, осуществляемой на основе Технологии OSTIS
 ⇐ *субъект*:*
Специализированная инженерия в области Искусственного интеллекта, осуществляемая на основе Технологии OSTIS
- *Корпоративная система OSTIS-сообщества образовательной деятельности в области Искусственного интеллекта, осуществляемой на основе Технологии OSTIS*
 ⇐ *корпоративная ostis-система*:*
OSTIS-сообщество Образовательной деятельности в области Искусственного интеллекта, осуществляемой на основе Технологии OSTIS
 ⇐ *субъект*:*
Образовательная деятельность в области Искусственного интеллекта, осуществляемая на основе Технологии OSTIS
- *Корпоративная система OSTIS-сообщества Бизнес-деятельности в области Искусственного интеллекта, осуществляемой на основе Технологии OSTIS*
 ⇐ *корпоративная ostis-система*:*
OSTIS-сообщество Бизнес-деятельности в области Искусственного интеллекта, осуществляемой на основе Технологии OSTIS
 ⇐ *субъект*:*
Бизнес-деятельность в области Искусственного интеллекта, осуществляемая на основе Технологии OSTIS

⇒ *примечание*:*

[Конвергенция и интеграция различных форм и направлений деятельности в области *Искусственного интеллекта* должна проходить через каждого персонального члена *Консорциума OSTIS* – желательно, чтобы большинство из них были одновременно:

- и участниками научно-исследовательской деятельности в области *Искусственного интеллекта* (аспирантами, докторантами и т.д.);
- и участниками совершенствования (развития) целостного комплекса методов и средств проектирования и реализации *интеллектуальных компьютерных систем*;
- и разработчиками различных прикладных *интеллектуальных компьютерных систем*;
- и преподавателями, участвующими в подготовке молодых специалистов в области *Искусственного интеллекта*.

]

: = [OSTIS-сообщество субъектов всех форм и направлений деятельности в области Искусственного интеллекта]
 ⇐ часть*:

Экосистема OSTIS

```

:= [Научно-техническое и учебное объединение специалистов и организаций, работающих в области Искусственного интеллекта]
⇒ перспективы*:
[Создание Консорциума OSTIS на основе широкого применения Технологии OSTIS может и должно осуществляться с поэтапным расширением состава участников и поэтапным повышением уровня автоматизации деятельности Консорциума OSTIS. Ключевыми направлениями деятельности Консорциума OSTIS являются:

□ Существенное повышение темпов эволюции Ядра Технологии OSTIS, темпов перехода на все более совершенные версии стандартов интеллектуальных компьютерных систем, проектных библиотек и средств автоматизации проектирования интеллектуальных компьютерных систем;
□ Разработка компьютеров нового поколения, ориентированных на интерпретацию логико-семантических моделей интеллектуальных компьютерных систем;
□ Разработка иерархического семейства семантически совместимых специализированных технологий проектирования различных классов интеллектуальных компьютерных систем;
□ Создание условий для развития технологий искусственного интеллекта в направлении унификации интеллектуальных компьютерных систем для обеспечения их конвергенции и семантической совместимости.

]
} /* Завершили Сегмент “Перспективы использования Технологии OSTIS для повышения качества человеческой деятельности в области Искусственного интеллекта” */

```

```

/* Сегмент *****/

```

Уточнение Понятия Экосистемы OSTIS

```

⊃=
{
:= [Использование Технологии OSTIS для повышения качества и, в частности, уровня автоматизации всех областей человеческой деятельности]
:= [Понятие Экосистемы OSTIS как формы реализации smart-общества, представляющего собой сеть взаимодействующих людей, интеллектуальных компьютерных систем, "умных" домов, "умных" предприятий, "умных" больниц, "умных" учебных заведений, "умных" городов, "умных" транспортных систем и т.п.]
⇒ рассматриваемые вопросы*:
{
• [Какова архитектура Экосистемы OSTIS]
• [Какова архитектура ostis-сообщества, входящего в состав Экосистемы OSTIS]
• [Как взаимодействуют между собой различные ostis-сообщества в рамках Экосистемы OSTIS]
• [Как интегрируется деятельность различных ostis-сообществ и результаты этой деятельности]
• [Какова типология ostis-сообществ и по каким признакам классификации можно эту типологию проводить]
• [Можно ли опыт автоматизации деятельности в области Искусственного интеллекта с помощью Технологии OSTIS расширить на все многообразие областей и видов человеческой деятельности]
• [Как выглядит систематизация областей и видов человеческой деятельности]
• [Как осуществляется конвергенция и интеграция различных областей и видов человеческой деятельности]
• [Как взаимодействуют ostis-системы, осуществляющие автоматизацию различных областей видов человеческой деятельности]
• [Как может выглядеть комплексная автоматизация всех областей и видов человеческой деятельности с помощью Технологии OSTIS]
}
⇒ план изложения*:
{
• [Что такое Экосистема OSTIS]
• [Структура Экосистемы OSTIS]
• [Что такое ostis-система, являющаяся агентом Экосистемы OSTIS]
• [Что такое ostis-сообщество, являющееся агентом Экосистемы OSTIS]
• [Что такое Проект создания Экосистемы OSTIS]
•
}

```


- [Цель создания и основные свойства *Экосистемы OSTIS*]
- [Как структурируется *человеческая деятельность*]
- [Как выглядит *рынок знаний*, реализуемый в рамках *Экосистемы OSTIS*]
- [Чем определяется качество *человеческой деятельности*]
- [Что такое эффективная автоматизация *человеческой деятельности*]
- [Почему повышение эффективности *человеческой деятельности* невозможно без *интеллектуальных компьютерных систем*]
- [Какие достоинства имеет *Экосистема OSTIS*]

/*-*/

Экосистема OSTIS

⇒ *вопрос**:

[Какова структура *Экосистемы OSTIS*]

⇒ *пояснение**:

[Популяция

- семантически совместимых
- эволюционируемых
- активно взаимодействующих
- способных координировать(согласовывать) свою деятельность с другими субъектами

интеллектуальных компьютерных систем (*ostis-систем*). При этом указанная популяция *ostis-систем* поддерживает децентрализованное управление собственной деятельностью, а также деятельностью людей(пользователей *ostis-систем*) и человеко-машинных сообществ (*ostis-сообществ*), обеспечивая тем самым автоматизацию системной интеграции любых новых субъектов (*ostis-систем*, людей, *ostis-сообществ*) в состав *Экосистемы OSTIS*.]

⇒ *принципы, лежащие в основе**:

- [Экосистема *OSTIS* представляет собой сеть *ostis-сообществ*]
 - [Каждому *ostis-сообществу* взаимно однозначно соответствует *корпоративная ostis-система* этого *ostis-сообщества*, которая:
 - обеспечивает координацию деятельности членов соответствующего *ostis-сообщества*;
 - является "представителем" этого *ostis-сообщества* в других *ostis-сообществах*, членом которых указанное *ostis-сообщество* является.
 -]
- [Каждое *ostis-сообщество* может входить в состав любого другого *ostis-сообщества* по своей инициативе. Формально это означает, что *корпоративная ostis-система* первого *ostis-сообщества* является членом другого *ostis-сообщества*.]
 - [Каждому специалисту, входящему в состав *Экосистемы OSTIS* ставится во взаимнооднозначное соответствие его *персональный ostis-ассистент*, который трактуется как *корпоративная ostis-система* вырожденного *ostis-сообщества*, состоящего из одного человека.]

*следует отличать**

- ⇒ {
- *корпоративная ostis-система**
:= [корпоративная *ostis-система* данного *ostis-сообщества**]
 - *корпоративная ostis-система*
← *второй домен**:
*корпоративная ostis-система**
 - *член ostis-сообщества**
 - *персональный ostis-ассистент**
:= [персональный *ostis-ассистент* данного специалиста*]
С *корпоративная ostis-система**
 - *персональный ostis-ассистент*
← *второй домен**:
*персональный ostis-ассистент**
С *корпоративная ostis-система*
- }

есть сходства*

- \ni {
 - Экосистема OSTIS
 - *ostis-сообщество* \ni Экосистема OSTIS
- \Rightarrow пояснение*:
 [Экосистема OSTIS является максимальным *ostis-сообществом*, включающим в себя все существующее *ostis-сообщества*]

Экосистема OSTIS

- \equiv [Максимальное иерархическое *ostis-сообщество*, обеспечивающее комплексную автоматизацию всех видов человеческой деятельности]
- \equiv [Максимальное *ostis-сообщество* такое *ostis-сообщество*, для которого не существует другого *ostis-сообщества*, содержащее указанное выше *ostis-сообщество* в качестве своего члена]
- \equiv [Симбиоз людей и компьютерных систем (точнее, *ostis-систем*) являющийся вариантом реализации *smart-общества*]
- \in иерархическое *ostis-сообщество*
- \equiv [такое *ostis-сообщество*, по крайней мере одним из членов которого является некоторое другое *ostis-сообщество*]
- \subset *ostis-сообщество*
- \Leftrightarrow следует отличать*:
 коллектив *ostis-систем*
- \leftarrow основной продукт*:
 Технология OSTIS
- \leftarrow вариант реализации*:
smart-общество

агент Экосистемы OSTIS

- \equiv [субъект Экосистемы OSTIS]
- \equiv [субъект, входящий в состав Экосистемы OSTIS]
- \supset когнитивный агент Экосистемы OSTIS
- \Rightarrow разбиение*:
 - индивидуальная *ostis-система* Экосистемы OSTIS
 - \equiv [индивидуальная *ostis-система*, входящая в состав Экосистемы OSTIS]
 - *ostis-сообщество* Экосистемы OSTIS
 - \Rightarrow разбиение*:
 - простое *ostis-сообщество* Экосистемы OSTIS
 - иерархическое *ostis-сообщество* Экоистемы OSTIS
 - пользователь Экоистемы OSTIS
- \Rightarrow правила поведения*:
 Правила поведения агентов Экоистемы OSTIS
- $=$ {
 - [Согласовывать денотационную семантику всех используемых знаков(в первую очередь понятий)]
 - [Согласовывать терминологию, соответствующую введенным знакам устранять противоречия и информационные дыры]
 - [Постоянно бороться с синонимией и омонимией как на уровне sc-элементов (внутренних знаков), так и на уровне соответствующих им терминов и прочих внешних идентификаторов(внешних обозначений)]
 - [Каждый агент Экоистемы OSTIS по своей инициативе может стать членом любого *ostis-сообщества* Экоистемы OSTIS после соответствующей регистрации]

Правила поведения агентов Экоистемы OSTIS

- \Rightarrow примечание*:
 [Существенно подчеркнуть, что все правила функционирования(поведения) агентов в рамках Экоистемы OSTIS должны соблюдать не только *ostis-системы*, являющиеся агентами(субъектами) этой Экоистемы, но и люди, которые являются её агентами. И здесь возникают очень важные проблемы, обусловленные

человеческим фактором. Дело в том, что убедить человека соблюдать правила, пусть даже те которые направлены на максимальную его самореализацию и в совершенствовании которых он может реально участвовать, очень непросто, поскольку любые подобные правила многими воспринимаются как ограничение их творческой свободы. Другими словами, корректное поведение *ostis-систем* в роли агентов *Экосистемы OSTIS* значительно проще обеспечить, чем корректное поведение людей в качестве таких агентов. Поведение пользователей (естественных агентов) *Экосистемы OSTIS* необходимо внимательно мониторить и контролировать, постоянно способствуя повышению уровня их квалификации как агентов *Экосистемы OSTIS*, а также повышению уровня их мотивации, целенаправленности, самореализации.]

*следует отличать**

- ⇒ { • агент *Экосистемы OSTIS*
• член *ostis-сообщества**
}

Экосистема OSTIS

⇒ *архитектура**:

[В *Экосистеме OSTIS* можно выделить следующие уровни иерархии:

- индивидуальные компьютерные системы (*индивидуальные ostis-системы* и люди, являющиеся конечными пользователями *ostis-систем*);
- иерархическая система *ostis-сообществ*, членами каждого из которых могут быть *индивидуальные ostis-системы*, люди, а также другие *ostis-сообщества*;
- *Максимальное ostis-сообщество Экосистемы OSTIS*, не являющееся членом никакого другого *ostis-сообщества*, входящего в состав *Экосистемы OSTIS*.

Подчеркнем, что качество *Экосистемы OSTIS* во многом определяется эффективностью взаимодействия каждой *ostis-системы* (в том числе и каждого *ostis-сообщества*), а также каждого человека со своей *внешней средой**, а также качеством (чистотой), самой *внешней среды**. Но внешняя среда каждого субъекта каждой *ostis-системы* и каждого человека, входящего в *Экосистему OSTIS* – это не только *материальная внешняя среда**, но и *информационная внешняя среда**, представляющая собой виртуальный распределенный информационный ресурс, являющийся интеграцией (объединением) информации, хранящейся в текущий момент в памяти всех других (остальных) *субъектов*, входящих в *Экосистему OSTIS*. Основной целью *Экосистемы OSTIS* является повышение качества (в том числе чистоты) *информационной внешней среды** для всех *субъектов*, входящих в *Экосистему OSTIS*. Фактически речь идет об ***Информационной экологии человеческого общества.***]

Информационная экология человеческого общества

⇒ *примечание**:

[Говоря об *Информационной экологии человеческого общества* необходимо заметить следующее. Современные подходы к развитию взаимодействия с информационной средой человеческого общества можно разбить на два направления:

- на разработку средств приспособления к недостаткам текущего состояния этой среды
- на устранение этих недостатков путем наведения порядка в устной информационной среде и её систематизации.

Технология OSTIS и реализация *Экосистемы OSTIS* целенаправленно и в известной степени радикально ориентирована на второе направление, памятуя искусственный (рукотворный) характер происхождения этой информационной среды.]

}

/*****/

ostis-система, являющаяся агентом Экосистемы OSTIS

⇒ *Примечание**:

[Вокруг каждой (!) *ostis-системы* формируется коллектив её разработчиков, несущих ответственность (1) за её качественную эксплуатацию и (2) за её перманентное совершенствование в ходе эксплуатации. При этом интеллект *ostis-системы* должен быть использован для максимально возможной автоматизации указанной деятельности разработчиков (для мониторинга состояния эксплуатируемой *ostis-системы*, для своевременной реакции на подозрительные ситуации или события), а также для координации деятельности разработчиков. Таким образом, каждая *ostis-система* (в том числе каждая персональная *ostis-система*) для своих разработчиков фактически является корпоративной системой. Следовательно, для каждой *ostis-системы* следует отличать (1) координируемый (управляемый, обслуживаемый) ею коллектив конечных пользователей, управляемых *ostis-сообществ* и *ostis-систем* (2) и координируемый ею коллектив разработчиков, в состав

которого входит также и "породившая" эту *ostis-систему* *ostis-система* автоматизации проектирования и реализации *ostis-систем* соответствующего класса (эта система должна рассматриваться как полноценный активный член коллектива разработчиков, как *ostis-система*, несущая полную ответственность за все те *ostis-системы*, которые она "породила"). Таким образом, в рамках *Экосистемы OSTIS* можно выделить (1) иерархическую сеть *ostis-систем*, обеспечивающую автоматизированную комплексную реализацию человеческой деятельности с помощью указанных *ostis-систем* и (2) сеть *ostis-систем*, предназначенную для поддержки их эффективной эксплуатации и их совершенствования (постоянного обновления), включая постоянное совершенствование *баз знаний ostis-систем*, осуществляемое не столько специалистами в области *Искусственного интеллекта*, сколько специалистами в соответствующих предметных областях. Ярким примером *ostis-систем*, совершенствуемых не только специалистами в областях *Искусственного интеллекта*, являются порталы научно-технических знаний. Подчеркнём, что ответственность за качество сети *ostis-систем*, предназначенной для разработчиков, постоянно совершенствующих эти *ostis-системы* несёт *Консорциум OSTIS*.]

персональный ostis-ассистент

:= [ostis-система, являющаяся персональным ассистентом для соответствующей персоны, входящей в состав *Экосистемы OSTIS*]

:= ["официальный" представитель соответствующей персоны во всех *ostis-сообществах*, членом которых эта персона является]

:= [*ostis-система*, являющаяся посредником соответствующей персоны в его взаимодействии с членами всех коллективов (*ostis-сообществ*), в состав которых эта персона входит]

⇒ *примечание**:

[Каждой персоне, входящей в состав *Экосистемы OSTIS* взаимно однозначно соответствует его личный (персональный) ассистент в виде персонального *ostis-ассистента*. Таким образом, количество персональных *ostis-ассистентов*, входящих в состав *Экосистемы OSTIS*, совпадает с числом персон, входящих в состав *Экосистемы OSTIS*.]

⇒ *примечание**:

[Коллектив, состоящий из персоны и соответствующего ей *персонального ostis-ассистента*, фактически является *минимальным ostis-сообществом*, которое также можно назвать терминальным *ostis-сообществом*. Следовательно, персонального *ostis-ассистента* можно условно(!) считать *корпоративной ostis-системой* минимального *ostis-сообщества*, которая, как и любая другая корпоративная *ostis-система* (1) организует "внутреннюю" деятельность соответствующего *ostis-сообщества* и (2) осуществляет "внешнее" взаимодействие этого *ostis-сообщества* (как единого целого) с другими агентами (субъектами) в рамках *ostis-сообществ*, находящихся на следующем (более высоком) уровне иерархии.]

⇒ *примечание**:

[Строго говоря, все *ostis-сообщества*, кроме *минимальных ostis-сообществ*, являются не коллективами, состоящими из *персон и ostis-систем*, а коллективами, состоящими только из *ostis-систем*, поскольку формально в неминимальное *ostis-сообщество* входят не персоны, а соответствующие им, организующие их деятельность *персональные ostis-ассистенты*.]

:= [корпоративная система минимального *ostis-сообщества*]

⇔ *следует отличать**:

корпоративная ostis-система

:= [корпоративная (центральная) *ostis-система* либо неминимального *ostis-сообщества*, либо коллектива *ostis-систем*]

⇔ *следует отличать**:

ostis-система массового обслуживания индивидуальных пользователей

:= [ostis-система, не осуществляющая координацию деятельность своих конечных пользователей]]

:= [ostis-сообщества массового обслуживания групп пользователей]

:= [ostis-сообщества, поддерживающая взаимодействия своих конечных пользователей, не только в рамках самостоятельных групп пользователей]

корпоративная ostis-система

:= [ostis-система осуществляющая поддержку организации деятельности некоего количества, а также поддержка эволюции (совершенствования) этой деятельности управления проектами]

⇒ *принципы, лежащие в основе**:

[В основе взаимодействия и взаимосвязи корпоративных *ostis-систем*, входящих в состав *Экосистемы OSTIS* лежат следующие принципы:

- глубокая конвергенция различных научных дисциплин, формальных моделей различных видов деятельности и все, трансдисциплинарность, что может быть сделано одинаково и должно быть сделано одинаково.

- обобщение различных видов деятельности, построение частной иерархии различных моделей.

]

ostis-система, являющаяся агентом Экосистемы OSTIS

▷ персональный ostis-ассистент

⇔ следует отличать*:
персональный ostis-ассистент*

:= [быть персональным ostis-ассистентом данной персоны*]

▷ корпоративная ostis-система

⇔ следует отличать*:
корпоративная ostis-система*

:= [быть корпоративной ostis-системой данного ostis-сообщества*]

▷ ostis-портал научно-технических знаний

:= [ostis-портал знаний по некоторой научно-технической дисциплине]

▷ ostis-система автоматизации проектирования

▷ ostis-система автоматизации производства

:= [ostis-система управления производством]

▷ ostis-система автоматизации образовательной деятельности

▷ обучающаяся ostis-система

▷ корпоративная ostis-система виртуальной кафедры

:= [корпоративная ostis-система, обеспечивающая интеграцию деятельности кафедр одинакового профиля и, возможно, различных вузов]

▷ ostis-система автоматизации бизнес-деятельности

▷ ostis-система автоматизации управления

▷ ostis-система управления проектами соответствующего вида

▷ ostis-система сенсо-моторной координации при выполнении определённого вида сложных действий во внешней среде

▷ ostis-система управления самостоятельным перемещением робота по пересеченной местности

обучающаяся ostis-система

⇒ Примечание*:

[Поскольку качество эксплуатации каждой ostis-системы зависит не только от неё, но и от квалификации пользователя (семантическая совместимость, знания о возможностях системы), каждая ostis-система должна быть способна обучать пользователя знаниям и навыкам эффективного её использования.]

/*****

ostis-сообщество, являющееся агентом Экосистемы OSTIS

:= [это естественный этап перехода (эволюции) творчески ориентированных коллективов людей в принципиально новое существенно более интеллектуальное и, соответственно, более позитивное качество]

:= [устойчивый фрагмент Экосистемы OSTIS, обеспечивающий (1) комплексную автоматизированную определенную часть коллективной человеческой деятельности (2) перманентное повышение эффективности (в т.ч. уровня автоматизации) указанной части человеческой деятельности]

⇒ примечание*:

[Каждому ostis-сообществу, являющемуся агентом, Экосистемы OSTIS соответствует своя *корпоративная ostis-система*, обеспечивающая интеграцию соответствующих корпоративных знаний и поддержку взаимодействий всех ostis-систем и пользователей, входящих в это ostis-сообщество]

ostis-сообщество

⇐ перманентно-решаемая задача*:

- перманентная поддержка семантической совместимости членов ostis-сообщества
- перманентная поддержка высокого качества базы знаний, доступной всем членам ostis-сообщества

⇒ пояснение*:

[Имеется в виду поддержка непротиворечивости (корректности отсутствия синонимов, омонимов и противоречий), полноты (отсутствия информационных дыр) и чистоты (отсутствия информационного мусора)]

- перманентная поддержка мониторинга эффективности распределения работ между членами ostis-сообщества и контроля исполнительской дисциплины

- *перманентная поддержка мониторинга динамики роста квалификации каждого члена ostis-сообщества*

коллектив людей

:= [человеческое общество]

⇒ *примечание**:

[Низкий уровень интеллекта современных коллективов людей определяется (1) низким уровнем качества организации общей памяти каждого такого коллектива (общей памяти всех его членов) (2) низким уровнем качества знаний, хранимых в этой памяти (как минимум это наличие большого количества синонимов, омонимов и противоречий). Поэтому переход от современных коллективов людей к соответствующим ostis-сообществам существенно повышает уровень интеллекта этих коллективов]

/*****/

Экосистема OSTIS

∈ *ostis-сообщество*

← *субъект**:

Объединенная человеческая деятельность, осуществляемая на основе Технологии OSTIS

⇒ *корпоративная ostis-система**:

Корпоративная система Экосистемы OSTIS

⇒ *пояснение**:

[Основным назначением *Корпоративной системы Экосистемы OSTIS* является организация общего взаимодействия при выполнении самых различных видов и *областей человеческой деятельности*, которые могут быть либо полностью автоматизированными, либо частично автоматизированными, либо вообще неавтоматизированными. Из этого следует, что база знаний *Корпоративной системы Экосистемы OSTIS* должна содержать *Общую формальную теорию человеческой деятельности*, включающей в себя типологию видов и *областей человеческой деятельности*, а также *общую методологию этой деятельности*.]

Деятельность в области Искусственного интеллекта, осуществляемая на основе Технологии OSTIS

⇒ *субъект**:

Консорциум OSTIS

⇒ *корпоративная ostis-система**:

Корпоративная ostis-система Консорциума OSTIS

⇒ *основной продукт**:

Экосистема OSTIS

⇒ *частное ostis-сообщество**:

Консорциум OSTIS

⇒ *член ostis-сообщества**:

Консорциум OSTIS

⇒ *примечание**:

{

член ostis-сообщества*

⊂ *частное ostis-сообщество**

:= [ostis-сообщество, входящее в состав заданного либо непосредственно (в качестве члена), либо в качестве члена члена заданного ostis-сообщества и т. д.]

}

:= [Проект, основной целью (продуктом) которого является создание *Экосистемы OSTIS*]

⇒ *часто используемый sc-идентификатор**:

Проект Экосистемы OSTIS

:= [Деятельность, направленная на создание и перманентное развитие *Экосистемы OSTIS*]

:= [Проект, направленный на проектирование, производство и реинжиниринг *ostis-систем*, входящих в сеть *Экосистемы OSTIS*, а также на проектирование и реинжиниринг *Экосистемы OSTIS* в целом (как сети *ostis-систем* и их пользователей)]

⇒ *подпроект**:

- *Проект IMS.ostis*

- *Проект программной реализации абстрактной sc-машины*

- *Проект разработки универсального sc-компьютера*

⇒ *примечание**:

[В состав *Проекта Экосистемы OSTIS* входит большое количество *проектов (подпроектов*)*, направленных на проектирование и производство *ostis-систем* самого различного назначения]

⇒ *пояснение**:

[Распространим предлагаемый нами подход к повышению эффективной и человеческой *Деятельности в области Искусственного интеллекта* на всю *Объединенную человеческую деятельность* в целом, т.е. рассмотрим структуру *Глобального ostis-сообщества (Экосистемы OSTIS)*]

⇒ *эпиграф**:

[От *Консорциума OSTIS* к *Экосистеме OSTIS*]

Экосистема OSTIS

⇒ *примечание**:

[Подчеркнем, что *Экосистема OSTIS* является:

- с одной стороны, *основным продуктом** человеческой *Деятельности в области Искусственного интеллекта, осуществляемой на основе Технологии OSTIS* (эту Деятельность мы также будем называть *Проектом Экосистемы OSTIS*),
- а, с другой стороны, *субъектом** *Объединенной человеческой деятельности, осуществляемой на основе Технологии OSTIS*.

Особо подчеркнем то, что продуктом человеческой *Деятельности в области Искусственного интеллекта, осуществляемой на основе Технологии OSTIS*, является не просто множество *ostis-систем* различного назначения, а *Экосистема, состоящая из взаимодействующих ostis-систем и их пользователей*]

⇒ *пояснение**:

[Принципиальным является то, что продуктом (результатом применения) *Технологии OSTIS* является не просто множество *ostis-систем*, а целая система, состоящая из *ostis-систем* и их пользователей, взаимодействующих между собой и осуществляющих комплексную автоматизацию всех *видов человеческой деятельности*, а также комплексное повышение уровня эффективности организации человеческой деятельности (и, в частности, повышение уровня автоматизации этой деятельности).]

/*****/

Экосистема OSTIS

⇒ *вопрос**:

- *Каковы основные свойства Экосистемы OSTIS*
- *Какова основная цель создания Экосистемы OSTIS*

:= [Экосистема *ostis-систем* и их пользователей]

← *общий создаваемый продукт**:

Технология OSTIS

:= [Расширяемый коллектив эволюционируемых, семантически совместимых и взаимодействующих *ostis-систем* и их пользователей]

∈ *многоагентная система*

:= [Многоагентная система, агентами которой являются *ostis-системы*, а также их конечные пользователи и разработчики]

Экосистема интеллектуальных компьютерных систем

:= [Smart-сообщество]

:= [Smart-сообщество интеллектуальных компьютерных систем и людей]

:= [Интеллектуальная многоагентная система, состоящая из интеллектуальных компьютерных систем и людей]

⇒ *примечание**:

[Многоагентная система может состоять из кибернетических систем, не являющихся интеллектуальными.]

⇒ *примечание**:

[Многоагентная система может состоять из интеллектуальных систем, но сама не быть интеллектуальной. Количество далеко не всегда переходит в нужное качество.]

:= [Экосистема *ostis-систем*, а также их разработчиков и пользователей]

:= [Эволюционируемая сеть *ostis-систем*, обеспечивающая конвергенцию и интеграцию всех видов человеческой деятельности]

Экосистема OSTIS

:= [Глобальная компьютерная сеть *ostis-систем*, обеспечивающая комплексную автоматизацию всевозможных видов и областей человеческой деятельности и отражающая иерархию уровней этой деятельности]

- ⇒ [Глобальная *многоагентная система*, состоящая из людей и семантически совместимых *интеллектуальных компьютерных систем*, построенных по *Технологии OSTIS*, которые, взаимодействуя между собой и с людьми, обеспечивают существенное повышение уровня автоматизации всех видов *человеческой деятельности* и существенное повышение эффективности человеческого взаимодействия]
- ⇒ [Предлагаемых нами подход к реализации smart-общества]
- ⇒ [Smart-общество, построенное на основе Технологии OSTIS]
- ⇒ [следующий этап развития человеческого общества, обеспечивающий существенное повышение уровня общественного (коллективного) интеллекта путем преобразования человеческого общества в экосистему, состоящую из людей и семантически совместимых интеллектуальных систем]
- ⇒ *обоснование**:
[Необходимо обеспечить не только повышение уровня автоматизации человеческой деятельности (как информационной (умственной), так и "физической"), но и существенное повышение уровня интеллекта человеческого общества как социальной кибернетической системы путем создания многоагентной кибернетической системы, состоящей из *интеллектуальных компьютерных систем* и людей и имеющей *высокий уровень интеллекта*. Человечества пока не умеет создавать интеллектуальные сообщества (коллективы) людей и, тем более, интеллектуальные общества людей и интеллектуальных компьютерных системы. Уровень интеллекта каждого такого сообщества обычно определяется уровнем интеллекта его руководителя (лица, принимающего решение). А надо, чтобы уровень интеллекта сообщества был результатом интеграции интеллектуального потенциала всех его членов. При этом следует помнить, что интеллект определяется не только и не столько множеством решаемых задач, а скоростью расширения этого множества.]
- ⇒ *примечание**:
[Предметом инженерной деятельности в области *искусственного интеллекта* следует считать не множество *интеллектуальных компьютерных систем* (например, *ostis-систем*), а весь комплекс взаимодействующих между собой *интеллектуальных компьютерных систем*. Назовём такой комплекс *Экосистемой интеллектуальных компьютерных систем* (в нашем случае – это Экосистема OSTIS – Экосистема взаимодействующих *ostis-систем*) здесь важно построить архитектуру таковой экосистемы, в основе которой должна лежать комплексная формальная модель всевозможных видов человеческой деятельности, автоматизируемых с помощью интеллектуальных компьютерных систем (*ostis-систем*). Указанная комплексная модель человеческой деятельности является необходимой основой создания smart-общества (общества 5.0)]
- ⇐ *общий создаваемый продукт**:
Технология OSTIS
- ⇒ [Общий (объединенный, интегрированный) продукт использования *Технологии OSTIS*, представляющий собой глобальную сеть *ostis-систем*, обеспечивающий комплексную автоматизацию и интеграцию всевозможных *видов человеческой деятельности* и, в частности, включающий в себя (в виде соответствующего *ostis-сообщества*) *консорциум OSTIS*, т.е. инфраструктуру, направленную на перманентное развитие *Технологии OSTIS* (как Ядра Технологии OSTIS, так и иерархического семейства *специализированных ostis-технологий*)]
- ⇒ *следовательно**:
[*Экосистема OSTIS* представляет собой саморазвивающуюся сеть *ostis-систем*]
- ⇒ *пояснение**:
[Сверхзадачей *Экосистемы OSTIS* является не просто комплексная автоматизация всех *видов человеческой деятельности* (разумеется, только тех видов деятельности, автоматизация которых целесообразна), но и существенное повышение уровня интеллекта различных человеческих (точнее человеко-машинных) сообществ и всего человеческого общества в целом. Это потребует соблюдения ряда требований, предъявляемых не только к *интеллектуальным компьютерным системам*, но и к людям, входящим в состав *Экосистемы OSTIS*]
- ⇒ *пояснение**:
[*Экосистема OSTIS* представляет собой открытый коллектив взаимодействующих интеллектуальных систем, состав которого входят *ostis-системы* и их пользователи (конечные пользователи и разработчики, участвующие в совершенствовании этих *ostis-систем*). Особое место среди *ostis-систем*, входящих в состав *экосистемы OSTIS*, занимают *корпоративные ostis-системы*, через которое осуществляется координация и эволюция деятельности некоторых групп *ostis-систем* и их пользователей. Основная цель корпоративных *ostis-систем* – локализовать базы знаний указанных групп *ostis-систем*, перевести их из статуса виртуальных в статус реальных и автоматизировать их эволюцию.]
- ⇒ [Сообщество *ostis-систем* или людей, обеспечивающее принципиально новый уровень автоматизации человеческой деятельности и принципиально новый уровень интеллекта человеческого общества]

Экосистема OSTIS

⇒ *примечание**:

[Очень важно проектировать не только саму *Экосистему OSTIS*, ну и процесс поэтапного перехода от современной глобальной сети *компьютерных систем* к глобальной сети *ostis-систем* (т.е. к *Экосистеме OSTIS*). В рамках такого переходного периода *ostis-системы* могут выполнять роль системных интеграторов различных ресурсов и сервисов, реализованных современными *компьютерными системами*, поскольку уровень интеллекта *ostis-систем* позволяет им с любой степенью детализации специфицировать интегрируемые *компьютерные системы* и, следовательно, достаточно адекватно "понимать", что знает и/или умеет каждая из них, а также достаточно качественно координировать их деятельность и обеспечивать "релевантный" поиск нужного ресурса и сервиса. Кроме того системы могут выполнять роль интеллектуальных help-систем – помощников и консультантов по вопросам эффективной эксплуатации различных *компьютерных систем* со сложными функциональными возможностями, имеющими пользовательский интерфейс с нетривиальной семантикой и использующимися в сложных предметных областях. Такие интеллектуальные help-системы можно сделать интеллектуальными посредниками между соответствующими компьютерными системами их пользователями. При этом пользователь может работать одновременно и с help-системой и с соответствующей эксплуатируемой компьютерной системой, консультируясь с help-системой в затруднительных для него ситуациях. Основными недостатками такого варианта является то, что: (1) пользователь должен использовать два разных интерфейса и (2) help-система не может мониторить деятельность пользователя и, следовательно, пользователь сам должен сообщать системе о своих затруднительных ситуациях. Указанные недостатки можно устранить, если компьютерную систему, которая построена по современным технологиям и эксплуатация которой нуждается в качественной консультационной (help-овой) поддержке, интегрировать с соответствующей ей help-системой, построенной по стандартам технологии OSTIS, так, чтобы пользовательским интерфейсом такой интегрированной системы стал пользовательский интерфейс, соответствующий стандартам технологии OSTIS и важнейшим достоинством которого является чёткая формализация семантики всех элементов управления пользовательским интерфейсом. Благодаря этому взаимодействие пользователя с пользовательским интерфейсом *ostis-систем* становится, во-первых, осмысленным и, во-вторых, позволяющим легко переносить опыт интерфейсного взаимодействия с одной *ostis-системы* на другую *ostis-систему*.]

/*****

Объединенная человеческая деятельность

- := [максимальная область человеческой деятельности]
- := [вся человеческая деятельность]
- := [человеческая деятельность в целом]
- := [объединение всевозможных областей человеческой деятельности]
- ∈ *человеческая деятельность*
 - := [область человеческой деятельности]
 - := [система целенаправленных действий некоторого количества (возможно одного) людей над некоторыми объектами с помощью некоторых инструментов]
 - С *деятельность*
 - := [трудно выполнимое сложное действие]
 - := [область деятельности]
 - := [система целенаправленных действий некоторых (возможно одного) субъектов над некоторыми объектами с помощью некоторых инструментов]

следует отличать*

- Э {
 - *Объединенная человеческая деятельность*
 - := [человеческая деятельность в целом]
 - := [максимальная область человеческой деятельности]
 - *область человеческой деятельности*
 - := [фрагмент (часть, раздел) человеческой деятельности]
 - := [человеческая деятельность]
 - := [деятельность, осуществляемая либо одним человеком (индивидуальная человеческая деятельность), либо коллективом людей]
 - С *деятельность*
 - С *действие*
 - ⊃ *индивидуальная человеческая деятельность*
 - ⊃ *коллективная человеческая деятельность*
 - := [множество всевозможных областей человеческой деятельности]
 - *вид человеческой деятельности*

:= [класс однотипных областей человеческой деятельности, которому можно поставить в соответствие некоторую технологию]
 C вид деятельности
 C класс действий

}

следует отличать*

- Э { • *Объединенная человеческая деятельность*
- := [максимальный процесс человеческой деятельности, включающий в себя деятельность всех людей и всех сообществ]
- *человеческая деятельность*
- := [множество всевозможных целостных, целенаправленных фрагментов *Объединенной человеческой деятельности*]
- ⇒ *примечание**:
 [На данном множестве заданы такие отношения, как *часть**, *декомпозиция**. Т.е. конкретный экземпляр (элемент) данного множества может быть *частью** (входить в состав) другой конкретной человеческой деятельности. Более того, целесообразно рассматривать достаточно сложную иерархию процессов человеческой деятельности.]
- := [конкретный процесс человеческой деятельности]
- := [бизнес-процесс]
- := [деятельность, основными субъектами которой являются люди и различные сообщества людей]
- C *деятельность*
 C *действие*
- ⇒ *примечание**:
 [Если для автоматизации человеческой деятельности используются интеллектуальные компьютерные системы, то эти системы также становятся достаточно самостоятельными полноценными субъектами этой деятельности, мнение которых обязательно принимается во внимание, но при этом интеллектуальные компьютерные системы не становятся основными субъектами человеческой деятельности.]
- Э *объединенная человеческая деятельность*
- := [максимальная человеческая деятельность, для которой не существует никакой другой конкретной человеческой деятельности, частью* которой указанная Максимальная человеческая деятельность является.]
- ⇒ *примечание**:
 [каждая конкретная человеческая деятельность (каждый бизнес-процесс) может быть:
- либо полностью автоматизирована – от человека требуется только корректно сформулировать соответствующую команду (цель инициируемого действия);
 - либо автоматизирована, но требующая от человека управления функционированием соответствующего одного инструментального средства;
 - либо состоящая из фрагментов (подпроцессов, частных бизнес-процессов), некоторые из которых автоматизированы, а некоторые нет;
 - либо полностью неавтоматизирована (т.е. выполняется "вручную")
-]
- ⇒ *примечание**:
 [Когда речь идет о спецификации конкретной человеческой деятельности (конкретного бизнес-процесса), важно провести четкую грань между теми действиями, которые выполняются автоматически (в том числе интеллектуальными компьютерными системами), и действиям, которые выполняются людьми "вручную" - это как минимум действия по "формулировке" команд, которые адресуются соответствующим инструментальным средствам (язык и, соответственно, интерфейс формулировки таких команд для разных инструментальных средств может сильно отличаться). Отсутствие унификации языка взаимодействия (интерфейсы) между людьми и различными инструментальными средствами (автомобилями, станками, холодильниками, газовыми плитами, микроволновками, компьютерными системами различного назначения) существенно снижает комплексную эффективность автоматизации человеческой деятельности, т.к. вынуждает людей тратить много времени на усвоение не сути (смысла) автоматизации, а формы (синтаксиса) своей деятельности по организации использования различных средств автоматизации.]
- *вид человеческой деятельности*
- := [класс (множество однотипных) процессов человеческой деятельности]

:=	[класс бизнес-процессов]
:=	[множество всевозможных классов бизнес-процессов]
С	<i>вид деятельности</i>
⇒	<i>примечание*</i> : [Каждый конкретный вид человеческой деятельности (т. е. каждый элемент множества " <i>вид человеческой деятельности</i> ") является <i>подмножеством*</i> множества "человеческая деятельность". Каждому виду человеческой деятельности соответствует своя <i>технология человеческой деятельности</i> , т.е. свой набор <i>методов</i> и <i>средств</i> , обеспечивающих выполнение каждой конкретной деятельности, принадлежащей этому виду.]
•	<i>область человеческой деятельности</i>
С	<i>человеческая деятельность</i>
:=	[достаточно крупный фрагмент человеческой деятельности]
:=	[раздел человеческой деятельности]
}	

Экосистема OSTIS

⇒ *примечание**:

[Содержательную типологию *ostis-систем*, входящих в состав *Экосистемы OSTIS* следует проводить на основе глубокого анализа содержательной структуры человеческой деятельности, требующей взаимодействия человека с другими людьми и даже с организациями. Очевидно, что эффективность такого взаимодействия во многом определяется качеством организации информационного взаимодействия, уровнем взаимопонимания, уровнем квалификации участников, оперативностью получения качественной консультативной помощи по любому (!) вопросу.]

вид человеческой деятельности, продуктом которой является информационная модель некоторого объекта или класса объектов

:=	[вид человеческой деятельности, направленной на построение описания (спецификации) некоторого объекта исследования или класса таких объектов]
С	<i>вид человеческой деятельности</i>
Э	<i>научно-исследовательская деятельность</i>
Э	<i>Научно-исследовательская деятельность в области Искусственного интеллекта</i>
:=	[разработка Общей теории интеллектуальных систем]
Э	<i>разработка теории искусственных объектов заданного класса</i>
Э	<i>Разработка Общей теории интеллектуальных компьютерных систем</i>
:=	[разработка стандарта интеллектуальных компьютерных систем]
Э	<i>разработка теории проектирования искусственных объектов заданного класса</i>
:=	[разработка системы проектных действий для искусственных объектов (артефактов) заданного класса]
Э	<i>разработка теории проектирования интеллектуальных компьютерных систем</i>
:=	[разработка стандарта организации коллективных проектных действий для проектирования интеллектуальных компьютерных систем]
Э	<i>разработка теории производства спроектированных искусственных объектов заданного класса</i>
:=	[разработка стандарта системы производственных действий, методов и инструментов, обеспечивающих производство спроектированных артефактов заданного класса]
Э	<i>Разработка Теории производства спроектированных интеллектуальных компьютерных систем</i>
Э	<i>проектирование искусственного объекта заданного класса</i>
:=	[проектная деятельность, направленная на построение такой информационной модели (спецификации) искусственно создаваемого объекта (артефакта) заданного класса, которой достаточно для производства этого объекта]
⊃	<i>проектирование конкретной интеллектуальной компьютерной системы</i>
:=	[процесс проектирования некоторой компьютерной системы по заданной технологии проектирования]

⇒ *пояснение**:

[Данный вид человеческой деятельности характерен следующими особенностями:

- очень часто продукт этой деятельности (создаваемая информационная конструкция) имеет высокую степень сложности и, следовательно, указанная деятельность не может быть индивидуальной, а несет коллективный характер;
- основными факторами качественного коллективного построения сложной информационной конструкции являются семантическая совместимость (взаимопонимание) авторов, а также согласованность их действий;

- важнейшим направлением автоматизации коллективной деятельности, объект и продукт которой представляет собой сложную информационную конструкцию, является автоматизация редактирования коллективно создаваемого информационного объекта, а также автоматизация обеспечения семантической совместимости и согласованности продуктов индивидуальной деятельности всех соавторов;
- указанную автоматизацию легко реализовать с помощью корпоративной интеллектуальной компьютерной системы, объединяющей всех соавторов создаваемого информационного объекта и снабженной мощными средствами поддержки коллективного проектирования различных разделов базы знаний этой системы. Примерами таких систем являются интеллектуальные порталы различного вида знаний.

]

вид человеческой деятельности, продуктом которой является информационная модель некоторого объекта или класса объектов

⇒ *пояснение**:

[Здесь речь идет о коллективной человеческой деятельности, которая принципиально не может быть полностью автоматизирована (исследовательская, проектная), то основной проблемой ее автоматизации являются

- недостаточный уровень семантической совместимости и взаимопонимания между людьми и отсутствие сознания серьезности этой проблемы;
- недостаточный уровень договоренности и отсутствия понимания серьезности этой проблемы;
- отсутствие четкой методики согласования точек зрения и отсутствие понимания серьезности этой проблемы.

Интеллектуальные компьютерные системы могут и должны создать корпоративную среду для решения этих проблем. По сути это не что иное, как поддержка коллективного проектирования соответствующих разделов баз знаний интеллектуальной компьютерной системы, реализуемая на семантическом уровне, когда интеллектуальная компьютерная система становится самостоятельным полноправным участником деятельности, в обязанности которого входит:

- анализ семантической совместимости точек уровня различных участников,
 - □ выявление противоречий и альтернатив
- фиксация авторства
- отмена современной формы представления интеллектуального продукта (статьи, книги, документы)

Недостаточно высокий уровень семантической согласованности используемых понятий приводит к огромному количеству искусственно создаваемых противоречий. При этом следует отличать семантические противоречия (например, синонимии вводимых знаков) и, соответственно, методику их устранения или разногласия по поводу системы вводимых понятий от терминологических разногласий, методика устранения которых может и должна быть максимально простой и лишенной эмоциональной окраски. Излишнее увлечение терминологическими спорами существенно тормозит творческий процесс, но и несерьезное отношение к постоянному совершенствованию и соблюдению правил построения терминов также недопустимо.]

/*****/

Рынок знаний, реализуемый в рамках Экосистемы OSTIS

⇒ *пояснение**:

[Важнейшим видом предметно-независимой человеческой деятельности, осуществляемой в рамках *Экосистемы OSTIS* является перманентный реинжиниринг всех *ostis-систем*, входящих в *Экосистему OSTIS*. Указанная деятельность должна быть направлена на перманентную и быструю эволюцию всех *ostis-систем* и, самое важное, на эволюцию *Экосистемы OSTIS* в целом. Особо следует подчеркнуть, что эволюция *ostis-систем* и *Экосистемы OSTIS* в целом представляет собой весьма сложный творческий, коллективный процесс, который принципиально может быть автоматизирован только частично. При этом от людей, участвующих в этом процессе требуется высокая квалификация, высочайшая системная культура на уровне глубокого знания общей теории систем, высокая математическая культура – культура формализации, высокая культура конвергенции (обнаружения сходств, доведение их до формальных аналогий), высокая культура глубокой интеграции, высокий уровень договороспособности.

Кроме указанных требований необходим высочайший уровень мотивации к тому, чтобы эволюция отдельных компонентов *Экосистемы OSTIS* (в частности, отдельных *ostis-систем*) не осуществлялась в ущерб эволюции *Экосистемы OSTIS* в целом, например, путём привнесения эклектичности, многообразия форм решения похожих проблем, путем ослабления фундаментального требования максимально возможной простоты и логичности принципов, лежащих в основе *Экосистемы OSTIS*.

Существенно подчеркнуть, что эволюция *ostis-систем* и *Экосистем OSTIS* в целом сводится к коллективному реинжинирингу *баз знаний ostis-систем*, что, в свою очередь сводится к:

- "ручной" генерации предлагаемых дополнительных (новых) знаний в базу знаний указываемой *ostis-системы*;
- "ручной" генерации предлагаемых изменений текущего состояния базы знаний указываемой *ostis-системы*;
- автоматическому назначению компетентных и заинтересованных рецензентов;
- "ручному" рецензированию каждого поступившего предложения, результатом чего является:
 - либо полное одобрение;
 - либо полное неодобрения с предлагаемой аргументацией;
 - либо детальная рекомендация доработки, предположения;
- автоматическому назначению достаточно широкого круга компетентных и заинтересованных специалистов для утверждения поступившего предложения (после получения одобрения от всех назначенных экспертов);
- автоматическому принятию решения по одобрению поступившего предложения на основании мнения всех привлечённых экспертов и специалистов.

Таким образом в *базе знаний* каждой *ostis-системы* можно (и нужно!) фиксировать весь процесс обсуждения каждого поступившего предложения с указанием (1) моментов времени всех привлечённых событий; (2) участников каждого события (авторов предложений, авторов рецензий участников голосования).

Кроме того, каждая *ostis-система*, анализируя процесс использования хранимых ею знаний в процессе эксплуатации, может оценивать частоту непосредственного и опосредованного использования этих знаний, т.е. может оценить степень востребованности этих знаний.

Следовательно, в перспективе *Экосистема OSTIS* может с достаточно высокой степенью объективности может оценивать объем и значимость вклада каждого специалиста в развитие распределенной базы знаний *Экосистемы OSTIS*. Это является фундаментальной основой для формирования достаточно объективного (честного) *рынка знаний*.]

Рынок знаний, реализуемый в рамках Экосистемы OSTIS

⇒ *правило для авторов в рамках Экосистемы OSTIS**:

[знания, предлагаемые для рецензирования, согласования, утверждения и публикации в базе знаний соответствующей *ostis-системы* должны быть специфицированы (указана *ostis-система*, атомарный раздел базы знаний, дата и время, автор, новый вид публикации, рынок знаний, защита авторского права не на уровне документов, а на уровне смысла.)]

⇒ *коллективное совершенствование базы знаний**:

[Абсолютно идеальных решений (в том числе проектных) не бывает. Поэтому (1) не надо бояться ошибок и (2) надо минимизировать степень ошибочности за счёт (2.1) оперативности исправления ошибок и (2.2) повышения качества (уровня) анализа при принятии решения путем (2.2.1) коллективного характера экспертизы, (2.2.2) достаточного количества привлекаемых экспертов и (2.2.3) учёта уровня осведомленности (квалифицированности и погруженности в соответствующую предметную область и онтологию). Для каждого эксперта, привлекаемого к принятию решения нужен постоянно уточняемый, по объективным критериям коэффициент осведомленности-авторитетности каждого эксперта к каждой конкретной предметной области.]

⇒ *правила редактирования Общей базы знаний коллектива интеллектуальной системы**:

[

- Если Вы в рамках базы знаний разрабатываемой Вами *ostis-системы* хотите ввести знак новой ранее не описываемой сущности, то Вы должны проверить, что эта сущность действительно не описывалась в рамках виртуальной базы знаний всей *Экосистемы OSTIS*
 - Если в результате такой проверки выяснилось, что указанная сущность уже рассматривалась, то Вы должны использовать введенный ранее основной внешний идентификатор этой сущности (Если он Вам не нравится, можете предложить, но пока не использовать, свой)
 - Если сущность не рассматривалась, нужно специфицировать, связать с семантически близким (особенно для понятий)

От толковых словарей и энциклопедий – к стройной семантической сети таких спецификации всех описываемых сущностей, которые позволяют установить (желательно автоматически) наличие или отсутствие в рамках технического состояния базы знаний синонимичного знака для любого нового знака, вводимого в базу знаний.]

⇒ *структура качественной спецификации**:

[Нужно стремиться:

- к однозначности такой спецификации;
- "координаты" в пространстве декомпозиций
- к семантической близости;
- сходства, отличия;

]

качество человеческой деятельности

:= [качество деятельности человеческого общества]

⇒ *пояснение**:

[Поскольку человеческого общество в целом является кибернетической системой, (которая принадлежит классу иерархических многоагентных систем, качество деятельности человеческого общества можно оценивать по критериям качества кибернетических систем.

На основании этих критериев можно оценивать:

- качество информационной среды, формируемой человеческим обществом, т.е. качество накапливаемой и общедоступной информации;
- качество текущего состояния общечеловеческих знаний;
- качество методов и технологий, используемых для решения задач как в рамках накопленных человеческим обществом знаний, так и в рамках внешней среды человеческого общества;
- качество организаций человеческой деятельности в целом;
- обучаемость (темпы эволюции) человеческого общества в целом.

Современный этап развития науки и техники характерен тем, что при оценке качества научно-технических результатов акцентируется внимание на новизне результатов, на их отличиях от текущего положения дел. Это создает почву и для имитации этой новизны и для увеличения барьеров между различными дисциплинами, что существенно препятствует конвергенции и интеграции различных дисциплин. Указанная конвергенция и интеграция, в частности, необходима для комплексной автоматизации всех видов человеческой деятельности в рамках smart-общества. Очевидно, что основной такой комплексной автоматизации должна быть **Общая формальная теория человеческой деятельности.**]

уровень конвергенции и интерации человеческой деятельности и её результатов

⇐ *свойство-предпосылка**:

Качество человеческой деятельности

⇒ *пояснение**:

[Повышение уровня конвергенции и интеграции различных видов человеческой деятельности и, соответственно, результатов этой деятельности является важнейшим фактором (важнейшим направлением) повышения качества (эффективности) человеческой деятельности, а, следовательно, и качества самого человеческого общества как сложной распределенной социотехнической кибернетической системы.]

⇒ *вопрос**:

[Что является главным препятствием существенному повышению уровня конвергенции и интеграции человеческой деятельности.]

⇒ *ответ**:

[Главным препятствием повышению уровня конвергенции и интеграции человеческой деятельности является то, что на текущем этапе эволюции человеческого общества основным механизмом эволюции является конкуренция. Конкуренция предполагает противопоставление результатов своей деятельности результатам конкурентов. т.е. акцентирует внимание на отличиях, новизне, преимуществах своих результатов по отношению к результатам своих конкурентов. При этом мысль о целесообразности объединения усилий со своими конкурентами чаще обусловлена стремлением повысить уровень конкурентоспособности и прибыли по отношению к другим более сильным конкурентам и значительно реже обусловлена искренним стремлением получить более качественный результат.

Таким образом, повышение уровня конвергенции и интеграции всех видов человеческой деятельности требует весьма сложного перехода от использования механизма конкуренции в её современном виде к созданию мощной технологической основы, обеспечивающей широкое взаимовыгодное сотрудничество и гарантированные возможности самореализации каждого человека и каждого коллектива. Фундаментом указанной технологической основы может и должен стать общечеловеческий рынок знаний, который построен на базе сети интеллектуальных компьютерных систем и в рамках которого фиксируется и объективно оценивается значимость вклада каждого человека и каждого коллектива.]

/*-*****-*/

автоматизация человеческой деятельности

⇒ *вопрос**:

- [В чем заключаются проблемы комплексной автоматизации человеческой деятельности]
- [Как автоматизировать участие человека одновременно в нескольких разных действиях (разных областях деятельности), принадлежащих в общем случае разным видам деятельности]

Экосистема OSTIS

⇒ *примечание**:

[Во многом разработка принципов организации взаимодействия интеллектуальных компьютерных систем (ostis-систем) и людей, входящих в состав *Экосистемы OSTIS* должна опираться на анализ того, как взаимодействие осуществляется между людьми, когда основные проблемы возникают из-за:

- отсутствия взаимопонимания (семантической совместимости),
- противоречий между целями различных субъектов,
- имитации целенаправленных действий,
- нарушений каких-либо соглашений, договоренностей и даже законов (правил поведения и обязанностей).

Общая (общедоступная) база знаний всей *Экосистемы OSTIS*, а также корпоративная база знаний каждого *ostis-сообщества*, входящего в состав *Экосистемы OSTIS*, является распределенной, но при этом обязательно целостной. Она поддерживается группой специальных *ostis-систем*, являющихся *порталами знаний* по самым различным областям. Для *Технологии OSTIS* роль такого портала знаний выполняет *Метасистема IMS.ostis*. *Экосистема OSTIS* представляет собой многоагентную социотехническую систему, в которой каждая *индивидуальная ostis-система*, входящая в состав *Экосистемы OSTIS*, каждый пользователь указанных *ostis-систем*, а также каждое *ostis-сообщество*, входящее в *Экосистему*, является её самостоятельным *субъектом** (когнитивным агентом). При этом каждый субъект *Экосистемы OSTIS* должен соблюдать определенные правила, обеспечивающие качественную (эффективную) эксплуатацию и эволюцию *Экосистемы OSTIS*.]

⇒ *резюме**:

[Сама идея комплексной автоматизации всех видов человеческой деятельности предполагает необходимость:

- разработки достаточно детальных формальных теорий всех видов человеческой деятельности, причем, теорий, доведенных до уровня разделов баз знаний соответствующих корпоративных компьютерных систем – это, фактически, строгое описание стандартов различных видов человеческой деятельности, доведенное до такого уровня, чтобы соответствующая корпоративная система "*понимала*", в какой деятельности она участвует, и могла быть активным и полноценным субъектом (участником) этой деятельности;
- серьезного отношения и научного подхода к формализации различных видов человеческой деятельности, к разработке самых различных стандартов;
- глубокой конвергенции различных областей (разделов) человеческой деятельности и, соответственно, различных видов человеческой деятельности, осуществляемой в условиях достигнутого уровня автоматизации этой деятельности. Это предполагает необходимость рассмотрения каждого вида человеческой деятельности в контексте *Общей теории человеческой деятельности* в условиях текущего состояния уровня автоматизации этой деятельности;
- обеспечения высоких темпов эволюции и, следовательно, высокого уровня *гибкости Общей теории человеческой деятельности* и теорий (стандартов) каждого *вида деятельности* в силу их большой зависимости от текущего уровня автоматизации;
- автоматизации взаимодействия субъектов не только внутри каждой области (раздела) человеческой деятельности, но и между этими областями (разделами), что предполагает автоматизацию представительства каждой области (раздела) человеческой деятельности во множестве всех таких областей;
- понимания того, что эффективность человеческой деятельности во многом определяется скоординированностью, адекватностью, грамотностью поведения каждого субъекта. Поэтому автоматизация человеческой деятельности должна быть направлена на более глубокую координацию этой деятельности на основе учета смысла и целей этой деятельности. А это "превращает" средства автоматизации в полноценных субъектов коллективной деятельности

]

автоматизация человеческой деятельности

⇒ *примечание**:

[Рассмотрение комплексной автоматизации человеческой деятельности в области *Искусственного интеллекта* естественным образом можно расширить (обобщить) до рассмотрения комплексной автоматизации человеческой деятельности в целом.]

⇒ *проблемы текущего состояния**:

[Для того, чтобы обеспечить качественную автоматизацию любой области человеческой деятельности с помощью интеллектуальных компьютерных систем, необходимо построить формальную модель этой области деятельности и довести эту модель до такого уровня формализации, чтобы она могла стать частью базы знаний интеллектуальной компьютерной системы, используемой для автоматизации указанной области человеческой деятельности. Очевидно, что, чем субъекты, участвующие в какой-либо коллективной деятельности, (люди и интеллектуальные компьютерные системы) лучше понимают суть, цели, критерии качества указанной коллективной деятельности, тем выше качество выполнения этой деятельности. В состав формальной модели автоматизируемой области человеческой деятельности входит спецификация:

- объектов деятельности;
- среды деятельности;
- инструментов (инструментальных средств) деятельности;
- субъектов деятельности;
- текущего состояния деятельности (как процесса) – в том числе, спецификация действий (целей, задач), выполняемых в текущий момент;
- формулировка различного рода учитываемых закономерностей – в том числе, правил поведения субъектов деятельности;
- спецификация всех используемых субъектами деятельности методов выполнения сложных действий (решения задач).

Примерами автоматизируемых областей человеческой деятельности являются:

- процесс взаимодействия "умного" дома с его жильцами и посетителями;
- процесс взаимодействия "умного" предприятия, выпускающего определенного вида продукцию, с его сотрудниками;
- процесс взаимодействия студентов и преподавателей в рамках "умной" кафедры, осуществляющей подготовку молодых специалистов по какой-либо инженерной специальности;
- процесс взаимодействия постояльцев, посетителей и сотрудников "умного" отеля;
- процесс взаимодействия посетителей и сотрудников "умного" музея;
- процесс взаимодействия пациентов и медицинского персонала "умной поликлиники", "умной" больницы;
- процесс взаимодействия граждан и чиновников в рамках "умной" администрации некоторого региона;
- процесс взаимодействия жителей и гостей в рамках "умного" города;
- и т. д.

Но для комплексной автоматизации человеческой деятельности в целом (Объединенной человеческой деятельности) автоматизации отдельных областей человеческой деятельности явно не достаточно, поскольку тесные связи между различными областями человеческой деятельности требуют автоматизации не только деятельности внутри каждой из этих областей, но и внешней деятельности, обусловленной необходимостью взаимодействия между различными областями деятельности, например, в рамках более крупных областей деятельности. Так, например, каждое предприятие взаимодействует со своими поставщиками и потребителями. Очевидно, что автоматизация такой внешней деятельности и, тем более, автоматизация с использованием интеллектуальных компьютерных систем существенно упрощается, если будут совпадать (будут унифицированы) принципы, лежащие в основе автоматизации каждой области деятельности, а также принципы автоматизации крупных областей деятельности, в состав которых входит некоторое количество более "мелких" областей человеческой деятельности.

Таким образом, для комплексной автоматизации человеческой деятельности в целом с применением интеллектуальных компьютерных систем и для обеспечения эффективной интеграции различных областей человеческой деятельности необходима разработка *Общей формальной теории человеческой деятельности*, которая объединила бы формальные модели всевозможных областей человеческой деятельности, а также включила бы:

- формальные теории различных видов человеческой деятельности. Поскольку каждый вид человеческой деятельности – это класс однотипных областей человеческой деятельности, формальная теория

каждого вида человеческой деятельности – это формальное представление стандарта соответствующего класса областей человеческой деятельности. Так, например, можно говорить о формальной модели конкретного предприятия рецептурного производства (например, предприятие "Савушкин продукт", выпускающего молочную продукцию), но можно говорить и о формальной теории всего класса предприятий рецептурного производства – о формальном представлении стандарта ISA-88. Формальная теория каждого вида человеческой деятельности включает в себя формальное описание технологии, обеспечивающей осуществление каждой области (фрагмента) человеческой деятельности, принадлежащей указанному виду деятельности. В описание технологии входит описание используемых методов, средств и основных объектов и субъектов деятельности;

- четкую иерархическую декомпозицию *Объединенной человеческой деятельности* по нескольким признакам. Основными признаками такой декомпозиции являются региональный признак и целевая направленность деятельности. По региональному признаку на высшем уровне иерархии выделяются такие области человеческой деятельности, как Деятельность Франции, Деятельность Германии и далее деятельность всех стран. По признаку целевой направленности на высшем уровне иерархии выделяются: Научно-исследовательская деятельность человечества, Проектная деятельность человечества, Производственная деятельность человечества, Образовательная деятельность человечества, Здравоохранительная деятельность человечества, Природоохранная деятельность человечества, Административная деятельность человечества и др. Дальнейшая декомпозиция областей человеческой деятельности по признаку целевой направленности выделяет такие области деятельности, как:

- □ Научно-исследовательская деятельность человечества в области Математики
- □ Научно-исследовательская деятельность человечества в области Лингвистики
- □ и др.

Заметим при этом, что, в отличие от "чисто" научных дисциплин, дисциплины научно-технического типа (например, дисциплина *Искусственный Интеллект*) представляют собой симбиоз фрагментов (областей) деятельности, принадлежащих разным видам деятельности:

- □ научно-исследовательской деятельности;
- □ деятельности по разработке технологии проектирования (CAD);
- □ деятельности по разработке технологии производства (CAM);
- □ проектная деятельность;
- □ производство спроектированного объекта;
- □ образовательной деятельности;
- □ бизнес-деятельности.

Кроме указанных областей человеческой деятельности выделяются области, соответствующие различным сочетаниям значений указанных признаков декомпозиции областей человеческой деятельности. Примерами таких областей являются: Научно-исследовательская деятельность Франции, Образовательная деятельность Германии. Подчеркнем то, что количество областей человеческой деятельности, выделенных в результате указанной иерархической декомпозиции *Объединенной человеческой деятельности*, является, хоть и не очень большим, но конечным в каждый момент времени.

При этом Общая формальная теория человеческой деятельности должна быть ориентирована:

- на унификацию формального описания самых различных технологий для самых различных областей человеческой деятельности;
- на унификацию формального описания всевозможных видов человеческой деятельности;
- на унификацию формального описания связей между различными областями и видами человеческой деятельности, различными субъектами деятельности, объектами, средствами (инструментами);
- на глубокую конвергенцию всех видов человеческой деятельности, областей человеческой деятельности, используемых методов.

1
⇒ что делать*:

- [Необходим переход от локальной автоматизации различных областей и видов человеческой деятельности путем независимой друг от друга разработки систем автоматизации бизнес-процессов даже близких по виду деятельности предприятий к комплексной автоматизации человеческой деятельности в целом прежде всего для обеспечения совместимости различных областей деятельности и исключения ужасающего и никому не нужного дублирования (многообразия форм) автоматизации аналогичных бизнес-процессов]
- [Все многообразие человеческой деятельности необходимо четко стратифицировать, доведя эту стратификацию до строгого формального представления]

- [Необходимо
 - четко выделить все виды человеческой деятельности, соответствующие текущему уровню развития человеческого общества;
 - построить четкую иерархию этих видов на основании отношения, связывающего виды человеческой деятельности с их подвидами;
 - унифицировать человеческую деятельность в рамках каждого выделенного вида, разработав соответствующие стандарты, для каждого из которых построить четкую систему используемых понятий;
 - довести указанные стандарты до такого уровня формализации, чтобы они стали частью базы знаний интеллектуальной системы автоматизации соответствующего вида человеческой деятельности.
-]
 - [Необходимо обеспечить конвергенцию, семантическую совместимость и глубокую интеграцию различных видов и областей человеческой деятельности путем:
 - согласования систем понятий, соответствующих стандартам разных видов человеческой деятельности, и особенно согласования систем понятий между стандартами видов и подвидов человеческой деятельности;
 - представления стандарта каждого вида человеческой деятельности в виде формальной онтологии;
 - построения такой иерархической системы формальных онтологий, соответствующих всевозможным видам человеческой деятельности, в которой обеспечивалась бы конвергенция и семантическая совместимость онтологий, входящих в эту систему, а также наследование свойств от онтологии каждого вида человеческой деятельности к онтологии каждого подвида этого вида человеческой деятельности.

)
⇒

следовательно*:

[Таким образом, в целях повышения эффективности автоматизации человеческой деятельности и, в первую очередь, в целях существенного снижения трудозатрат на такую автоматизацию необходимо с точки зрения общей теории систем фундаментально переосмыслить современную организацию человеческой деятельности, поскольку автоматизация беспорядка приводит к ещё большему беспорядку. На этом пути имеется только одно препятствие – противодействие лени с высоким уровнем эгоизма, которым современный беспорядок организации человеческой деятельности выгоден.]

следует отличать

- ∃ {
 - вид человеческой деятельности
 - ∃ пример':
 - научно-исследовательская деятельность
 - проектирование
 - := [проектная деятельность]
 - ⊃ проектирование интеллектуальной компьютерной системы
 - ⊃ проектирование ostis-системы
 - область человеческой деятельности
 - ∃ пример':
 - Научно-исследовательская деятельность в области Искусственного интеллекта
 - ∈ научно-исследовательская деятельность
 - ⇒ часть*:
 - Научно-исследовательская деятельность РАИИ
 - Проектирование Metасистемы IMS.ostis
 - ∈ проектирование ostis-системы
 - подвид человеческой деятельности*
 - ⊂ включение*
 - := [подмножество*]
 - подобласть человеческой деятельности*
 - ⊂ часть*

информационная технология

⇒ пояснение*:

[Множество технологий, связанных с проектированием и производством компьютерных систем и их компонентов, с эксплуатацией компьютерных систем, а также с их использованием в качестве инструмента в составе самых различных технологий. В рамках различных информационных технологий компьютерные системы рассматриваются как инструментальные средства, как вспомогательные субъекты, обеспечивающие автоматизацию соответствующих видов деятельности. Но в некоторых информационных технологиях компьютерные системы являются также и объектами автоматизируемых видов деятельности. Примерами таких технологий являются:

- технология проектирования компьютерных систем;
- технология реализации (сборки) компьютерных систем;
- технология обновления компьютерных систем.

]

- Э *Комплекс современных информационных технологий*
- Э *Комплекс современных технологий искусственного интеллекта*
- Э *Технология OSTIS*

автоматизация человеческой деятельности

:= [человеческая деятельность, направленная на повышения уровня автоматизации человеческой деятельности, а также на повышение качества (в том числе, уровня интеллекта) человеческого общества как многоагентной кибернетической системы]

С *вид человеческой деятельности*

⇒ *примечание*:*

[Важнейшим этапом автоматизации человеческой деятельности в перспективе должен стать переход к существенно более высокому уровню *интеллекта человеческого общества* как целостной кибернетической системы путем преобразования современного человеческого общества в сообщество взаимодействующих между собой людей и интеллектуальных компьютерных систем. Такое сообщество иногда называют smart-обществом, обществом 5.0.

Особо подчеркнем то, что переход к такому интеллектуальному обществу требует существенного переосмысления современной организации различных видов человеческой деятельности. Прежде всего, следует подчеркнуть, что эффективность (коэффициент полезного действия) современной организации человеческой деятельности в целом ужасающе низка, а, как известно, автоматизация беспорядка (даже с помощью интеллектуальных компьютерных систем) приводит к ещё большему беспорядку.]

вид человеческой деятельности

:= [Множество всевозможных видов человеческой деятельности]

⇒ *разбиение*:*

Разбиение Множества видов человеческой деятельности по степени их автоматизируемости

- = {
- *вид человеческой деятельности, который принципиально может быть автоматизирован полностью*
:= [Множество полностью автоматизируемых видов человеческой деятельности]
 - *вид человеческой деятельности, который может быть автоматизирован только частично*
:= [Множество частично автоматизируемых видов человеческой деятельности]
 - *вид человеческой деятельности, который принципиально никак не может быть автоматизирован*
:= [Множество неавтоматизируемых видов человеческой деятельности, которые могут быть выполнены только "вручную" (точнее самими людьми с возможным использованием каких-либо "пассивных" инструментов – топора, лопаты и т. п.)]
- }

вид человеческой деятельности, который принципиально может быть автоматизирован полностью

⇒ *примечание*:*

[Есть виды человеческой деятельности, которые принципиально могут быть автоматизированы полностью, но в текущий момент эта автоматизация не полна. Это, например, частично автоматизированная деятельность по производству спроектированных искусственных объектов. Здесь важна четкость распределения "обязанностей" между различными средствами автоматизации и поэтапное исключение неавтоматизированных действий, "вручную" выполняемых людьми (например, сотрудниками производственных предприятий), т. е. поэтапная автоматизация этих действий.]

/*****/

автоматизация человеческой деятельности⇒ *вопрос**:

[Почему для комплексной автоматизации человеческой деятельности целесообразно использовать семантически совместимые, договороспособные, самостоятельные интеллектуальные компьютерные системы, которым можно делегировать права на принятие некоторых решений.]

⇒ *вопрос**:

[Почему для повышения уровня комплексной *автоматизации человеческой деятельности* необходим переход от современных (традиционных) *компьютерных систем* и соответствующих им информационных технологий, а также от *современных интеллектуальных компьютерных систем* и соответствующих им современных технологий искусственного интеллекта к *интеллектуальным компьютерным системам нового поколения* и к соответствующей им Комплексной технологии проектирования таких систем.]

⇒ *ответ**:

[В силу отсутствия унификации представления обрабатываемой информации в традиционных компьютерных системах и, как следствие, отсутствия совместимости этих систем как на синтаксическом уровне, так и на семантическом уровне, принципиально не существует универсального метода системной интеграции традиционных компьютерных систем и, следовательно, невозможна полная автоматизация решения этой задачи. Системная интеграция традиционных компьютерных систем практически всегда осуществляется "вручную" с учетом индивидуальной специфики каждой интегрируемой системы и, следовательно, является весьма трудоемкой и требующей высокой квалификации разработчиков.

Но на данном этапе эволюции компьютерных систем крайне актуальной является полная автоматизация их интеграции без какого бы то ни было участия разработчиков и тем более конечных пользователей. Если компьютерные системы не приобретут способность самостоятельно взаимодействовать между собой в целях решения сложных комплексных задач, то эффективность использования человечеством интенсивно расширяемого многообразия весьма полезных и качественно реализованных информационных ресурсов и сервисов будет весьма низкой. Традиционно компьютерные технологии позволяют реализовать любую модель обработки информации (в том числе и любую интеллектуальную модель – нейросетевую, логическую и т.д.). Однако актуальным является не реализация самих этих моделей, а их интеграция, что требует обеспечения синтаксически и семантически совместимых компьютерных систем и полной автоматизации их системной интеграции. Следовательно необходим переход на принципиально новое поколение компьютерных технологий и, в частности, на принципиально новое поколение самих компьютеров, ориентированных на решение проблем совместимости компьютерных систем и полной автоматизации их системной интеграции. Таким образом, дальнейшее повышение уровня автоматизации различных видов человеческой деятельности потребует перехода на принципиально новый уровень информационных технологий – от современных (традиционных) *компьютерных систем* к компьютерным системам, имеющим существенно более *высокий уровень интеллекта* и способным не только индивидуально решать достаточно сложные (в том числе, интеллектуальные) задачи, но и эффективно самостоятельно взаимодействовать между собой, координируя свою деятельность при решении задач, принадлежащих априори неизвестным (заранее не предусмотренным) классам задач и требующих коллективного (корпоративного) решения.

Основные проблемы автоматизации *человеческой деятельности* в настоящее время лежат не в области разработки средств автоматизации решения различных конкретных классов задач (в том числе и весьма сложных, интеллектуальных, труднорешаемых задач), а в области системной интеграции этих средств в комплексы, компоненты которых способны самостоятельно кооперироваться для совместного (коллективного) решения сложных задач. Но для этого указанные компоненты должны уметь согласовывать, координировать свои действия, должны понимать друг друга, должны быть семантически совместимы.]

интеллектуальная компьютерная система⇒ *примечание**:

[Различные интеллектуальные компьютерные системы могут быть эффективно использованы в качестве средств автоматизации самых различных видов человеческой деятельности. Но, поскольку все виды человеческой деятельности взаимосвязаны (как минимум потому, что каждый человек может одновременно участвовать сразу в нескольких видах деятельности, причем в разные моменты времени этот набор видов деятельности для каждого человека может быть различным), интеллектуальная компьютерная система автоматизации каждого вида человеческой деятельности должна эффективно взаимодействовать с другими интеллектуальными компьютерными системами, осуществляющими автоматизацию других видов человеческой деятельности. Другими словами, необходимо переходить от автоматизации отдельных видов человеческой деятельности к автоматизации комплекса всех видов человеческой деятельности. Для этого необходимо:

- не просто достаточно детально разработать теорию каждого вида деятельности, выделив (1) все классы автоматизируемых действий, (2) все классы неавтоматизируемых действий, (3) соответствующие этим классам методы выполнения действий (в частности, это могут быть обобщенные бизнес-процессы), которые по сравнению с используемыми в настоящий момент могут потребовать существенного реинжиниринга бизнес-процессов;
- но и представить эти теории в формализованном и унифицированном виде в качестве фрагментов баз знаний соответствующих интеллектуальных компьютерных систем, обеспечив при этом высокую степень конвергенции этих теорий.

]

⇒ возможное амплуа*:

- *средство автоматизации проектирования*
 - ⊃ *средство автоматизации проектирования интеллектуальных компьютерных систем*
 - ⇒ *примечание*:*
[В силу большой сложности процесса проектирования интеллектуальных компьютерных систем для автоматизации этого процесса необходимо использовать именно интеллектуальные компьютерные системы.]
- *средство автоматизации производства*
- *средство повышения качества эксплуатации сложного объекта*
 - ⊃ *средство help-поддержки конечных пользователей*
 - ⊃ *средство управления процессом повышения качества деятельности конечных пользователей*
 - ⊃ *средство поддержки оптимальных эксплуатационных свойств эксплуатируемого объекта*
 - ⇒ *пояснение*:*
[Здесь имеется в виду мониторинг состояния эксплуатируемого объекта, контроль условий эксплуатации, своевременная профилактика и ремонт.]
 - ⊃ *средство поддержки совершенствования эксплуатируемого объекта в ходе его эксплуатации*
 - ⇒ *примечание*:*
[Для интеллектуальных компьютерных систем все средства повышения качества их эксплуатации целесообразно встраивать в эти системы. Имеется в виду слияние нескольких интеллектуальных компьютерных систем в одну интегрированную.]
- *средство автоматизации научно-исследовательской деятельности в рамках заданной научной дисциплины*
- *средство автоматизации образовательной деятельности*
 - ⇒ *примечание*:*
[Автоматизация образовательной деятельности может осуществляться в рамках:
 - заданной учебной дисциплины;
 - заданной учебной специальности;
 - заданного учебного заведения;
 - заданного государства.
- *средство автоматизации бизнес-деятельности в заданной научно-технической области*
 - ⇒ *примечание*:*
[Здесь важна автоматизация контроля за реализацией всех направлений организационной деятельности с учетом разработанных и постоянно уточняемых и корректируемых планов, а также с учетом согласованных приоритетов.]
- *средство автоматизации деятельности в области здравоохранения*
- *средство автоматизации административной деятельности*
- *средство автоматизации деятельности жилищно-коммунального хозяйства*
- *юриспруденция*
- *правоохранительная деятельность*
- *транспорт*

Экосистема OSTIS

⇒ *примечание*:*

[Экосистема OSTIS является основой для перевода уровня информатизации различных областей человеческой деятельности на принципиально новый уровень, а также для интеграции соответствующих проектов – "Общество 5.0", "Industry 4.0", "University 3.0", "Умный дом", "Умный город" и других (без интеллектуальных

компьютерных систем все эти проекты невозможны).

Все эти проекты должны быть приведены в единую стройную иерархическую систему взаимосвязанных проектов, охватывающих весь объем и многообразие человеческой деятельности.]

}

/ Завершили Сегмент “Уточнение Понятия Экосистемы OSTIS” */*

}

/ Завершили Раздел “Анализ методологических проблем современного состояния работ в области Искусственного интеллекта” */*

/* Раздел *****/

Предметная область и онтология информационных конструкций и языков

⊃=

{

∈

раздел базы знаний

⇒

дочерний раздел*:

- Описание внутреннего языка *ostis-систем*
- Предметная область и онтология внешних идентификаторов знаков, входящих в информационные конструкции внутреннего языка *ostis-систем*
- Описание языка графического представления информационных конструкций в *ostis-системах*
- Описание языка линейного представления информационных конструкций в *ostis-системах*
- Описание языка структурированного представления информационных конструкций в *ostis-системах*

⇒

аннотация*:

[Поскольку предлагаемая Вашему вниманию *Публикация Стандарта OSTIS-2021* представляет собой внешний текст основной части базы знаний *ostis-системы (Метасистемы IMS.ostis)*, необходимо сразу во вводном разделе этого Стандарта пояснить основные принципы, лежащие в основе внутреннего представления баз знаний в памяти *ostis-систем*, а также некоторые правила и условные обозначения, используемые в оформлении внешних текстов баз знаний *ostis-систем*.

Подчеркнем, что все эти принципы, правила и условные обозначения детально рассмотрены в соответствующих разделах *Стандарта OSTIS*, но некоторые из них необходимо пояснить до начала ознакомления с основными положениями *Технологии OSTIS*. Фактически речь идет о кратком руководстве конечных пользователей *ostis-систем*.

Прежде, чем рассматривать *внутренний язык** и *внешние языки**, используемые *ostis-системами*, необходимо уточнить понятие *информационной конструкции*, понятие *знака*, понятие *текста*, понятие *языка*.]

Предметная область информационных конструкций и языков

∈

предметная область

∋

максимальный класс объектов исследования¹:

информационная конструкция

∋

класс объектов исследования¹:

- дискретная информационная конструкция
- элемент дискретной информационной конструкции
- соответствие, заданное на множестве дискретных информационных конструкций
- знак
- знаковая конструкция
- язык
- семантически выделяемый класс дискретных информационных конструкций
- язык *ostis-системы*
- ограничитель
- идентификатор
- имя
- предложение
- *sc.g-текст*
- *sc.s-текст*
- *sc.n-текст*

∋

исследуемый класс классов первичных объектов исследования¹:

- отношение, заданное на множестве элементов дискретных информационных конструкций[∧]
- параметр, заданный на множестве дискретных информационных конструкций[∧]
- отношение, заданное на множестве знаков[∧]
- отношение, заданное на множестве знаковых конструкций[∧]
- класс знаковых конструкций[∧]
- параметр, заданный на множестве знаковых конструкций[∧]
- отношение, заданное на множестве языков[∧]
- параметр, заданный на множестве языков[∧]

∋

исследуемое отношение¹:

- операционная семантика информационной конструкции*
- алфавит*

- *ограничители**
- *разделители**
- *внешний идентификатор**
- *предложения**

ostis-система

:= [компьютерная система, построенная по Технологии OSTIS]

информационная конструкция

:= [информация]

:= [конструкция (структура), содержащая некоторые сведения о некоторых сущностях]

⇒ *примечание**:

[Форма представления ("изображения", "материализации"), форма структуризации (синтаксическая структура), а также *смысл** (денотационная семантика) *информационных конструкций* могут быть самыми различными.]

дискретная информационная конструкция

С *информационная конструкция*

⇒ *пояснение**:

[Каждая *дискретная информационная конструкция* — это *информационная конструкция*, смысл которой задается (1) множеством элементов (синтаксически атомарных фрагментов) этой информационной конструкции, (2) алфавитом этих элементов — семейством классов синтаксически эквивалентных элементов информационной конструкции, (3) принадлежностью каждого элемента информационной конструкции соответствующему классу синтаксически эквивалентных элементов информационной конструкции, (4) конфигурацией связей инцидентности между элементами информационной конструкции.]

⇒ *следствие**:

[Форма представления элементов дискретной информационной конструкции для анализа её смысла не требует уточнения. Главным является (1) наличие простой процедуры выделения (сегментации) элементов дискретной информационной конструкции (2) наличие простой процедуры установления синтаксической эквивалентности разных элементов дискретной информационной конструкции и (3) наличие простой процедуры установления принадлежности каждого элемента дискретной информационной конструкции соответствующему классу синтаксически эквивалентных элементов (т.е. соответствующему элементу алфавита).]

элемент дискретной информационной конструкции

:= [синтаксически атомарный фрагмент (символ), входящий в состав дискретной информационной конструкции]

⇒ *примечание**:

[Поскольку дискретные информационные конструкции могут иметь общие элементы (атомарные фрагменты) и даже некоторые из них могут быть частями других информационных конструкций, элемент дискретной информационной конструкции может входить в состав сразу нескольких информационных конструкций.]

← *второй домен**:

*элемент дискретной информационной конструкции**

отношение, заданное на множестве элементов дискретных информационных конструкций[^]

Э *элемент дискретной информационной конструкции**

:= [быть элементарным (синтаксически атомарным) фрагментом заданной дискретной информационной конструкции*]

:= [быть элементом (атомарным фрагментом) заданной дискретной информационной конструкции*]

:= [Бинарное ориентированное отношение, каждая пара которого связывает (1) знак некоторой дискретной информационной конструкции и (2) знак одного из элементов этой дискретной информационной конструкции*]

⇒ *второй домен**:

элемент дискретной информационной конструкции

Э *синтаксическая эквивалентность элементов дискретных информационных конструкций**

:= [быть синтаксически эквивалентными элементами (атомарными фрагментами) одной и той же или разных дискретных информационных конструкций, т.е. элементами, принадлежащими одному и тому же классу синтаксически эквивалентных элементов дискретных информационных конструкций*]

Э *инцидентность элементов дискретных информационных конструкций**

Є *бинарное ориентированное отношение[^]*

- ⇒ *примечание**:
 [Для *линейных информационных конструкций* – это последовательность элементов (символов), входящих в состав этих конструкций.
 Для дискретных информационных конструкций, конфигурация которых имеет нелинейный характер, отношение инцидентности их элементов может быть разбито на несколько частных отношений инцидентности, каждое из которых является подмножеством объединенного отношения инцидентности. Например, для двухмерных дискретных информационных конструкций это (1) инцидентность элементов информационных конструкций "по горизонтали" и (2) инцидентность элементов информационных конструкций "по вертикали".]
- Э *неэлементарный фрагмент дискретной информационной конструкции**
 := [быть дискретной информационной конструкцией, которая является подструктурой для заданной дискретной информационной конструкции, в состав которой входит (1) подмножество элементов заданной информационной конструкции и, соответственно, (2) подмножество пар инцидентности элементов заданной информационной конструкции.]
- Э *алфавит дискретной информационной конструкции**
 := [быть семейством попарно непересекающихся классов синтаксически эквивалентных элементов заданной дискретной информационной конструкции*]
- Э *первичная синтаксическая структура дискретной информационной конструкции**
 := [быть *графовой структурой*, которая полностью описывает конфигурацию заданной *дискретной информационной конструкции* и которая включает в себя: (1) знаки всех тех классов синтаксически эквивалентных элементов, которым принадлежат элементы описываемой дискретной информационной конструкции, (2) знаки всех элементов (атомарных фрагментов) описываемой информационной конструкции, (3) пары, описывающие инцидентность элементов описываемой информационной конструкции, (4) пары, описывающие принадлежность элементов описываемой информационной конструкции соответствующим классам синтаксически эквивалентных элементов этой информационной конструкции.]
- Э *синтаксическая эквивалентность дискретных информационных конструкций**
 ⇒ *определение**:
 [Дискретные информационные конструкции T_i и T_j являются синтаксически эквивалентными в том и только в том случае, если между конструкцией T_i и конструкцией T_j существует изоморфизм, в рамках которого каждому элементу конструкции T_i соответствует синтаксически эквивалентный элемент конструкции T_j , т.е. элемент, принадлежащий тому же классу синтаксически эквивалентных элементов дискретных информационных конструкций. И наоборот.]
- Э *копия дискретной информационной конструкции**
 С *синтаксическая эквивалентность дискретных информационных конструкций**
 := *определение**:
 [быть дискретной информационной конструкцией, которая не только синтаксически эквивалентна заданной, но и содержит информацию о форме представления элементов копируемой информационной конструкции*]
- Э *семантическая эквивалентность дискретных информационных конструкций**
 ⇒ *определение**:
 [Информационная конструкция T_i и информационная конструкция T_j являются семантически эквивалентными в том и только в том случае, если каждая сущность (в том числе, и каждая связь между сущностями), описываемая в информационной конструкции T_i описывается также и в информационной конструкции T_j . И наоборот.]
- Э *семантическое расширение дискретной информационной конструкции**
 ⇒ *определение**:
 [Информационная конструкция T_j является семантическим расширением информационной конструкции T_i в том и только в том случае, если каждая сущность, описываемая в T_i , описывается также и в T_j , но обратное неверно.]
- Э *синтаксис информационной конструкции**
 := [синтаксическая структура информационной конструкции*]
 := [быть синтаксической структурой для заданной информационной конструкции*]
 := [Бинарное ориентированное отношение, каждая пара которого связывает знак некоторой информационной конструкции с синтаксической структурой этой конструкции*]
 := [описание того, из каких частей состоит заданная информационная конструкция и как эти части (фрагменты) связаны между собой*]
 ⇒ *первый домен**:
информационная конструкция

Э	<i>смысл*</i>
	:= [смысл информационной конструкции*]
	:= [денотационная семантика информационной конструкции*]
	:= [семантическая (смысловая) структура информационной конструкции*]
	:= [быть семантической (смысловой) структурой для заданной информационной конструкции*]
	:= [быть смыслом заданной информационной конструкции*]
	:= [быть явным (формальным) представлением того, какие сущности описывает данная информационная конструкция и как эти сущности связаны между собой*]
	:= [Бинарное ориентированное отношение, каждая пара которого связывает некоторую информационную конструкцию с ее смыслом (смысловой структурой)*]
Э	<i>операционная семантика информационной конструкции*</i>
	:= [правила трансформации заданной информационной конструкции*]
	:= [описание того, на основании каких правил можно осуществлять действия по преобразования (обработке, трансформации) заданной информационной конструкции, оставляя ее в рамках класса синтаксически и семантически правильных информационных конструкций*]
	:= [Бинарное ориентированное отношение, каждая пара которого связывает знак некоторой информационной конструкции со множеством правил ее трансформации*]
	<i>операционная семантика информационной конструкции*</i>
⇒	<i>второй домен*</i> : <i>операционная семантика информационной конструкции</i>
	<i>соответствие, заданное на множестве дискретных информационных конструкций</i>
Э	<i>соответствие между синтаксической структурой информационной конструкции и смыслом этой конструкции*</i>
	:= [Множество ориентированных пар, первым компонентом которых является ориентированная пара, состоящая из (1) знака синтаксической структуры некоторой информационной конструкции и (2) знака смысловой структуры этой конструкции, а вторым компонентом которых является множество ориентированных пар, связывающих фрагменты синтаксической структуры заданной информационной конструкции (которые описывают либо структуру фрагментов заданной конструкции, либо связи между фрагментами этой конструкции) с теми фрагментами смысловой структуры заданной информационной конструкции, которые семантически эквивалентны либо синтаксически представленным фрагментам заданной информационной конструкции, либо синтаксически представленным связям между такими фрагментами*]
С	<i>соответствие*</i>
	<i>параметр, заданный на множестве дискретных информационных конструкций[^]</i>
Э	<i>размерность дискретных информационных конструкций[^]</i>
	:= [типология дискретных информационных конструкций, определяемая их размерностью]
⇒	<i>пояснение*</i> : [Это параметр дискретных информационных конструкций, определяющий характер <u>инцидентности</u> элементов таких конструкций.]
Э	<i>линейная информационная конструкция</i>
	:= <i>пояснение*</i> : [дискретная информационная конструкция, каждый элемент которой может иметь не более двух инцидентных ему элементов (один слева, другой справа)]
Э	<i>двухмерная информационная конструкция</i>
	:= <i>пояснение*</i> : [дискретная информационная конструкция, каждый элемент которой может иметь не более четырех инцидентных ему элементов (слева-справа, сверху-снизу)]
Э	<i>трехмерная информационная конструкция</i>
	:= <i>пояснение*</i> : [дискретная информационная конструкция, каждый элемент которой может иметь не более шести инцидентных ему элементов (слева-справа, сверху-снизу, сзади-спереди)]
Э	<i>четырёхмерная информационная конструкция</i>
	:= <i>пояснение*</i> : [дискретная информационная конструкция, каждый элемент которой может иметь не более восьми инцидентных ему элементов (например, слева-справа, сверху-снизу, сзади-спереди, раньше-позже)]

Э	<p><i>графовая информационная конструкция</i></p> <p>:= <i>пояснение*</i>:</p> <p>[дискретная информационная конструкция, множество элементов которой разбивается на два подмножества — связи и узлы. При этом узлы могут иметь <u>неограниченное</u> количество инцидентных им связей]</p> <p>⇒ <i>примечание*</i>:</p> <p>[А в некоторых графовых информационных конструкциях и связи могут иметь неограниченное количество инцидентных им других связей]</p>
Э	<p><i>типология дискретных информационных конструкций, определяемая их носителем</i>[^]</p> <p>Э <i>некомпьютерная форма представления дискретных информационных конструкций</i></p> <p>⊃ <i>аудио-сообщение</i></p> <p>:= [речевое сообщение]</p> <p>:= [информационная конструкция, представленная в звуковой форме]</p> <p>⊃ <i>информационная конструкция, представленная на языке жестов</i></p> <p>⊃ <i>информационная конструкция, представленная в письменной форме</i></p> <p>⇒ <i>примечание*</i>:</p> <p>[Конкретный вид носителя для письменной формы представления информации может быть разным — бумага, папирус, береста, камень...]</p>
Э	<p><i>файл</i></p> <p>:= [компьютерная ("электронная") форма (формат) представления и хранения информационных конструкций]</p> <p>⇒ <i>примечание*</i>:</p> <p>[Представление информационных конструкций в виде файлов ориентировано на представление <u>дискретных</u> (!) информационных конструкций. Поэтому "файловое" представление недискретных информационных конструкций (например, различного рода сигналов) предполагает "дискретизацию" таких конструкций, т.е. преобразование их в дискретные. Так преобразуются аудио-сигналы (в частности, речевые сообщения), изображения, видео-сигналы и др.]</p>
Э	<p><i>уровень унификации представления синтаксически эквивалентных элементов дискретных информационных конструкций</i>[^]</p> <p>:= [уровень "членораздельности" дискретных информационных конструкций]</p> <p>⇒ <i>примечание*</i>:</p> <p>[Чем выше уровень унификации представления элементов дискретных информационных конструкций, тем проще реализуется (1) процедура выделения (сегментации) элементов дискретной информационной конструкции, (2) процедура установления синтаксической эквивалентности этих элементов и (3) процедура их распознавания, т.е. процедура установления их принадлежности соответствующим классам синтаксически эквивалентных элементов.]</p> <p>Э <i>дискретная информационная конструкция с низким уровнем унификации представления элементов</i></p> <p>⊃ <i>аудио-сообщение</i></p> <p>⊃ <i>информационная конструкция, представленная на языке жестов</i></p> <p>⊃ <i>рукопись или её копия</i></p> <p>Э <i>дискретная информационная конструкция с высоким уровнем унификации представления элементов</i></p> <p>⊃ <i>печатный текст</i></p> <p>⊃ <i>файл</i></p>
знак	<p>⇒ <i>пояснение*</i>:</p> <p>[фрагмент информационной конструкции, который условно представляет (изображает) некоторую описываемую сущность, которую называют денотатом знака]</p> <p>⇒ <i>разбиение*</i>:</p> <p>{</p> <ul style="list-style-type: none"> • <i>знак, являющийся элементом дискретной информационной конструкции</i> • <i>знак, являющийся неатомарным фрагментом дискретной информационной конструкции</i> <p>}</p> <p>⇒ <i>примечание*</i>:</p> <p>[Отсутствие знака, обозначающего некоторую сущность, не означает отсутствие самой этой сущности. Это означает только то, что мы даже не догадываемся о её существовании и, следовательно, не приступили к её исследованию.]</p>

отношение, заданное на множестве знаков[^]

⇒ *примечание**:

[Поскольку все знаки являются дискретными информационными конструкциями, множество знаков является областью задания всех отношений, заданных на множестве дискретных информационных конструкций. Тем не менее есть как минимум одно отношение, специфичное для множества знаков.]

Э *синонимия знаков**

⇒ *определение**:

[Знаки являются синонимичными в том и только в том случае, если они обозначают одну и ту же сущность.]

⇒ *примечание**:

[Синонимичные знаки могут быть синтаксически эквивалентными, а могут и не быть таковыми.]

знаковая конструкция

С *дискретная информационная конструкция*

:= *пояснение**:

[дискретная информационная конструкция, которая в общем случае представляет собой конфигурацию знаков и специальных фрагментов информационной конструкции, обеспечивающих структуризацию конфигурации знаков — различного вида разделителей и ограничителей]

⇒ *примечание**:

[Для некоторых знаковых конструкций и даже для некоторых языков необходимость в разделителях и ограничителях может отсутствовать.]

отношение, заданное на множестве знаковых конструкций[^]

Э *знак**

:= [быть знаком для заданной знаковой конструкции*]

Э *разделитель знаковой конструкции**

Э *разделители знаковой конструкции**

:= [Множество всех разделителей, входящих в состав заданной знаковой конструкции*]

Э *ограничитель знаковой конструкции**

Э *ограничители знаковой конструкции**

Э *семантическая смежность знаковых конструкций**

⇒ *определение**:

[Знаковые конструкции T_i и T_j являются смежными в том и только в том случае, если существуют синонимичные знаки T_i и T_j , один из которых входит в состав конструкции T_i , а второй — в состав конструкции T_j]

Э *конкатенация знаковых конструкций**

:= [декомпозиция заданной знаковой конструкции на последовательность знаковых конструкций*]

класс знаковых конструкций[^]

Э *семантически элементарная знаковая конструкция*

:= [знаковая конструкция, описывающая некоторую (одну) связь между некоторыми сущностями]

Э *семантически связанная знаковая конструкция*

:= *определение**:

[знаковая конструкция, которую можно представить в виде конкатенации семантически элементарных знаковых конструкций, каждая из которых семантически смежна предшествующей и последующей семантически элементарной знаковой конструкции]

параметр, заданный на множестве знаковых конструкций[^]

Э *семантическая связность знаковых конструкций[^]*

Э *семантически связанная знаковая конструкция*

Э *семантически несвязная знаковая конструкция*

Э *пример'*:

[В огороде бузина, а в Киеве дядька]

Э *наличие разделителей и ограничителей[^]*

Э *знаковая конструкция, содержащая разделители и-или ограничители*

Э *знаковая конструкция без разделителей и ограничителей*

язык	
:=	<i>пояснение*</i> : [класс знаковых конструкций, для которого существуют (1) общие правила их построения и (2) общие правила их соотнесения с теми сущностями и конфигурациями сущностей, которые описываются (отражаются) указанными знаковыми конструкциями]
:=	[класс знаковых конструкций, эквивалентных с точки зрения правил их построения и правил их семантической интерпретации]
⇒	<i>разбиение*</i> : { • язык, у которого все знаки, входящие в состав его знаковых конструкций, являются элементарными фрагментами этих конструкций ⇒ <i>примечание*</i> : [Для языков такого типа существенно упрощаются методы обработки их текстов.] • язык, у которого знаки, входящие в состав его знаковых конструкций, в общем случае не являются элементарными фрагментами этих конструкций }
⇒	<i>разбиение*</i> : { • язык, знаковые конструкции которого содержат разделители и ограничители • язык, знаковые конструкции которого не содержат разделителей и ограничителей ⇒ <i>следствие*</i> : [Знаковые конструкции такого языка состоят <u>только</u> из знаков.] }
	отношение, заданное на множестве языков[^]
:=	[отношение, область определения которого включает в себя множество всевозможных языков]
Э	<i>текст заданного языка*</i>
:=	[знаковая конструкция, принадлежащая заданному языку*]
:=	[синтаксически правильная (правильно построенная) знаковая конструкция заданного языка*]
:=	[синтаксически корректная и целостная знаковая конструкция для заданного языка*]
=	(синтаксически корректная знаковая конструкция для заданного языка* \cap синтаксически целостная знаковая конструкция для заданного языка*)
Э	<i>синтаксически корректная знаковая конструкция для заданного языка*</i>
:=	[знаковая конструкция, не содержащая синтаксических ошибок для заданного языка]
Э	<i>синтаксически целостная знаковая конструкция для заданного языка*</i>
Э	<i>синтаксически неправильная знаковая конструкция для заданного языка*</i>
=	(синтаксически некорректная знаковая конструкция для заданного языка* \cup синтаксически нецелостная знаковая конструкция для заданного языка*)
⊃	<i>синтаксически некорректная знаковая конструкция для заданного языка*</i>
⊃	<i>синтаксически нецелостная знаковая конструкция для заданного языка*</i>
Э	<i>знание, представленное на заданном языке*</i>
:=	[семантически правильный текст заданного языка*]
=	(семантически корректный текст заданного языка* \cap семантически целостный текст заданного языка*)
:=	[истинный текст заданного языка*]
:=	[истинное высказывание, представленное на заданном языке*]
Э	<i>семантически корректный текст заданного языка*</i>
:=	[текст заданного языка, не содержащий семантических ошибок, противоречащих признанным закономерностям и фактам*]
Э	<i>семантически целостный текст заданного языка*</i>
:=	[текст заданного языка, содержащий достаточную информацию для установления его истинности*]
Э	<i>семантически неправильный текст для заданного языка*</i>
=	(семантически некорректный текст для заданного языка* \cup семантически нецелостный текст для заданного языка*)
⊃	<i>семантически некорректный текст для заданного языка*</i>
⊃	<i>семантически нецелостный текст для заданного языка*</i>
Э	<i>алфавит*</i>
:=	[алфавит заданной информационной конструкции или заданного языка*]
:=	[семейство классов, синтаксически эквивалентных элементов (элементарных фрагментов) заданной информационной конструкции или информационных конструкций заданного языка*]

- Э *семейство классов синтаксически эквивалентных разделителей**
 := [семейство классов синтаксически эквивалентных разделителей, входящих в состав заданной информационной конструкции или в состав информационных конструкций заданного языка*]
- Э *семейство классов синтаксически эквивалентных ограничителей**
 := [семейство классов синтаксически эквивалентных ограничителей, входящих в состав заданной информационной конструкции или в состав информационных конструкций заданного языка*]
- Э *синтаксис языка**
 := [быть теорией правильно построенных информационных конструкций, принадлежащих заданному языку*]
 := [определение понятия правильно построенной информационной конструкции заданного языка*]
 := [синтаксические правила заданного языка*]
 := [быть синтаксическими правилами заданного языка *]
 := [Бинарное ориентированное отношение, каждая пара которого связывает знак некоторого языка с описанием синтаксически выделяемых классов фрагментов конструкций заданного языка, с описанием отношений, заданных на этих классах и с конъюнкцией кванторных высказываний, являющихся синтаксическими правилами заданного языка, то есть правилами, которым должны удовлетворять все синтаксические правильные (правильно построенные) конструкции (тексты) указанного языка*]
 ⇒ *примечание**:
 [При представлении синтаксиса (синтаксических правил) заданного языка используются все те понятия, которые вводятся для представления синтаксических структур информационных конструкций, принадлежащих указанному языку. Это и синтаксически выделяемые классы фрагментов указанных информационных конструкций, и отношения, заданные на множестве таких фрагментов.]
 ⇒ *второй домен**:
синтаксис языка
- Э *описание синтаксических понятий языка**
 := [описание синтаксически выделяемых классов фрагментов конструкций заданного языка*]
 ⇒ *второй домен**:
описание синтаксических понятий языка
 ← *обобщенное включение**:
синтаксис языка
- Э *синтаксические правила языка**
 := [синтаксические правила заданного языка*]
 ⇒ *второй домен**:
синтаксические правила языка
- Э *денотационная семантика языка**
 := [быть теорией морфизмов, связывающих правильно построенные информационные конструкции заданного языка с описываемыми конфигурациями описываемых сущностей*]
- Э *денотационная семантика языка**
 := [семантические правила заданного языка*]
 := [быть семантическими правилами заданного языка *]
 := [Бинарное ориентированное отношение, каждая пара которого связывает знак некоторого языка с описанием основных семантических понятий заданного языка и конъюнкцией кванторных высказываний, являющихся семантическими правилами заданного языка, то есть правилами, которым должны удовлетворять семантически правильные смысловые информационные конструкции, соответствующие (семантические эквивалентные) синтаксически правильным конструкциям (текстам) заданного языка*]
 ⇒ *примечание**:
 [При формулировке семантических правил заданного языка используются понятия, введенные в рамках базовых онтологий (онтологий самого высокого уровня, в которых рассматриваются самые общие классы описываемых сущностей, самые общие отношения и параметры).]
 ⇒ *второй домен**:
денотационная семантика языка
- Э *описание семантических понятий языка**
 ⇒ *второй домен**:
описание семантических понятий языка
- Э *семантические правила языка**
 ⇒ *второй домен**:
семантические правила языка

Э	<p><i>семантическая эквивалентность языков*</i></p> <p>:= [быть семантически эквивалентными языками*]</p> <p>⇒ <i>определение*</i>:</p> <p>[Язык <i>Li</i> и язык <i>Lj</i> будем считать <i>семантически эквивалентными языками*</i> в том и только в том случае, если для каждого текста, принадлежащего языку <i>Li</i>, существует <i>семантически эквивалентный текст*</i>, принадлежащий языку <i>Lj</i>, и наоборот.]</p>
Э	<p><i>семантическое расширение языка*</i></p> <p>⇔ <i>обратное отношение*</i>:</p> <p><i>семантическое сужение языка*</i></p> <p>⇒ <i>определение*</i>:</p> <p>[Язык <i>Lj</i> будем считать <i>семантическим расширением*</i> языка <i>Li</i> в том и только в том случае, есть ли для каждого текста, принадлежащего языку <i>Li</i>, существует <i>семантически эквивалентный текст*</i>, принадлежащий языку <i>Lj</i>, но обратное неверно.]</p>
Э	<p><i>синтаксическое расширение языка*</i></p> <p>:= [быть семантически эквивалентным надмножеством заданного языка*]</p> <p>⇒ <i>определение*</i>:</p> <p>[Язык <i>Lj</i> будем считать <i>синтаксическим расширением*</i> языка <i>Li</i> в том и только в том случае, если:</p> <ul style="list-style-type: none"> □ $Lj \supset Li$ (то есть все тексты языка <i>Li</i> являются также и текстами языка <i>Lj</i>, но обратное неверно); □ Язык <i>Lj</i> и язык <i>Li</i> являются <i>семантически эквивалентными языками*</i>. <p>]</p>
Э	<p><i>синтаксическое ядро языка*</i></p> <p>:= [быть синтаксическим ядром заданного языка*]</p> <p>:= [быть семантически эквивалентным подмножеством заданного языка, имеющим минимальную синтаксическую сложность*]</p>
Э	<p><i>направление синтаксического расширения ядра заданного языка*</i></p> <p>:= [быть правилом трансформации информационных конструкций, принадлежащих заданному языку, которое описывает одно из направлений перехода от множества конструкций ядра этого языка ко множеству всех принадлежащих ему информационных конструкций*]</p>
Э	<p><i>операционная семантика языка*</i></p> <p>:= [Бинарное ориентированное отношение, каждая пара которого связывает знак некоторого языка со множеством правил трансформации текстов этого языка*]</p> <p>⇒ <i>второй домен*</i>:</p> <p><i>операционная семантика языка</i></p>
Э	<p><i>внутренний язык*</i></p> <p>:= [быть внутренним языком для заданной системы, основанной на обработке информации, или заданного множества таких систем*]</p> <p>:= [быть языком внутреннего представления информации в памяти заданной системы, основанной на обработке информации или заданного класса таких систем*]</p>
Э	<p><i>внешний язык*</i></p> <p>:= [быть внешним языком для заданной системы, основанной на обработке информации, или заданного множества таких систем*]</p> <p>:= [быть языком, используемым для обмена информацией заданной системы, основанной на обработке информации, или заданного множества таких систем с другими системами, основанными на обработке информации, (в том числе, и с себе подобными)*]</p>
Э	<p><i>используемый язык*</i></p> <p>= (<i>внутренний язык*</i> \cup <i>внешний язык*</i>)</p> <p>:= [язык, используемый заданной системой, основанной на обработке информации или заданного множества таких систем*]</p> <p>:= [язык, носителем которого является (которым владеет) данная система, основанная на обработке информации]</p>
Э	<p><i>используемые языки*</i></p>
	<p><i>параметр, заданный на множестве языков[^]</i></p> <p>:= [признак классификации языков]</p> <p>:= [свойство, которым обладают языки]</p> <p>:= [свойство языков, на основании которого можно осуществлять их классификацию]</p>

:=	[семейство классов эквивалентности языков, трактуемой в контексте того или иного свойства (характеристики), присущего языкам]
Э	<i>семантическая мощность языка</i> [^]
:=	[класс языков, семантически эквивалентных друг другу]
Э	<i>универсальный язык</i>
:=	[класс всевозможных универсальных языков]
⇒	<i>примечание*</i> : [Очевидно, что все универсальные языки (если они действительно таковыми являются, а не только претендуют на это) семантически эквивалентны друг другу, т.е. имеют одинаковую семантическую мощность.]
Э	<i>уровень синтаксической сложности представления знаков в текстах языка</i> [^]
Э	<i>язык, в текстах которого все знаки представлены синтаксически элементарными фрагментами</i>
Э	<i>язык, в текстах которого знаки в общем случае представлены синтаксически неэлементарными фрагментами</i>
Э	<i>использование разделителей и ограничителей в текстах языка</i> [^]
Э	<i>язык, в текстах которого не используются разделители и ограничители</i>
Э	<i>язык, в текстах которого используются разделители и ограничители</i>
Э	<i>уровень сложности процедуры установления синонимии знаков в текстах языка</i> [^]
Э	<i>язык, в рамках каждого текста которого синонимичные знаки отсутствуют</i>
⇒	<i>пояснение*</i> : [В текстах такого языка знак каждой описываемой сущности входит <u>однократно</u> .]
Э	<i>язык, в рамках которого синонимичные знаки представлены синтаксически эквивалентными фрагментами текстов</i>
Э	<i>флективный язык</i>
:=	[язык, в рамках которого синонимичные знаки могут быть представлены синтаксически неэквивалентными фрагментами, но фрагментами, являющимися модификациями некоторого "ядра" этих фрагментов (при склонении и спряжении этих знаков).]
Э	<i>язык, в рамках которого синонимичные знаки в общем случае могут быть представлены синтаксически неэквивалентными текстовыми фрагментами, структура которых носит непредсказуемый характер</i>
Э	<i>наличие омонимии в текстах языка</i> [^]
Э	<i>язык, в текстах которого присутствует омонимия знаков</i>
:=	[язык, в текстах которого присутствуют синтаксически эквивалентные, не синонимичные знаки]
Э	<i>язык, в текстах которого омонимия знаков отсутствует</i>
	семантически выделяемый класс дискретных информационных конструкций
Э	<i>синтаксическая структура информационной конструкции</i>
←	<i>второй домен*</i> : <i>синтаксис информационной конструкции*</i>
⊃	<i>первичная синтаксическая структура информационной конструкции</i>
⊃	<i>вторичная синтаксическая структура информационной конструкции</i>
Э	<i>синтаксис языка</i>
Э	<i>описание синтаксических понятий языка</i>
Э	<i>синтаксические правила языка</i>
Э	<i>денотационная семантика языка</i>
Э	<i>описание семантических понятий языка</i>
Э	<i>семантические правила языка</i>
Э	<i>операционная семантика языка</i>
Э	<i>смысл</i>
←	<i>второй домен*</i> : <i>смысл*</i>
:=	[смысловая информационная конструкция]
:=	[смысловое представление информационной конструкции]
:=	[явное (формальное) представление описываемых сущностей и связей между ними]
:=	[смысловое представление информации]
⇒	<i>примечание*</i> : [Для явного представления описываемых сущностей и связей между ними требуется существенное упрощение синтаксической структуры информационных конструкций]

язык *ostis-системы*

- := [язык, используемый *ostis-системами*]
- := [язык представления информационных конструкций в *ostis-системах*]
- := [Множество языков, которыми "владеют" *ostis-системы*]
- С *формальный язык*
- С *универсальный язык*
- ← *используемые языки**:
- ostis-система*
- Э *SC-код*
- := [Semantic Computer Code]
- ← *внутренний язык**:
- ostis-система*
- Є *универсальный язык*
- Э *SCg-код*
- := [Semantic Code graphical]
- := [*внешний язык** *ostis-систем*, тексты которого представляют собой графовые структуры общего вида с точно заданной *денотационной семантикой**]
- ← *внешний язык**:
- ostis-система*
- Є *универсальный язык*
- Э *SCs-код*
- := [Semantic Code string]
- := [*внешний язык** *ostis-систем*, тексты которого представляют собой строки (цепочки) символов]
- ← *внешний язык**:
- ostis-система*
- Є *универсальный язык*
- Э *SCn-код*
- := [Semantic Code natural]
- := [*внешний язык** *ostis-систем*, тексты которого представляют собой двухмерные матрицы символов, являющиеся результатом форматирования, двухмерной структуризации текстов *SCs-кода*.]
- ← *внешний язык**:
- ostis-система*
- Є *универсальный язык*
- ⇒ *пояснение**:
- [Для представления *баз знаний ostis-систем* используется целый ряд как *универсальных языков*, так и *специализированных языков*, как *формальных языков*, так и *естественных языков*, как *внутренних языков*, обеспечивающих представление информации в памяти *ostis-систем*, так и *внешних языков*, обеспечивающих представление информации, вводимой в память *ostis-систем*, либо выводимой из этой памяти. *Естественные языки* используются исключительно для представления *файлов*, хранимых в памяти *ostis-системы* и формально специфицируемых в рамках *базы знаний* этой *ostis-системы*.
- Для эксплуатации *интеллектуальных компьютерных систем*, построенных на основе *SC-кода*, кроме способа абстрактного внутреннего представления *баз знаний (SC-кода)* потребуются несколько способов внешнего изображения абстрактных *sc-текстов*, удобных для пользователей и используемых при оформлении исходных текстов *баз знаний* указанных интеллектуальных компьютерных систем и исходных текстов фрагментов этих *баз знаний*, а также используемых для отображения пользователям различных фрагментов *баз знаний* по пользовательским запросам. В качестве таких способов внешнего отображения *sc-текстов* и предлагаются указанные выше внешние языки *ostis-систем (SCg-код, SCs-код и SCn-код)*.
- Для описания перечисленных *языков*, используемых *ostis-системами*, в каждом из них мы выделим *ядро языка**, которое является *семантически эквивалентным языком** для соответствующего языка и имеет минимальную синтаксическую сложность. Описание каждого из указанных языков строится как описание нескольких направлений синтаксического расширения выделенного *языка-ядра*.]
- ⇒ *примечание**:
- [Все основные внешние формальные языки, используемые *ostis-системами (SCg-код, SCs-код, SCn-код)* являются различными вариантами внешнего представления текстов внутреннего языка *ostis-систем – SC-кода*. Указанные языки являются универсальными и, следовательно, *семантически эквивалентными языками**.]

- ⇒ *примечание**:
 [Каждая ostis-система может приобрести способность использовать любой внешний язык (как универсальный, так и специализированный, как естественный, так и искусственный), если синтаксис и денотационная семантика этого языка будут описаны в памяти ostis-системы на ее внутреннем языке (SC-коде).]
- следует отличать**
- ∃ { • *семантическое расширение языка**
 := [расширение семантической мощности языка*]
 • *синтаксическое расширение языка**
 := [расширение синтаксиса языка при сохранении его семантической мощности*]
 }
- ∃ { • *синтаксическое расширение языка**
 • *направление синтаксического расширения ядра заданного языка**
 := [одно из (или группа) правил синтаксической трансформации текстов заданного языка, расширяющих множество синтаксически правильных текстов этого языка]
 ⇒ *примечание**:
 [Таких направлений синтаксического расширения заданного языка может быть несколько.]
 }
- ∃ { • { • *синтаксис дискретной информационной конструкции**
 • *синтаксическая структура дискретной информационной конструкции*
 }
 ∈ *следует отличать**
 • { • *первичная синтаксическая структура дискретной информационной конструкции**
 • *первичная синтаксическая структура дискретной информационной конструкции*
 }
 ∈ *следует отличать**
 • { • *смысл**
 • *смысл*
 := [смысловая информационная конструкция]
 }
 ∈ *следует отличать**
 • { • *синтаксис языка**
 • *синтаксис языка*
 }
 ∈ *следует отличать**
 • { • *денотационная семантика языка**
 • *денотационная семантика языка*
 }
 ∈ *следует отличать**
 • { • *операционная семантика языка**
 • *операционная семантика языка*
 }
 ∈ *следует отличать**
 }
- ∃ { • *смысловое представление знака**
 := [Бинарное ориентированное отношение, каждая пара которого связывает фрагмент синтаксической структуры некоторой дискретной информационной конструкции (точнее, фрагмент смыслового представления этой синтаксической структуры) с элементом смыслового представления исходной дискретной информационной конструкции]
 • *смысл**
 := [смысловое представление дискретной информационной конструкции*]
 • *денотационная семантика языка*
 }
- ∃ { • *информационная трансформация*
 := [трансформация информационной конструкции]
 := [действие по преобразованию (обработке, трансформации) информационной конструкции]

- С *действие*
- *класс однотипных информационных конструкций*
 - *правило выполнения однотипных информационных трансформаций*
:= [обобщенная спецификация класса однотипных информационных трансформаций, описывающая обобщенное условие выполнения произвольной трансформации указанного класса, ее результат и соответствие между условием и результатом]
 - *программа выполнения однотипных информационных трансформаций*
:= [обобщенная декомпозиция произвольной информационной трансформации заданного класса на систему взаимосвязанных трансформаций более низкого уровня]
 - *агент выполнения однотипных информационных трансформаций*

следует отличать*

- Э {
- *денотационная семантика**
 - *описание денотационной семантики**
 - *операционная семантика**
 - *описание операционной семантики**
- }

следует отличать*

- Э {
- *денотационная семантика знака*
 - *денотационная семантика текста*
 - *денотационная семантика языка*
- }

следует отличать*

- Э {
- *синтаксис*
:= [синтаксис языка]
:= [отношение, связывающее тексты некоторого языка и соответствующие тексты того же или другого языка, описывающие синтаксическую структуру этих текстов]
 - *синтаксис**
:= [отношение, связывающее язык и его синтаксис]
 - *синтаксическая конструкция текста*
:= [информационная конструкция, описывающая синтаксическую структуру текста некоторого языка]
 - *описание синтаксиса*
:= [информационная конструкция, описывающая синтаксис некоторого языка и его свойства, включают правила построения синтаксически корректных конструкций данного языка]
- }

язык *ostis-системы*

⇒ *примечание**:

[Следует отличать:

- саму описываемую сущность;
- текст, являющийся описанием некоторой сущности;
- текст, являющийся описанием некоторого другого текста, а возможно и самого себя (т.е. текст может быть описываемой сущностью);
- знак (обозначение) описываемой сущности в рамках заданного текста;
- обозначение описываемой сущности в sc-тексте (это всегда sc-элемент того или иного вида);
- коммуникативный (внешний для *ostis-системы*) уникальный (основной) идентификатор (чаще всего строковый идентификатор-имя), обозначающий соответствующую описываемую сущность и являющийся внешним идентификатором (именем) для соответствующего синонимичного ему sc-элемента. Такие идентификаторы взаимно однозначно соответствуют sc-элементам, которые имеют такие идентификаторы;
- вспомогательные (неосновные) внешние идентификаторы sc-элементов. Такие идентификаторы могут обладать свойством омонимии (когда один идентификатор соответствует нескольким sc-элементам) и синонимии (когда разные идентификаторы соответствует одному sc-элементу);
- обозначение описываемой сущности в sc.g-тексте – это всегда графически представленный sc.g-элемент, являющийся изображением соответствующего sc-элемента;

- обозначение описываемой сущности в sc.s-предложении и в sc.n-предложении – это всегда строка символов (либо омонимичное изображение sc-коннекторов различного семантического типа, либо основной строковый идентификатор, соответствующий некоторому sc-элементу, либо выражение, являющееся неатомарным идентификатором, содержащим некоторую информацию о соответствующей именуемой сущности).

Подчеркнем, что каждое обозначение описываемой сущности в SCg-коде, SCs-коде, SCn-коде рассматривается нами как изображение соответствующего ему (синонимичного ему) sc-элемента, обозначающего ту же описываемую сущность. Таким образом, указанные языки (SCg-код, SCs-код; SCn-код) рассматриваются нами как различные варианты изображения текстов SCg-кода.]

⇒ *примечание**:

[Для формального описания различного рода языков, включая рассматриваемые нами языки (SCg-код, SCs-код, SCn-код) используется целый ряд метаязыковых понятий.

Перечислим некоторые из них: *идентификатор, класс синтаксически эквивалентных идентификаторов, имя, простое имя, выражение, внешний идентификатор**, *алфавит**, *разделители**, *ограничители**, *предложения**

⇒ *примечание**:

[Синтаксис *языков представления знаний в ostis-системах* может быть формально описан различными способами. Так, например, можно использовать метаязык Бэкуса-Наура для описания синтаксиса SCs-кода или его расширение для описания синтаксиса SCn-кода. Однако значительно более логично и целесообразно описывать синтаксис всех форм внешнего отображения sc-текстов с помощью самого SC-кода. Такой подход позволит ostis-системам самостоятельно понимать, анализировать и генерировать тексты указанных языков на основе принципов, общих для любых форм внешнего представления информации, в том числе нелинейных.]

алфавит*

:= [быть алфавитом для заданного множества текстов*]

:= [быть семейством максимальных множеств синтаксически однотипных элементарных (атомарных) фрагментов текстов, принадлежащих заданному множеству текстов*]

ограничители*

:= [Отношение, связывающее заданный класс информационных конструкций с соответствующим классом их ограничителей]

:= [быть ограничителями, используемыми в заданном множестве информационных конструкций*]

ограничитель

⊃ *sc.g-ограничитель*

⊃ *sc.s-ограничитель*

⊃ *sc.n-ограничитель*

⊃ *ограничитель, используемый в ея-файлах ostis-систем*

SCg-код

⇒ *ограничители**:

sc.g-ограничитель

:= [Множество ограничителей, используемых в sc.g-текстах]

разделители*

:= [быть разделителями, используемыми в заданном множестве информационных конструкций*]

⇒ *второй домен**:

разделитель

⊃ *sc.g-разделитель*

= *sc.g-коннектор*

⊃ *sc.s-разделитель*

⊃ *sc.n-разделитель*

⊃ *разделитель, используемый в ея-файлах ostis-систем*

идентификатор

⊃ *sc.s-идентификатор*

:= [структурированный знак соответствующей (обозначаемой) сущности, который чаще всего представляет собой строку (цепочку символов), которую будем называть именем соответствующей сущности.]

⇒ *примечание**:
 [В формальных текстах (в том числе текстах SC-кода, SCg-кода, SCs-кода, SCп-кода) основные используемые идентификаторы не должны быть омонимичными, то есть должны однозначно соответствовать идентифицируемым сущностям. Следовательно, каждая пара идентификаторов, имеющих одинаковую структуру, должны обозначать одну и ту же сущность.]

следует отличать*

⊃ { • *идентификатор*
 := [Множество всевозможных конкретных экземпляров, конкретных вхождений идентификаторов, имеющих различную структуру, во всевозможные тексты]
 • *класс синтаксически эквивалентных идентификаторов*
 := [класс идентификаторов, имеющих одинаковую структуру]
 := [Семейство всевозможных множеств, каждое из которых является максимальным множеством синтаксически эквивалентных идентификаторов]
 }

имя

⊂ *идентификатор*
 := [строковый идентификатор]
 := [идентификатор, представляющий собой строку (цепочку) символов]
 ⇒ *разбиение**:
 { • *простое имя*
 := [атомарное имя]
 := [имя, в состав которого другие имена не входят]
 • *выражение*
 := [неатомарное имя]
 }
 ⊃ *sc.s-идентификатор*

внешний идентификатор*

∈ *отношение*
 := [Бинарное ориентированное отношение, каждая связка (sc-дуга) которого связывает некоторый элемент с файлом, содержимым которого является внешний идентификатор (чаще всего, имя), соответствующий указанному элементу]
 := [быть внешним идентификатором*]
 := [внешний идентификатор sc-элемента*]
 ⇒ *второй домен**:
идентификатор
 ⇒ *примечание**:
 [Понятие внешнего идентификатора является понятием относительным и важным для ostis-систем, поскольку внутреннее для ostis-систем представление информации (в виде текстов SC-кода) оперирует не идентификаторами описываемых сущностей, а знаками, структура которых никакого значения не имеет]

следует отличать*

⊃ { • *sc-элемент, обозначающий файл ostis-системы*
 • *sc.g-элемент, обозначающий файл ostis-системы*
 • *простой sc.s-идентификатор, обозначающий файл ostis-системы*
 := [простое имя файла ostis-системы]
 • *изображение файла ostis-системы, ограниченное sc.g-рамкой*
 • *изображение файла ostis-системы, ограниченное sc.n-рамкой*
 • *изображение строки символов, ограниченное квадратными скобками*
 }

следует отличать*

⊃ { • *файл-экземпляр*
 • *файл-класс*
 := [файл, обозначающий класс файлов-экземпляров, синтаксически эквивалентных заданному образцу]
 }

предложения*

- := [быть множеством всех предложений заданного текста, не являющихся встроенными предложениями, то есть предложениями, входящими в состав других предложений*]
- ⇒ *второй домен**:
предложение

предложение

- ⇒ *пояснение**:
[минимальный семантически целостный фрагмент текста, представляющий собой конфигурацию знаков, входящих в этот фрагмент и связываемых между собой отношениями инцидентности (в частности, отношением непосредственной последовательности в строке), а также различного вида разделителями и ограничителями]

sc.g-текст

- := [текст SCg-кода]
- ⇒ *разбиение**:
{
• *семантически связный sc.g-текст*
:= [sc.g-текст, который семантически эквивалентен связному sc-тексту]
• *семантически несвязный sc.g-текст*
}
- ⇒ *разбиение**:
{
• *синтаксически связный sc.g-текст*
• *синтаксически несвязный sc.g-текст*
}

sc.s-текст

- := [текст SCs-кода]
- ⇒ *разбиение**:
{
• *семантически связный sc.s-текст*
:= [sc.s-текст, который семантически эквивалентен связному sc-тексту]
• *семантически несвязный sc.s-текст*
}
- ⇒ *разбиение**:
{
• *синтаксически связный sc.s-текст*
• *синтаксически несвязный sc.s-текст*
}

sc.n-текст

- := [текст SCn-кода]
- ⇒ *разбиение**:
{
• *семантически связный sc.n-текст*
:= [sc.n-текст, который семантически эквивалентен связному sc-тексту]
• *семантически несвязный sc.n-текст*
}
- ⇒ *разбиение**:
{
• *синтаксически связный sc.n-текст*
• *синтаксически несвязный sc.n-текст*
}

следует отличать*

- ∋ {
• *денотационная семантика**
• *описание денотационной семантики**
• *операционная семантика**
• *описание операционной семантики**
}
- ∋ {
• *денотационная семантика знака*
• *денотационная семантика текста*
• *денотационная семантика языка*
}

<p> } Э { </p>	<ul style="list-style-type: none"> • <i>синтаксис</i> := [синтаксис языка] := [отношение, связывающее тексты некоторого языка и соответствующие тексты того же или другого языка, описывающие синтаксическую структуру этих текстов] • <i>синтаксис*</i> := [Отношение, связывающее язык и его синтаксис] • <i>синтаксическая структура текста</i> := [информационная конструкция, описывающая синтаксическую структуру текста некоторого языка] • <i>описание синтаксиса</i> := [информационная конструкция, описывающая синтаксис некоторого языка и его свойства, включая правила построения синтаксически корректных конструкций данного языка] <p> } } /* Завершили Раздел “Предметная область и онтология информационных конструкций и языков” */ </p>
-------------------------------------	--

```

/* Раздел *****/
Описание внутреннего языка ostis-систем
⊃=
{
← конкатенация сегментов*:
{
• Основные положения внутреннего языка ostis-систем
• Описание Ядра SC-кода
• SC-код как синтаксическое расширение Ядра SC-кода
• Использование SC-кода для формального описания собственного синтаксиса
}
}

/* Сегмент *****/
Основные положения внутреннего языка ostis-систем
⊃=
{
SC-код
:= [Язык унифицированного смыслового представления знаний в памяти интеллектуальных компьютерных систем]
:= [Внутренний язык ostis-систем]
← внутренний язык*:
ostis-система
⇒ эпиграф*:
[Информация содержится не в самих знаках, а в конфигурации связей между ними.]
⇒ эпиграф*:
[Он вскочил на коня и поскакал во все стороны.]
⇒ основной внешний идентификатор sc-элемента*:
[SC-код]
∈ имя собственное
⇒ часто используемый неосновной внешний идентификатор sc-элемента*:
[sc-текст]
∈ имя нарицательное
∈ абстрактный язык
:= [Язык, для которого не уточняется способ представления символов (синтаксически элементарных фрагментов), входящих в состав текстов этого языка, а задается только алфавит* этих символов, то есть семейство классов символов, считающихся синтаксически эквивалентными друг другу.]
⇒ примечание*:
[Каждому абстрактному языку можно поставить в соответствие целое семейство реальных языков, обеспечивающих изоморфное реальное представление текстов указанного абстрактного языка путем уточнения способов представления (изображения, кодирования) символов, входящих в состав этих текстов, а также путем уточнения правил установления синтаксической эквивалентности этих символов. Очевидно, что во всём остальном синтаксис и денотационная семантика указанных реальных языков полностью совпадает с синтаксисом и денотационной семантикой соответствующего абстрактного языка.]
⇒ примечание*:
[Для SC-кода как абстрактного языка необходима разработка как минимум трех синтаксически и семантически эквивалентных ему реальных языков: (1) язык кодирования текстов SC-кода в памяти традиционных компьютеров; (2) язык кодирования текстов SC-кода в семантической ассоциативной памяти; (3) Ядро SCg-кода – язык, синтаксически и семантически эквивалентный SC-коду и обеспечивающий графическое представление текстов SC-кода.]
∈ графовый язык
⇒ пояснение*:
[язык, каждый текст которого
□ задается множеством входящих в него элементарных фрагментов (символов), которое, в свою очередь, состоит
□□ из множества узлов (вершин), возможно, синтаксически разного вида

```


- □ из множества связей, которые также могут принадлежать разным синтаксически выделяемым классам
- задается в общем случае несколькими отношениями инцидентности связей с компонентами этих связей (при этом указанными компонентами в общем случае могут быть не только вершины, но и другие связи)

]

:= [Универсальный язык, обеспечивающий внутреннее представление и структуризацию всех(!), используемых ostis-системой в процессе своего функционирования.]

:= [Универсальный язык, являющийся результатом унификации (уточнения) синтаксиса и денотационной семантики семантических сетей.]

⇒ *пояснение**:

[Универсальность SC-кода обеспечивается и тем, что элементами текстов SC-кода могут быть знаки описываемых сущностей любого вида, в том числе, и знаки связей между описываемыми сущностями и/или их знаками]

⇒ *следствие**:

[Тексты SC-кода являются графовыми структурами расширенного вида, в которых знаки описываемых связей могут соединять не только вершины (узлы) графовой структуры, но и знаки других связей.]

:= [Базовый универсальный язык внутреннего представления знаний в памяти ostis-систем.]

:= [Базовый внутренний язык ostis-систем.]

:= [Максимальный внутренний язык ostis-систем, по отношению к которому все остальные (специализированные) внутренние языки являются его подязыками (подмножествами)]

:= [Множество всевозможных текстов SC-кода]

∈ *имя собственное*

:= [текст SC-кода]

∈ *имя нарицательное*

⇒ *принципы, лежащие в основе**:

- [Знаки (обозначения) всех *сущностей*, описываемых в *sc-текстах* (текстах **SC-кода**) представляются в виде синтаксически элементарных (атомарных) фрагментов *sc-текстов* и, следовательно, не имеющих внутренней структуры, не состоящих из более простых фрагментов *текста*, как, например, имена (термины), которые представляют *знаки* описываемых *сущностей* в привычных языках и состоят из букв.]
- [Имена (термины), *естественно-языковые тексты* и другие информационные конструкции, не являющиеся *sc-текстами*, могут входить в состав *sc-текста*, но только как *файлы*, описываемые (специфицируемые) *sc-текстами*. Таким образом, в состав базы знаний *интеллектуальной компьютерной системы*, построенной на основе **SC-кода**, могут входить *имена* (термины), обозначающие некоторые описываемые *сущности* и представленные соответствующими *файлами*. Каждый *sc-элемент* будем называть внутренним обозначением некоторой *сущности*, а *имя* этой *сущности*, представленное соответствующим файлом, будем называть *внешним идентификатором* (внешним обозначением) этой *сущности*. При этом каждый именуемый (идентифицируемый) *sc-элемент* связывается дугой, принадлежащей отношению “быть *внешним идентификатором**”, с *узлом*, содержимым которого является *файл* идентификатора (в частности, *имени*), обозначающего ту же *сущность*, что и указанный выше *sc-элемент*. *Внешним идентификатором* может быть не только *имя* (термин), но и иероглиф, пиктограмма, озвученное имя, жест. Особо отметим, что *внешние идентификаторы* описываемых *сущностей* в *интеллектуальной компьютерной системе*, построенной на основе **SC-кода**, используются только (1) для анализа информации, поступающей в эту систему из вне из различных источников, и ввода (понимания и погружения) этой информации в *базу знаний*, а также (2) для синтеза различных *сообщений*, адресуемых различным субъектам (в т.ч. пользователям).]
- [Тексты **SC-кода** (*sc-тексты*) имеют в общем случае нелинейную (графовую) структуру, поскольку (1) *знак* каждой описываемой сущности входит в состав *sc-текста* однократно и (2) каждый такой *знак* может быть инцидентен неограниченному числу других *знаков*, поскольку каждая описываемая *сущность* может быть связана неограниченным числом связей с другими описываемыми *сущностями*.]
- [База знаний, представленная текстом **SC-кода**, является *графовой структурой* специального вида, алфавит элементов которой включает в себя множество *узлов*, множество *ребер*, множество *дуг*, множество *базовых дуг* – дуг специально выделенного типа, обеспечивающих структуризацию *баз знаний*, а также множество специальных *узлов*, каждый из которых имеет содержимое, являющееся

файлом, хранящимся в памяти интеллектуальной компьютерной системы. Структурная особенность данной *графовой структуры* заключается в том, что ее *дуги* и *ребра* могут связывать не только *узел* с *узлом*, но и *узел* с *ребром* или *дугой*, *ребро* или *дугу* с другим *ребром* или *дугой*.]

- [Все элементы (*sc-элементы*) указанной выше *графовой структуры* (текста *SC-кода*), т.е. все ее узлы (*sc-узлы*), ребра (*sc-ребра*) и дуги (*sc-дуги*) являются обозначениями различных сущностей. При этом ребро является обозначением бинарной неориентированной связи между двумя сущностями, каждая из которых либо представлена в рассматриваемой графовой структуре соответствующим знаком, либо является самим этим знаком. Дуга является обозначением бинарной ориентированной связи между двумя сущностями. Дуга специального вида (*базовая дуга*) является знаком связи между узлом, обозначающим некоторое множество элементов рассматриваемой графовой структуры, и одним из элементов этой графовой структуры, который принадлежит указанному множеству. Узел, имеющий содержимое (узел, для которого содержимое существует, но может в текущий момент быть неизвестным) является знаком файла, который является содержимым этого узла. Узел, не являющийся знаком файла, может обозначать какой-либо материальный объект, первичный абстрактный объект (например, число, точку в некотором абстрактном пространстве), какую-либо бинарную связь, какое-либо множество (в частности, понятие, структуру, ситуацию, событие, процесс). При этом сущности, обозначаемые элементами рассматриваемой графовой структуры, могут быть постоянными (существующими всегда) и временными (сущностями, которым соответствует отрезок времени их существования). Кроме того, сущности, обозначаемые элементами рассматриваемой графовой структуры, могут быть константными (конкретными) сущностями и переменными (произвольными) сущностями. Каждому элементу рассматриваемой графовой структуры, являющемуся обозначением переменной сущности, ставится в соответствие область возможных значений этого обозначения. Область возможных значений каждого переменного ребра является подмножеством множества всевозможных константных ребер, область возможных значений каждой переменной дуги является подмножеством множества всевозможных константных дуг, область возможных значений каждого переменного узла является подмножеством множества всевозможных константных узлов.]
- [В рассматриваемой графовой структуре, являющейся представлением базы знаний в *SC-коде*, могут, но не должны существовать разные элементы графовой структуры, обозначающие одну и ту же сущность. Если пара таких элементов обнаруживается, то эти элементы склеиваются (отождествляются). Таким образом, синонимия внутренних обозначений в базе знаний интеллектуальной компьютерной системы, построенной на основе *SC-кода*, запрещена. При этом синонимия внешних обозначений считается нормальным явлением. Формально это означает, что из некоторых элементов рассматриваемой графовой структуры выходит несколько дуг, принадлежащих отношению "быть *внешним идентификатором**". Из всех указанных дуг, принадлежащих отношению "быть *внешним идентификатором**" и выходящих из одного элемента рассматриваемой графовой структуры, обязательно выделяется одна (очень редко две) путем включения их в число дуг, принадлежащих отношению "быть *основным внешним идентификатором**". Это означает, что указываемый таким образом внешний идентификатор не является омонимичным, т.е. не может быть использован как внешний идентификатор, соответствующий другому элементу рассматриваемой графовой структуры.]
- [Кроме файлов, представляющих различные внешние обозначения (имена, иероглифы, пиктограммы), в памяти интеллектуальной компьютерной системе, построенной на основе *SC-кода*, могут храниться файлы различных текстов (книг, статей, документов, примечаний, комментариев, пояснений, чертежей, рисунков, схем, фотографий, видео-материалов, аудио-материалов).]
- [Любую сущность, требующую описания, в тексте *SC-кода* можно обозначить в виде *sc-элемента*. Это является одним из факторов, обеспечивающих универсальность *SC-кода*. Особо подчеркнем, что *sc-элементы* являются не просто обозначениями различных описываемых сущностей, а обозначениями, которые являются элементарными (атомарными) фрагментами знаковой конструкции, т.е. фрагментами, детализация структуры которых не требуется для "прочтения" и понимания этой знаковой конструкции.]
- [Текст *SC-кода*, как и любая другая графовой структура, является абстрактным математическим объектом, не требующим детализации (уточнения) его кодирования в памяти компьютерной системы (например, в виде матрицы смежности, матрицы инцидентности, списковой структуры). Но такая детализация потребуется для технической реализации памяти, в которой хранятся и обрабатываются *sc-тексты*.]
- [Важнейшим дополнительным свойством *SC-кода* является то, что он удобен не просто для внутреннего представления знаний в памяти интеллектуальной компьютерной системы, но также и для внутреннего представления информации в памяти компьютеров, специально предназначенных

для интерпретации семантических моделей интеллектуальных компьютерных систем. Т.е., *SC-код* определяет синтаксические, семантические и функциональные принципы организации памяти компьютеров нового поколения, ориентированных на реализацию интеллектуальных компьютерных систем, – принципы организации графодинамической ассоциативной семантической памяти.]

- [SC-код рассматривается нами как объединение трех его подязыков, в число которых входит *Ядро SC-кода*, подязык *SC-кода*, обеспечивающий представление текстов *SC-кода* (*sc-текстов*) в форме орграфов классического вида, являющихся подразделениями текстов *Ядра SC-кода* и, соответственно, использующих явное представление пар инцидентности элементов *sc-текстов* (*sc-элементов*), синтаксическое *Расширение Ядра SC-кода*, обеспечивающее представление в памяти *ostis-системы* информационных конструкций инородного для *SC-кода* вида.]

} /* Завершили Описание принципов, лежащих в основе *SC-кода* */

SC-код

⇒ примечание*:

[Следует особо подчеркнуть, что унификация и максимально возможное упрощение *синтаксиса* и *денотационной семантики* внутреннего языка интеллектуальных компьютерных систем прежде всего необходимы потому, что подавляющий объем *знаний*, хранимых в составе базы знаний интеллектуальной компьютерной системы, представляют собой *метазнания*, описывающие свойства других знаний. К *метазнаниям*, в частности, следует отнести и различного вида логические высказывания и всевозможного вида программы, описания методов (навыков), обеспечивающих решение различных классов задач. Необходимо исключить зависимость формы представляемого знания от вида этого знания. Форма (структура) внутреннего представления знания любого вида должна зависеть только(!) от смысла этого знания. Более того, конструктивное (формальное) развитие теории интеллектуальных компьютерных систем невозможно без уточнения (унификации, стандартизации) и обеспечения семантической совместимости различных видов знаний, хранимых в базе знаний интеллектуальной компьютерной системы. Очевидно, что многообразие форм представления семантически эквивалентных знаний делает разработку общей теории интеллектуальных компьютерных систем практически невозможной.]

SC-пространство

⇒ примечание*:

[Понятие *SC-пространства* наряду с понятием *SC-кода* играет важнейшую роль для уточнения и формализации понятия смысла информационных конструкций, для унификации смыслового представления информации и для максимально возможного исключения субъективизма в трактовке понятия смысла. Смысл информационной конструкции в конечном счете определяется (1) конфигурацией смыслового представления этой конструкции и (2) и "местоположением"(контекстом) смыслового представления указанной информационной конструкции в рамках смыслового пространства, т.е. в рамках объединенного смыслового представления всевозможных информационных конструкций, либо в рамках объединенного смыслового представления информации, накопленной к заданному моменту времени некоторым индивидуальным субъектом или коллективом субъектов. Таким объединенным смысловым представлением информации, в частности, является смысловое представление глобальной базы всех знаний, накопленных человечеством к текущему моменту.]

⇒ пояснение*:

[Объединение (вместилище) всевозможных унифицированных семантических сетей (текстов *SC-кода*)]

⇒ примечание*:

[При теоретико-множественном объединении текстов *SC-кода* семантически эквивалентные (синонимичные) элементы (синтаксически элементарные фрагменты) этих текстов считаются совпадающими элементами и при объединении указанных текстов "склеиваются".]

⇐ объединение*:

SC-код

:= [Унифицированное смысловое пространство]

⇒ достоинство*:

[Важнейшим достоинством *SC-пространства* является возможность уточнения таких понятий, как понятие аналогичности (сходства и отличия) различных описываемых "внешних"сущностей, аналогичности унифицированных семантических сетей (текстов *SC-кода*), понятие семантической близости описываемых сущностей (в том числе, и текстов *SC-кода*).]

} /* Завершили Сегмент "Основные положения внутреннего языка *ostis-систем*" */

```

/* Сегмент *****/
Описание Ядра SC-кода
⊃=
{

Синтаксис Ядра SC-кода
⊃=
{
Синтаксис Ядра SC-кода
⇒ примечание*:
    [Синтаксис Ядра SC-кода задается: (1) Алфавитом Ядра SC-кода, (2) Отношением инцидентности
    sc-коннекторов*, (3) Отношением инцидентности входящих sc-дуг*]
⇐ синтаксис*:
    Ядро SC-кода

Ядро SC-кода
⇒ множество всех элементов конструкций данного языка*:
    sc-элемент
    := [элемент конструкции Ядра SC-кода]
    := [синтаксически элементарный (атомарный) фрагмент дискретной информационной конструкции,
    принадлежащей Ядру SC-кода]
    := [Класс элементов конструкций Ядра SC-кода]
    := [Множество всех элементов всевозможных конструкций Ядра SC-кода]
⇒ алфавит*:
    Алфавит Ядра SC-кода^

Алфавит Ядра SC-кода^
:= [Множество (Семейство) всех классов синтаксически эквивалентных sc-элементов Ядра SC-кода]
:= [класс синтаксически эквивалентных sc-элементов Ядра SC-кода]
:= [класс синтаксически эквивалентных элементов конструкций Ядра SC-кода]
:= [элемент Алфавита Ядра SC-кода]
:= [синтаксический тип sc-элемента Ядра SC-кода]
= {
    • sc-узел общего вида
    • sc-ребро общего вида
    • sc-дуга общего вида
    • базовая sc-дуга
    }

sc-элемент
⇒ разбиение*:
    Алфавит Ядра SC-кода^
    ⇒ примечание*:
        [Алфавит Ядра SC-кода является одним из признаков классификации sc-элементов.]
    ⇒ примечание*:
        [В процессе обработки текстов Ядра SC-кода синтаксический тип sc-элементов может меняться –
        sc-узел может трансформироваться в sc-ребро, sc-ребро – в sc-дугу, sc-дуга общего вида – в базовую
        sc-дугу.]

синтаксически выделяемый класс sc-элементов в рамках Ядра SC-кода^
:= [класс sc-элементов, определяемый на основе Алфавита Ядра SC-кода]
⊃ sc-коннектор
⊃ sc-дуга
⊃ Алфавит Ядра SC-кода^

sc-дуга
⇒ разбиение*:
    {
    • sc-дуга общего вида
    • базовая sc-дуга
    }

```

sc-коннектор

⇒ разбиение*:
 { • sc-ребро общего вида
 • sc-дуга общего вида
 }

Синтаксическая классификация sc-элементов в рамках Ядра SC-кода

⊇=
 {

sc-элемент

⇒ разбиение*:
 { • sc-узел общего вида
 • sc-коннектор
 ⇒ разбиение*:
 { • sc-ребро общего вида
 • sc-дуга
 ⇒ разбиение*:
 { • sc-дуга общего вида
 • базовая sc-дуга
 }
 }
 }

/* Завершили представление Синтаксической классификации sc-элементов в рамках Ядра SC-кода */

⇒ примечание*:

[Все классы *sc-элементов*, входящие в состав синтаксической классификации *sc-элементов* являются синтаксически выделяемыми классами *sc-элементов*]

инцидентность sc-коннекторов*

:= определение*:

[Бинарное ориентированное отношение, первым компонентом каждой ориентированной пары которого является некоторый *sc-коннектор*, а вторым компонентом является один из *sc-элементов*, соединяемых указанным *sc-коннектором* с некоторым другим *sc-элементом*, который указывается в другой паре инцидентности для этого же *sc-коннектора*]

инцидентность входящих sc-дуг*

:= определение*:

[Бинарное ориентированное отношение, первым компонентом каждой ориентированной пары которого является некоторая *sc-дуга*, а вторым компонентом — *sc-элемент*, в который указанная *sc-дуга* входит, т.е. *sc-элемент*, который является вторым компонентом, соединяемым (связываемым) указанной *sc-дугой*]

Ядро SC-кода

⇒ синтаксические правила*:

Синтаксические правила Ядра SC-кода

⊇= { • {
 инцидентность sc-коннекторов*
 ⊃ инцидентность входящих sc-дуг*
 ∈ бинарное ориентированное отношение
 }
 •

- [Для каждого *sc-коннектора* существует две и только две пары *инцидентности sc-коннекторов**, указанный *sc-коннектор* является первым связующим компонентом. При этом для каждой *sc-дуги* из двух указанных пар инцидентности одна должна принадлежать отношению инцидентности *входящей sc-дуги**.]
- [Пары инцидентности *sc-коннекторов* могут быть кратными. То есть *sc-коннектор* может соединять (связывать) *sc-элемент* с самим собой. Такие *sc-коннекторы* будем называть петлевыми *sc-коннекторами* (петлевыми *sc-ребрами* и петлевыми *sc-дугами*).]
- [Само *Отношение инцидентности sc-коннекторов** и, следовательно, *Отношение инцидентности входящих sc-дуг** не имеет кратных пар инцидентности. То есть *sc-коннектор* не может быть инцидентен самому себе.]

	<ul style="list-style-type: none"> [В область определения <i>Отношения инцидентности sc-коннекторов*</i> и <i>Отношения инцидентности входящих sc-дуг*</i> входят не только sc-узлы общего вида, но и sc-коннекторы. Это значит, что sc-коннектор может соединять (связывать) не только sc-узел с sc-узлом, но также sc-узел с sc-коннектором и даже sc-коннектор с sc-коннектором.]
	} /* Завершили перечень синтаксических правил Ядра SC-кода */
}	/* Завершили изложение Синтаксиса Ядра SC-кода */
Денотационная семантика Ядра SC-кода	
⊇=	
{	
Ядро SC-кода	
⇒	денотационная семантика*: Денотационная семантика Ядра SC-кода
:=	[Описание соответствия информационных конструкций, принадлежащих Ядру SC-кода, и сущностей, описываемых этими конструкциями]
параметр, заданный на множестве sc-элементов	
⊃	Алфавит Ядра SC-кода [^]
⊃	Алфавит SC-кода [^]
⊃	Структурная типология sc-элементов [^]
⊃	Типология sc-элементов по признаку константности [^]
⊃	Типология sc-элементов по признаку постоянства обозначаемой сущности [^]
⊃	Типология sc-элементов по признаку доступности sc-элемента в процессе эксплуатации и эволюции базы знаний [^]
Семантическая классификация sc-элементов	
⊇=	
{	
sc-элемент	
:=	[обозначение описываемой сущности]
⇒	разбиение*: Структурная типология sc-элементов[^]
=	{ <ul style="list-style-type: none"> • обозначение терминальной сущности <ul style="list-style-type: none"> ⇒ разбиение*: { <ul style="list-style-type: none"> • обозначение материальной сущности <ul style="list-style-type: none"> ⇒ примечание*: [К материальным сущностям относятся физические тела, поля, биологические объекты, технические системы и многое другое.] • обозначение абстрактной терминальной сущности <ul style="list-style-type: none"> ⇒ примечание*: [Примерами абстрактных терминальных сущностей являются предельно малые физические тела, точки различных пространств, числа.] • обозначение дискретной информационной конструкции, не принадлежащей SC-коду <ul style="list-style-type: none"> := [обозначение информационной конструкции, не являющейся конструкцией SC-кода и тем более Ядра SC-кода] := [обозначение "инородной" для SC-кода информационной конструкции] ⊃ обозначение файла <ul style="list-style-type: none"> := [обозначение внешней информационной конструкции, представленной в электронной форме]
	} <ul style="list-style-type: none"> • обозначение множества <ul style="list-style-type: none"> ⇒ разбиение*: { <ul style="list-style-type: none"> • обозначение связки • обозначение класса • обозначение структуры
}	}

обозначение связки

⇒ разбиение*:
 { • обозначение небинарной связки
 • обозначение sc-пары
 ⇒ разбиение*:
 { • обозначение неориентированной sc-пары
 • обозначение ориентированной пары неизвестной направленности
 • обозначение ориентированной sc-пары
 ⊃ обозначение sc-пары принадлежности
 ⇒ разбиение*:
 { • обозначение позитивной sc-пары принадлежности
 • обозначение негативной sc-пары принадлежности
 • обозначение нечеткой sc-пары принадлежности
 }
 }
 }

sc-элемент

⇒ разбиение*:
Типология sc-элементов по признаку константности[^]
 = { • sc-константа
 := [константный sc-элемент]
 := [обозначение конкретной (фиксированной) сущности]
 • sc-переменная
 := [переменный sc-элемент]
 := [обозначение произвольной сущности из некоторого множества сущностей]
 := [sc-элемент, имеющий (принимающий) произвольное значение из некоторого множества sc-элементов]
 }
 ⇒ разбиение*:
Типология sc-элементов по постоянству обозначаемых сущностей[^]
 = { • обозначение постоянной сущности
 • обозначение временной сущности
 := [обозначение нестационарной сущности, факт существования которой зависит от времени]
 ⇒ разбиение*:
 { • обозначение прошлой сущности
 := [обозначение сущности, существовавшей до текущего момента времени]
 • обозначение настоящей сущности
 := [обозначение сущности, существующей в текущий момент времени]
 • обозначение будущей сущности
 := [обозначение сущности, существование которой прогнозируется или планируется в будущем]
 }
 }
 ⇒ разбиение*:
Типология sc-элементов по признаку доступности sc-элемента в процессе эксплуатации и эволюции базы знаний[^]
 = { • удаленный sc-элемент
 := [sc-элемент, считающийся логически удаленным, но присутствующим в описании истории эксплуатации и эволюции базы знаний]
 • настоящий sc-элемент
 := [sc-элемент, входящий в состав эксплуатируемой части базы знаний]
 • будущий sc-элемент
 := [sc-элемент, планируемый для включения в состав эксплуатируемой части базы знаний]
 }

обозначение множества

:= [обозначение множества sc-элементов]
 ⇒ разбиение*:
 { • произвольное множество
 := [sc-переменная, обозначающая произвольное множество из некоторого семейства множеств]
 := [переменное множество]
 • множество
 := [конкретное (константное, фиксированное) множество sc-элементов]
 }

множество

:= [множество sc-элементов]
 ⇒ разбиение*:
 { • множество sc-констант
 := [множество, элементами которого являются только sc-константы]
 • множество sc-переменных
 := [множество, элементами которого являются только sc-переменные]
 ⊃ sc-переменная
 := [множество, элементами которого являются всевозможные sc-переменные и только они]
 ⊃ произвольное множество
 := [sc-переменная, значениями которой являются всевозможные обозначения множеств и только они]
 • множество sc-констант и sc-переменных
 := [множество, в число элементов которого входят как sc-константы, так и sc-переменные]
 ⊃ обозначение множества
 := [множество, элементами которого являются всевозможные sc-переменные и sc-константы, обозначающие множества и только они]
 }

обозначение связки

:= [обозначение связи между sc-элементами и/или обозначаемыми ими сущностями]
 ⇒ разбиение*:
 { • произвольная связка
 := [sc-переменная, значениями которой являются обозначения связок]
 • связка
 := [конкретная связка sc-элементов]
 }
 ⊃ обозначение sc-пары
 := [обозначение связки двух sc-элементов либо одного sc-элемента с самим собой]
 ⊃ sc-пара
 := [конкретная sc-пара]
 ⊃ sc-константа
 ⊃ sc-коннектор
 ⊃ ориентированная sc-пара
 ⊃ sc-пара принадлежности
 ⇒ разбиение*:
 { • позитивная sc-пара принадлежности
 ⊃ позитивная постоянная sc-пара принадлежности
 ⊃ базовая sc-дуга
 • негативная sc-пара принадлежности
 • нечеткая sc-пара принадлежности
 }

обозначение класса

:= [обозначение множества sc-элементов, которые в соответствующем смысле эквивалентны друг другу, т.е. имеют одинаковые свойства]


```

⇒ разбиение*:
{
• произвольный класс
  C sc-переменная
  E sc-константа
• класс
  C sc-константа
}

класс
⇒ разбиение*:
{
• класс терминальных сущностей
• класс множеств
  ⇒ разбиение*:
  {
  • класс связок
    ⊃ sc-отношение
  • класс классов
    ⊃ параметр
  • класс структур
    ⊃ sc-язык
    := [специализированный язык, являющийся подязыком SC-кода, и обеспечивающий представление всевозможных знаний в рамках соответствующей предметной области, которая, в свою очередь, специфицируется соответствующей комплексной онтологией]
  }
}

Э обозначение множества
  ⊃ множество
Э множество
Э обозначение связки
  ⊃ связка
Э связка
Э обозначение класса
  ⊃ класс
Э класс
Э обозначение структуры
  ⊃ структура
Э структура
Э обозначение дискретной информационной конструкции
  ⊃ дискретная информационная конструкция
    ⊃ файл
      ⊃ файл ostis-системы
      ⊃ внутренний файл ostis-системы
  ⊃ обозначение структуры
⇒ примечание*:
[Все семантически и синтаксически выделяемые классы sc-элементов, а также всевозможные подклассы этих классов являются экземплярами (элементами) класса]

обозначение структуры
:= [обозначение множества, не являющегося ни связкой, ни классом]
⇒ разбиение*:
{
• произвольная структура
  C sc-переменная
• структура
  := [конкретная структура]
  C sc-константа
}
} /* Завершили представление Семантической классификации sc-элементов */

```

Соотношение между семантически и синтаксически выделяемыми классами sc-элементов в рамках Ядра SC-кода

⊃=
{

семантически выделяемый класс sc-элементов

- := [класс sc-элементов, определяемый сущностями, которые обозначаются этими sc-элементами, также доступностью (активностью использования) sc-элементов в процессе эксплуатации и эволюции базы знаний]
- ∈ *обозначение терминальной сущности*
 ⊃ **sc-узел общего вида**
- ∈ *обозначение небинарной связи*
 ⊃ **sc-узел общего вида**
- ∈ *обозначение sc-пары*
 ⇔ *пара пересекающихся множеств**:
 sc-узел общего вида
 ⊃ **sc-коннектор**
 ⇒ *примечание**:
 [обозначение sc-пары может быть представлено либо sc-узлом общего вида, либо sc-коннектором. При этом каждый sc-коннектор представляет собой обозначение sc-пары.]
- ∈ *обозначение неориентированной sc-пары*
 := [обозначение бинарной неориентированной связи между sc-элементами]
 ⇔ *пара пересекающихся множеств**:
 • **sc-узел общего вида**
 • **sc-ребро общего вида**
 ⇒ *примечание**:
 [обозначение неориентированной sc-пары может быть представлено либо sc-узлом общего вида, либо sc-ребром. При этом не каждое sc-ребро представляет обозначение неориентированной sc-пары. Некоторые из них представляют обозначения ориентированных sc-пар неизвестной направленности.]
- ∈ *обозначение ориентированной sc-пары неизвестной направленности*
 ⇔ *пара пересекающихся множеств**:
 • **sc-узел общего вида**
 • **sc-ребро общего вида**
- ∈ *обозначение ориентированной sc-пары*
 := [обозначение бинарной ориентированной связи между sc-элементами]
 ⇔ *пара пересекающихся множеств**:
 • **sc-узел общего вида**
 • **sc-ребро общего вида**
 ⊃ *sc-дуга общего вида*
- ∈ *константная постоянная позитивная sc-пара принадлежности*
 ⇔ *пара пересекающихся множеств**:
 • **sc-узел общего вида**
 • **sc-ребро общего вида**
 • **sc-дуга общего вида**
 ⊃ **базовая sc-дуга**
 ⇐ *пересечение множеств**:
 { • *sc-константа*
 • *обозначение постоянной сущности*
 • *обозначение sc-пары принадлежности*
 }
- ∈ *обозначение класса*
 C **sc-узел общего вида**
- ∈ *обозначение структуры*
 C **sc-узел общего вида**
- } /* Завершили Описание Соотношения между семантически и синтаксически выделяемыми классами sc-элементов в рамках Ядра SC-кода. В этом описании жирным курсивом выделены идентификаторы (имена) синтаксически выделяемых классов sc-элементов */

Ядро SC-кода

⇒ семантические правила*:

Семантические правила Ядра SC-кода

- $$\supseteq \{$$
- [Каждый sc-элемент является знаком (обозначением) некоторой описываемой сущности.]
 - [Любая сущность может быть обозначена sc-элементом и, соответственно, описана в виде конструкции Ядра SC-кода.]
 - [С помощью sc-элементов можно описать любые связи между sc-элементами и/или между сущностями, которые обозначаются этими sc-элементами. При этом указанные связи трактуются как множества связываемых sc-элементов и обозначаются sc-ребрами, sc-дугами, а в случае небинарных связей – sc-узлами.]
 - [Поскольку каждый sc-коннектор семантически трактуется как обозначение пары sc-элементов, связываемых (соединяемых) этим sc-коннектором, каждая пара инцидентности sc-коннектора семантически интерпретируется как обозначение пары принадлежности, связывающей sc-коннектор с одним из элементов обозначаемой им пары sc-элементов.]
 - [Любая описываемая сущность может быть обозначена sc-узлом общего вида, но обратное неверно, т.к. некоторые сущности могут быть обозначены sc-ребрами общего вида, sc-дугами общего вида, базовыми sc-дугами.]
 - [Каждое sc-ребро является обозначением либо бинарной неориентированной связи между sc-элементами, либо бинарной ориентированной связи неизвестной направленности между sc-элементами.]
 - [Любая бинарная неориентированная связь между sc-элементами может быть обозначена sc-ребром, но обратное неверно.]
- $$\}$$

} /* Завершили описание Денотационной семантики Ядра SC-кода */

Правила синтаксической трансформации sc-элементов в рамках Ядра SC-кода

:= [Правила модификации синтаксического типа sc-элементов в рамках Ядра SC-кода]

- $$\supseteq \{$$
- [Если sc-узел общего вида является обозначением sc-пары, то он трансформируется в sc-коннектор]
 - [Если sc-узел общего вида является обозначением неориентированной sc-пары или обозначением ориентированной sc-пары неизвестной направленности, то он трансформируется в sc-ребро общего вида]
 - [Если sc-узел общего вида или sc-ребро общего вида являются обозначением ориентированной sc-пары и при этом дополнительно указана направленность этой sc-пары, то она трансформируется в sc-дугу общего вида.]
 - [Если sc-узел общего вида или sc-ребро общего вида или sc-дуга общего вида являются константными постоянными позитивными парами принадлежности, то они трансформируются в базовую sc-дугу.]
- $$\}$$

следует отличать*

- $$\ni \{$$
- синтаксически выделяемый класс sc-элементов в рамках Ядра SC-кода
 - синтаксически выделяемый класс sc-элементов в рамках SC-кода
 - семантически выделяемый класс sc-элементов
- $$\}$$

} /* Завершили Сегмент “Описание Ядра SC-кода” */

/* Сегмент *****/

SC-код как синтаксическое расширение Ядра SC-кода

:=

{

Сравнение SC-кода и Ядра SC-кода

:=

{

SC-код

← синтаксическое расширение языка*:

Ядро SC-кода

:= [Синтаксическое расширение Ядра SC-кода]
 ⇒ *примечание**:
 [Синтаксическое расширение Ядра SC-кода заключается во введении дополнительного класса синтаксически эквивалентных элементарных фрагментов конструкций Ядра SC-кода – sc-элементов, обозначающих внутренние файлы, хранимые в памяти ostis-системы]

⇔ *семантическая эквивалентность языков**:
 Ядро SC-кода

⇒ *примечание**:
 [Семантическая эквивалентность SC-кода и Ядра SC-кода является следствием того, что SC-код является синтаксическим расширением Ядра SC-кода]

:= [Результат введения в Ядро SC-кода sc-узлов, имеющих содержимое и обозначающих файлы, хранимые в памяти ostis-системы, т.е. внутренние файлы ostis-системы]

⇒ *примечание**:
 [Результатом просмотренного расширения Ядра SC-кода является расширение Алфавита Ядра SC-кода]

⇒ *примечание**:
 [Все файлы, представляющие собой электронные образы инородных для SC-кода информационных конструкций, можно представить в SC-коде с помощью графовых структур, в которых sc-элементы обозначают буквы текстов или пиксели изображений. Но такой вариант кодирования внешних для ostis-системы информационных конструкций не дает возможности непосредственно использовать накопленный человечеством арсенал электронных информационных ресурсов.]

⇒ *примечание**:
 [Важнейшим видом внутренних файлов ostis-систем являются внутренние файлы внешних идентификаторов sc-элементов (в частности, имен sc-элементов), представляющих sc-элементы в текстах внешних языков (в том числе, в текстах SCs-кода и SCn-кода)]

{ • Ядро SC-кода
 • SC-код
 }

⇒ *сравнение**:
 [Множество всех элементов конструкций Ядра SC-кода и Множество всех элементов конструкций SC-кода полностью совпадают, т.к. для каждого элемента конструкции Ядра SC-кода существует синонимичный ему элемент конструкции SC-кода и наоборот. Из этого следует, что семантическая классификация элементов информационных конструкций SC-кода и Ядра SC-кода также полностью совпадают.

Семантика SC-кода ничем не отличается от семантики Ядра SC-кода. То есть все, что может быть обозначено и описано текстами SC-кода, может быть обозначено и описано текстами Ядра SC-кода. Отличие SC-кода от Ядра SC-кода заключается только в том, что в SC-код добавляется новый синтаксически выделяемый класс sc-элементов – класс sc-элементов, являющихся знаками конкретных (константных) файлов, хранимых в памяти ostis-системы.

Такие "внутренние" файлы необходимы для того, чтобы в памяти ostis-системы можно было хранить и обрабатывать информационные конструкции, не являющиеся текстами SC-кода, что необходимо при вводе (восприятии) информации, поступающей извне, а также при генерации информационных конструкций, передаваемых другим субъектам.

Включение в SC-код специальных синтаксически выделяемых sc-узлов, обозначающих хранимые в sc-памяти электронные образы (файлы) различного вида информационных конструкций, не являющихся конструкциями SC-кода, дает возможность непосредственно в памяти ostis-системы, то есть в одной и той же запоминающей среде обрабатывать не только конструкции SC-кода, но и "инородные" для него конструкции, что для необходимо для реализации интерфейса ostis-системы, обеспечивающего ее взаимодействие с внешней средой.

Без такой реализации интерфейса ostis-системы невозможно реализовать синтаксический анализ, семантический анализ и понимание, а также невозможно реализовать синтез (генерацию) внешних информационных конструкций, принадлежащих заданному внешнему языку и семантически эквивалентных заданному смыслу.

Поскольку все синтаксические и семантические свойства SC-кода и Ядра SC-кода являются весьма близкими, при описании SC-кода акцентируется внимание на его отличия от Ядра SC-кода, а также на более детальное рассмотрение семантической классификации элементов.]

} /* Завершили Сравнение SC-кода и Ядра SC-кода */

Синтаксис SC-кода
 $\supset =$
 $\{$
SC-код
 \Rightarrow *синтаксис**:

синтаксис SC-кода
 \Rightarrow *примечание**:

[*Синтаксис SC-кода* отличается от *Синтаксиса Ядра SC-кода* только тем, что в *Алфавит SC-кода* дополнительно вводится класс *sc-узлов*, являющихся знаками *файлов*, хранимых в памяти *ostis-системы*]

 \Rightarrow *множество всех экземпляров конструкций данного языка**:

sc-элемент
 $:=$ [элемент конструкции SC-кода]

 \Rightarrow *примечание**:

[Множество всех элементов конструкций SC-кода совпадает со множеством всех элементов конструкций Ядра SC-кода. Просто в конструкциях SC-кода некоторые *sc-элементы*, имеющие "синтаксическую метку" (синтаксический тип) *sc-узла общего вида*, будут иметь "метку" *sc-узла*, являющегося знаком *внутреннего файла*, хранимого в памяти *ostis-системы*]

 \Rightarrow *пояснение**:

[*Синтаксис SC-кода* задается

- типологией (алфавитом) *sc-элементов* (атомарных фрагментов текстов SC-кода);
- правилами соединения (инцидентности) *sc-элементов* (например, *sc-элементы* каких типов не могут быть инцидентными друг другу);
- типологией конфигураций *sc-элементов* (связки, классы, структуры), связями между конфигурациями *sc-элементов* (в частности, теоретико-множественными)

 $]$
Алфавит SC-кода
 \Leftarrow *алфавит**:

SC-код
 \Rightarrow *разбиение**:

sc-элемент
 $=$ (*Алфавит Ядра SC-кода* \cup {*внутренний файл ostis-системы*})

 $=$ {

- *sc-узел общего вида*
- ***внутренний файл ostis-системы***
- *sc-ребро общего вида*
- *sc-дуга общего вида*
- *базовая sc-дуга*

 $}$
Алфавит SC-кода
 $:=$ [Алфавит *sc-элементов* в рамках SC-кода]

 $:=$ [Семейство всех максимальных множеств синтаксически эквивалентных (в рамках SC-кода) *sc-элементов*]

 $:=$ [Семейство классов синтаксически эквивалентных *sc-элементов* SC-кода]

 $:=$ [Семейство всех множеств, в каждое из которых входят все синтаксически эквивалентные друг другу (в рамках SC-кода) *sc-элементы* и только они]

 $:=$ [Фактор-множество отношения "синтаксическая эквивалентность *sc-элементов* в рамках SC-кода"]

 $=$ *фактор-множество*(синтаксическая эквивалентность sc-элементов в рамках SC-кода*)*
 \in *сложный внешний идентификатор sc-элемента*
 $:=$ [Семейство множеств *sc-элементов*, являющихся результатом разбиения максимального множества *sc-элементов* SC-кода (класса всевозможных *sc-элементов*) по признаку синтаксической эквивалентности *sc-элементов*]

 $:=$ [Признак (параметр) синтаксической эквивалентности *sc-элементов*]
внутренний файл ostis-системы
 $:=$ [*sc-узел*, имеющий содержимое]

 $:=$ [*sc-ссылка*]

- := [множество всевозможных sc-узлов, имеющих содержимое, хранимое в памяти ostis-системы]
- := [внутренний файл, хранимый в памяти ostis-системы]
- := [внутренний файл для заданной ostis-системы (той ostis-системы, в памяти которой хранится sc-узел, обозначающий этот файл)]
- := [sc-узел, являющийся знаком конкретного файла, хранимого в той же sc-памяти (в качестве содержимого sc-узла), в которой находится и сам указанный sc-узел]
- := [файл, знак которого находится в той же sc-памяти, в которой находится и сам файл]
- := ["свой" файл ostis-системы]
- С *внутренняя информационная конструкция*

синтаксически выделяемый класс sc-элементов в рамках SC-кода

- := [класс sc-элементов, определяемый на основе Алфавита SC-кода]
- ⊃ *Алфавит SC-кода*
- ∋ *sc-узел, не являющийся знаком внутреннего файла ostis-системы*

Синтаксическая классификация sc-элементов в рамках SC-кода

- = {
 - sc-элемент*
 - ⇒ *разбиение**:
 - {
 - *sc-узел общего вида*
 - ⇒ *разбиение**:
 - *sc-узел, не являющийся знаком внутреннего файла ostis-системы*
 - *внутренний файл ostis-системы*
 - *sc-коннектор*
 - ⇒ *разбиение**:
 - *sc-ребро общего вида*
 - *sc-дуга*
 - ⇒ *разбиение**:
 - *sc-дуга общего вида*
 - *базовая sc-дуга*
- }
 - ⇒ *примечание**:
 - [Данная Синтаксическая классификация sc-элементов от Синтаксической классификации sc-элементов Ядра SC-кода отличается только дополнительным уточнением синтаксической типологии sc-узлов]
- } /* Завершили представление Синтаксиса SC-кода */

Денотационная семантика SC-кода

- ⊃=
- {
- Денотационная семантика SC-кода*
- ⇒ *аннотация**:
 - [Денотационную семантику SC-кода рассмотрим как расширение и уточнение Денотационной семантики Ядра SC-кода (смотрите предыдущий сегмент "Описание Ядра SC-кода"). Изложение построим как последовательное уточнение следующих понятий:
 - *sc-переменная*
 - *обозначение дискретной информационной конструкции*
 - *дискретная информационная конструкция* (рассмотрим различные параметры и отношения, заданные на множестве дискретных информационных конструкций)
 - *знание* (как частный вид дискретных информационных конструкций)
 - *файл* (как *sc-константа*, являющаяся *обозначением файла*)
 - *внутренний файл ostis-системы*
 - *структура* (как *дискретная информационная конструкция*, принадлежащая *SC-коду*)
-]

SC-код

⇒ денотационная семантика*:
 Денотационная семантика SC-кода
 ⇒ пояснение*:
 [Денотационная семантика SC-кода задается
 □ семантической интерпретацией sc-элементов и их конфигураций;
 □ семантической интерпретацией инцидентности sc-элементов;
 □ иерархической системой предметных областей;
 □ структурой используемых понятий в каждой предметной области (исследуемые классы объектов, исследуемые отношения, исследуемые классы объектов отношений из смежных предметных областей, ключевые экземпляры исследуемых классов объектов);
 □ онтологиями предметных областей;
]

Классификация sc-переменных

⊃=
 {

sc-переменная

:= [sc-элемент, представляющий собой обозначение произвольной (переменной) сущности из некоторого дополнительно уточняемого множества обозначений других сущностей, которые считаются возможными значениями указанной произвольной сущности]

⇐ область задания*:
 значение переменной*

:= [Бинарное ориентированное отношение, связывающее sc-переменные с их возможными значениями*]

⇒ пояснение*:

[Это одно из отношений, заданных на множестве sc-переменных]

⇒ разбиение*:

Структурная типология sc-переменных

= { • произвольная терминальная сущность
 := [sc-переменная, обозначающая терминальную сущность]
 := [sc-переменная, значением или значением значения и т.д. которой является терминальная сущность]
 := [sc-переменная, "конечным" значением которой является терминальная сущность]
 := [обозначение произвольной терминальной сущности]
 • произвольное множество sc-элементов
 }

⇒ разбиение*:

{ • sc-переменная, у которой логический уровень всех ее значений одинаков
 ⊃ первичная sc-переменная
 := [sc-переменная, все значения которой являются sc-константами]
 ⊃ вторичная sc-переменная
 := [sc-переменная, все значения которой являются первичными sc-переменными]
 ⊃ sc-переменная третьего уровня
 := [sc-переменная, все значения которой являются вторичными sc-переменными]
 • sc-переменная, значения которой имеют различный логический уровень
 }

⇒ разбиение*:

{ • sc-переменная, у которой синтаксический тип всех её значений одинаков
 ⊃ переменный sc-узел
 := [sc-переменная, все значения которой являются sc-узлами]
 ⊃ переменное sc-ребро
 ⊃ переменная sc-дуга
 • sc-переменная, значения которой имеют различный синтаксический тип
 }

} /* Завершили Классификации sc-переменных */

обозначение дискретной информационной конструкции

- ⇒ разбиение*:
- обозначение дискретной информационной конструкции, не принадлежащей SC-коду
 - **обозначение структуры**
- := [обозначение дискретной информационной конструкции, принадлежащей SC-коду]
- := [обозначение sc-конструкции]
- }
- ⇒ разбиение*:
- произвольная дискретная информационная конструкция
- := [sc-переменная, обозначающая дискретную информационную конструкцию]
- **дискретная информационная структура**
- := [sc-константа, обозначающая конкретную дискретную информационную конструкцию]
- }

Описание параметров и отношений, заданных на дискретных информационных конструкциях

⊇=
{

параметр, заданный на множестве дискретных информационных конструкций[^]

- ⊃
- ⊃ типология дискретных информационных конструкций, определяемая их носителем[^]
 - ⊃ некомпьютерная форма представления дискретных информационных конструкций[^]
 - ⊃ файл
- := [компьютерная форма представления дискретных информационных конструкций в линейной адресной памяти]
- ⊃ структура
- := [компьютерная форма представления дискретных информационных конструкций в графодинамической ассоциативной памяти]
- := [представление дискретных информационных конструкций в виде конструкций SC-кода в памяти ostis-систем]
- ⊃ типология дискретных информационных конструкций, определяемая их соотношением с памятью ostis-систем[^]
- ⊃ внешняя дискретная информационная конструкция ostis-системы
- := [дискретная информационная конструкция, которая находится вне памяти той ostis-системы, в которой находится sc-узел, обозначающий эту информационную конструкцию]
- ⇒ разбиение*:
- некомпьютерная форма представления дискретных информационных конструкций
- ⇒ примечание*:
- [Очевидно, что информационные конструкции такого вида принципиально не могут быть внутренними информационными конструкциями ostis-систем, хранимыми в их памяти.]
- внешний файл ostis-системы
- ⇒ разбиение*:
- файл компьютерной системы, которая не является ostis-системой
- := [файл, который не хранится в памяти данной ostis-системы, но о которой известно, какая система, не являющаяся ostis-системой, им "владеет" и как его "скачать"]
- := [внешний файл ostis-системы, принадлежащий компьютерной системе, которая не является ostis-системой]
- файл другой ostis-системы
- := [файл, который не является внутренним файлом данной ostis-системы, в памяти которой находится знак этого файла, но является внутренним знаком другой ostis-системы]
- := [внешний файл ostis-системы, принадлежащий другой ostis-системе]
- }
- внешняя структура ostis-системы
- := [структура, хранимая в памяти другой ostis-системы]
- := [структура другой ostis-системы]
- }

- ⇒ *внутренняя информационная конструкция ostis-системы*
 := [внутренняя для заданной ostis-системы информационная конструкция]
 := [внутренняя информационная конструкция той ostis-системы, в памяти (sc-памяти) которой хранится знак (sc-узел) этой информационной конструкции]
 ⇒ *примечание**:
 [Внутренние информационные конструкции ostis-систем (т.е. конструкции, обрабатываемые в их памяти) могут быть только дискретными, хотя и не обязательно знаковыми.]
 ⇒ *разбиение**:
 {
 • *внутренний файл ostis-системы*
 • *внутренняя структура*
 := [структура, которой в памяти данной ostis-системы соответствует не только знак этой структуры, но и она сама]
 := [структура, хранимая и обрабатываемая в памяти данной ostis-системы]
 := [внутренняя структура ostis-системы]
 }
 ⇒ *разбиение**:
 {
 • *сформированная внутренняя информационная конструкция ostis-системы*
 • *частично сформированная внутренняя информационная конструкция ostis-системы*
 • *внутренняя информационная конструкция ostis-системы на начальной стадии формирования*
 }

⇒ *типология дискретных информационных конструкций, определяемая правилами, которым они должны удовлетворять*[^]

- ⇒ *информационная конструкция Русского языка*
 ⇒ *информационная конструкция Английского языка*
 ⇒ *структура*

- := [информационная конструкция SC-кода]
 ⇒ *sc.g-конструкция*
 := [информационная конструкция SCg-кода]
 ⇒ *sc.s-конструкция*
 := [информационная конструкция SCs-кода]
 ⇒ *sc.n-конструкция*
 := [информационная конструкция SCn-кода]

⇒ *наличие синтаксической связности*[^]

- ⇒ *синтаксически связная дискретная информационная конструкция*
 := *определение**:
 [дискретная информационная конструкция, у которой для каждой пары её элементов существует маршрут, соединяющий эти элементы и проходящий по связям их инцидентности]
 ⇒ *синтаксически несвязная дискретная информационная конструкция*
 ⇒ *примечание**:
 [Можно оценивать "силу" синтаксической связности – наличие и число "мостов" в графе инцидентности элементов дискретной информационной конструкции, наличие и число точек "сочленения", минимальное число элементов конструкции, удаление которых приводит к несвязности. Можно также оценивать уровень синтаксической несвязности дискретной информационной конструкции числом компонентов связности этой конструкции.]

⇒ *наличие семантической связности*[^]

- ⇒ *примечание**:
 [Свойством семантической связности могут обладать только знаковые конструкции.]
 ⇒ *семантически связная дискретная информационная конструкция*
 ⇒ *определение**:
 [Это конструкция, которая обладает следующим свойством: для любой ее декомпозиции на два синтаксически правильных компонента всегда найдется пара синонимичных знаков, один из которых находится в одном компоненте, а другой – в другом.]
 ⇒ *семантически несвязная дискретная информационная конструкция*

отношение, заданное на множестве дискретных информационных конструкций[^]

- ⇒ *дискретная информационная конструкция заданного языка*^{*}
 ∈ *отношение, заданное на множестве языков*[^]
 ⇒ *разбиение**:
 {
 • *синтаксически неправильная дискретная информационная конструкция заданного языка*^{*}

	<pre> ⇐ объединение*: { • синтаксически некорректная дискретная информационная конструкция данного языка* • синтаксически нецелостная дискретная информационная конструкция данного языка* } • текст данного языка* ⇐ пересечение*: { • синтаксически корректная дискретная информационная конструкция данного языка* • синтаксически целостная дискретная информационная конструкция данного языка* } ⇒ разбиение*: { • семантически неправильный текст данного языка* ⇐ объединение*: { • семантически некорректный текст данного языка* • семантически нецелостный текст данного языка* } • знание, представленное в данном языке* ⇐ пересечение*: { • семантически корректный текст данного языка* • семантически целостный текст данного языка* } } } </pre>	<p>/* Завершили Описание параметров и отношений, заданных на дискретных информационных конструкци- ях */</p>
обозначение файла	<pre> ⇒ разбиение*: { • произвольный файл := [sc-переменная, каждым значением которой является обозначение файла] := [обозначение произвольного файла] := [sc-переменная, обозначающая файл] • файл := [знак конкретного файла] := [sc-константа, обозначающая конкретный файл] } ⇒ разбиение*: { • обозначение внешнего файла ostis-системы ⇒ разбиение*: { • произвольный внешний файл ostis-системы • внешний файл ostis-системы } • обозначение внутреннего файла ostis-системы ⇒ разбиение*: { • произвольный внутренний файл ostis-системы • внутренний файл ostis-системы } } </pre>	
файл	<pre> := [sc-узел, обозначающий файл] := [знак файла] ⇒ разбиение*: { • ея-файл := [естественно-языковой файл] • файл, являющийся текстом формального языка ⊃ sc.g-файл } </pre>	

	⊃	<i>sc.s-файл</i>
	⊃	<i>sc.n-файл</i>
	•	<i>файл, отражающий процесс изменения sc.g-текста</i>
	•	<i>графический файл</i>
	•	<i>файл-изображение</i>
	•	<i>видео-файл</i>
	•	<i>аудио-файл</i>
	}	
⇒		<i>разбиение*</i> :
	{	• <i>файл-экземпляр</i>
		:= [файл, являющийся конкретным электронным документом или электронным образом конкретной внешней информационной конструкции]
	•	<i>файл-образец</i>
		:= [файл-класс ostis-системы]
		:= [файл, являющийся одновременно также и знаком множества всевозможных экземпляров (копий) этого файла]
	}	
⇒		<i>разбиение*</i> :
	{	• <i>внешний файл ostis-системы</i>
		• <i>внутренний файл ostis-системы</i>
	}	

внутренний файл ostis-системы

∈ *синтаксически выделяемый класс sc-элементов в рамках SC-кода*

∈ *семантически выделяемый класс sc-элементов в рамках SC-кода*

⇒ *примечание**:

[Данный класс sc-элементов, являющихся знаками файлов, хранимых в памяти ostis-систем, в отличие от других синтаксически выделяемых классов sc-элементов, представляет собой одновременно синтаксически и семантически выделяемый класс sc-элементов. Это обусловлено (1) тем, что каждый экземпляр данного класса sc-элементов является знаком конкретного файла, хранимого в памяти ostis-системы, и (2) тем, что каждый файл, хранимый в памяти ostis-системы, может и должен быть обозначен только таким sc-элементом, который является экземпляром рассматриваемого класса sc-элементов.]

⇒ *примечание**:

[sc-узел может быть знаком файла, находящегося в памяти другой ostis-системы (не в той, в которой хранится этот sc-узел). Но в этом случае указанный sc-узел не будет принадлежать рассматриваемому классу sc-узлов.]

:= [знак файла ostis-системы, хранимого в "моей" памяти]

⇒ *примечание**:

[Следует отличать синтаксическую эквивалентность файлов, семантическую эквивалентность файлов и совпадение файлов (когда речь идет об одном и том же файле). Т.е. копия файла и один и тот же файл – это разные вещи.]

Классификация структур

⊃=

{

структура

:= [структура, элементами которой являются sc-элементы]

:= [множество sc-элементов (множество), не являющиеся ни связкой (связкой sc-элементов), ни классом (множеством всех sc-элементов, эквивалентных в определенном смысле)]

:= [знак конкретной (константной) структуры]

:= [Класс всех тех и только тех sc-элементов, каждый из которых является знаком конкретной структуры]

:= [Знак класса всех sc-элементов, являющихся знаками конкретных структур]

:= [Константный sc-элемент (точнее, sc-узел), являющийся знаком конкретного класса всех sc-элементов, являющихся знаками конкретных структур]

:= [sc-конструкция]

:= [информационная конструкция, принадлежащая SC-коду]

⊃ *sc-текст*

:= [структура, удовлетворяющая синтаксическим правилам SC-кода]

	\supset знание := [семантически корректный и целостный sc-текст]
⇒	<i>разбиение*</i> : Наличие sc-переменных, входящих в состав структуры = { <ul style="list-style-type: none"> • структура, в составе которой sc-переменные не входят • структура, в составе которой входят sc-переменные ⇒ <i>примечание*</i> : [Такие структуры при представлении логических высказываний в SC-коде являются аналогами атомарных логических формул.]
⇒	} <i>разбиение*</i> : Темпоральная характеристика структур = { <ul style="list-style-type: none"> • <i>ситуативная структура</i> := [ситуация, представленная в SC-коде] := [ситуация] := [структура, в состав которой входят знаки временных сущностей и которая сама является временной сущностью (при этом время существования такой структуры совпадает с временем одновременного существования всех временных сущностей, знаки которых входят в состав этой ситуативной структуры).] ⇒ <i>разбиение*</i>: { <ul style="list-style-type: none"> • ситуация во внешней среде • ситуация в sc-памяти } • структура, не содержащая знаков временных сущностей • динамическая структура ⇒ <i>пояснение*</i>: [В отличие от ситуативной структуры конфигурация динамической структуры меняется во времени в зависимости от момента появления и момента завершения существования каждой временной сущности (в том числе временной связи), знак которой входит в состав динамической структуры. Каждой динамической структуре можно поставить в соответствие темпоральную последовательность состояний (ситуаций) и событий.] }
⇒	<i>разбиение*</i> : Наличие связности структур = { <ul style="list-style-type: none"> • <i>связная структура</i> ⇒ <i>разбиение*</i>: <i>Связность структур</i>[^] := [Минимальное число sc-элементов, удаление которых преобразует связную структуру в несвязную] ∈ <i>одно-связная структура</i> ⇐ <i>разбиение*</i>: <i>Признак классификации структур по числу точек сочленения</i>[^] ⇐ <i>разбиение*</i>: <i>Признак классификации структур по числу мостов</i>[^] • <i>несвязная структура</i> ⇐ <i>разбиение*</i>: <i>Признак классификации структур по числу компонентов связности</i>[^] С <i>тривиальная структура</i> := [структура, в состав элементов которой sc-коннекторы не входят] }
⇒	<i>примечание*</i> : [Важнейшей особенностью SC-кода является то, что для конструкций SC-кода (для структур) нет необходимости противопоставлять синтаксическую и семантическую связность, то есть все синтаксически связанные структуры являются также и семантически связными и наоборот.]
⇒	<i>разбиение*</i> : Рефлексивность структур = { <ul style="list-style-type: none"> • <i>рефлексивная структура</i> := [структура, в число элементов которой входит sc-узел, обозначающий саму эту структуру] С <i>рефлексивное множество</i>

	<ul style="list-style-type: none"> • <i>нерефлексивная структура</i>
⇒	<i>разбиение*</i> : Целостность структур по связкам
=	<ul style="list-style-type: none"> • <i>структура, содержащая все компоненты всех своих связей</i> • <i>структура, не содержащая все компоненты всех своих связей</i>
⇒	<i>разбиение*</i> : <ul style="list-style-type: none"> • <i>синтаксически неправильная структура</i>
	:= [синтаксически неправильно построенная структура]
⇐	<i>объединение*</i> : <ul style="list-style-type: none"> • <i>синтаксически некорректная структура</i>
	:= [структура, содержащая фрагменты, противоречащие <i>Синтаксическим правилам SC-кода</i> (ошибочные фрагменты)]
	<ul style="list-style-type: none"> • <i>синтаксически нецелостная структура</i>
	:= [структура, в которой имеется синтаксически выявленная недостаточность, неполнота (то есть имеется некоторое количество информационных дыр)]
⇒	<i>примечание*</i> : [Разделение <i>Синтаксических правил SC-кода</i> на правила анализа синтаксической корректности и правила анализа синтаксической целостности (полноты) существенно упрощает процедуру синтаксического анализа <i>структур</i> .]
	<ul style="list-style-type: none"> • <i>sc-текст</i>
	:= [синтаксически правильная структура]
	:= [синтаксически правильно построенная структура]
⇐	<i>пересечение*</i> : <ul style="list-style-type: none"> • <i>синтаксически корректная структура</i> • <i>синтаксически целостная структура</i>
⇒	<i>разбиение*</i> : <ul style="list-style-type: none"> • <i>семантически неправильный sc-текст</i>
	⇐ <i>объединение*</i> : <ul style="list-style-type: none"> • <i>семантически некорректный sc-текст</i> • <i>семантически нецелостный sc-текст</i>
	<ul style="list-style-type: none"> • <i>знание</i>
	:= [семантически правильно построенный sc-текст]
⇐	<i>пересечение*</i> : <ul style="list-style-type: none"> • <i>семантически корректный sc-текст</i> • <i>семантически целостный sc-текст</i>
	}
}	/* Завершили <i>Классификацию структур</i> */
	знание
:=	[дискретная информационная конструкция, являющаяся знанием, представленная в некотором (дополнительно уточняемом) языке]
⇐	<i>второй домен*</i> : <i>знание, представленное в заданном языке*</i>
⊂	<i>знаковая конструкция</i>
⇒	<i>примечание*</i> : [Каждое знание является знаковой конструкцией, но не каждая знаковая конструкция является знанием, а только та, смысловое представление которой удовлетворяет определенным требованиям корректности и целостности.]
∈	<i>семантически выделяемый класс дискретных информационных конструкций</i> [^]
	следует отличать*
∃	<ul style="list-style-type: none"> • { <ul style="list-style-type: none"> • <i>дискретная информационная конструкция</i> • <i>текст</i>

```

    • знание
    }
    ∈ следует отличать*
    • { • структура
      • sc-текст
      • знание
      }
    ∈ следует отличать*
  }
} /* Завершили представление Денотационной семантики SC-кода */

} /* Завершили Сегмент “SC-код как синтаксическое расширение Ядра SC-кода” */

/* Сегмент *****/
Использование SC-кода для формального описания собственного синтаксиса
⊃=
{
[В предыдущем сегменте “SC-код как синтаксическое расширение Ядра SC-кода” рассмотрен Синтаксис SC-кода
путём:


- введения синтаксически выделяемых классов sc-элементов в рамках SC-кода;
- описания теоретико-множественных связей между указанными классами sc-элементов (к такому описанию, в частности, относится Синтаксическая классификация sc-элементов в рамках SC-кода);
- введения двух отношений инцидентности sc-элементов – Отношения инцидентности sc-коннекторов* и Отношения инцидентности входящих sc-дуг*;
- описания Синтаксических правил SC-кода, которые, прежде всего, описывают формальные свойства указанных выше отношений инцидентности sc-элементов.


Однако для того, чтобы получить возможность все (!) Синтаксические правила SC-кода записать средствами самого SC-кода, необходимо иметь явное представление пар отношений инцидентности sc-элементов в виде sc-дуг, принадлежащим этим отношениям. В случае, если указанные sc-дуги инцидентности являются sc-переменными, логико-семантических проблем не возникнет. И этого, кстати, вполне достаточно, чтобы Синтаксические правила SC-кода, сформулированные в виде логических высказываний, записать средствами SC-кода. Но, если разрешить sc-дугам инцидентности быть sc-константами, то, во-первых, в Синтаксические правила SC-кода необходимо добавить Правило удаления константной sc-дуги инцидентности, если эта инцидентность представлена неявно, а, во-вторых, в Правила синтаксической трансформации sc-элементов необходимо добавить Правило трансформации (замены) константной sc-дуги инцидентности на неявное представление этой инцидентности.
В теоретическом и, возможно, даже в практическом плане может быть интересна такая синтаксическая модификация (синтаксическое расширение) SC-кода, в котором:


- все неявно представленные пары инцидентности sc-элементов заменяются на константные sc-дуги инцидентности – неявно представленными парами инцидентности остаются только пары инцидентности константных sc-дуг инцидентности с компонентами этих sc-дуг;
- В Алфавит SC-кода вводятся два новых синтаксически выделяемых класса sc-элементов – класс sc-дуг инцидентности sc-коннекторов, а также класс sc-дуг инцидентности входящих sc-дуг.


В результате такого преобразования конструкций SC-кода конструкции SC-кода перестают быть графовыми конструкциями нетрадиционного вида, в которых рёбра, гиперрёбра, дуги могут быть инцидентны другим рёбрам, гиперребрам и дугам, а становятся классическими графами с двумя типами дуг (с sc-дугами инцидентности sc-коннекторов и с sc-дугами инцидентности входящих sc-дуг) и с пятью типами вершин (с вершинами, представляющими sc-узлы общего вида, с вершинами, представляющими sc-узлы, являющиеся знаками внутренних файлов ostis-системы, с вершинами, представляющими sc-рёбра общего вида, с вершинами, представляющими sc-дуги общего вида, с вершинами, представляющими базовые sc-дуги).
Рассмотренное преобразование конструкций SC-кода в теории графов называется подразделением или подразбие-нием графа (Смотрите Trudeau.R.J.IntroGT-1993кн.)]
} /* Завершили Сегмент “Использование SC-кода для формального описания собственного синтаксиса” */

} /* Завершили Раздел “Описание внутреннего языка ostis-систем” */

```

```

/* Раздел *****/
Предметная область и онтология внешних идентификаторов знаков, входящих в информационные конструкции внутреннего языка ostis-систем
⊃=
{
← конкатенация сегментов*:
{
• Понятие внешнего идентификатора sc-элемента
• Понятие простого идентификатора sc-элемента
• Понятие сложного идентификатора sc-элемента
}
}

Предметная область внешних идентификаторов знаков, входящих в информационные конструкции внутреннего языка ostis-систем
∈ предметная область
∋ максимальный класс объектов исследования':
sc-идентификатор
∋ класс объектов исследования':
• основной sc-идентификатор
• строковый sc-идентификатор
• системный sc-идентификатор
• нетранслируемый sc-идентификатор
• простой sc-идентификатор
• простой строковый sc-идентификатор
• имя нарицательное
• имя собственное
• sc-выражение
• ограничитель sc-выражений
∋ исследуемое отношение':
• sc-идентификатор*

/* Сегмент *****/
Понятие внешнего идентификатора sc-элемента
⊃=
{
sc-идентификатор
:= [строка символов или пиктограмма, взаимно однозначно представляющая соответствующий sc-элемент, хранимый в sc-памяти]
:= [внешний идентификатор sc-элемента]
⇒ пояснение*:
[Внешние идентификаторы sc-элементов (или, сокращенно sc-идентификаторы) необходимы ostis-системе исключительно для того, чтобы осуществлять обмен информацией с другими ostis-системами или со своими пользователями. Для того чтобы представить свою базу знаний, решать самые различные задачи, связанные с анализом текущего состояния и эволюцией своей базы знаний, задачи, связанные с анализом текущего состояния (текущих ситуаций) окружающей среды, принятием соответствующих решений (целей) и организацией соответствующих действий, направленных на выполнение принятых решений (на достижение поставленных целей), ostis-системе не нужны никакие внешние идентификаторы (в частности, имена) соответствующие sc-элементам. Но для понимания сообщений, принимаемых от других субъектов (что для ostis-системы означает построение sc-текста, семантически эквивалентного принятому сообщению) и для анализа сообщений, передаваемых другим субъектам (что для ostis-системы означает синтез внешнего текста, семантически эквивалентного заданному sc-тексту и удовлетворяющего некоторым дополнительным требованиям, например, эмоционального характера) ostis-системе необходимо знать, как в принимаемом или передаваемом сообщении изображаются (представляются) знаки, синонимичные sc-элементам, которые уже хранятся или могут храниться в составе базы знаний ostis-системы. В качестве внешних идентификаторов sc-элементов чаще всего используются имена (термины) соответствующих (обозначаемых) сущностей, представленные отдельными словами или словосочетаниями на различных естественных языках, но также могут использоваться иероглифы, условные обозначения, пиктограммы.

В общем случае sc-элементу может соответствовать несколько синонимичных ему имен на разных естественных языках. Более того, sc-элементу может соответствовать несколько синонимичных ему

```

имен на одном и том же *естественном языке*. В этом случае одно из этих имен объявляется как основной внешний идентификатор для соответствующего *sc-элемента* и соответствующего *естественного языка*. Основное требование, предъявляемое к таким внешним идентификаторам это отсутствие как синонимов, так и омонимов в рамках множества основных внешних идентификаторов *sc-элементов* для каждого *естественного языка*.

Каждый внешний идентификатор *sc-элемента*, используемый *ostis-системой*, может быть описан (представлен) в её памяти в виде *внутреннего файла ostis-системы*, т.е. в виде электронного образа всевозможных вхождений данного внешнего идентификатора во всевозможные внешние тексты соответствующего внешнего языка. В некоторых случаях явное представление в памяти не требуется, например, в случае *нетранслируемых sc-идентификаторов*.]

⇒ разбиение*:

- *простой sc-идентификатор*
:= [простой внешний идентификатор *sc-элемента*]
- *sc-выражение*
:= [сложный внешний идентификатор *sc-элемента*, в состав которого входит один или несколько идентификаторов других *sc-элементов*]

⇒ разбиение*:

- *основной sc-идентификатор*
:= [основной *sc-идентификатор* для носителей дополнительно указываемого языка общения (например, соответствующего *естественного языка*)]
⇒ *примечание**:
[*основной sc-идентификатор* является уникальным (не имеет синонимов и омонимов) в рамках соответствующего *естественного языка*]
- ⊃ *основной международный sc-идентификатор*
⇒ *примечание**:
[В качестве *основных sc-идентификаторов* могут использоваться также общепринятые международные условные обозначения некоторых сущностей, например, обозначения часто используемых функций (sin, cos, tg, log, и т.д.), единиц измерения, денежных единиц и многое другое. Формально каждый *основной международный sc-идентификатор* считается *основным sc-идентификатором* также и для каждого *естественного языка*, несмотря на то, что символы, используемые в *основных международных sc-идентификаторах*, могут не соответствовать алфавиту некоторых или даже всех *естественных языков*.]
- *неосновной sc-идентификатор*
⇒ *примечание**:
[С помощью *неосновных sc-идентификаторов* указываются возможные *синонимы** соответствующего *основного sc-идентификатора*, которые в частности, могут пояснять или даже определять обозначаемую сущность, указывает на важные свойства этой сущности.]
- ⊃ (*неосновной sc-идентификатор* ∩ *пояснение*)
:= [неосновной *sc-идентификатор*, являющийся одновременно и пояснением обозначаемой сущности]
⊃ (*неосновной sc-идентификатор* ∩ *определение*)
:= [неосновной *sc-идентификатор*, являющийся одновременно и определением обозначаемого понятия]
- ⊃ *неосновной часто используемый sc-идентификатор*
⇒ *пояснение**:
[Для некоторых *sc-элементов* могут часто использоваться не только *основные*, но и *неосновные sc-идентификаторы* (особенно в неформальных текстах – в пояснениях, примечаниях и т.п.). Явное выделение такого класса *sc-идентификаторов* позволяет упростить семантический анализ исходных текстов баз знаний.]
- *системный sc-идентификатор*
⇒ *пояснение**:
[*системный sc-идентификатор* – это *sc-идентификатор*, являющийся уникальным в рамках всей базы знаний *Экосистемы OSTIS (Глобальной базы знаний)*. Данный *sc-идентификатор*, как правило, используется в исходных текстах базы знаний, при обмене сообщениями между *ostis-системами*, а также для взаимодействия *ostis-системы* с компонентами, реализованными с использованием средств, внешних с точки зрения *Технологии OSTIS*, например, программ, написанных на традиционных языках программирования. Алфавит *системных sc-идентификаторов* максимально упрощен для того, чтобы обеспечить удобство автомати-

ческой обработки таких sc-идентификаторов с использованием современных технических средств, в частности, запрещены пробелы и различные специальные символы.]

⇒ *примечание**:

[В качестве указанного языка общения между ostis-системами может использоваться SCs-код.]

}

⇒ *примечание**:

[Каждому sc-элементу может соответствовать целое семейство внешних идентификаторов этого sc-элемента, которые обычно являются терминами, именующими обозначаемую сущность. Среди этих внешних идентификаторов для каждого идентифицируемого sc-элемента выделяется один как основной идентификатор. А неосновные термины (имена), соответствующие этим sc-элементам (в том числе и классам), поясняют денотационную семантику указанного sc-элемента.]

{
основной sc-идентификатор

С *файл-образец ostis-системы*

}

⇔ *семантическая эквивалентность**:

[Все основные идентификаторы sc-элементов в памяти ostis-системы оформляются в виде копируемых файлов-образцов ostis-системы.]

⇒ *пояснение**:

[Копии основных sc-идентификаторов входят в состав внешних текстов различных языков (SCg-кода, SCs-кода, SCп-кода), а также в различных падежах, склонения, спряжениях в состав файлов ostis-систем.]

⇒ *примечание**:

[Аналогичное утверждение справедливо и для неосновных часто используемых sc-идентификаторов. Все остальные неосновные sc-идентификаторы считаются вспомогательными файлами-экземплярами.]

sc-идентификатор

⇒ *разбиение**:

{ • *строковый sc-идентификатор*

:= [sc-идентификатор, представленный строкой символов, которая является именем обозначаемой сущности]

:= [имя сущности, обозначаемой идентифицируемым sc-элементом]

:= [имя (термин, словосочетание), синонимичное соответствующему (идентифицируемому) sc-элементу и представленное в соответствующем алфавите символов]

• *нестроковый sc-идентификатор*

⇒ *пояснение**:

[В общем случае в качестве *sc-идентификатора* некоторого *sc-элемента* может выступать произвольный *внутренний файл ostis-системы*, например, пиктограмма, условное обозначение или даже аудиофрагмент.]

}

⇒ *примечание**:

[Введенные нами sc-идентификаторы используются во всех внешних языках, близких SC-коду – в SCg-коде, в SCs-коде и в SCп-коде.]

строковый sc-идентификатор

:= [имя, приписываемое идентифицируемому sc-элементу]

:= [имя сущности, обозначаемой идентифицируемым sc-элементом]

:= [строка символов, синонимичная соответствующему идентифицируемому sc-элементу]

⊃ *основной строковый sc-идентификатор*

:= [уникальное для каждого естественного языка внешнее имя, приписываемое идентифицируемому sc-элементу]

⊃ *основной русскоязычный sc-идентификатор*

⊃ *системный sc-идентификатор*

⊃ *основной англоязычный sc-идентификатор*

⊃ *основной германоязычный sc-идентификатор*

⊃ *основной франкоязычный sc-идентификатор*

⊃ *основной италияязычный sc-идентификатор*

⊃ *основной китайскоязычный sc-идентификатор*
 ⊃ *системный sc-идентификатор*

sc-идентификатор

⇒ *примечание**:

[Представление знаков в виде неатомарных фрагментов информационных конструкций (в частности, в виде имен обозначаемых сущностей, построенных в фиксированном алфавите), взаимно однозначно соответствующих обозначенным сущностям, необходимо только для того, чтобы иметь простую процедуру установления синонимии знаков, входящих в состав одной или разных информационных конструкций.]

системный sc-идентификатор

⇒ *правила построения**:

{ • [Символами, используемыми в *системном sc-идентификаторе*, могут быть буквы латинского алфавита, цифры, знак нижнего подчеркивания и знак тире. Для обеспечения интернационализации рекомендуется формировать *системные sc-идентификаторы* на основании основных англоязычных *sc-идентификаторов*. Таким образом, наиболее целесообразно формировать *системный sc-идентификатор sc-элемента* из основного англоязычного путем замены всех символов, не входящих в описанный выше алфавит на символ нижнее подчеркивание (“_”). Кроме того, заглавные буквы чаще всего заменяются на соответствующие строчные.]

⇒ *пример применения**:

{
Раздел. Предметная область и онтология множеств

⇒ *основной sc-идентификатор**:

[Section. Subject domain and ontology of sets]

∈ *Английский язык*

∈ *основной sc-идентификатор*

⇒ *системный sc-идентификатор**:

[section_subject_domain_and_ontology_of_sets]

∈ *системный sc-идентификатор*

}

• [Для именования *sc-элементов*, являющихся знаками *ролевых отношений*, вместо знака “/” в *системном sc-идентификаторе* используется приставка “rrel” и далее после нижнего подчеркивания записывается имя *ролевого отношения*.]

⇒ *пример применения**:

{

слагаемое'

⇒ *системный sc-идентификатор**:

[rrel_summand]

∈ *системный sc-идентификатор*

}

• [Для именования *sc-элементов*, являющихся знаками *неролевых отношений*, вместо знака “*” в *системном sc-идентификаторе* используется приставка “nrel” и далее после нижнего подчеркивания записывается имя *неролевого отношения*.]

⇒ *пример применения**:

{

включение*

⇒ *системный sc-идентификатор**:

[nrel_inclusion]

∈ *системный sc-идентификатор*

}

• [Для именования *sc-элементов*, являющихся знаками классов *понятий* в *системном sc-идентификаторе* используется приставка “concept” и далее после нижнего подчеркивания записывается имя *класса*.]

⇒ *пример применения**:

{

треугольник

⇒ *системный sc-идентификатор**:

[concept_triangle]

∈ *системный sc-идентификатор*

}

- [Для именованя *sc*-элементов, являющихся знаками *структур* в *системном sc-идентификаторе* используется приставка “struct” и далее после нижнего подчеркивания записывается имя *структуры*.]
⇒ *пример применения**:

```

{
  Треугольник(ТочкаА;ТочкаВ;ТочкаС)
  ⇒ системный sc-идентификатор*:
  [struct_Triangle_A_B_C]
  ∈ системный sc-идентификатор
}

```
- [Для именованя *sc*-элементов, являющихся знаками *связок* в *системном sc-идентификаторе* используется приставка “tuple” и далее после нижнего подчеркивания записывается имя *связки*.]
}

нетранслируемый *sc*-идентификатор

C системный *sc*-идентификатор

:= [*sc*-идентификатор, не представляемый в базе знаний *ostis*-системы]

:= [*sc*-идентификатор, существующий только вне базы знаний *ostis*-системы]

⇒ *пояснение**:

[*нетранслируемые sc-идентификаторы* используются только в рамках исходных текстов *баз знаний* (в том числе, *sc.s*-текстов) и при обмене сообщениями между *ostis*-системами в тех случаях, когда необходимо в нескольких фрагментах исходного текста *базы знаний* или передаваемого сообщения использовать имя одного и того же *sc*-элемента, но при этом указанный *sc*-элемент не имеет *системного sc-идентификатора* и вводит его нецелесообразно. Использование *нетранслируемых sc-идентификаторов* позволяет повысить читабельность и структурированность исходных текстов *баз знаний*, а также позволяет обратиться к одному и тому же именованному (в рамках базы знаний) *sc*-элементу в разных файлах исходных текстов *баз знаний* или в разных сообщениях, передаваемых между *ostis*-системами. В качестве таких *sc*-элементов часто выступают знаки *структур* и *связок*.

Таким образом, *нетранслируемые sc-идентификаторы* существуют только вне *базы знаний ostis-системы* и при формировании базы знаний из исходных текстов или при погружении в базу знаний полученного сообщения соответствующий им *внутренний файл ostis-системы* не создается.]

⇒ *правило построения**:

[*нетранслируемые sc-идентификаторы* строятся по тем же принципам, что и системные *sc*-идентификаторы, но в начале *нетранслируемого sc-идентификатора* ставится одна или несколько точек (“.”), количество которых определяет область видимости данного *нетранслируемого sc-идентификатора*.]

⇒ *примечание**:

[Данное правило справедливо и для построения *sc-идентификаторов sc-переменных*]

⇒ *пояснение**:

[..var1]

⊃ глобальный нетранслируемый *sc*-идентификатор

⇒ *пояснение**:

[*глобальные нетранслируемые sc-идентификаторы* считаются уникальными в рамках всего имеющегося набора исходных текстов *баз знаний* и/или передаваемых между *ostis*-системами сообщений. Таким образом, *sc*-элементы, имеющие на уровне исходных текстов (в том числе, в разных файлах исходных текстов) одинаковые *глобальные нетранслируемые sc-идентификаторы*, считаются синонимичными и в *базе знаний ostis-системы* представляются одним и тем же *sc*-элементом.

Можно сказать, что областью видимости *глобальных нетранслируемых sc-идентификаторов* является весь набор (репозиторий) исходных текстов некоторой базы знаний.]

⇒ *правило построения**:

[В начале *глобальных нетранслируемых sc-идентификаторов* ставится одна точка]

⇒ *пример*':

- [operator1]

- [node123]

⊃ локальный нетранслируемый *sc*-идентификатор

⇒ *пояснение**:

[*локальные нетранслируемые sc-идентификаторы* считаются уникальными в рамках конкретного файла исходных текстов *баз знаний* и/или передаваемого между *ostis*-системами сообщения. Таким образом, *sc*-элементы, имеющие в рамках одного файла исходных текстов одинаковые *локальные нетранслируемые sc-идентификаторы*, считаются синонимичными, в то время как *sc*-элементы,

имеющие в рамках разных файлов исходных текстов одинаковые *локальные нетранслируемые sc-идентификаторы* будут считаться разными *sc-элементами*.

Можно сказать, что областью видимости *локальных нетранслируемых sc-идентификаторов* является конкретный файл исходных текстов базы знаний.]

⇒ *правило построения**:

[В начале *глобальных нетранслируемых sc-идентификаторов* ставится две точки]

Э *пример'*:

- [..operator1]
- [..node123]

⊃ *уникальный нетранслируемый sc-идентификатор*

= ![...]!

⇒ *пояснение**:

[*уникальные нетранслируемые sc-идентификаторы* используются для однократного обозначения конкретного *неименованного sc-элемента* в рамках исходных текстов *баз знаний* и/или передаваемых между *ostis-системами* сообщений.

Кроме того, *уникальные нетранслируемые sc-идентификаторы* могут использоваться при визуализации баз знаний в виде, например, *sc.s-текстов* или *sc.n-текстов*, в тех случаях, когда необходимо визуализировать *sc-элемент*, не имеющий *sc-идентификатора*.

Каждое вхождение *уникального нетранслируемого sc-идентификатора* в исходный текст *базы знаний* или передаваемое сообщение считается обозначением разных *sc-элементов* (чаще всего, *sc-узлов*).]

sc-идентификатор*

⊃ *основной sc-идентификатор**

:= [Бинарное ориентированное отношение, каждая пара которого связывает *sc-элемент* с внутренним файлом *ostis-системы*, который содержит *основной sc-идентификатор* указанного *sc-элемента*.]

⇒ *примечание**:

[Отношение, связывающее *sc-элементы* с файлами, содержащими их *основные sc-идентификаторы*, само имеет свой *основной sc-идентификатор*, который представляет собой строку символов, имеющую вид “основной *sc-идентификатор**”. Заметим, что в состав первого домена отношения “основной *sc-идентификатор**” входит также и *sc-узел*, обозначающий само это отношение.]

⊃ *системный sc-идентификатор**

:= [Бинарное ориентированное отношение, каждая пара которого связывает *sc-элемент* с внутренним файлом *ostis-системы*, который содержит *системный sc-идентификатор* указанного *sc-элемента*.]

следует отличать*

- Э {
- *системный sc-идентификатор*
 - *основной sc-идентификатор*
- }

⇒ *отличие**:

[Системные идентификаторы отличаются от основных, во-первых, требованием к уникальности в рамках всей базы знаний *Экосистемы OSTIS* (а, значит, независимостью от внешнего языка), а во-вторых, более простым алфавитом, удобным для автоматической обработки.]

- Э {
- *системный sc-идентификатор*
 - *нетранслируемый sc-идентификатор*
- }

⇒ *сходство**:

[*системные sc-идентификаторы* и *нетранслируемые sc-идентификаторы* выполняют схожие функции, связанные с именованием *sc-элементов* на уровне исходных текстов *баз знаний* или передаваемых между *ostis-системами* сообщений.]

⇒ *отличие**:

[Каждый *системный sc-идентификатор* представляется в базе знаний в виде *внутреннего файла ostis-системы* и связан с соответствующим *sc-элементом* парой отношения *системный sc-идентификатор**. *нетранслируемые sc-идентификаторы* не представляются в рамках *базы знаний*, не имеют соответствующих *внутренних файлов ostis-системы* и на уровне *базы знаний* никак не связаны с идентифицируемыми ими *sc-элементами*.]

} /* Завершили Сегмент “Понятие внешнего идентификатора *sc-элемента*” */

```

/* Сегмент *****/
Понятие простого идентификатора sc-элемента
⊃=
{
простой sc-идентификатор
:= [идентификатор sc-элемента, в состав которого идентификаторы других sc-элементов не входят и который не содержит транслируемую в SC-код информацию об обозначаемой им сущности]
:= [простой идентификатор sc-элемента]
⊃ простой строковый sc-идентификатор
:= [простой sc-идентификатор, представляющий собой строку (цепочку) символов, которая является именем (названием) той же сущности, что и идентифицируемый sc-элемент]
⇒ примечание*:
   [простые строковые sc-идентификаторы являются наиболее распространенным видом идентификаторов, приписываемых sc-элементам]

простой строковый sc-идентификатор
⊃ системный sc-идентификатор
⊃ простой строковый идентификатор sc-переменной
   ⊃ пример':
     [_var1]
⊃ простой строковый sc-идентификатор неролевого отношения
   := [простой строковый идентификатор sc-узла, являющегося знаком неролевого отношения]
   ⊃ пример':
     [включение множеств*]
⊃ простой строковый sc-идентификатор ролевого отношения
   := [простой строковый идентификатор sc-узла, являющегося знаком ролевого отношения]
   ⊃ пример':
     [слагаемое']
⊃ простой строковый sc-идентификатор класса классов
   := [простой строковый идентификатор sc-узла, являющегося знаком класса классов]
   ⊃ пример':
     [длина^]
⊃ sc-идентификатор внешнего файла ostis-системы
   := [URL-идентификатор]
   ⇒ пояснение*:
     [sc-идентификаторы данного класса предназначены для описания местоположения внешних файлов ostis-систем и представляют собой строку символов, которая строится в соответствии со стандартом URL, а затем берется в двойные кавычки. Кавычки нужны для однозначности определения того, где начинается и заканчивается данный sc-идентификатор, поскольку в общем случае в URL разрешены пробелы. Целесообразность этого обусловлена тем, что sc-идентификаторы данного типа часто используются в файлах исходных текстов баз знаний ostis-систем.

     sc-идентификаторы внешних файлов ostis-систем с точки зрения Технологии OSTIS являются простыми строковыми sc-идентификаторами, хотя и могут содержать специальные символы, например "%" или "/". Это связано с тем, что указанные идентификаторы не несут в себе семантически значимой информации о свойствах самого sc-элемента, обозначаемого таким sc-идентификатором, а только информацию о его расположении в текущем состоянии внешнего мира ostis-системы.]
   ⊃ пример':
     ["file:///home/user/image1.png"]
     ⇒ примечание*:
       [Данный sc-идентификатор описывает абсолютный путь к файлу под названием "image1.png"]
   ⊃ пример':
     ["file://image1.png"]
     ⇒ примечание*:
       [Данный sc-идентификатор описывает относительный путь к файлу под названием "image1.png"]
   ⊃ пример':
     ["https://conf.ostis.net/content/image1.png"]

```

⇒ *примечание**:

[Данный sc-идентификатор описывает путь к файлу под названием "image1.png" расположенному на удаленном сервере.]

имя нарицательное

:= [простой строковый sc-идентификатор, являющийся именем нарицательным]
 ∈ *имя нарицательное*

:= [Множество всевозможных имен нарицательных]
 ∈ *имя собственное*

:= [имя некоторого класса сущностей (а, точнее, имя класса sc-элементов, обозначающих сущности некоторого класса), которое, с одной стороны, является знаком всего указанного класса, а, с другой стороны, соответствует (может быть приписано) любому экземпляру этого класса]

⇒ *примеры**:

- { • [треугольник]
- [отношение^]
- [разбиение*]
- [параметр^]
- [длина^]
- [язык^]
- [sc-текст]
- [sc-константа]
- [sc-переменная]
- [персона]
- [публикация]
- [город]
- }

⇒ *примечание**:

[*имя нарицательное* всегда начинается с маленькой буквы]

имя собственное

:= [простой строковый sc-идентификатор, являющийся именем собственным]
 ∈ *имя нарицательное*

:= [Множество всевозможных имен собственных]
 ∈ *имя собственное*

⊃ *sc-идентификатор внешнего файла ostis-системы*

:= *пояснение**:

[имя, которое либо не является обозначением какого-либо класса сущностей, либо является обозначением (именем) некоторого класса сущностей, но построенным без использования нарицательного имени этого класса, либо является именем некоторого класса сущностей, построенным с использованием нарицательного имени этого класса либо путем преобразования имени нарицательного во множественное число, либо путем дополнительного использования в начале формулируемого имени таких слов или словосочетаний, как "Знак класса", "Класс", "Знак множества", "Множество", "Знак множества всевозможных", "Множество всевозможных"]

⇒ *примечание**:

[*имя собственное* всегда начинается с большой буквы]

⇒ *примеры**:

- { • [Иванов Петр Николаевич]
- [Минск]
- ⇔ *синонимия внешних идентификаторов**:
 [Город/Минск]
- [Точка/A]

- [SC-код]
 - ⇔ синонимия внешних идентификаторов*:
[Знак множества всевозможных sc-текстов]
 - ⇔ синонимия внешних идентификаторов*:
[Множество всевозможных sc-текстов]
 - ⇔ синонимия внешних идентификаторов*:
[Класс sc-текстов]
 - ⇔ синонимия внешних идентификаторов*:
[sc-текст]
 - ∈ имя нарицательное

}

простой строковый sc-идентификатор

- ⇒ правила построения*:
Правила построения простых строковых sc-идентификаторов
- ⇒ пояснение*:

- [Правила построения простых строковых sc-идентификаторов включают в себя:
- Алфавит символов, используемых в простых строковых sc-идентификаторах;
 - Префиксы и суффиксы, используемые в простых строковых sc-идентификаторах;
 - Разделители и ограничители, используемые в простых строковых sc-идентификаторах;
 - Правила построения имен собственных и имен нарицательных, являющихся простыми строковыми sc-идентификаторами;
 - Правила построения простых строковых sc-идентификаторов, определяемые различными классами идентифицируемых sc-элементов.

]

- = { • [Общим правилом построения *простых sc-идентификаторов* является стремление максимально возможным образом использовать сложившуюся терминологию. Но при этом следует подчеркнуть, что необходимость исключения омонимии в *sc-идентификаторах* требует строгого формального уточнения семантической интерпретации каждого используемого термина. Особо подчеркнем то, что в *ostis-системах* процесс построения новых терминов (*sc-идентификаторов*) и процесс совершенствования существующей терминологии по отношению к процессу эволюции *баз знаний*, представленных в *SC-коде*, с технической точки зрения абсолютно не зависят друг от друга. Кроме того, следует помнить, что далеко не все sc-элементы, входящие в состав базы знаний *ostis-системы*, должны иметь соответствующие им sc-идентификаторы (быть идентифицированными). Существенно подчеркнуть то, что далеко не все sc-элементы должны иметь *простые sc-идентификаторы* по той простой причине, что для многих sc-элементов в случае необходимости можно достаточно легко построить идентифицирующее их *sc-выражение* (sc-выражение). Тем не менее, для целого ряда сущностей (например, для понятий, исторических событий и т.п.) обойтись без простых sc-идентификаторов очень сложно. Очевидно, что идентифицированными (именованными) должны быть все используемые понятия, вводимые в соответствующих предметных областях и специфицируемые соответствующими онтологиями. Идентифицированными также должны быть обладающие особыми свойствами ключевые экземпляры (элементы) некоторых понятий, различные социально значимые объекты (персоны, населенные пункты, географические объекты, страны, организации, библиографические источники, исторические события и многое другое).]

- {
 - простой sc-идентификатор**
 - ⇒ алфавит*:
Алфавит символов, используемых в простых строковых sc-идентификаторах
 - ⊃ Объединение заглавных и строчных букв алфавитов всевозможных естественных языков
 - ⊃ Алфавит символов, используемых в URL
 - ⊃ цифра
 - Э запятая
 - Э точка
 - Э прямые кавычки
 - = !["]!

	:=	[ограничитель метафорических словосочетаний]
	⇒	<i>примечание*</i> : [указанный символ также может использоваться и во внутренних файлах ostis-систем, и в рамках нетранслируемых комментариев]
Э		<i>знак подчеркивания</i>
	:=	[знак нижнего подчеркивания]
	⇒	<i>примечание*</i> : [используется в начале sc-идентификаторов sc-переменных. При этом □ если в начале sc-идентификатора sc-переменной стоит один знак подчеркивания, то по умолчанию такая sc-переменная считается <i>первичной sc-переменной</i> , если явно не указано иное; □ если в начале sc-идентификатора sc-переменной стоит подряд два знака подчеркивания, то по умолчанию такая sc-переменная считается <i>вторичной sc-переменной</i> ; □ если необходимо использовать <i>sc-переменную третьего уровня</i> или <i>sc-переменную, значения которой имеют различный логический уровень</i> , то в начале sc-идентификатора такой sc-переменной ставится один знак подчеркивания, а ее принадлежность соответствующему классу sc-переменных указывается явно
]
Э		<i>звездочка</i>
	:=	[надстрочная звездочка]
	=	![*]!
	⇒	<i>примечание*</i> : [используется в конце <i>простых sc-идентификаторов</i> , соответствующих <i>неролевым отношениям</i>]
Э		<i>штрих</i>
	=	![']!
	:=	[прямой апостроф]
	⇒	<i>примечание*</i> : [используется в конце <i>простых sc-идентификаторов</i> , соответствующих <i>ролевым отношениям</i>]
Э		<i>циркумфлекс</i>
	=	![^]!
	:=	["крышка"]
	⇒	<i>примечание*</i> : [используется в конце <i>простых sc-идентификаторов</i> , соответствующих <i>классам классов</i>]
Э		<i>дефис</i>
	=	![-]!
Э		<i>косая черта</i>
	=	![/]!
	:=	[слэш]
	⇒	<i>примечание*</i> : [используется при формировании <i>простых sc-идентификаторов</i> для экземпляров некоторого заданного <i>класса</i> путем конкатенации <i>sc-идентификатора</i> указанного <i>класса</i> и некоторого <i>sc-идентификатора</i> (необязательно уникального, часто условного), соответствующего рассматриваемому экземпляру указанного <i>класса</i>]
	⇒	<i>пример применения*</i> : • [Точка/A] • [Множество/Si] • [Город/Минск] • [Озеро/Нарочь]
	}	

- [Первым символом каждого *простого строкового sc-идентификатора*, идентифицирующего *sc-переменную* (переменный *sc-элемент*), является знак подчеркивания. При этом, если после указанного знака подчеркивания указывается *имя нарицательное* некоторого *класса*

sc-элементов, то рассматриваемый *простой строковый sc-идентификатор* становится уже *sc-выражением*, содержащим информацию о том, что *областью возможных значений** идентифицируемой *sc-переменной* является указанный *класс sc-элементов*.]

- [Последним символом простого sc-идентификатора, идентифицирующего sc-узел, обозначающий неролевое отношение, заданное на множестве sc-элементов, является надстрочная *звездочка*.]
- [Последним символом простого sc-идентификатора, идентифицирующего sc-узел, обозначающий заданное на множестве sc-элементов ролевое отношение (т.е. отношение, являющееся подмножеством отношения принадлежности), является *итрих* (прямой апостроф)]
- [Последним символом простого sc-идентификатора, идентифицирующего sc-узел, обозначающий понятие, являющееся классом классов (такowymi в частности, являются различного рода параметры – длина, площадь, объем, масса) является символ *циркумфлекс* ("крышка"). Однако, если отсутствует омонимичный идентификатор без этого суффикса, то указанный символ можно не приписывать.]
- [Простой строковый sc-идентификатор может рассматриваться как последовательное сокращение sc-идентификаторов одного и того же sc-элемента с преобразованием имен собственных в имена нарицательные и наоборот.]

⇒ *пример**:

```
{
множество
:= [SC-узел, являющийся знаком множества всевозможных множеств, элементами
    которых являются sc-элементы]
  ⊃ имя собственное
  ⇒ примечание*:
    [здесь можно убрать слова "SC-узел, являющийся"]
:= [Знак множества всевозможных множеств sc-элементов]
  ⊃ имя собственное
  ⇒ примечание*:
    [здесь можно убрать слово "Знак"]
:= [Множество всевозможных множеств sc-элементов]
  ⊃ имя собственное
  ⇒ примечание*:
    [здесь можно убрать слова "Множество всевозможных"]
:= [множество sc-элементов]
  ⊃ имя нарицательное
:= [sc-множество]
  ⊃ имя нарицательное
:= основной sc-идентификатор*:
  [множество]
  ⇒ примечание*:
    [такое сокращение возможно, т.к. любое множество можно представить
    в виде sc-множества]
}
```

- [При наличии синонимичных *имен собственных* и *имен нарицательных* при выборе *основного sc-идентификатора* преимущество имеют *имена нарицательные*. Так, например, основным идентификатором sc-узла, обозначающего *SC-код*, является термин "*sc-текст*", а термин "*SC-код*", являющийся *именем собственным*, считается *неосновным часто используемым sc-идентификатором*. Подчеркнем при этом, что имя нарицательное – это всегда имя некоторого класса сущностей (в частности, понятия). В конце этого имени может быть указан либо признак класса классов, либо признак неролевого отношения (класса связок), либо признак ролевого отношения (подмножества отношения принадлежности), либо не указано ничего. Последнее означает, что именуется класс, не являющийся ни классом классов, ни классом связок. А это, в свою очередь, означает, что именуемым классом является либо класс первичных (терминальных) сущностей, либо класс структур.]
- [Одно и тоже словосочетание, которому приписываются разные дополнительные признаки, может быть основой для внешних идентификаторов разных sc-элементов]

⇒ *пример**:

```
{
```

*следует отличать**

```

∃ {
  • элемент информационной конструкции
  := пояснение*:
    [Множество всевозможных элементов всевозможных информаци-
    онных конструкций]
  • Элемент информационной конструкции/i
  := пояснение*:
    [Некоторый конкретный элемент некоторой информационной
    конструкции]
  • элемент информационной конструкции*
  := пояснение*:
    [Отношение, связывающее информационные конструкции с мно-
    жествами элементов этих информационных конструкций]
  • Элемент информационной конструкции*/i
  := пояснение*:
    [Некоторая конкретная пара (связка), принадлежащая Отноше-
    нию, связывающему информационные конструкции с множества-
    ми элементов этих информационных конструкций]
  := пояснение*:
    [Некоторая конкретная пара (связка), связывающая некоторую
    информационную конструкцию с множеством ее элементов]
  • элемент информационной конструкции'
  := пояснение*:
    [Ролевое отношение, связывающее информационные конструкции
    с конкретными элементами этих информационных конструкций]
  • Элемент информационной конструкции'/i
  := пояснение*:
    [Пара (связка) отношения принадлежности, связывающая некото-
    рую конкретную информационную конструкцию с ее конкретным
    элементом]
  • _элемент информационной конструкции/i
  := пояснение*:
    [sc-переменная, значениями которой могут быть произвольные
    элементы произвольных информационных конструкций]
  • _элемент информационной конструкции*/i
  := пояснение*:
    [sc-переменная, значениями которой могут пары (связки) отно-
    шения “элемент информационной конструкции*”]
  • _элемент информационной конструкции'/i
  := пояснение*:
    [sc-переменная, значениями которой могут пары принадлежности,
    принадлежащие ролевому отношению “элемент информаци-
    онной конструкции'”]
}
}

```

- [В рамках *SC*-кода целесообразно вводить правила унифицированного построения *простых sc-идентификаторов* и целого ряда других классов идентифицируемых сущностей – *персон, библиографических источников (публикаций), разделов баз знаний ostis-систем, файлов ostis-систем, самих ostis-систем.*]

} /* Завершили Сегмент “Понятие простого идентификатора *sc*-элемента” */

/* Сегмент *****/

Понятие сложного идентификатора *sc*-элемента

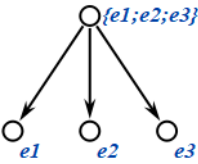
```

⊃=
{

```

sc-выражение

- := [идентификатор, который не только обозначает соответствующую сущность, но также содержит информацию, представляющую собой по возможности однозначную спецификацию указанной сущности]
- ⇒ *примечание**:
[Однозначную спецификацию сущности, которая является понятием, называют определением этого понятия.]
- := [имя соответствующей (именуемой) сущности построенное по принципу "та (тот), которая (который) указываемым образом связана с другими указываемыми сущностями"]
- := [выражение, идентифицирующее sc-элемент]
- := [идентификатор sc-элемента, в состав которого входят другие идентификаторы и денотационная семантика которого точно определяется конкретным набором правил построения таких сложных (комплексных) идентификаторов, состоящих из определенным образом связанных между собой других идентификаторов]
- := [сложный идентификатор, состоящий из других идентификаторов]
- := [идентификатор, который представляет собой конструкцию, состоящую из нескольких других идентификаторов, а также из некоторых разделителей и ограничителей, и денотационная семантика которого однозначно задается конфигурацией указанной конструкции]
- := [сложный sc-идентификатор]
- := [сложный (составной) внешний идентификатор sc-элемента]
- := [выражение, идентифицирующее sc-элемент]
- := [sc-идентификатор, в состав которого входит один или несколько простых sc-идентификаторов]
- ⇒ *разбиение**:
- { • *sc-выражение неориентированного множества*
 - := [sc-выражение в фигурных скобках]
 - := [sc-выражение, ограниченное фигурными скобками]
 - ⇒ *разбиение**:
 - { • *sc-выражение множества, заданного перечислением*
 - := [sc-выражение, обозначающее множество, все элементы которого перечислены своими sc-идентификаторами]
 - ⇒ *пояснение**:
[В качестве разделителя sc-идентификаторов sc-элементов, входящих в указанное множество может выступать точка с запятой (";") или круглый маркер ("•").]
 - ∋ *пример'*:
[$\{e1; e2; e3\}$]
 - ⇔ *семантическая эквивалентность**:
[Дано множество с элементами $e1, e2, e3$.]
 - ⇔ *семантическая эквивалентность**:
[

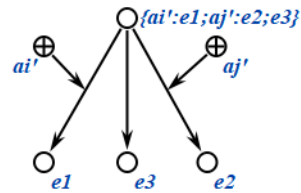


```

graph TD
    A["{e1; e2; e3}"] --> B["e1"]
    A --> C["e2"]
    A --> D["e3"]
          
```
-]
- ∈ *sc.g-текст*
- ⊃ *sc-выражение множества, заданного перечислением элементов с указанием их роли*
- ⇒ *примечание**:
[В рамках такого sc-выражения можно не только перечислять sc-идентификаторы элементов множества, обозначаемого рассматриваемым sc-выражением, но и указывать роль этих элементов в рамках обозначаемого множества.]
- ∋ *пример'*:
[$\{ai': e1; aj': e2; e3\}$]
- ⇔ *семантическая эквивалентность**:
[Дано множество с элементами $e1, e2, e3$. При этом элемент $e1$ в рамках указанного множества имеет атрибут (роль) ai' , а элемент $e2$ – атрибут (роль) aj' .]

⇔ семантическая эквивалентность*:

[



]

∈ *sc.g-текст*

- *sc-выражение структуры*

:= [sc-выражение, обозначающее структуру, представленную на любом известном и легко определяемом языке (Русском, Английском, SCg-коде, SCs-коде, SCn-коде)]

⇒ *пояснение**:

[sc-выражение структуры обозначает структуру, содержащую sc-текст, семантически эквивалентный тому тексту (на некотором известном языке), который заключен в фигурные скобки. Чаще всего такой текст записывается на формальном языке, например, SCs-коде, и может быть автоматически однозначно интерпретирован. Возможна ситуация, когда указанный текст записан на менее неформальном языке, например, Русском, но в этом случае его автоматическая интерпретация значительно усложняется и в общем случае не всегда однозначна.]

⇒ *примечание**:

[В текущей реализации средств разработки исходных текстов баз знаний в соответствии с более старой версией правил построения sc-выражений вместо фигурных скобок *sc-выражение структуры* ограничивается квадратными скобками со звездочками ("[*"и "*""])]

}

- *sc-выражение ориентированного множества*

:= [sc-выражение кортежа]

:= [sc-выражение, ограничиваемое угловыми скобками и обозначающее упорядоченное (ориентированное) множество sc-элементов, порядок которых задаётся последовательностью перечисляемых их sc-идентификаторов]

⇒ *пример'*:

{e1; e2; e3; e4}

⇔ семантическая эквивалентность*:

{1': e1; 2': e2; 3': e3; 4': e4}

⇒ *примечание**:

[Здесь 1' – это ролевое отношение “быть первым компонентом кортежа”, 2' – ролевое отношение “быть вторым компонентом кортежа”.]

- *sc-выражение внутреннего файла ostis-системы*

:= [sc-выражение, ограниченное квадратными скобками]

:= [sc-выражение, обозначающее *внутренний файл ostis-системы*, визуальное представление (изображение) которого ограничивается квадратными скобками]

⊂ *sc-идентификатор внутреннего файла ostis-системы*

⇒ *пояснение**:

[*sc-выражение внутреннего файла ostis-системы* обозначает *внутренний файл ostis-системы*, содержимое которого заключено в квадратные скобки, ограничивающие данное sc-выражение.]

⇒ *примечание**:

[Дополнительная спецификация *внутреннего файла ostis-системы* легко осуществляется с помощью SC-кода. Сюда входит язык, на котором представлена *информационная конструкция*, являющаяся содержимым *файла*, формат кодирования, *автор** и многое другое.]

- *sc-выражение, обозначающее файл-образец ostis-системы*

:= [sc-выражение, ограниченное квадратными скобками с восклицательными знаками]

⇒ *пояснение**:

[Использование *sc*-выражений позволяет существенно сократить число "придумываемых" *sc*-идентификаторов, каковыми в конечном счете становятся только простые *sc*-идентификаторы, поскольку, зная то, как связан идентифицируемый *sc*-элемент с теми *sc*-элементами, которые уже имеют *sc*-идентификаторы, во многих случаях можно построить *sc*-выражение, идентифицирующее указанный *sc*-элемент. Кроме того, каждое *sc*-выражение, являясь внешним идентификатором, является также и транслируемым формальным текстом, содержащим некоторую информацию об обозначаемой ею сущности.]

⇒ *правило интерпретации**:

[Очевидно, что некоторые *sc*-выражения могут интерпретироваться неоднозначно. Например, два одинаковых *sc*-выражения, идентифицирующих *sc*-коннектор, могут изображать как один и тот же *sc*-коннектор, так и кратные *sc*-коннекторы одного и того же вида, связывающие одни и те же два *sc*-элемента. Аналогичная неоднозначность может возникать при использовании *sc*-выражений, построенных на основе бинарного отношения (подмножество*(*S*)) и ряда других *sc*-выражений.

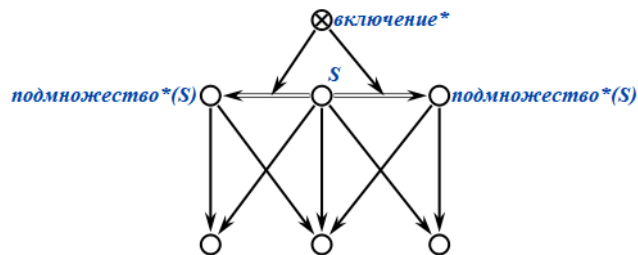
В то же время, некоторые *sc*-выражения являются однозначными, то есть всегда идентифицируют одну и ту же сущность в любом тексте, в состав которого входят. Например выражение "*sin(x)*" является однозначным (при условии, что "*x*" в разных текстах обозначает одно и то же). Однако, однозначность или неоднозначность *sc*-выражений зависит от их семантики, таким образом установить факт однозначности на уровне внешнего текста достаточно сложно.

Для решения проблемы неоднозначности интерпретации *sc*-выражений такого рода будем считать, что каждое вхождение какого-либо *sc*-выражения в различные тексты (например, *sc.s*-тексты) обозначает разные *sc*-элементы. После трансляции таких текстов в *sc*-текст может оказаться, что некоторые из указанных *sc*-выражений на самом деле обозначали одну и ту же сущность, в этом случае соответствующие *sc*-элементы будут "склеены" но уже на уровне *sc*-текста, а не на уровне внешнего текста.

Следует отметить, что факт совпадения *sc*-выражений в рамках некоторого внешнего текста может являться поводом для анализа идентифицируемых этими *sc*-выражениями *sc*-элементов на возможную синонимичность и явно фиксироваться, например, на этапе погружения внешнего текста в *sc*-память.]

⇒ *пояснение**:

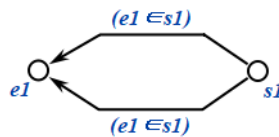
[



]

⇒ *пояснение**:

[



]


ограничитель *sc*-выражений

:= [ограничитель, используемый в *sc*-выражениях]

:= [ограничитель, ограничивающий *sc*-выражения или их компоненты]

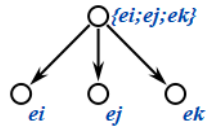
⇒ *разбиение**:

- *левый ограничитель sc-выражений*
 - := [начальный ограничитель *sc*-выражений]
 - := [открывающий ограничитель *sc*-выражений]
- *правый ограничитель sc-выражений*
 - := [конечный ограничитель *sc*-выражений]
 - := [закрывающий ограничитель *sc*-выражений]

- ⇔ пара пересекающихся множеств*:
 фигурная скобка
 = $(![\{ \}]! \cup ![]!)$
 ⇒ пояснение*:
 [Фигурные скобки ограничивают sc-выражение, обозначающее либо множество sc-элементов, задаваемое путем перечисления sc-идентификаторов его элементов (при необходимости с указанием ролей, под которыми эти элементы входят в указанное множество), либо множество sc-элементов, входящих в состав sc-текста, который является результатом трансляции в SC-код текста (например, sc.s-текста или ея-текста), ограниченного фигурными скобками.]
- ⇔ пара пересекающихся множеств*:
 квадратная скобка
 = $(![[]]! \cup ![]]!)$
 ⇒ пояснение*:
 [Квадратные скобки ограничивают sc-выражение, обозначающее файл-экземпляр ostis-системы, содержимое (тело) которого изображается и ограничивается указанными квадратными скобками.]
- ⇔ пара пересекающихся множеств*:
 квадратная скобка со звездочкой
 ∈ устаревшее понятие
 = $(![[*]]! \cup ![] [*]!)$
 ⇒ пояснение*:
 [Данный ограничитель в текущей версии средств разработки исходных текстов баз знаний используется в качестве ограничителя sc-выражений sc-структур вместо фигурных скобок. Такой способ записи sc-выражений sc-структур является устаревшим, однако используется в настоящее время.]
- ⇔ пара пересекающихся множеств*:
 квадратная скобка с восклицательным знаком
 = $(![![]]! \cup ![] ![]!)$
 ⇒ пояснение*:
 [Данный ограничитель sc-выражений ограничивает sc-выражение, обозначающее файл-класс ostis-системы, содержимое (тело) которого изображается и ограничивается указанными sc-ограничителями.]
- ⇔ пара пересекающихся множеств*:
 круглая скобка
 = $(![()]! \cup ![] ()]!)$
 ⇒ пояснение*:
 [В sc-выражениях круглые скобки могут ограничивать:
- перечень (через точку с запятой) sc-идентификаторов, обозначающих аргументы заданного квазибинарного ориентированного отношения в рамках sc-выражений, построенных на основе бинарных отношений и sc-выражений, построенных на основе алгебраических операций;
 - sc-выражение, идентифицирующее sc-коннектор.
-]
- ⇒ примечание*:
 [Очевидно, что многие ограничители sc-выражений могут использоваться также в рамках внутренних файлов ostis-систем, содержащих естественно-языковые тексты или математические выражения.]
- sc-выражение**
- ∋ пример':
 $[(ei \ ki \ ej)]$
 ∈ sc-выражение, идентифицирующее sc-коннектор
 ⇔ семантическая эквивалентность*:
 [
- 
-]
- ∋ пример':
 $[{\bullet}ei \bullet ej \bullet ek]$
 ∈ sc-выражение множества, заданного перечислением
 ⇔ семантическая эквивалентность*:
 $[{ei; ej; ek}]$

⇔ семантическая эквивалентность*:

[



]

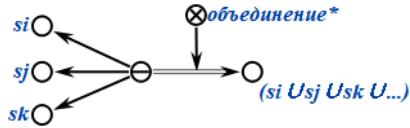
⇒ пример':

[(si ∪ sj ∪ sk ∪ ...)]

∈ sc-выражение, построенное на основе алгебраической операции

⇔ семантическая эквивалентность*:

[



]

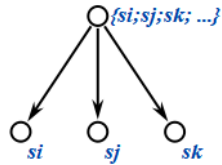
⇒ пример':

[[si; sj; sk; ...]]

∈ sc-выражение множества, заданного перечислением

⇔ семантическая эквивалентность*:

[



]

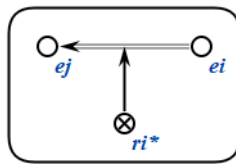
⇒ пример':

[[ei => ri : ej; ;]]

∈ sc-выражение структуры

⇔ семантическая эквивалентность*:

[



]

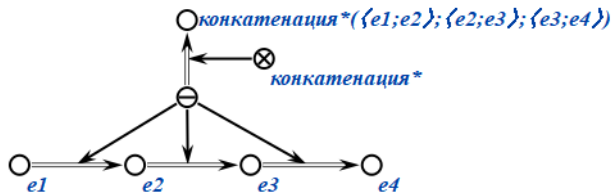
⇒ пример':

[конкатенация*({e1; e2}; {e2; e3}; {e3; e4})]

∈ sc-выражение, построенное на основе квазибинарного отношения

⇔ семантическая эквивалентность*:

[



]

∃

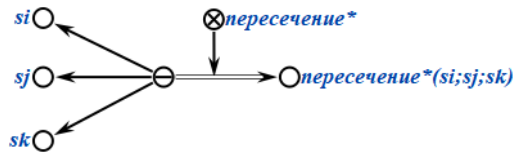
пример':

[пересечение*(si; sj; sk)]

∈ sc-выражение, построенное на основе квазибинарного отношения

⇔ семантическая эквивалентность*:

[



]

∃

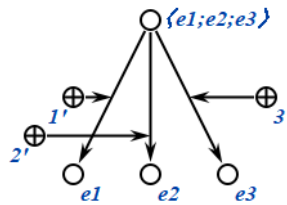
пример':

[(e1; e2; e3)]

∈ sc-выражение в угловых скобках

⇔ семантическая эквивалентность*:

[



]

{
sc-выражение

∃

пример':

[степень*(x, n)]

∃

пример':

[корень*(y, n)]

∃

пример':

[log*(x, y)]

}

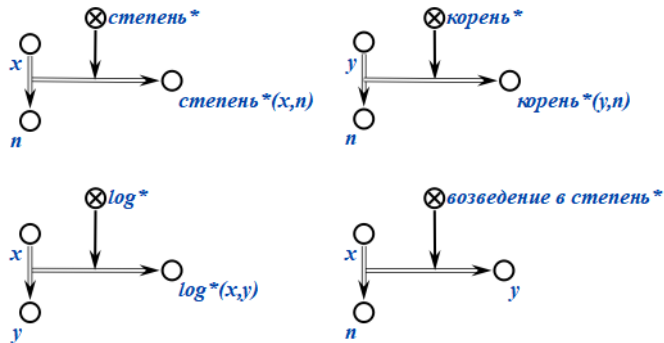
⇒

примечание*:

[В данном примере представлены три разных, но логически эквивалентных друг другу квазибинарных отношения: степень*, корень* и log*. В базе знаний соответствующая конструкция записывается с использованием отношения возведение в степень*.]

⇒ пояснение*:

[



]

{
sc-выражение

∃

пример':

[подмножество*(si) / i]

⇒ пример':
 [надмножество*(sj) /i]
 }
 ⇒ примечание*:
 [Отношения *подмножество** и *надмножество** не являются функциональными, таким образом необходимо явно уточнить при помощи символа “/” и некоторого уточняющего индекса (в данном случае просто “i”), что имеется в виду некоторое конкретное подмножество множества *si* и некоторое конкретное надмножество множества *sj*.]
 ⇒ примечание*:
 [В данном примере представлены два отношения (*подмножество** и *надмножество**), которые по отношению друг к другу являются *обратными отношениями**. В базе знаний соответствующая конструкция записывается при помощи отношения *включение множеств**.]
 ⇒ пояснение*:
 [

]

Правила построения sc-идентификаторов

⇒ примечание*:
 [Правила построения sc-идентификаторов легко записываются на языке Бэкуса-Наура, который является формальным метаязыком, ориентированным на описание синтаксиса всевозможных линейных языков (языков, текстами которых являются строки символов-линейные информационные конструкции). Но, поскольку в рамках Технологии OSTIS любые виды знаний (в том числе и описания синтаксиса различных языков) представляются в виде текстов SC-кода, в рамках SC-кода должен существовать подязык, семантически эквивалентный языку Бэкуса-Наура.]

следует отличать*

⇒ {

- sc-элемент
- sc-идентификатор

 := [внешний идентификатор sc-элемента]
 := [имя, соответствующее (приписываемое) sc-элементу]

⇒ {

- простой sc-идентификатор
- sc-выражение

⇒ {

- sc-выражение структуры
- sc-выражение внутреннего файла ostis-системы

⇒ примечание*:
 [Для каждого sc-выражения структуры, существует sc-выражение внутреннего файла ostis-системы, отличающееся от sc-выражения структуры только заменой фигурных скобок на квадратные.]

⇒ {

- sc-идентификатор
- Правила построения sc-идентификаторов

⇒ {

- [si ⇒ пересечение*: {sj, sk}]
- [si = (sj ∩ sk)]
- [si := [(sj ∩ sk)]]

⇒ пояснение*:
 [Из перечисленных трех текстов первый и второй семантически эквивалентны друг другу и логически следуют из третьего, в котором дополнительно указывается, что у sc-элемента, имеющий основной sc-идентификатор "si", добавляется еще один, но не основной sc-идентификатор "(sj ∩ sk)".]

} /* Завершили Сегмент “Понятие сложного идентификатора sc-элемента” */
 } /* Завершили Раздел “Предметная область и онтология внешних идентификаторов знаков, входящих в информационные конструкции внутреннего языка ostis-систем” */

```

/* Раздел *****/
Описание языка графического представления информационных
конструкций в ostis-системах
:=
{
:= [Описание SCg-кода]
← конкатенация сегментов*:
{
• Основные положения языка графического представления знаний ostis-систем
• Описание Ядра SCg-кода
• Описание Первого расширения Ядра SCg-кода
• Описание Второго расширения Ядра SCg-кода
• Описание Третьего расширения Ядра SCg-кода
• Описание Четвертого расширения Ядра SCg-кода
• Описание Пятого расширения Ядра SCg-кода
• Описание Шестого расширения Ядра SCg-кода
• Описание Седьмого расширения Ядра SCg-кода
}
}

/* Сегмент *****/
Основные положения языка графического представления знаний
ostis-систем
:=
{
SCg-код
:= [Semantic Code graphical]
:= [Язык визуального (графического) представления баз знаний ostis-систем]
∈ графовый язык
⇒ пояснение*:
[SCg-код представляет собой способ визуализации sc-текстов (информационных конструкций SC-кода)
в виде рисунков этих абстрактных конструкций. Подчеркнем, что абстрактная графовая структура
и её рисунок (графическое изображение) – это не одно и то же даже если они изоморфны друг другу.
SCg-код рассматривается нами как объединение Ядра SCg-кода, обеспечивающего изоморфное графическое
изображение любого sc-текста, а также нескольких направлений расширения этого ядра, обеспечивающих
повышение компактности и "читабельности" текстов SCg-кода (sc.g-текстов).]
⇒ примечание*:
[SC-код – это рассмотрение множества всевозможных графически представленных (визуализированных)
графовых структур как универсального языка представления знаний с соответствующим синтаксисом и
семантикой.]

SCg-код
⇒ пояснение*:
[Основная цель SCg-кода – иметь четкие синтаксические графические признаки изображения sc.g-элементов,
позволяющие легко выделить и различать такие классы sc.g-элементов, как:
□ sc.g-константы (знаки константных сущностей) и sc.g-переменные (изображения переменных, значения-
ми которых являются соответствующие sc-элементы);
□ sc.g-переменные, значениями которых являются sc-константы, и sc.g-переменные, значениями которых
являются sc-переменные;
□ знаки постоянных (стабильных) сущностей и знаки временных (нестабильных, временно существующих,
ситуативных) сущностей;
□ sc.g-коннекторы (знаки бинарных связей) и sc.g-элементы, не являющиеся sc.g-коннекторами;
□ неориентированные sc.g-коннекторы (sc.g-ребра) и ориентированные (sc.g-дуги);
□ sc.g-дуги принадлежности и sc.g-дуги, не являющиеся таковыми;
□ sc.g-дуги позитивной принадлежности, негативной принадлежности и нечеткой принадлежности.

В файле “SCg-текст. Алфавит SCg-кода” приведен перечень элементов Алфавита SCg-кода. Этот
перечень оформлен в виде sc.g-текста и представляет собой изображение примеров всех введенных
видов sc.g-элементов (по одному примеру каждого вида). При этом, указанные примеры sc.g-элементов
```

разбиты на пять групп (*SCg-текст. Алфавит SCg-кода*). Первая группа (верхняя строка) включает в себя *sc.g-элементы*, для которых константность и постоянство обозначаемых ими сущностей требует дополнительного уточнения. Остальные четыре группы *sc.g-элементов* аналогичны друг другу и включают в себя соответственно:

- знаки константных постоянных сущностей;
- знаки константных временных сущностей;
- изображения *sc-переменных*, значениями которых или значениями значений которых (в случае, если значениями переменных являются переменные) являются знаки константных постоянных сущностей;
- изображения *sc-переменных*, значениями которых или значениями значений которых (в случае, если значениями переменных являются переменные) являются знаки константных временных сущностей.

Особое место в *SCg-коде* занимает изображение *sc-элементов*, являющихся *обозначениями пар принадлежности**, путём явного использования этого семантически выделяемого класса sc-элементов. Данный *sc.g-элемент* используется тогда, когда нам необходимо изобразить *sc-дугу*, о которой известно, что она является *обозначением пары принадлежности**, но неизвестно о какой принадлежности идет речь – о константной или переменной, о постоянной или временной, о позитивной, негативной или нечеткой.

Кроме *sc.g-элементов*, перечисленных в файле "*SCg-текст. Алфавит SCg-кода*", в состав *Алфавита SCg-кода* входят также следующие *sc.g-элементы*:

- внешние идентификаторы *sc-элементов*, идентичные (приписываемые) соответствующим *sc.g-элементам*.
- *sc.g-контур*, каждый из которых является знаком некоторого *sc-текста* (структуры, состоящей из *sc-элементов*). Каждый такой *sc-текст* может быть:
 - либо константной постоянной структурой;
 - либо константной временной структурой (ситуацией);
 - либо переменной структурой, значениями которой являются постоянные структуры изоморфной конфигурации;
 - либо переменной структурой, значениями которой являются временные структуры (ситуации) изоморфной конфигурации;
- *sc.g-рамки* увеличенного размера являются ограничителями изображения различных файлов, хранимых в памяти *ostis-системы*;
- *sc.g-шины*, являющиеся обозначениями тех же сущностей, что и инцидентные им *sc.g-элементы*.

Заметим также, что, кроме всех перечисленных элементов *Алфавита SCg-кода*, каждый из которых имеет вполне определенную денотационную семантику, для формализации синтаксиса *SCg-кода* необходимо ввести целый ряд более "мелких" синтаксических объектов, например:

- точек инцидентности *sc.g-коннекторов* с *sc.g-узлами*, с другими *sc.g-коннекторами*, с *sc.g-контурами*, с *sc.g-рамками*;
- точек инцидентности *sc.g-шин*;
- точек излома линейных *sc.g-элементов* (*sc.g-коннекторов*, *sc.g-контуров*, *sc.g-рамок*, *sc.g-шин*).

]

*следует отличать**

- ∃ { • { • *SCg-код*
• *Введение в язык графического представления знаний ostis-систем*
} }
• { • *Ядро SCg-кода*
• *Описание Ядра SCg-кода*
} }
• { • *Первое направление расширения Ядра SCg-кода*
• *Описание Первого направления расширения Ядра SCg-кода*
} }
• { • *следует отличать**
• *следует отличать**
• *следует отличать**
} }
⇒ примечание*:

[Следует отличать сами описываемые сущности (в данном случае – различные языки) от текстов, описывающих эти сущности.]

SCg-текст. Алфавит SCg-кода

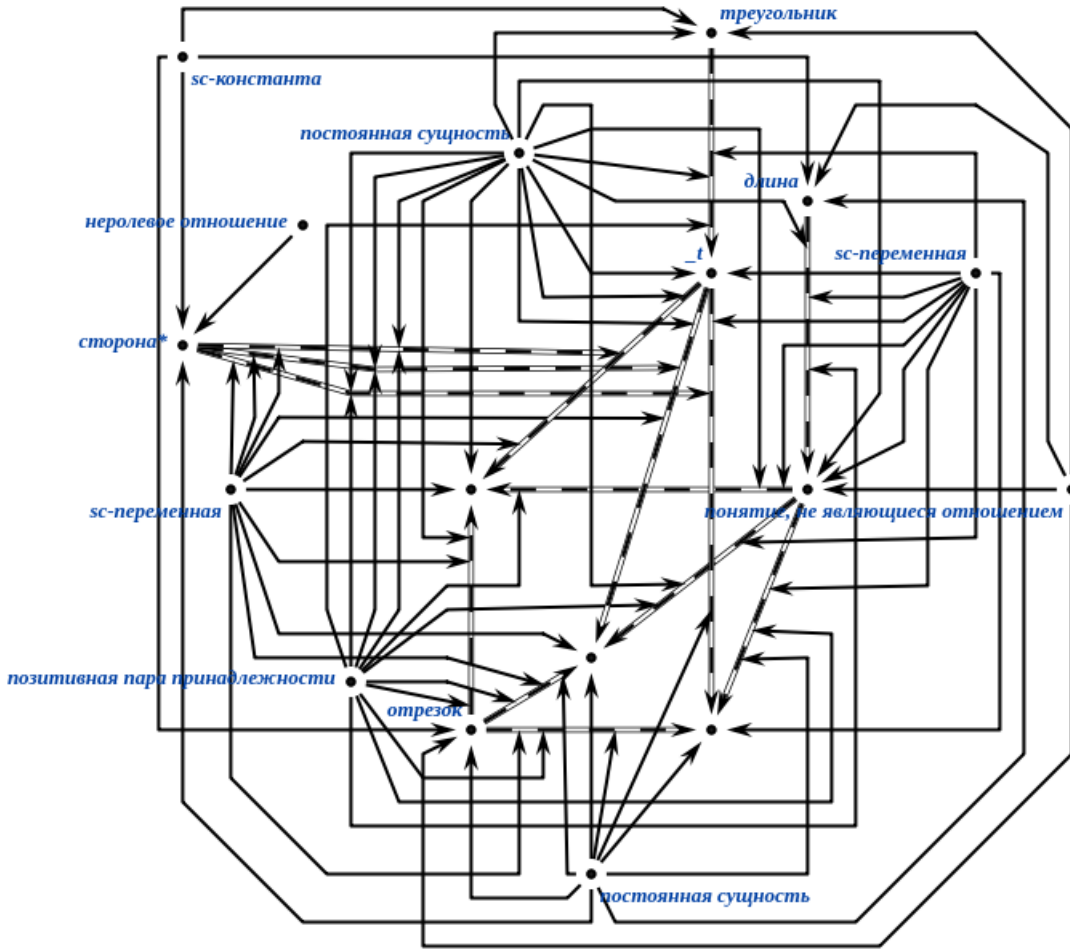
=
[

⊗ обозначение пары принадлежности*

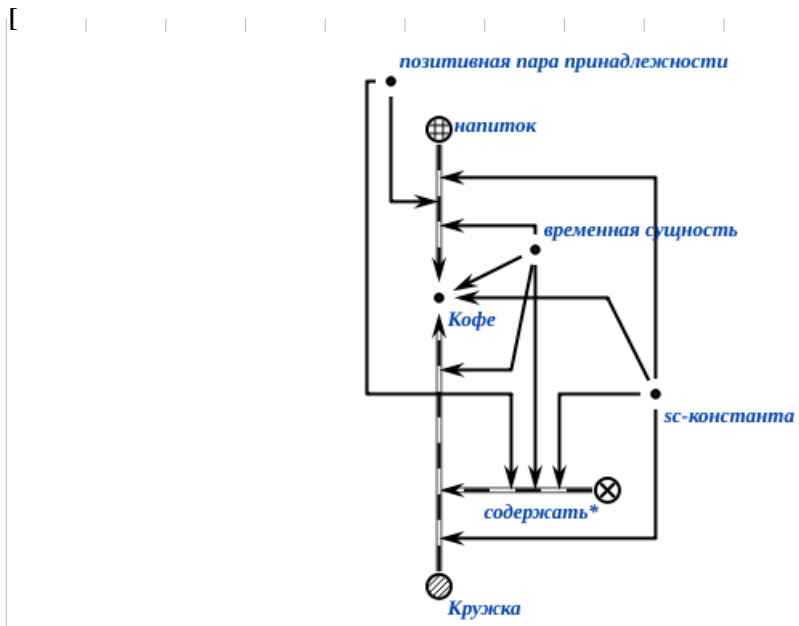
]

Примеры sc.g-текстов, трансформируемых по различным направлениям расширений SCg-кода

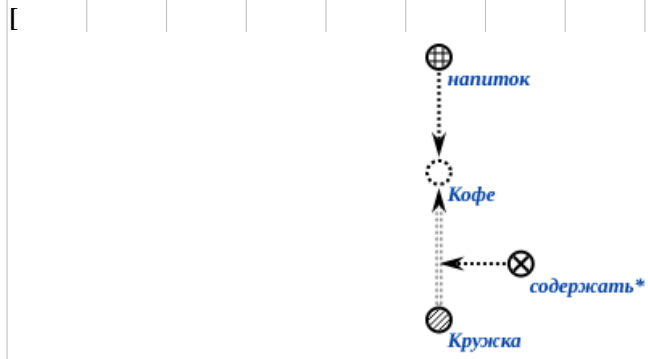
$\supset =$
 $\{ \bullet$ [

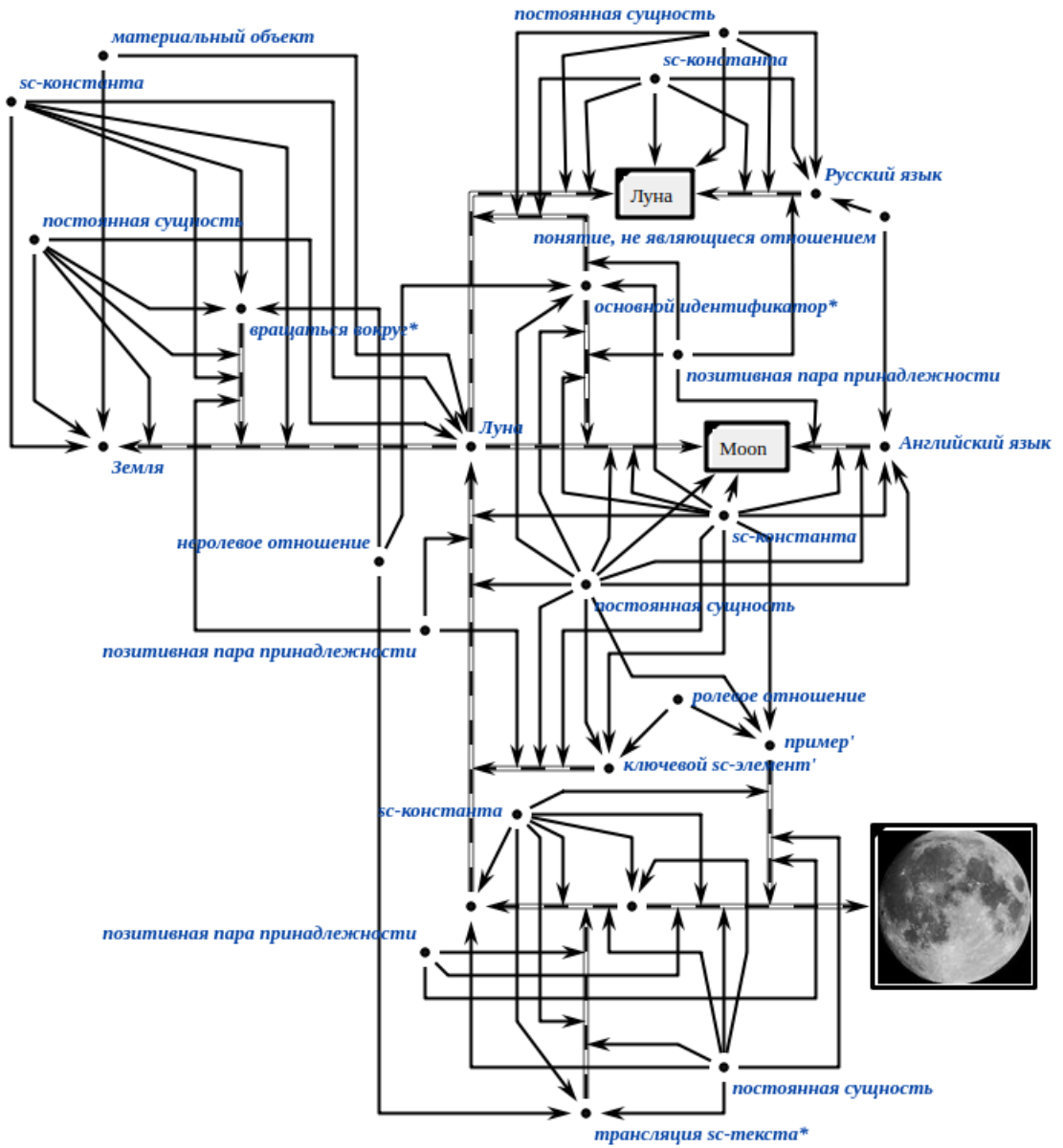


] \Rightarrow синтаксическая трансформация*:

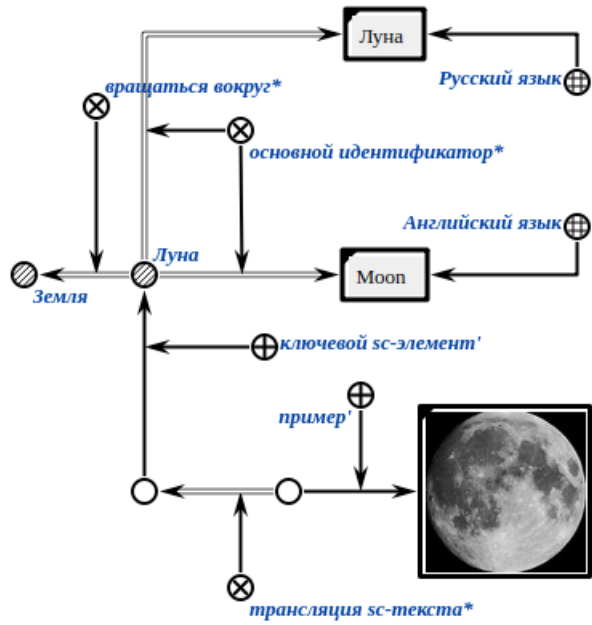


⇒ синтаксическая трансформация*:

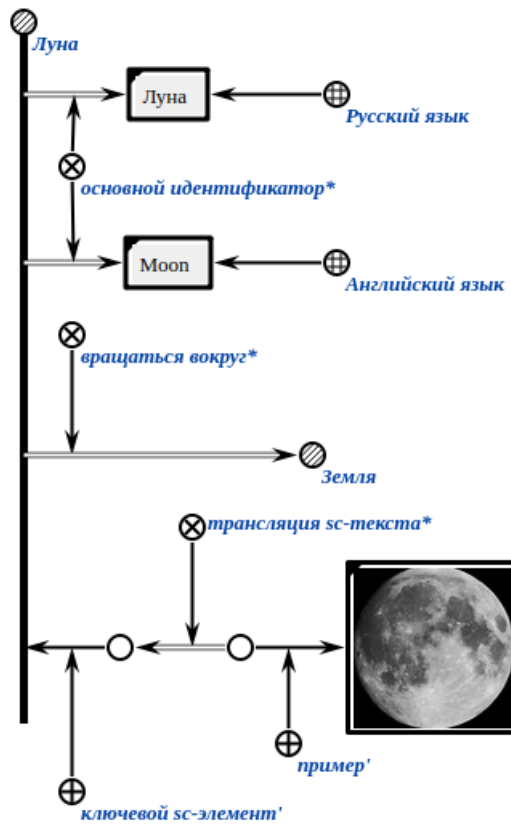




] => синтаксическая трансформация*:



⇒ синтаксическая трансформация*:



/* Завершили Сегмент “Основные положения языка графического представления знаний ostis-систем” */

/* Сегмент *****/

Описание Ядра SCg-кода

:=
{

Алфавит Ядра SCg-кода

:= [Алфавит sc.g-элементов, графически изображающих sc-элементы]

← *ключевой знак**:






Таблица. Алфавит Ядра SCg-кода

⇒ *пояснение**:

[Алфавит Ядра SCg-кода взаимно однозначно соответствует Алфавиту SC-кода. Указанное соответствие представлено в файле “Таблица. Алфавит Ядра SCg-кода”.]

Таблица. Алфавит Ядра SCg-кода

=
[

Алфавит SC-кода	Алфавит Ядра SCg-кода	Изображение sc.g-элемента
sc-узел общего вида	sc.g-узел общего вида	
sc-ребро общего вида	sc.g-ребро общего вида	
sc-дуга общего вида	sc.g-дуга общего вида	
базовая sc-дуга	базовая sc.g-дуга	
внутренний файл ostis-системы	sc.g-узел с содержимым	

]

sc.g-узел общего вида

:= [sc.g-элемент, являющийся графическим изображением sc-узла общего вида]

⇒ *пояснение**:

[Все sc-узлы, не являющиеся знаками файлов, в тексте (конструкции) Ядра SCg-кода, изображаются в виде небольших чёрных кругов одинакового диаметра, который обозначим через d , и точная величина которого зависит от масштаба отображения sc.g-текста.]

sc.g-ребро общего вида

:= [sc.g-элемент, являющийся графическим изображением sc-ребра общего вида]

⇒ *пояснение**:

[Каждое sc-ребро в Ядре SCg-кода изображается в виде широкой линии, в которой чередуются фрагменты со сплошной заливкой и без заливки, не имеющей самопересечений и имеющей общую толщину, равную примерно $0.7d$.]

sc.g-дуга общего вида

:= [sc.g-элемент, являющийся графическим изображением sc-дуги общего вида]

⇒ *пояснение**:

[Каждая sc-дуга в Ядре SCg-кода изображается в виде широкой линии, в которой чередуются фрагменты со сплошной заливкой и без заливки, не имеющей самопересечений, имеющей общую толщину, равную примерно $0.7d$ и имеющей изображение стрелочки на одном из концов этой линии.]

базовая sc.g-дуга

:= [sc.g-элемент, являющийся графическим изображением базовой sc-дуги]

⇒ *пояснение**:

[Каждая входящая в состав sc-текста базовая sc-дуга в Ядре SCg-кода изображается в виде линии произвольной формы, не имеющей самопересечений, имеющей толщину $0.4d$, и имеющей изображение стрелочки на одном из ее концов.]

внутренний файл ostis-системы

:= [sc-узел с содержимым]

∈ *часто используемый sc-идентификатор*

```

:= [sc-узел, являющийся знаком внутреннего файла ostis-системы]
:= [sc-знак внутреннего файла ostis-системы]
sc.g-узел с содержимым
:= [sc.g-узел, имеющий содержимое]
:= [sc.g-узел, являющийся знаком внутреннего файла ostis-системы]
:= [sc.g-знак внутреннего файла ostis-системы]
:= [sc.g-рамка, ограничивающая изображение (представление) внутреннего файла ostis-системы, обозначаемого этой sc.g-рамкой]
:= [sc.g-рамка]
⇒ примечание*:
    [sc.g-рамка – это всегда прямоугольник, максимальный размер которого не ограничивается, но минимальный фиксируется и соответствует sc.g-рамке, внутри которой обозначаемый ею файл не отображается.]
⇒ пояснение*:
    [Каждый входящий в sc-текст sc-узел, имеющий содержимое, в Ядре SCg-кода изображается в виде прямоугольника произвольного размера с толщиной линии 0.6d. Внутри этого прямоугольника отображается файл, обозначаемый изображаемым sc-узлом. Если нет необходимости изображать в тексте сам файл, то sc-узел, обозначающий такой файл, в sc.g-тексте изображается в виде прямоугольника со сторонами 2d по вертикали и 4d по горизонтали.]

```

Алфавит Ядра SCg-кода

```

⇒ примечание*:
    [Трудно сразу поверить, что на основе такого простого алфавита можно построить удобный и универсальный графовый язык. В рамках Документации Технологии OSTIS мы постараемся Вас в этом убедить. Кроме того, нас не должна настораживать простота алфавита, поскольку человечество имеет большой опыт кодирования, хранения в памяти и передачи по каналам связи самых различных информационных ресурсов, используя алфавит, состоящий только из двух классов элементов – единиц и нулей.

    Мы ведем речь о принципиально ином (графовом) способе кодирования информации в компьютерных системах, но стараемся при этом свести это кодирование к достаточно простому алфавиту хотя бы для того, чтобы искусственно не усложнять проблему создания нового поколения компьютеров, основанных на указанном способе кодирования информации.

    Расширения Ядра SCg-кода рассмотрим как направления перехода от текстов Ядра SCg-кода к более компактным текстам. Но, поскольку это приводит к усложнению Синтаксиса SCg-кода и, в первую очередь, к расширению Алфавита SCg-кода, делать такие расширения необходимо обоснованно с учетом частоты встречаемости в рамках баз знаний ostis-систем соответствующих фрагментов.]

} /* Завершили Сегмент “Описание Ядра SCg-кода” */

```

```

/* Сегмент *****/
Описание Первого направления расширения Ядра SCg-кода
⊃=
{

```

Первое направление расширения Ядра SCg-кода

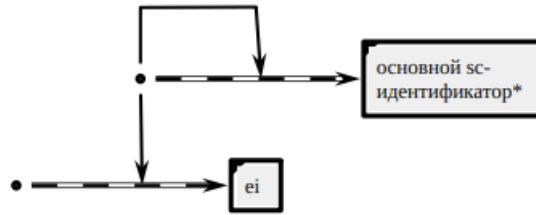
```

⇒ пояснение*:
    [Первое направление синтаксического расширения Ядра SCg-кода – это приписывание некоторым sc.g-элементам основных sc-идентификаторов* (чаще всего - строковых идентификаторов, то есть имен) sc-элементов, изображаемых этими sc.g-элементами. Указываемые идентификаторы являются уникальным для каждого идентифицируемого (именуемого) sc.g-элемента. Приписывание sc.g-элементам уникальных идентификаторов дает возможность в рамках одного sc.g-текста дублировать (копировать) некоторые sc.g-узлы при условии, если всем таким копиям будут приписаны соответствующие идентификаторы. Такое дублирование sc.g-узлов является дополнительным средством наглядного размещения sc.g-текстов. Кроме того, приписывание sc.g-элементу соответствующего ему основного (уникального) sc-идентификатора* представляет собой более компактный вариант изображения sc.g-текстов.]

```

Пример sc.g-текста, трансформируемого по Первому направлению расширения Ядра SCg-кода

=
[



]

∈ sc.g-текст
⇒ пояснение*:

[Здесь (в левом нижнем углу приведенного sc.g-текста) представлен sc.g-узел общего вида, изображающий sc-узел общего вида, которому соответствует основной sc-идентификатор* в виде строки “ei”]
⇒ трансформация sc.g-текста по Первому направлению расширения Ядра SCg-кода*:

[



]

∈ sc.g-текст
⇒ пояснение*:

[sc.g-узлу общего вида изображающему sc-узел, внешним идентификатором которого является строка “основной sc-идентификатор*” и который, соответственно является знаком бинарного ориентированного отношения, каждая пара которого связывает идентифицируемый sc-элемент с его основным внешним sc-идентификатором, приписывается указанный внешний идентификатор изображаемого им sc-элемента.]

⇒ трансформация sc.g-текста по Первому направлению расширения Ядра SCg-кода*:

[

• ei

]

∈ sc.g-текст
⇒ пояснение*:

[В результате данной трансформации исходный sc.g-текст трансформируется в один sc.g-общего вида, которому приписывается основной sc-идентификатор “ei”.]

трансформация sc.g-текста по Первому направлению расширения Ядра SCg-кода*

:= [Бинарное ориентированное отношение, каждая пара которого связывает исходный вид трансформируемого sc.g-текста и результат этой трансформации]

⇒ примечание*:

[Подчеркнем, что рассматриваемая трансформация преобразует исходный текст Ядра SCg-кода в текст, семантически эквивалентный, но принадлежащий не Ядру SCg-кода, а его расширению.]

синтаксическая трансформация*

:= [синтаксическая трансформация информационной конструкции*]

⊃ синтаксическая трансформация sc.g-текста*

⊃ трансформация sc.g-текста по Первому направлению расширения Ядра SCg-кода*

} /* Завершили Сегмент “Описание первого направления расширения Ядра SCg-кода” */

/* Сегмент *****/

Описание Второго направления расширения Ядра SCg-кода

⊃=
{

Второе направление расширения Ядра SCg-кода

⇒ пояснение*:

[Второе направление расширения Ядра SCg-кода – это уточнение типологии константных постоянных сущностей и расширение Алфавита Ядра SCg-кода, позволяющее типологию константных постоянных сущностей привести в соответствие с синтаксической типологией новых вводимых элементов Алфавита SCg-кода. Рассмотрим подробнее sc.g-элементы, знаки константных постоянных сущностей различного вида. Графическим признаком константных постоянных sc-узлов в конструкциях SCg-кода является их изображение в виде окружностей диаметра $3d$, где d – диаметр sc.g-узла общего вида. Такое изображение является более компактной записью факта принадлежности заданного sc-узла (назовем его vi) классу sc-констант и классу обозначений постоянных сущностей. Запись этого факта в Ядре SCg-кода потребует (1) явного изображения sc-узла, обозначающего класс всевозможных константных sc-элементов (класс sc-констант), (2) явного изображения базовой sc-дуги, соединяющего изображение sc-узла, обозначающего класс sc-констант, с изображением заданного константного sc-узла, (3) явного изображения sc-узла, обозначающего класс всевозможных sc-элементов, обозначающих постоянные сущности, (4) явного изображения базовой sc-дуги, соединяющего изображение sc-узла, обозначающего класс обозначений постоянных сущностей с изображением рассматриваемого sc-узла vi (Смотрите *Файл. Изображение спецификации sc.g-элемента средствами Ядра SCg-кода и Первого расширения Ядра SCg-кода*).

Общепринятая запись данного факта выглядит следующим образом:

“sc-константа $\ni vi$; постоянная сущность $\ni vi$;”

Константные постоянные sc-ребра в конструкциях SCg-кода изображаются в виде двойной линии, каждая из которых имеет толщину примерно $d/7$, а расстояние между ними равно примерно $3d/7$.

Константные постоянные sc-дуги изображаются в виде такой же двойной линии, но со стрелочкой. Все базовые sc-дуги, а также все sc-узлы, имеющие содержимое, по определению являются константными постоянными sc-элементами.

Константные sc.g-узлы, изображаемые окружностями диаметра $3d$ и толщиной границы $d/5$, обозначают константные постоянные сущности, о которых мало что известно, но известно то, что они не являются парами (то есть множествами, мощность* которых равна 2) и, следовательно, не могут быть изображены в виде sc.g-дуг или sc.g-рёбер. Но, если при этом об обозначаемой константной постоянной сущности (vi) известно, что она является классом сущностей, то явное указание принадлежности sc-элемента vi всевозможных классов можно заменить на специальное графическое изображение sc-элемента vi , предполагаемое указанную принадлежность. Это приводит к расширению Алфавита SCg-кода (см. *Примеры sc.g-текстов, трансформируемых по Второму направлению расширения Ядра SCg-код*).

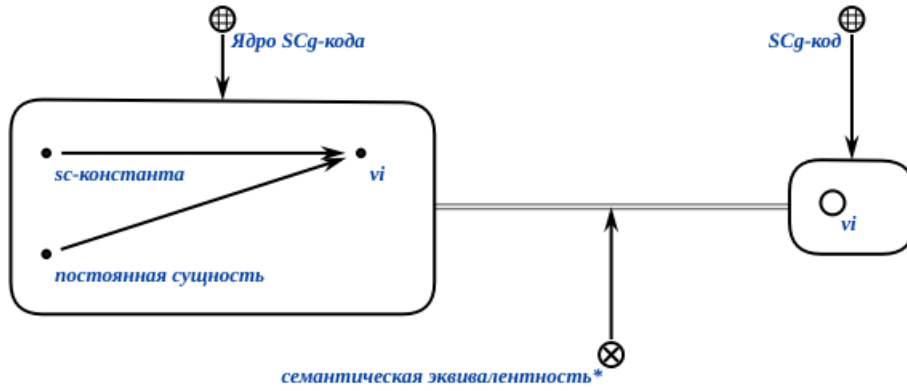
Аналогичным образом (см. *Примеры sc.g-текстов, трансформируемых по Второму направлению расширения Ядра SCg-код*) вводятся:

- sc.g-узел, являющийся изображением класса;
- sc.g-узел, являющийся изображением класса классов;
- sc.g-узел, являющийся изображением отношения;
- sc.g-узел, являющийся изображением ролевого отношения;
- sc.g-узел, являющийся изображением структуры;
- sc.g-узел, являющийся изображением небинарной связи;
- sc.g-узел, являющийся изображением первичной сущности (терминальной сущности, которая не является множеством, а также файлом, хранимым в памяти ostis-системы).

Важное место среди константных постоянных сущностей занимают константные постоянные пары принадлежности, обозначаемое соответствующими sc.g-дугами. Такие пары принадлежности и обозначающие их sc.g-дуги бывают позитивными, негативными и нечеткими. Константная постоянная позитивная sc.g-дуга принадлежности есть ничто иное, как базовая sc.g-дуга. Константная постоянная негативная sc.g-дуга принадлежности изображается в виде базовой sc.g-дуги, перечеркнутой штриховыми черточками. Константная постоянная нечеткая sc.g-дуга принадлежности изображается в виде "недочеркнутой" базовой sc.g-дуги, с каждой стороны которой отображаются штрихи, по длине равные половине от длины штрихов, которыми перечеркнута константная постоянная негативная sc.g-дуга.]

Файл. Изображение спецификации sc.g-элемента средствами Ядра SCg-кода и Второго направления расширения Ядра SCg-кода

=
[



]

Примеры sc.g-текстов, трансформируемых по Второму направлению расширения Ядра SCg-кода

⊃=
{•

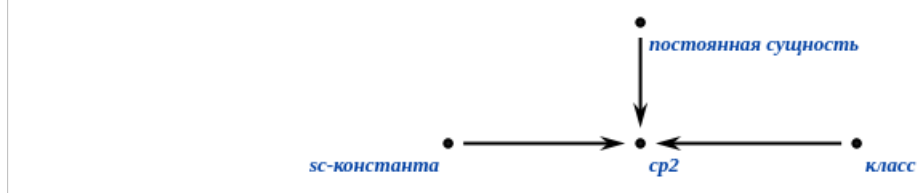


⇒ синтаксическая трансформация*:
[



⇒ пояснение*:
[Здесь вводится новый синтаксический вид sc.g-элементов – константный постоянный sc.g-узел общего вида, изображаемый окружностью диаметра 3d и толщиной границы d/5.]

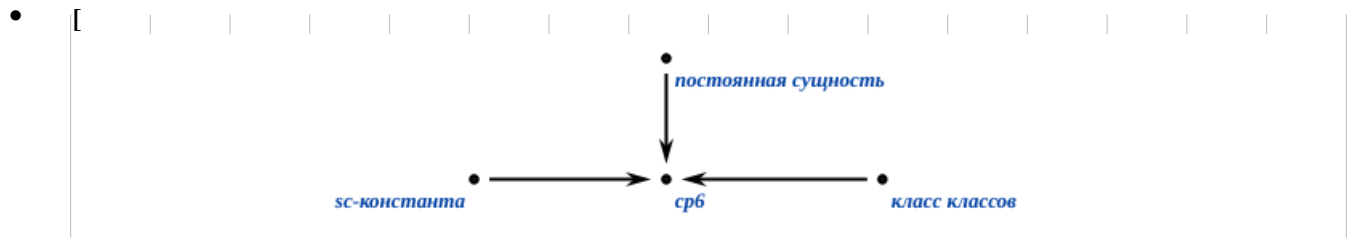
•



⇒ синтаксическая трансформация*:
[



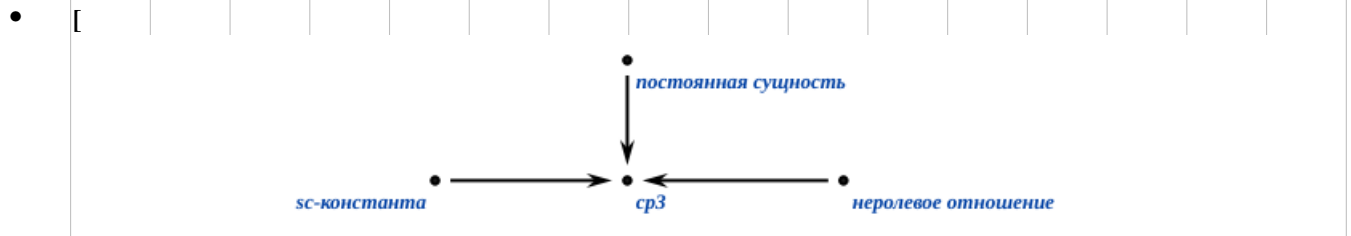
⇒ пояснение*:
[Здесь вводится новый синтаксический вид sc.g-элементов – константный постоянный sc.g-узел, обозначающий класс, изображаемый как константный постоянный sc.g-узел с "решеткой" внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный постоянный sc.g-узел*, обозначающий *класс классов*, изображаемый как *константный постоянный sc.g-узел* с направленным вверх углом внутри.]



⇒ синтаксическая трансформация*:



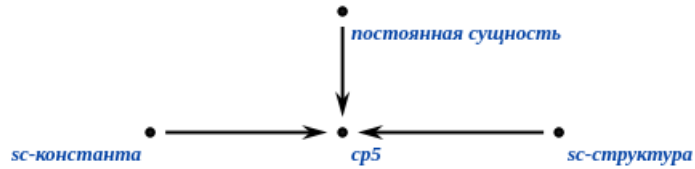
⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный постоянный sc.g-узел*, обозначающий *неролевое отношение*, изображаемый как *константный постоянный sc.g-узел* с "крестиком" внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный постоянный sc.g-узел*, обозначающий *ролевое отношение*, изображаемый как *константный постоянный sc.g-узел* с "плюсом" внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный постоянный sc.g-узел*, обозначающий *структуру*, изображаемый как *константный постоянный sc.g-узел* с точкой внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный постоянный sc.g-узел*, обозначающий *первичную сущность*, изображаемый как *константный постоянный sc.g-узел* с кривой штриховкой внутри.]

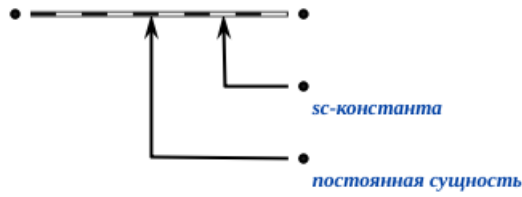


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный постоянный sc.g-узел*, обозначающий *небинарную связку*, изображаемый как *константный постоянный sc.g-узел* с горизонтальной линией внутри.]

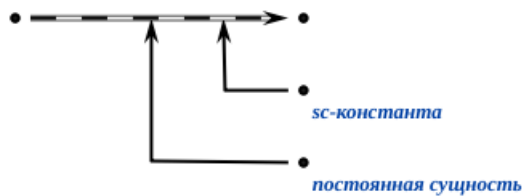


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *константное постоянное sc.g-ребро*, изображаемое двумя непрерывными параллельными линиями.]

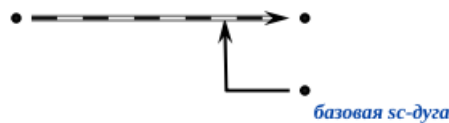


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *константная постоянная sc.g-дуга*, изображаемая двумя непрерывными параллельными линиями с общей стрелкой на одном из концов.]

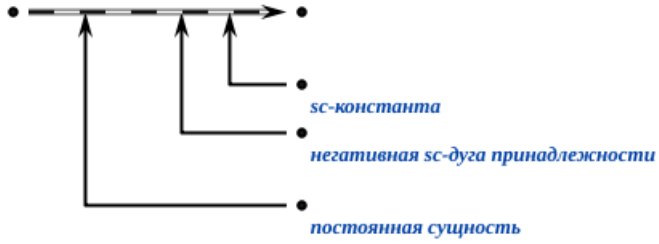


⇒ синтаксическая трансформация*:



⇒ пояснение*:

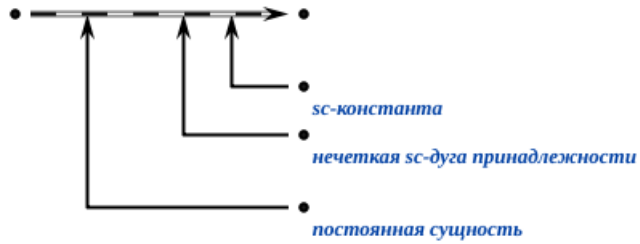
[*Константная постоянная позитивная sc.g-дуга принадлежности* есть ничто иное, как *базовая sc.g-дуга*.]



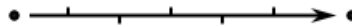
⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константная постоянная негативная sc.g-дуга принадлежности*, изображается в виде *базовой sc.g-дуги*, перечеркнутой штриховыми черточками.]



⇒ синтаксическая трансформация*:

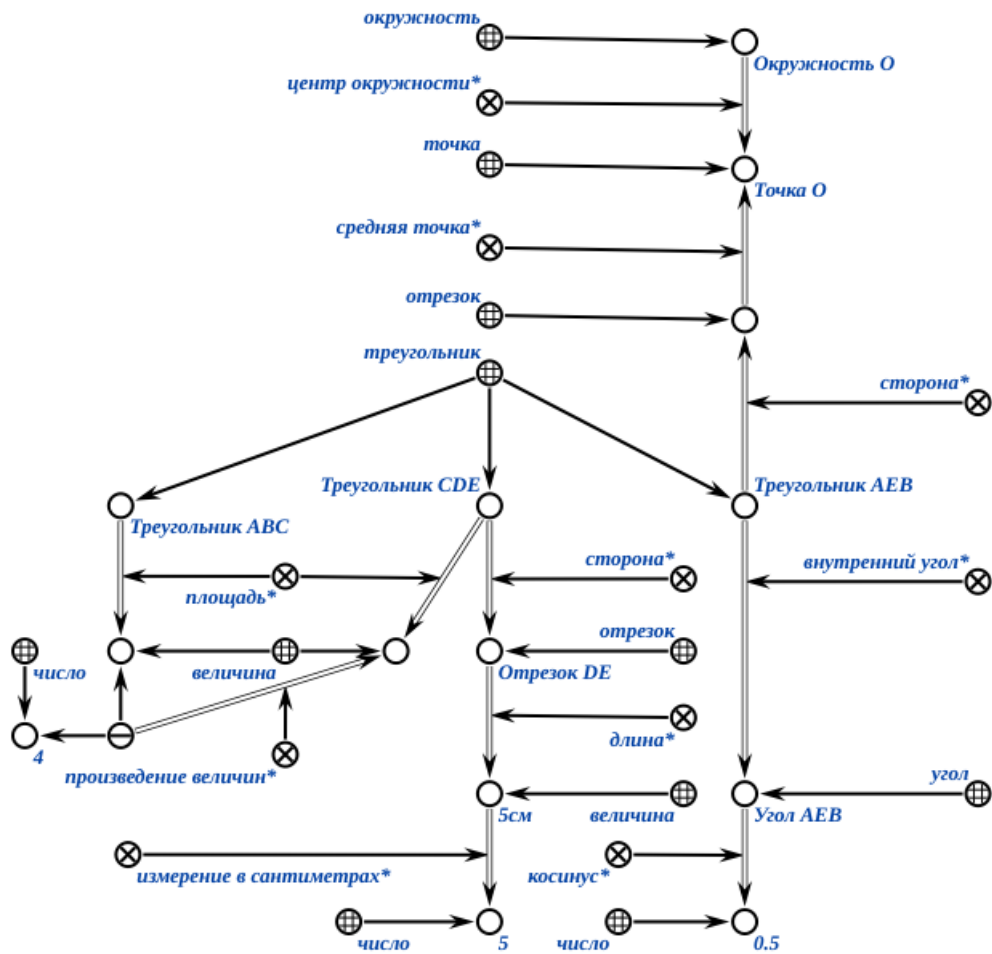


⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константная постоянная нечеткая sc.g-дуга принадлежности*, которая изображается в виде "недочеркнутой" *базовой sc.g-дуги*, с каждой стороны которой отображаются штрихи, по длине равные половине от длины штрихов, которыми перечеркнута *константная постоянная негативная sc.g-дуга*.]

}

Примеры sc.g-текста, записанного средствами Второго направления расширения Ядра SCg-кода

⊃=
{• [



]

∈

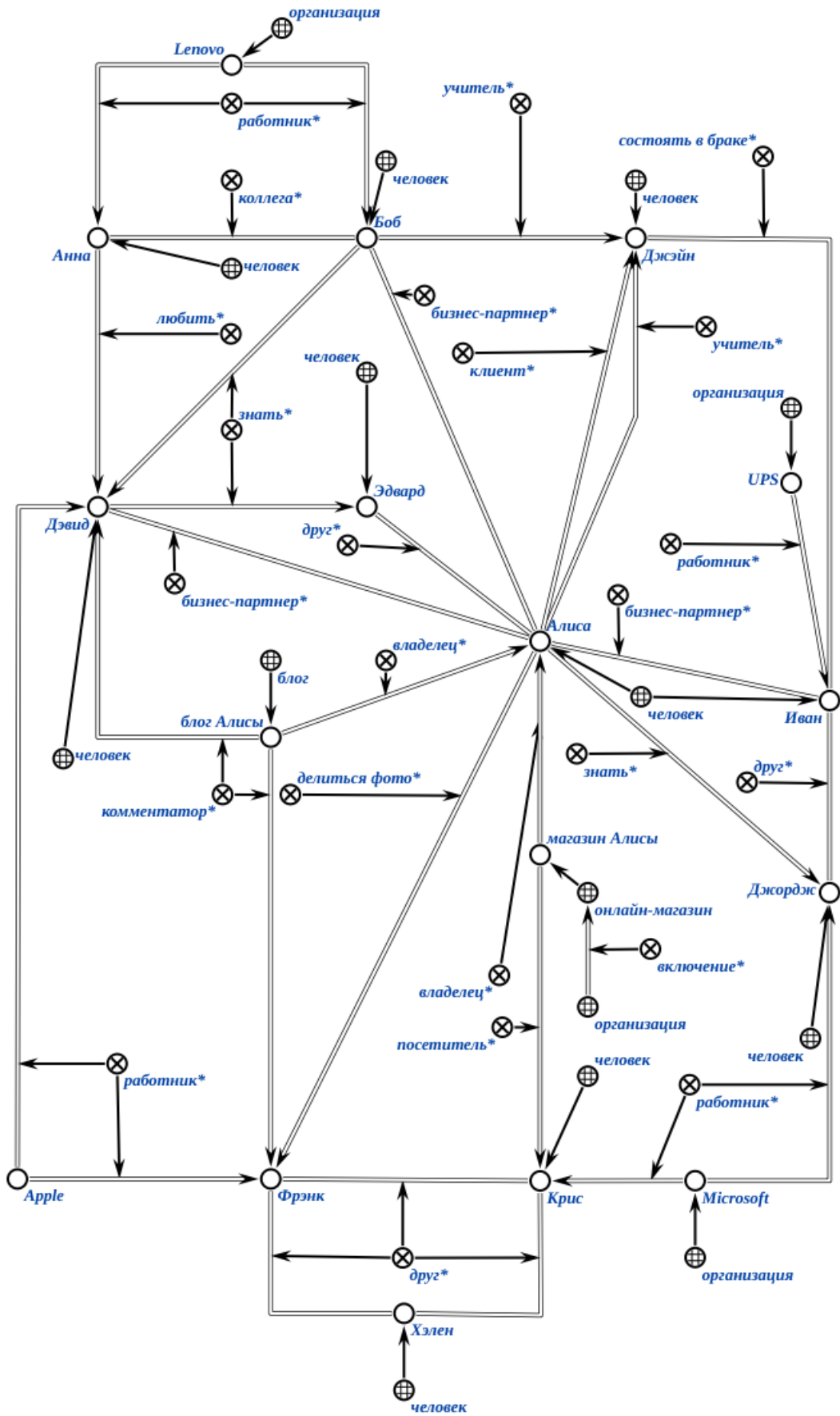
⇒

пример':
sc.g-текст
пояснение*:

[Данный sc.g-текст содержит следующую информацию:

- Сущности *Треугольник ABC* и *Треугольник CDE* являются треугольниками (принадлежат классу *треугольников*). При этом известно, что площадь *Треугольника CDE* в 4 раза больше, чем площадь *Треугольника ABC*, но конкретные значения площадей не известны;
- Сущность *Отрезок DE* является отрезком (принадлежит классу *отрезков*) и является стороной *Треугольника CDE*. Кроме того, у *Отрезка DE* есть длина, измерение которой в сантиметрах составляет 5. Обратите внимание, что в данном случае для упрощения понимания использовано бинарное отношение *длина**, которое является *неосновным понятием* и в базе знаний заменяется на *базовую sc-дугу*, связывающую величину как класс эквивалентности с конкретной сущностью, входящей в данный класс, в данном случае – *Отрезок DE*;
- Сущность *Треугольник AEB* является треугольником и имеет *внутренний угол** *Угол AEB*. В свою очередь, *Угол AEB* является *углом* и имеет *косинус**, равный 0,5;
- *Треугольник AEB* имеет *сторону** (не указывается, какая именно из сторон имеется в виду), *средней точкой** которой является *Точка O*. В свою очередь, *Точка O* является центром некоторой *Окружности O*, которая относится к классу *окружностей*.

]



⇒ *пояснение**:
 [Данный sc.g-текст основан на популярном примере, наглядно иллюстрирующем понятие семантической сети, известном как “Социальная сеть Алисы”. Как видно из примера, данный текст описывает различные взаимосвязи персоны по имени Алиса, при этом некоторые из используемых отношений являются ориентированными (например, “работник*”), а некоторые – неориентированными (например, “друг*”).]

}

/* Завершили Сегмент “Описание Второго направления расширения Ядра SCg-кода” */

/* Сегмент *****/

Описание Третьего направления расширения Ядра SCg-кода

⊃=
 {

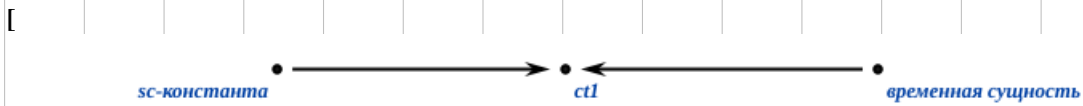
Третье направление расширения Ядра SCg-кода

⇒ *пояснение**:
 [Третье направление расширения Ядра SCg-кода – это расширение его алфавита путем введения дополнительных sc.g-элементов, обозначающих *константные временные сущности* различного вида. Признаком sc.g-элементов, обозначающих *константные временные сущности* являются точечные линии (линии, состоящие из точек, размер которых равен размеру изображаемой линии и которые близко расположены друг к другу на расстоянии, равном половине их размера), с помощью которых рисуются окружности при изображении sc-узлов, а также линии при изображении sc-коннекторов.

Результатом Третьего направления расширения Ядра SCg-кода является введение следующих видов sc.g-элементов (см. *Примеры sc.g-текстов, трансформируемых по Третьему направлению расширения Ядра SCg-кода*).

Примеры sc.g-текстов, трансформируемых по Третьему направлению расширения Ядра SCg-кода

⊃=
 {•

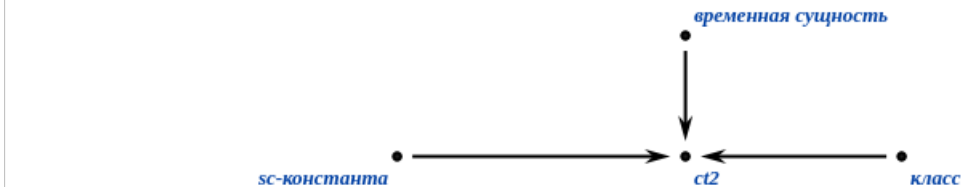


⇒ *синтаксическая трансформация**:
 [



⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид sc.g-элементов – *константный временный sc.g-узел общего вида*, изображаемый точечной окружностью диаметра $3d$ и толщиной границы $d/5$.]

•

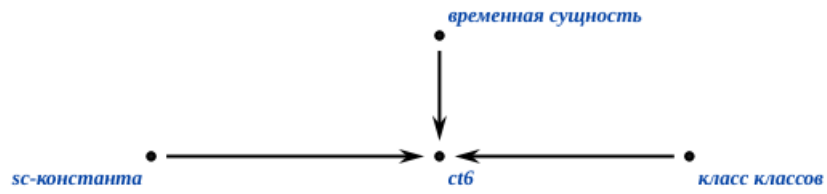


⇒ *синтаксическая трансформация**:
 [



]

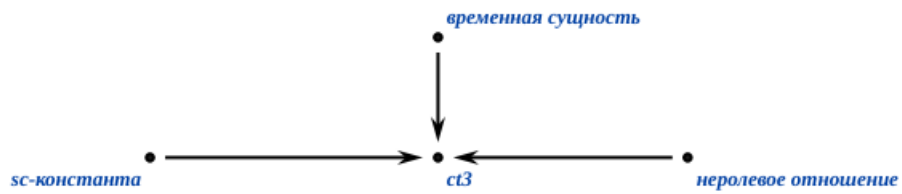
⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный временный sc.g-узел*, обозначающий *класс*, изображаемый как *константный временный sc.g-узел* с "решеткой" внутри.]



⇒ *синтаксическая трансформация**:
 [



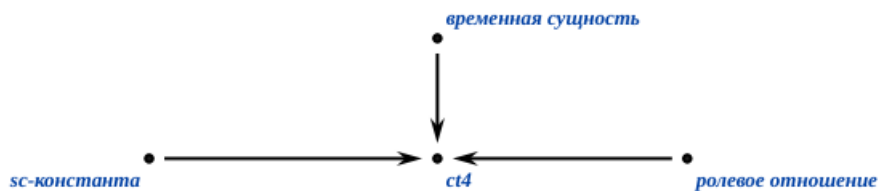
⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный временный sc.g-узел*, обозначающий *класс классов*, изображаемый как *константный временный sc.g-узел* с направленным вверх углом внутри.]



⇒ *синтаксическая трансформация**:
 [



⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный временный sc.g-узел*, обозначающий *неролевое отношение*, изображаемый как *константный временный sc.g-узел* с "крестиком" внутри.]

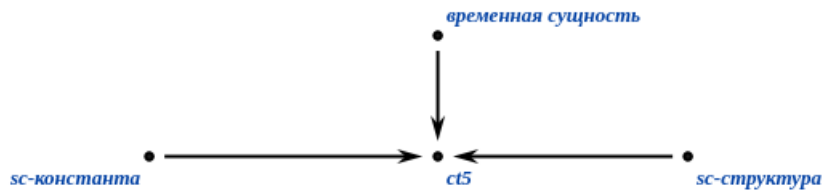


⇒ *синтаксическая трансформация**:
 [



]

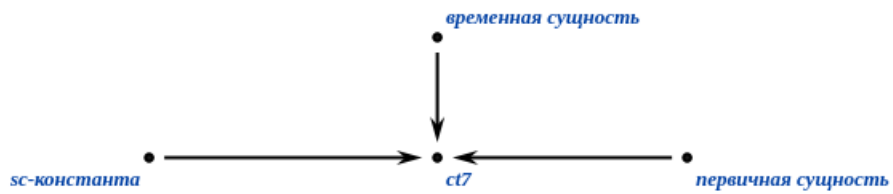
⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный временный sc.g-узел*, обозначающий *ролевое отношение*, изображаемый как *константный временный sc.g-узел* с "плюсом" внутри.]



⇒ *синтаксическая трансформация**:
 [



⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный временный sc.g-узел*, обозначающий *структуру*, изображаемый как *константный временный sc.g-узел* с точкой внутри.]



⇒ *синтаксическая трансформация**:
 [



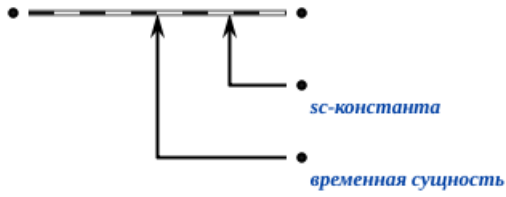
⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный временный sc.g-узел*, обозначающий *первичную сущность*, изображаемый как *константный временный sc.g-узел* с косою штриховкой внутри.]



⇒ *синтаксическая трансформация**:
 [



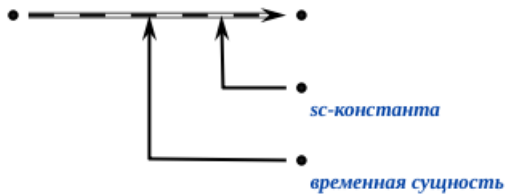
⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константный временный sc.g-узел*, обозначающий *небинарную связку*, изображаемый как *константный временный sc.g-узел* с горизонтальной линией внутри.]



⇒ *синтаксическая трансформация**:



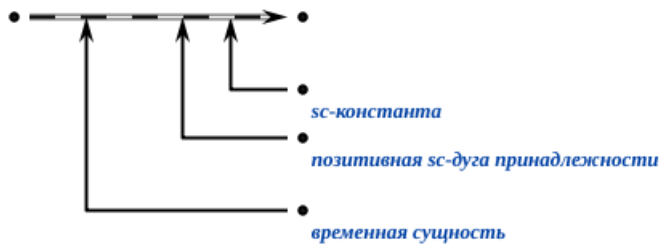
⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константное временное sc.g-ребро*, изображаемое двумя точечными параллельными линиями.]



⇒ *синтаксическая трансформация**:



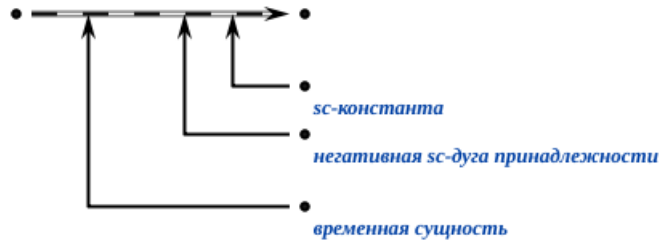
⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константная временная sc.g-дуга*, изображаемая двумя точечными параллельными линиями с общей стрелкой на одном из концов.]



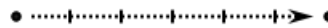
⇒ *синтаксическая трансформация**:



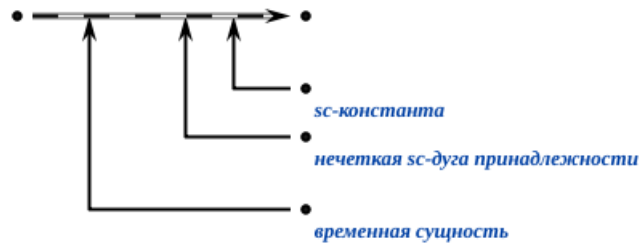
⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константная временная позитивная sc.g-дуга принадлежности*, изображаемая точечной линией со стрелкой на конце.]



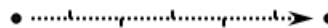
] ⇒ *синтаксическая трансформация**:
 [



] ⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константная временная негативная sc.g-дуга принадлежности*, изображаемая точечной линией, перечеркнутой штриховыми черточками, со стрелкой на конце.]



] ⇒ *синтаксическая трансформация**:
 [



] ⇒ *пояснение**:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *константная временная нечеткая sc.g-дуга принадлежности*, которая изображается в виде "недочеркнутой" константной временной позитивной *sc.g-дуги принадлежности*, с каждой стороны которой отображаются штрихи, по длине равные половине от длины штрихов, которыми перечеркнута константная постоянная негативная *sc.g-дуга*. -1]

}
 } /* Завершили Сегмент "Описание Третьего направления расширения Ядра SCg-кода" */

/* Сегмент *****
 Описание Четвёртого направления расширения Ядра SCg-кода

⊃=
 {

Четвёртое направление расширения Ядра SCg-кода

⇒ *пояснение**:
 [Четвёртое направление расширения Ядра SCg-кода – это расширение его алфавита путем введения дополнительных элементов, обозначающих *переменные постоянные сущности* различного вида. Признаком

sc.g-элементов, обозначающих сущности указанного класса, являются квадратики для изображения обозначений *переменных постоянных сущностей*, не являющихся бинарными связями, а также пунктирные и штрих-пунктирные линии для изображения *переменных постоянных бинарных связей*.

Подчеркнем, что *переменные постоянные сущности* могут отличаться друг от друга по характеру их *области значений**. Этими значениями в общем случае могут быть как *константные постоянные сущности*, так и *переменные постоянные сущности*. В любом случае, значение *переменной сущности* является либо *константной сущностью*, либо *переменной сущностью*. Если каждое значение переменной является константой, то такую переменную будем называть *переменной первого уровня*. Если каждое значение переменной является *переменной первого уровня*, то такую переменную будем называть *переменной второго уровня*.

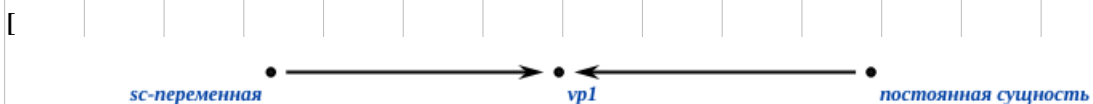
Переменная постоянная сущность первого уровня (первичная sc-переменная), не являющаяся бинарной связью – это переменная, каждым значением которой является *константная постоянная сущность*, не являющаяся бинарной связью. Такая переменная изображается квадратиком, который ориентирован по вертикали и горизонтали.

переменная постоянная сущность второго уровня (вторичная sc-переменная), не являющаяся бинарной связью, изображается квадратиком, повернутым на 45°.

Указанная выше семантика таких изображений приписывается по умолчанию. Это означает, что, если обозначаемая sc-переменная имеет более сложную структуру области её значений (является sc-переменной третьего и выше уровня или sc-переменной, значения которой имеют различный логический уровень), то эта область должна быть специфицирована явно, при этом такая sc-переменная в SCg-коде изображается так же, как первичная sc-переменная.]

Примеры sc.g-текстов, трансформируемых по Четвертому направлению расширения Ядра SCg-кода

⊃=
{•

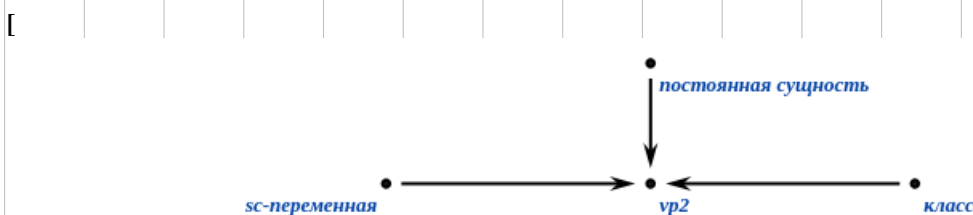


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид sc.g-элементов – *переменный постоянный sc.g-узел общего вида*, изображаемый квадратиком сj со стороны длины 3d и толщиной границы d/5.]

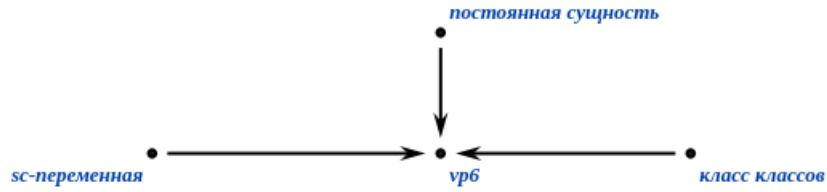


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид sc.g-элементов – *переменный постоянный sc.g-узел, обозначающий класс*, изображаемый как *переменный постоянный sc.g-узел* с "решеткой" внутри.]

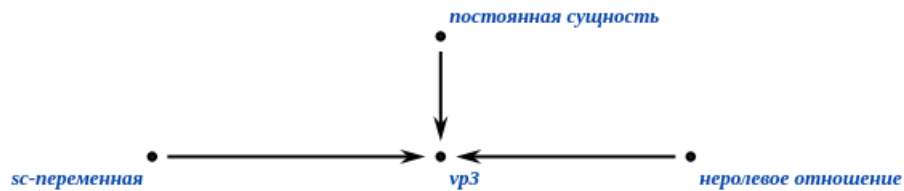


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный постоянный sc.g-узел*, обозначающий *класс классов*, изображаемый как *переменный постоянный sc.g-узел* с направленным вверх углом внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный постоянный sc.g-узел*, обозначающий *неролевое отношение*, изображаемый как *переменный постоянный sc.g-узел* с "крестиком" внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный постоянный sc.g-узел*, обозначающий *ролевое отношение*, изображаемый как *переменный постоянный sc.g-узел* с "плюсом" внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный постоянный sc.g-узел*, обозначающий *структуру*, изображаемый как *переменный постоянный sc.g-узел* с точкой внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный постоянный sc.g-узел*, обозначающий *первичную сущность*, изображаемый как *переменный постоянный sc.g-узел* с крестиком внутри.]

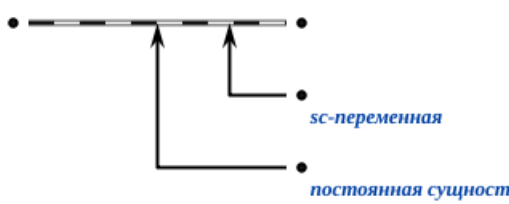



⇒ синтаксическая трансформация*:

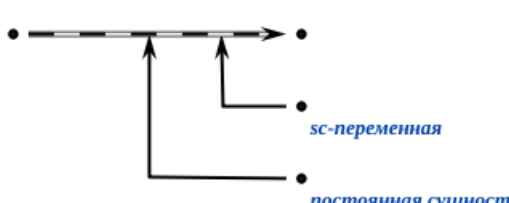



⇒ пояснение*:



[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный постоянный sc.g-узел*, обозначающий *небинарную связку*, изображаемый как *переменный постоянный sc.g-узел* с горизонтальной линией внутри.]

- [
 - 
 -] ⇒ синтаксическая трансформация*:
 - [
 
 -] ⇒ пояснение*:

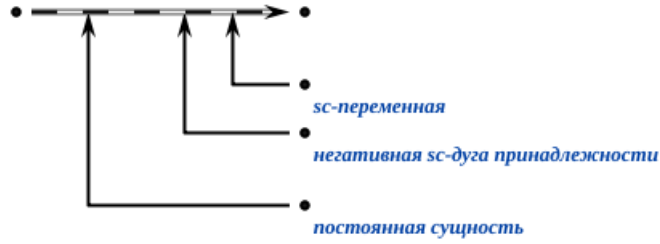
[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменное постоянное sc.g-ребро*, изображаемое двумя пунктирными параллельными линиями.]

- [
 - 
 -] ⇒ синтаксическая трансформация*:
 - [
 
 -] ⇒ пояснение*:

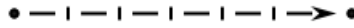
[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменная постоянная sc.g-дуга*, изображаемая двумя пунктирными параллельными линиями с общей стрелкой на одном из концов.]

- [
 - 
 -] ⇒ синтаксическая трансформация*:
 - [
 
 -] ⇒ пояснение*:

[*переменная постоянная позитивная sc.g-дуга принадлежности*, изображаемая пунктирной линией со стрелкой на конце.]

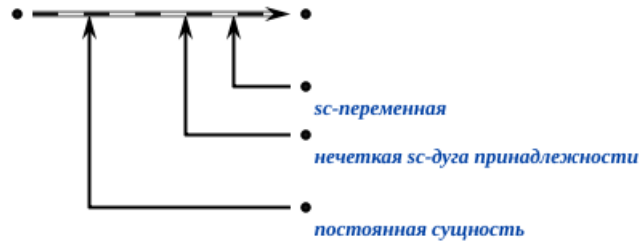


⇒ синтаксическая трансформация*:

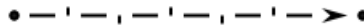


⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменная постоянная негативная sc.g-дуга принадлежности*, изображается в виде *переменной постоянной позитивной sc.g-дуги принадлежности*, перечеркнутой штриховыми черточками.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменная постоянная нечеткая sc.g-дуга принадлежности*, которая изображается в виде "недочеркнутой" *переменной постоянной позитивной sc.g-дуги принадлежности*, с каждой стороны которой отображаются штрихи, по длине равные половине от длины штрихов, которыми перечеркнута *переменная постоянная негативная sc.g-дуга*.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метапеременный постоянный sc.g-узел общего вида*, изображаемый квадратиком, повернутым на 45 градусов.]

- [

] ⇒ синтаксическая трансформация*:
 - [

] ⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременный постоянный sc.g-узел*, обозначающий *класс*, изображаемый как *метаемременный постоянный sc.g-узел* с "решеткой" внутри.]

- [

] ⇒ синтаксическая трансформация*:
 - [

] ⇒ пояснение*:

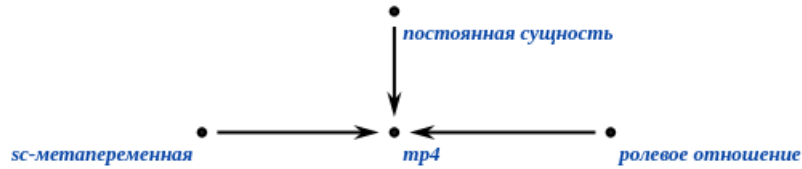
[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременный постоянный sc.g-узел*, обозначающий *класс классов*, изображаемый как *метаемременный постоянный sc.g-узел* с направленным вверх углом внутри.]

- [

] ⇒ синтаксическая трансформация*:
 - [

] ⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременный постоянный sc.g-узел*, обозначающий *нероловое отношение*, изображаемый как *метаемременный постоянный sc.g-узел* с "крестиком" внутри.]

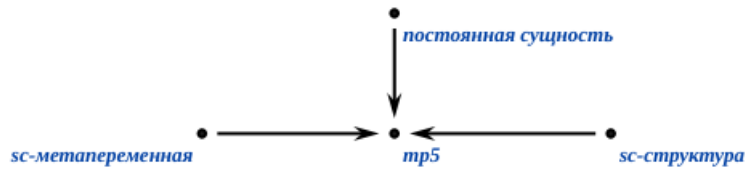


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременный постоянный sc.g-узел*, обозначающий *ролевое отношение*, изображаемый как *метаемременный постоянный sc.g-узел* с "плюсом" внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременный постоянный sc.g-узел*, обозначающий *структуру*, изображаемый как *метаемременный постоянный sc.g-узел* с точкой внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременный постоянный sc.g-узел*, обозначающий *первичную сущность*, изображаемый как *метаемременный постоянный sc.g-узел* с косой штриховкой внутри.]

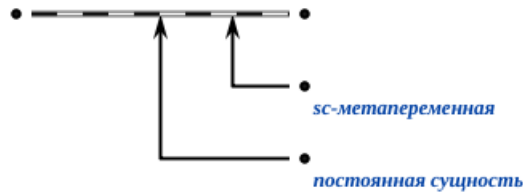


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременный постоянный sc.g-узел*, обозначающий *небинарную связку*, изображаемый как *метаемременный постоянный sc.g-узел* с горизонтальной линией внутри.]

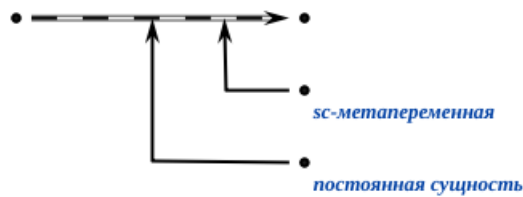


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременное постоянное sc.g-ребро*, изображаемое двумя штрих-пунктирными параллельными линиями.]

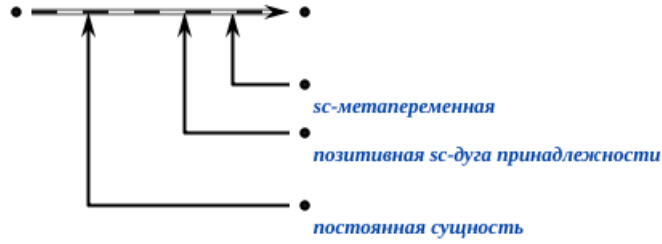


⇒ синтаксическая трансформация*:

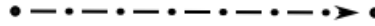


⇒ пояснение*:

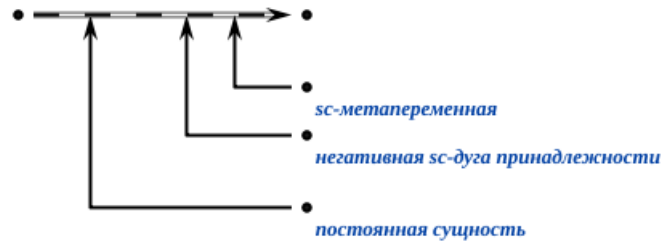
[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременная постоянная sc.g-дуга*, изображаемая двумя штрих-пунктирными непрерывными параллельными линиями с общей стрелкой на одном из концов.]



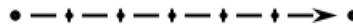
⇒ синтаксическая трансформация*:



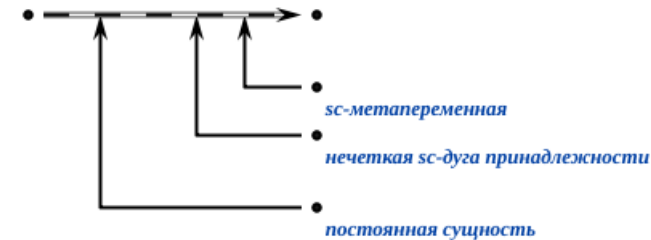
⇒ пояснение*:
 [метапеременная постоянная позитивная sc.g-дуга принадлежности, изображаемая штрих-пунктирной линией со стрелкой на конце.]



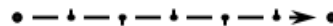
⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид sc.g-элементов – метапеременная постоянная негативная sc.g-дуга принадлежности, изображается в виде метапеременной постоянной позитивной sc.g-дуги принадлежности, перечеркнутой штриховыми черточками.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид sc.g-элементов – метапеременная постоянная нечеткая sc.g-дуга принадлежности, которая изображается в виде "недочеркнутой" мета-

переменной постоянной позитивной *sc.g*-дуги принадлежности, с каждой стороны которой отображаются штрихи, по длине равные половине от длины штрихов, которыми перечеркнута метаваренная постоянная негативная *sc.g*-дуга.]

```

}
} /* Завершили Сегмент "Описание Четвёртого направления расширения Ядра SCg-кода" */

```

/* Сегмент ***
Описание Пятого направления расширения Ядра SCg-кода**

```

▷=
{

```

Пятое направление расширения Ядра SCg-кода

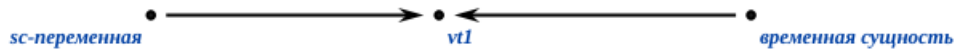
⇒ *пояснение**:
[Пятое направление расширения Ядра SCg-кода – это расширение его алфавита путем введения дополнительных *sc.g*-элементов, обозначающих *переменные временные сущности* различного вида. Указанные дополнительные *sc.g*-элементы аналогичны тем, которые введены в рамках Четвертого направления расширения Ядра SCg-кода, и отличаются только тем, что в Пятом направлении расширения Ядра SCg-кода речь идёт о переменных временных сущностях, а в Четвертом направлении расширения Ядра SCg-кода – о переменных постоянных сущностях.]

Примеры *sc.g*-текстов, трансформируемых по Пятому направлению расширения Ядра SCg-кода

```

▷=
{•

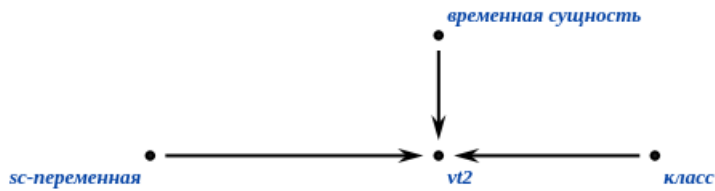
```



] ⇒ *синтаксическая трансформация**:



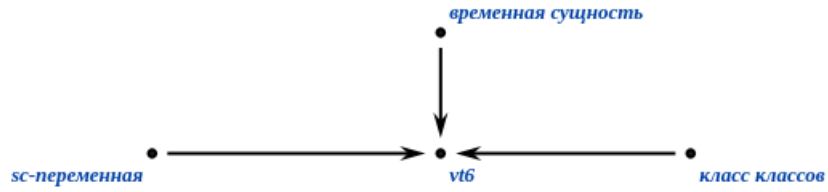
] ⇒ *пояснение**:
[Здесь вводится новый синтаксический вид *sc.g*-элементов – *переменный временный sc.g-узел общего вида*, изображаемый точечным квадратиком диаметра $3d$ и толщиной границы $d/5$.]



] ⇒ *синтаксическая трансформация**:



] ⇒ *пояснение**:
[Здесь вводится новый синтаксический вид *sc.g*-элементов – *переменный временный sc.g-узел, обозначающий класс*, изображаемый как *переменный временный sc.g-узел* с "решеткой" внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный временный sc.g-узел*, обозначающий *класс классов*, изображаемый как *переменный временный sc.g-узел* с направленным вверх углом внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный временный sc.g-узел*, обозначающий *неролевое отношение*, изображаемый как *переменный временный sc.g-узел* с "крестиком" внутри.]

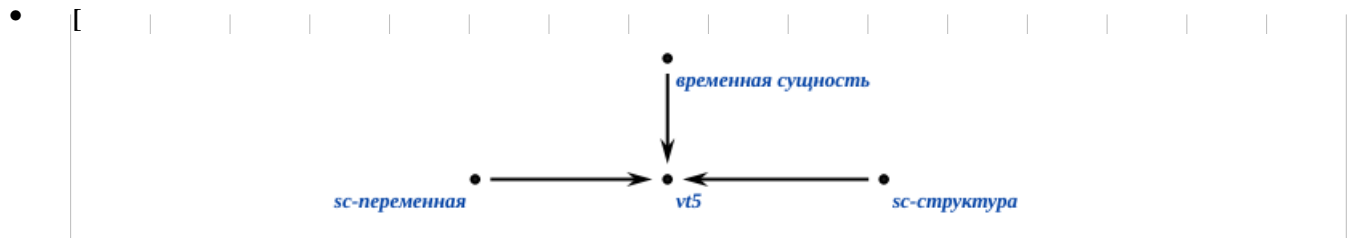


⇒ синтаксическая трансформация*:



⇒ пояснение*:

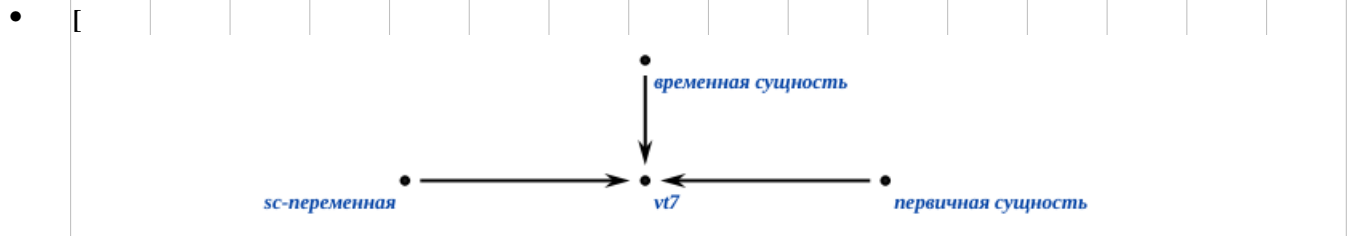
[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный временный sc.g-узел*, обозначающий *ролевое отношение*, изображаемый как *переменный временный sc.g-узел* с "плюсом" внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный временный sc.g-узел*, обозначающий *структуру*, изображаемый как *переменный временный sc.g-узел* с точкой внутри.]



⇒ синтаксическая трансформация*:



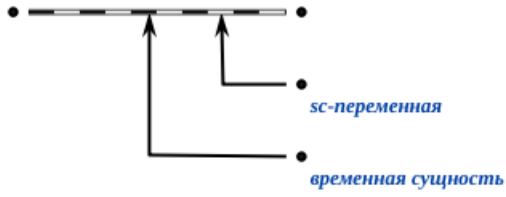
⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный временный sc.g-узел*, обозначающий *первичную сущность*, изображаемый как *переменный временный sc.g-узел* с косой штриховкой внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменный временный sc.g-узел*, обозначающий *небинарную связку*, изображаемый как *переменный временный sc.g-узел* с горизонтальной линией внутри.]

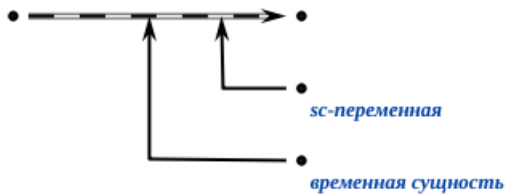


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменное временное sc.g-ребро*, изображаемое двумя пунктирными точечными параллельными линиями.]

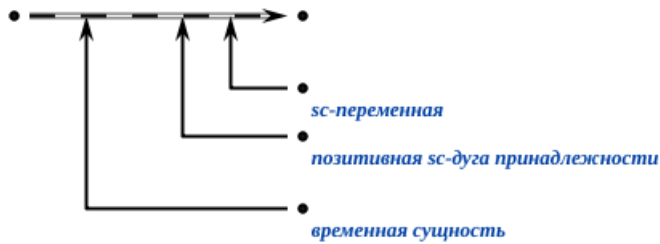


⇒ синтаксическая трансформация*:

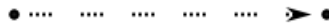


⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменная временная sc.g-дуга*, изображаемая двумя пунктирными точечными параллельными линиями с общей стрелкой на одном из концов.]

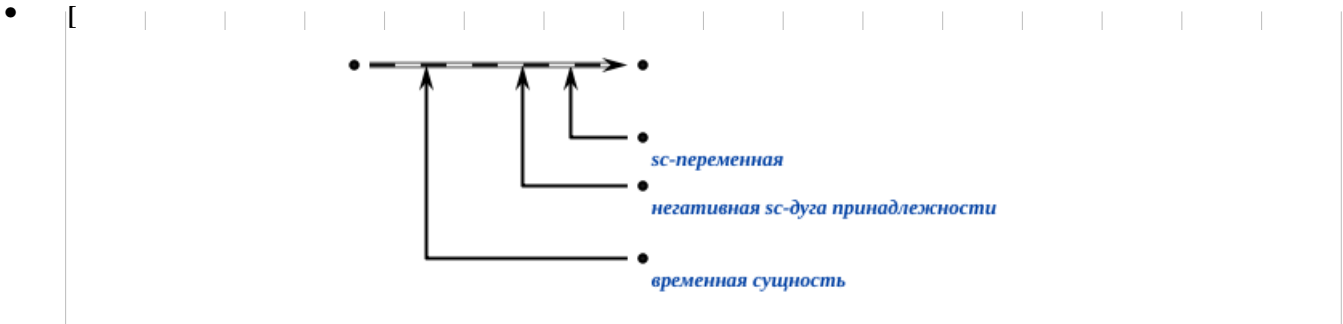


⇒ синтаксическая трансформация*:



⇒ пояснение*:

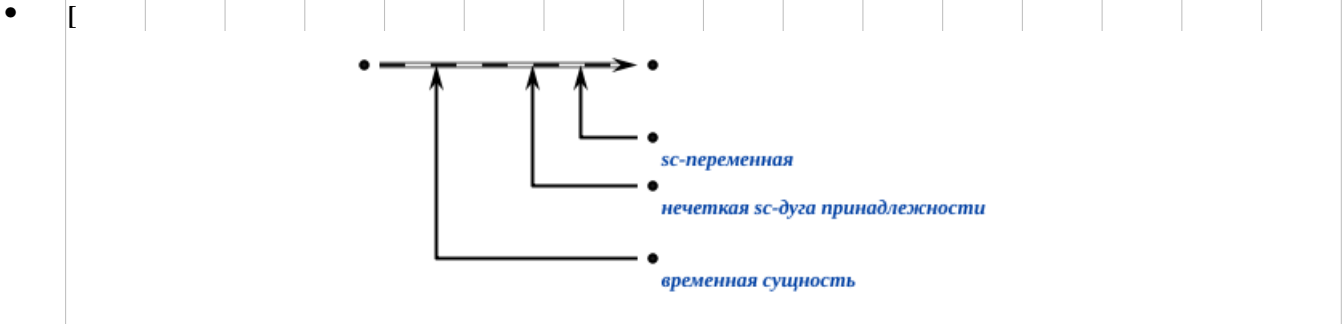
[*Переменная временная позитивная sc.g-дуга принадлежности*, изображаемая в виде пунктирной точечной линией со стрелкой на конце.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменная временная негативная sc.g-дуга принадлежности*, изображается в виде *переменной временной позитивной sc.g-дуги*, перечеркнутой штриховыми черточками.]



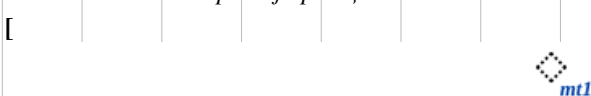
⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *переменная временная нечеткая sc.g-дуга принадлежности*, которая изображается в виде "недочеркнутой" *переменной временной позитивной sc.g-дуги*, с каждой стороны которой отображаются штрихи, по длине равные половине от длины штрихов, которыми перечеркнута *переменная временная негативная sc.g-дуга*.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид *sc.g-элементов* – *метавременный временный sc.g-узел общего вида*, изображаемый точечным квадратиком, повернутым на 45 градусов.]

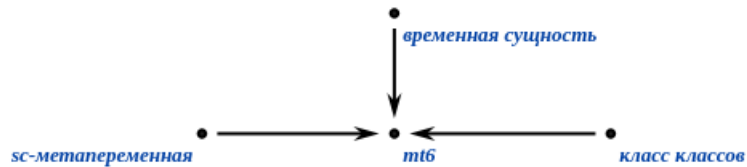


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метапеременный временный sc.g-узел*, обозначающий *класс*, изображаемый как *метапеременный временный sc.g-узел* с "решеткой" внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метапеременный временный sc.g-узел*, обозначающий *класс классов*, изображаемый как *метапеременный временный sc.g-узел* с направленным вверх углом внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метапеременный временный sc.g-узел*, обозначающий *неролевое отношение*, изображаемый как *метапеременный временный sc.g-узел* с "крестиком" внутри.]

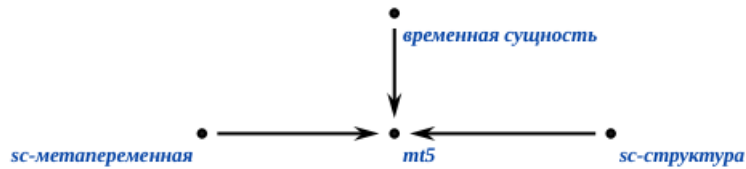


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метапеременный временный sc.g-узел*, обозначающий *ролевое отношение*, изображаемый как *метапеременный временный sc.g-узел* с "плюсом" внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метапеременный временный sc.g-узел*, обозначающий *структуру*, изображаемый как *метапеременный временный sc.g-узел* с точкой внутри.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метапеременный временный sc.g-узел*, обозначающий *первичную сущность*, изображаемый как *метапеременный временный sc.g-узел* с косой штриховкой внутри.]

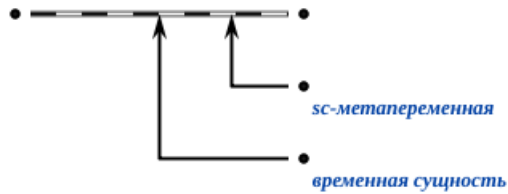


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременный временный sc.g-узел*, обозначающий *небинарную sc-связку*, изображаемый как *метаемременный временный sc.g-узел* с горизонтальной линией внутри.]

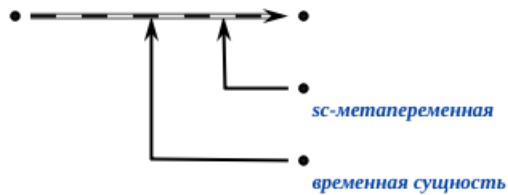


⇒ синтаксическая трансформация*:



⇒ пояснение*:

[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременное временное sc.g-ребро*, изображаемое двумя штрих-пунктирными параллельными линиями.]

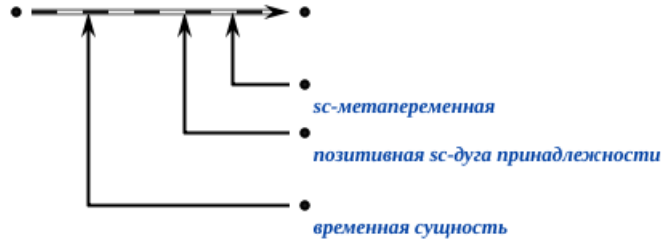


⇒ синтаксическая трансформация*:



⇒ пояснение*:

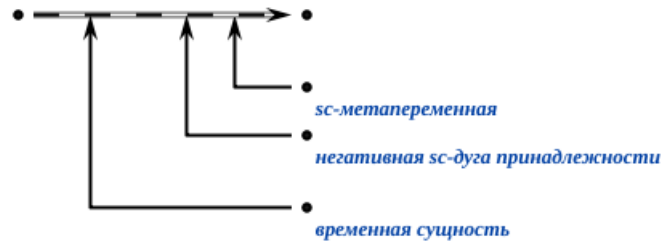
[Здесь вводится новый синтаксический вид *sc.g-элементов* – *метаемременная временная sc.g-дуга*, изображаемая двумя штрих-пунктирными параллельными линиями с общей стрелкой на одном из концов.]



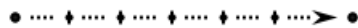
⇒ синтаксическая трансформация*:



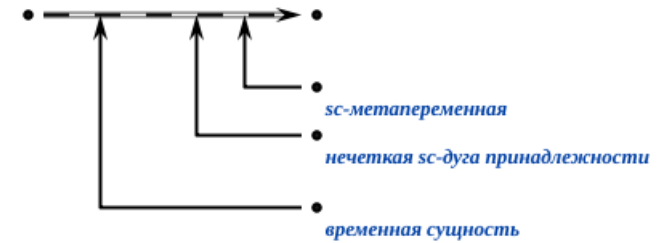
⇒ пояснение*:
 [метапеременная временная позитивная sc.g-дуга принадлежности, изображаемая штрих-пунктирной точечной линией со стрелкой на конце.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид sc.g-элементов – метапеременная временная негативная sc.g-дуга принадлежности, изображается в виде метапеременной временной позитивной sc.g-дуги, перечеркнутой штриховыми черточками.]



⇒ синтаксическая трансформация*:



⇒ пояснение*:
 [Здесь вводится новый синтаксический вид sc.g-элементов – метапеременная временная нечеткая sc.g-дуга принадлежности, которая изображается в виде "недочеркнутой" метапе-

ременной временной позитивной *sc.g-дуги*, с каждой стороны которой отображаются штрихи, по длине равные половине от длины штрихов, которыми перечеркнута *метаемременная временная негативная sc.g-дуга*.]

```
} /* Завершили перечень Примеров sc.g-текстов, трансформируемых по Пятому направлению расширения Ядра SCg-кода */
```

```
} /* Завершили Сегмент “Описание Пятого направления расширения Ядра SCg-кода” */
```

```
/* Сегмент *****/
```

Описание Шестого направления расширения Ядра SCg-кода

```
⊃=
{
```

Шестое направление расширения Ядра SCg-кода

⇒ *пояснение**:

[**Шестое направление расширения Ядра SCg-кода** – это введение в SCg-код *sc.g-контуров* и *sc.g-шин* как средств структуризации *sc.g-текстов* и повышения наглядности при их размещении. Подчеркнем, что и *sc.g-контур*, и *sc.g-шина*, и *sc.g-рамки* являются специальными видами *sc.g-элементов*. При этом *sc.g-контур* и *sc.g-рамки* являются *sc.g-ограничителями* (ограничителями SCg-кода).]

sc.g-контур

⇒ *пояснение**:

[Каждый *sc.g-контур* изображается (в 2D-модификации) в виде замкнутой ломаной линии со скругленными изломами, ограничивающей некоторый фрагмент *sc.g-текста* и обозначает множество всех *sc-элементов*, *sc.g-изображения* которых оказались внутри этого контура. Толщина указанной линии составляет примерно **0.4d**, где **d** - диаметр *sc.g-узла общего вида*.

Обозначение множества *sc-элементов*, изображаемое *sc.g-контуром*, может быть как константным, так и переменным. Соответственно этому линия, изображающая *sc.g-контур* может быть:

- сплошной непунктирной линией,
- точечной непунктирной линией,
- сплошной пунктирной линией,
- точечной пунктирной линией.

Семантическим эквивалентом *sc.g-контур* является *sc.g-узел*, обозначающий структуру. Использование *sc.g-контур* вместо указанного *sc.g-узла* исключает необходимость явно изображать SC-дуги принадлежности, выходящие из этого *sc.g-узла*. Это существенно повышает уровень наглядности *sc.g-текста*.

Если представленный внутри *sc.g-контур* текст не является *sc.g-текстом*, то считается, что что на самом деле внутренностью *sc.g-контур* является *sc.g-текст*, являющийся результатом перевода предоставленного текста в SCg-код.]

sc.g-шина

⇒ *пояснение**:

[Каждая *sc.g-шина* представляет собой замкнутую или незамкнутую линию толщиной примерно равной диаметру *sc.g-узла общего вида*, которая инцидентна только одному *sc.g-элементу* и семантически ему эквивалентна. Идея введения *sc.g-шин* заключается в увеличении «размеров» *sc.g-элементов* для расширения их области инцидентности. Особенно актуально это для *sc.g-узлов*, имеющих большое число инцидентных им *sc.g-коннекторов*.]

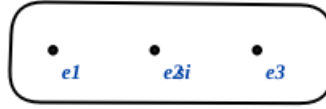
Примеры *sc.g-текстов*, трансформируемых по Шестому направлению расширения Ядра SCg-кода

```
⊃=
{•
```

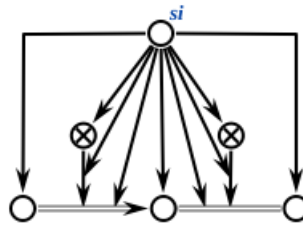


```
]
```

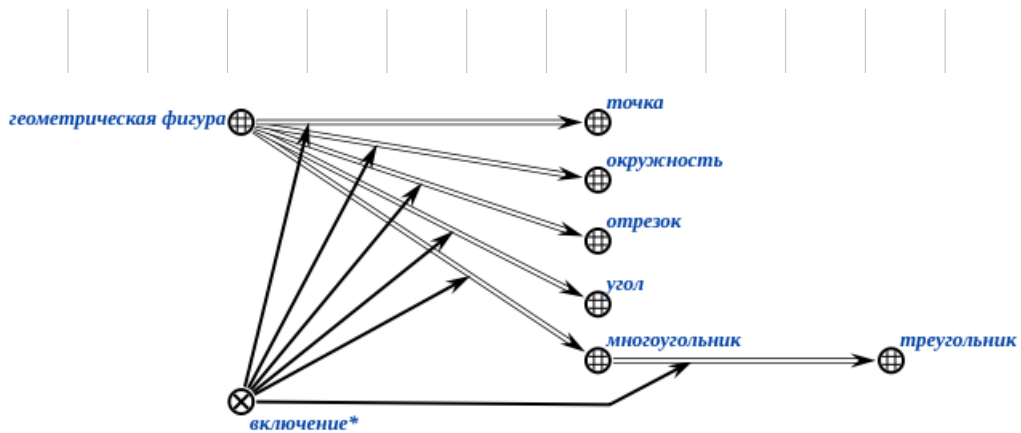
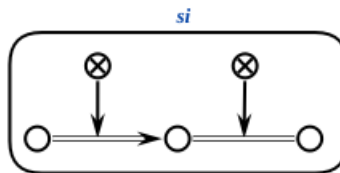
∈ *sc.g-текст*
 ⇒ *пояснение**:
 [В данном примере представлена тривиальная *структура si*, которая содержит три *sc-элемента* – *e1*, *e2*, *e3*.]
 ⇒ *трансформация sc.g-текста по Шестому направлению расширения Ядра SCg-кода**:
 [



]
 ⇒ *пояснение**:
 [В результате трансформации *sc.g-узел*, являющийся изображением структуры *si* заменен на *sc.g-контур*, внутри которого изображены *sc.g-узлы*, изображающие *sc-элементы e1*, *e2*, *e3*, при этом соответствующие *sc.g-дуги* не изображаются. Как видно из приведенного *sc.g-текста*, при необходимости *sc.g-контур* также может ставиться в соответствие идентификатор, который изображается в произвольном месте вблизи границы *sc.g-контура*.]
 • [



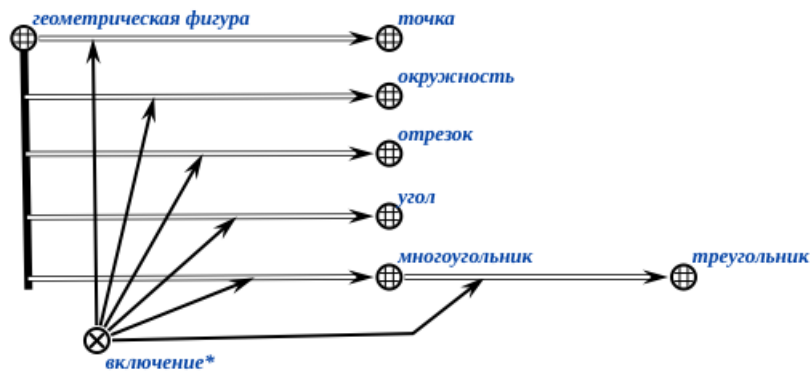
]
 ∈ *sc.g-текст*
 ⇒ *пояснение**:
 [В данном примере *структура* содержит *sc-элементы* различных типов.]
 ⇒ *трансформация sc.g-текста по Шестому направлению расширения Ядра SCg-кода**:
 [



]
 ∈ *sc.g-текст*
 ⇒ *пояснение**:
 [В приведенном примере из *sc.g-узла* “*геометрическая фигура*”, выходит несколько *sc.g-дуг*, с увеличением количества которых *sc.g-текст* становится менее читабельным.]

⇒ трансформация sc.g-текста по Шестому направлению расширения Ядра SCg-кода*:

[



]

} /* Завершили перечень Примеров sc.g-текстов, трансформируемых по Шестому направлению расширения Ядра SCg-кода */

} /* Завершили Сегмент “Описание Шестого направления расширения Ядра SCg-кода” */

/* Сегмент *****/

Описание Седьмого направления расширения Ядра SCg-кода

⊃=

{

Седьмое направление расширения Ядра SCg-кода

⇒ пояснение*:

[Седьмое направление синтаксического расширения Ядра SCg-кода – это переход от 2D-изображений sc.g-текстов к 3D-изображениям. Одним из вариантов трехмерного изображения sc.g-текстов является следующий:

- все sc.g-узлы изображаются, как и ранее, плоскими графическими примитивами. При изменении точки просмотра они всегда "поворачиваются" параллельно плоскости экрана, но их масштаб (размер на экране) при удалении от точки просмотра уменьшается;
- аналогичным "плоским" образом изображаются sc.g-рамки с их "внутренним" содержанием, а также внешние идентификаторы, приписываемые sc.g-элементам;
- sc.g-коннекторы изображаются непересекающимися линиями в трехмерном пространстве (заметим, что при изображении sc.g-текстов на плоскости пересечение sc.g-коннекторов часто снижает наглядность, "читабельность" sc.g-текстов). Т.е. sc.g-коннекторы, которые на плоскости изображаются двойными линиями, в пространстве цилиндрическими, "трубчатыми линиями" с находящейся внутри тонкой, но просвечивающейся осевой линией;
- sc.g-контур в пространстве визуализируется несколькими (!) специального вида точками – например там, где есть точки инцидентности sc.g-контура с внешними sc.g-коннекторами. При этом sc.g-контур становится виден только по команде просмотра указываемого контура (указание контура – это указание одной из его точек инцидентности). По этой команде цветом выделяются все граничные точки контура (точки инцидентности) и все внутренние sc.g-элементы контура. Если просматривается несколько контуров, то используется несколько цветов.

Вторым вариантом 3D-визуализации sc.g-текстов является размещение sc.g-текстов на параллельных плоскостях (слоях) с “прошивками” между этими слоями, соединяющими синонимичные sc.g-узлы, т.е. sc.g-узлы, имеющие одинаковые приписываемые им внешние идентификаторы. Такой вариант плоской, но многослойной визуализации sc.g-текстов дает возможность широко использовать те средства просмотра и редактирования sc.g-текстов, которые разработаны для плоской их визуализации.]

} /* Завершили Сегмент “Описание Седьмого направления расширения Ядра SCg-кода” */

} /* Завершили Раздел “Описание языка графического представления информационных конструкций в ostis-системах” */


```

/* Раздел *****/
Описание языка линейного представления информационных конструкций в ostis-системах
⊃=
{
:= [Описание SCs-кода]
⇐ конкатенация сегментов*:
{
• Описание Алфавита SCs-кода
• Описание sc.s-разделителей и sc.s-ограничителей
• Описание sc.s-предложений
• Описание Ядра SCs-кода и различных направлений его расширения
}
}

SCs-код
:= [Semantic Code string]
:= [Язык линейного представления знаний ostis-систем]
:= [Множество всевозможных текстов SCs-кода]
:= [Тексты SCs-кода]
∈ имя собственное
:= [текст SCs-кода]
∈ имя нарицательное
:= [sc.s-текст]
∈ линейный язык
⇒ алфавит*:
Алфавит SCs-кода
⇒ разделители*:
sc.s-разделитель
⇒ ограничители*:
sc.s-ограничитель
⇒ предложения*:
sc.s-предложение
⇒ неоднозначные обозначения описываемых сущностей*:
неоднозначное sc.s-изображение sc-элемента
:= пояснение*:
[Множество линейных текстов (sc.s-текстов), каждый из которых состоит из предложений (sc.s-предложений),
разделенных друг от друга двойной точкой с запятой (разделителем sc.s-предложений). При этом
sc.s-предложение представляет собой последовательность sc-идентификаторов, являющихся именами опи-
сываемых сущностей и разделяемых между собой различными sc.s-разделителями и sc.s-ограничителями]

неоднозначное sc.s-изображение sc-элемента
⇔ пара пересекающихся множеств*:
sc-выражение
:= [условное обозначение именуемой (неидентифицируемой) сущности]
⊃ sc.s-коннектор
:= [неоднозначное sc.s-изображение sc-коннектора, являющееся также одновременно одним из видов
sc.s-разделителей]
⊃ sc.s-разделитель
⊃ неоднозначное sc.s-изображение sc-узла
⊃ условное обозначение именуемого множества sc-элементов
⇒ пояснение*:
[условное обозначение именуемого множества sc-элементов в SCs-коде представляется
строкой из двух символов – левой фигурной скобки и правой фигурной скобки.]
⊃ условное обозначение именуемого кортежа sc-элементов
⇒ пояснение*:
[В SCs-коде такое обозначение представляется двух-символьной строкой, состоящей из левой
угловой скобки и правой угловой скобки]

```

- ▷ условное обозначение неименованного файла-экземпляра *ostis-системы*
⇒ пояснение*:
[В SCs-коде такое обозначение представляется двух-символьной *строкой*, состоящей из *левой квадратной скобки* и *правой квадратной скобки*]
- ▷ условное обозначение неименованного файла-образца *ostis-системы*
⇒ пояснение*:
[В SCs-коде такое обозначение представляется *строкой*, состоящей из *восклицательного знака*, *левой квадратной скобки*, *правой квадратной скобки* и еще одного *восклицательного знака*]

/* Сегмент *****/

Описание Алфавита SCs-кода

▷=
{

Алфавит SCs-кода

:= [Алфавит символов SCs-кода]

:= [множество символов SCs-кода]

:= [символ, используемый в текстах SCs-кода]

⇐ объединение*:

- Алфавит символов, используемых в *sc.s-разделителях*
- Алфавит символов, используемых в *sc.s-ограничителях*
- Алфавит символов, используемых в *sc-идентификаторах*

⇐ объединение*:

- Алфавит символов, используемых в *простых строковых sc-идентификаторах*
- Алфавит символов, используемых в *sc-выражениях*

- Алфавит символов, используемых в *неоднозначных sc.s-изображениях sc-узлов*

⇒ принципы*:

- [Алфавит SCs-кода строится на основе современных общепринятых наборов символов, что позволяет упростить разработку средств для работы с *sc.s-текстами* с использованием современных технологий.]
- [В состав *sc.s-текстов*, как и в состав текстов любых других языков, являющихся вариантами внешнего отображения текстов SC-кода, могут входить различные файлы, в том числе естественно-языковые или даже файлы, содержащие другие *sc.s-тексты*. В общем случае в таких файлах могут использоваться самые разные символы, в связи с чем будем считать, что в Алфавит SCs-кода эти символы не включаются.]

Алфавит символов, используемых в *sc.s-разделителях*

Э пробел; точка с запятой; двоеточие; круглый маркер; знак равенства

▷ Алфавит символов, используемых в *sc.s-разделителях*, изображающих связь инцидентности *sc-элементов*

Э ![<]! ; ![>]! ; ![|]! ; ![-]!

▷ Алфавит символов, используемых в *sc.s-коннекторах*

▷ Расширенный алфавит символов, используемых в *sc.s-коннекторах*

:= [Расширенный алфавит *sc.s-коннекторов*]

Э ![∈]! ; ![∋]! ; ![∉]! ; ![≠]! ; ![⊆]! ; ![⊇]! ; ![⊂]! ; ![⊃]! ; ![≤]! ; ![≥]! ; ![⇐]! ; ![⇒]! ; ![⇔]! ; ![←]! ; ![→]! ; ![↔]!

▷ Базовый алфавит символов, используемых в *sc.s-коннекторах*

:= [Базовый алфавит *sc.s-коннекторов*]

Э ![~]! ; знак подчеркивания; знак равенства; ![>]! ; ![<]! ; двоеточие; ![-]! ; ![|]! ; ![/]!

⇒ примечание*:

[Как в Базовом, так и в Расширенном Алфавитах *sc.s-коннекторов* используются следующие общие признаки, характеризующие тип изображаемого *sc-коннектора*:

- знак подчеркивания как признак изображений переменных *sc-коннекторов* (один знак подчеркивания для *sc-коннекторов*, являющихся первичными *sc-переменными*, два знака подчеркивания для *sc-коннекторов*, являющихся вторичными *sc-переменными* (*sc-метапеременными*));
- вертикальная черта ("") как признак изображений негативных *sc-дуг* принадлежности;
- косая черта ("/") как признак изображений нечетких *sc-дуг* принадлежности;

□ тильда ("~") как признак изображений временных sc-дуг принадлежности

При необходимости комбинации указанных признаков перечисленные символы комбинируются так, как показано в сегменте "*Описание sc.s-разделителей и sc.s-ограничителей*".]

- { • *Расширенный алфавит символов, используемых в sc.s-коннекторах*
- *Базовый алфавит символов, используемых в sc.s-коннекторах*
- }

⇒ *пояснение**:

[Для упрощения процесса разработки исходных текстов баз знаний с использованием SCs-кода и создания соответствующих средств вводятся два алфавита символов. *Базовый алфавит символов, используемых в sc.s-коннекторах* включает только символы, входящие в переносимый набор символов (portable character set) и имеющиеся на стандартной современной клавиатуре. Таким образом, для разработки исходных текстов баз знаний, использующих только *Базовый алфавит символов, используемых в sc.s-коннекторах* достаточно обычного текстового редактора. *Расширенный алфавит символов, используемых в sc.s-коннекторах* включает также дополнительные символы, которые позволяют сделать sc.s-тексты (и sc.n-тексты) более читабельными и наглядными. Для визуализации и разработки sc.s-текстов с использованием расширенного алфавита требуется наличие специализированных средств.]

Алфавит символов, используемых в sc.s-ограничителях

⊃ ![(! ; ! [)] ! ; ! [*] !

Алфавит символов, используемых в неоднозначных sc.s-изображениях sc-узлов

⊃ ![{ ! ; ! [}] ! ; ! [-] ! ; ! [!] ! ; ! [[] ! ; ! []] !

} /* Завершили Сегмент "Описание Алфавита SCs-кода" */

/* Сегмент *****

Описание sc.s-разделителей и sc.s-ограничителей

⊃ =

{

sc.s-разделитель

:= [разделитель, используемый в sc.s-текстах]

⇒ *разбиение**:

- { • *sc.s-разделитель, используемый для структуризации sc.s-предложений*

⊃ *sc.s-коннектор*

⊃ *sc.s-разделитель, изображающий связь инцидентности sc-элементов*

⊃ *двоеточие*

= ![:]!

⇒ *примечание**:

[Разделяет sc-идентификатор бинарного отношения и второй компонент одной из связок этого отношения в случае, если указанное бинарное отношение и его связка связаны константной sc-дугой принадлежности.]

⊃ *двойное двоеточие*

= ![: :]!

⇒ *примечание**:

[Разделяет sc-идентификатор бинарного отношения и второй компонент одной из связок этого отношения в случае, если указанное бинарное отношение и его связка связаны переменной sc-дугой принадлежности.]

- *sc.s-разделитель sc.s-предложений*

= ![;]!

:= [двойная точка с запятой]

}

sc.s-ограничитель

⊃ *sc.s-ограничитель присоединенных sc.s-предложений*

= (![(*]! ∪ ![*]!)

⇒ *пояснение**:
 [Круглые скобки со звездочкой ограничивают присоединенные sc.s-предложения, которые, в свою очередь, могут иметь в своем составе другие присоединенные sc.s-предложения.]

sc.s-коннектор

:= [изображение *sc-коннектора* во внешнем тексте SCs-кода или SCn-кода]

C *sc.s-разделитель*

⇒ *примечание**:

[типология sc.s-коннекторов полностью соответствует типологии sc.g-коннекторов, и, тем более, sc-коннекторов, т.к. она учитывает устоявшиеся традиции изображения связей целого ряда конкретных отношений]

⇒ *разбиение**:

- *ориентированный sc.s-коннектор*
- *неориентированный sc.s-коннектор*

⇒ *разбиение**:

- *sc.s-коннектор, соответствующий sc.g-дуге принадлежности*
- *sc.s-коннектор, соответствующий sc.g-коннектору, который не является sc.g-дугой принадлежности*

sc.s-разделитель, изображающий связь инцидентности sc-элементов

⇒ *разбиение**:

- *знак инцидентности “правого” sc-коннектора*
 := [знак инцидентности sc-коннектора, sc-идентификатор которого находится справа]
 = ![|]!
- *знак инцидентности “левого” sc-коннектора*
 := [знак инцидентности sc-коннектора, sc-идентификатор которого находится слева]
 = ![-]!
- *знак инцидентности входящей sc-дуги справа*
 := [знак инцидентности sc-дуги, sc-идентификатор который находится справа]
 = ![<]!
- *знак инцидентности входящей sc-дуги слева*
 := [знак инцидентности sc-дуги, sc-идентификатор который находится слева]
 = ![>]!

⇒ *пояснение**:

[На множестве sc-элементов задано бинарное ориентированное отношение инцидентности sc-элементов, а так же подмножество этого отношения – отношение инцидентности входящих sc-дуг, каждая пара которого связывает sc-дугу с тем sc-элементом, в который она входит. В SC-коде sc-коннекторы могут соединять между собой не только sc-узел с sc-узлами, но и sc-узел с sc-коннектором и даже sc-коннектор с sc-коннектором. В последнем случае, указывая инцидентность sc-коннекторов, необходимо уточнить, какой из них является соединяемым (связываемым), а какой-соединяющим (связующим). Поэтому отношение инцидентности, заданное на множестве sc-элементов является ориентированным. Первый компонент пары этого отношения – связующий sc-коннектор, а второй – связуемый sc-элемент. Очевидно, что связующий sc-элемент всегда является sc-коннектором, а sc-узел может быть только связуемым.]

⇒ *примечание**:

[указанные sc.s-разделители с точки зрения синтаксической структуры sc.s-предложений аналогичны sc.s-коннекторам, но с точки зрения их денотационной семантики в отличие от sc.s-коннекторов они не являются изображениями соответствующих sc-коннекторов]

Описание изображения sc.s-коннекторов в Базовом и Расширенном алфавите

⊃=
 {

константная постоянная позитивная sc.s-дуга принадлежности

:= [базовая sc.s-дуга]

⇒ *изображение в базовом алфавите sc.s-коннекторов**:

- ![->]!
- ![<-]!

⇒ *изображение в расширенном алфавите sc.s-коннекторов**:

- ![⊃]!

⇒ • $![\in]!$
изображение в SCg-коде*:

• []
• []
]



константная постоянная негативная sc.s-дуга принадлежности

⇒ изображение в базовом алфавите sc.s-коннекторов*:

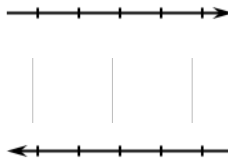
• $![-|>]!$
• $![<|-]!$

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

• $![\neq]!$
• $![\notin]!$

⇒ изображение в SCg-коде*:

• []
• []
]



константная постоянная нечеткая sc.s-дуга принадлежности

⇒ изображение в базовом алфавите sc.s-коннекторов*:

• $![-|>]!$
• $![<|-]!$

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

• $![/ \ni]!$
• $![\in /]!$

⇒ изображение в SCg-коде*:

• []
• []
]



константная временная позитивная sc.s-дуга принадлежности

⇒ изображение в базовом алфавите sc.s-коннекторов*:

• $![\sim >]!$
• $![< \sim]!$

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

• $![\sim \ni]!$
• $![\in \sim]!$

⇒ изображение в SCg-коде*:

• []
]





константная временная негативная sc.s-дуга принадлежности

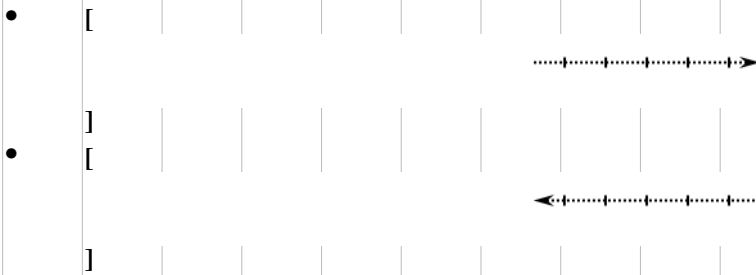
⇒ изображение в базовом алфавите sc.s-коннекторов*:

- $![\sim \triangleright]!$
- $![\triangleleft \sim]!$

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

- $![\sim \not\triangleright]!$
- $![\not\triangleleft \sim]!$

⇒ изображение в SCg-коде*:



константная временная нечеткая sc.s-дуга принадлежности

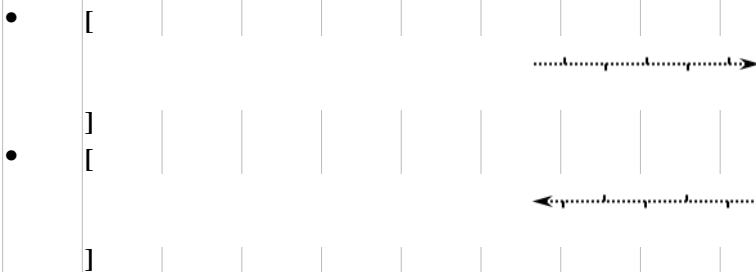
⇒ изображение в базовом алфавите sc.s-коннекторов*:

- $![\sim / \triangleright]!$
- $![\triangleleft / \sim]!$

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

- $![\sim / \ni]!$
- $![\in / \sim]!$

⇒ изображение в SCg-коде*:



переменная постоянная позитивная sc.s-дуга принадлежности

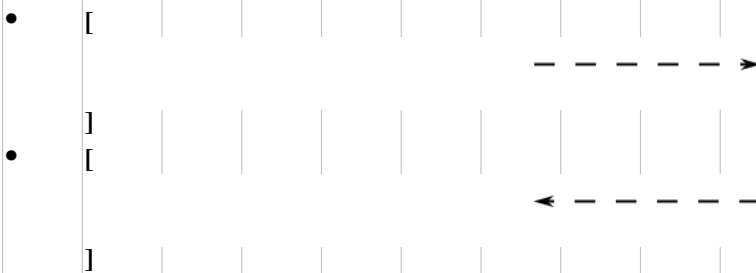
⇒ изображение в базовом алфавите sc.s-коннекторов*:

- $![_ \triangleright]!$
- $![_ \triangleleft]!$

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

- $![_ \ni]!$
- $![_ \in]!$

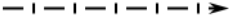
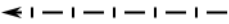
⇒ изображение в SCg-коде*:



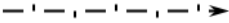
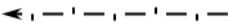
переменная постоянная негативная sc.s-дуга принадлежности

⇒ изображение в базовом алфавите sc.s-коннекторов*:


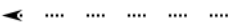
- $![_ \triangleright]!$

- $! [< | - _] !$
- \Rightarrow изображение в расширенном алфавите *sc.s*-коннекторов*:
- $! [_ \neq] !$
- $! [\notin _] !$
- \Rightarrow изображение в SCg-коде*:
- []
- []

переменная постоянная нечеткая *sc.s*-дуга принадлежности

- \Rightarrow изображение в базовом алфавите *sc.s*-коннекторов*:
- $! [_ / >] !$
- $! [< / - _] !$
- \Rightarrow изображение в расширенном алфавите *sc.s*-коннекторов*:
- $! [_ / \ni] !$
- $! [\in / _] !$
- \Rightarrow изображение в SCg-коде*:
- []
- []

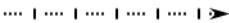

переменная временная позитивная *sc.s*-дуга принадлежности

- \Rightarrow изображение в базовом алфавите *sc.s*-коннекторов*:
- $! [_ \sim >] !$
- $! [< \sim _] !$
- \Rightarrow изображение в расширенном алфавите *sc.s*-коннекторов*:
- $! [_ \sim \ni] !$
- $! [\in \sim _] !$
- \Rightarrow изображение в SCg-коде*:
- []
- []

переменная временная негативная *sc.s*-дуга принадлежности

- \Rightarrow изображение в базовом алфавите *sc.s*-коннекторов*:
- $! [_ \sim >] !$
- $! [< | \sim _] !$
- \Rightarrow изображение в расширенном алфавите *sc.s*-коннекторов*:
- $! [_ \sim \neq] !$
- $! [\notin \sim _] !$

⇒ *изображение в SCg-коде*:*

- []
- []

переменная временная нечеткая sc.s-дуга принадлежности

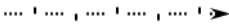

⇒ *изображение в базовом алфавите sc.s-коннекторов*:*

- $![_\sim/>]!$
- $![</\sim_]!$

⇒ *изображение в расширенном алфавите sc.s-коннекторов*:*

- $![_\sim/\exists]!$
- $![\in/\sim_]!$

⇒ *изображение в SCg-коде*:*

- []
- []

метаварiable постоянная позитивная sc.s-дуга принадлежности



⇒ *изображение в базовом алфавите sc.s-коннекторов*:*

- $![_>]!$
- $![<_]!$

⇒ *изображение в расширенном алфавите sc.s-коннекторов*:*

- $![_ \exists]!$
- $![\in _]!$

⇒ *изображение в SCg-коде*:*

- []
- []

метаварiable постоянная негативная sc.s-дуга принадлежности

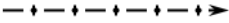
⇒ *изображение в базовом алфавите sc.s-коннекторов*:*

- $![_>]!$
- $![<_]!$

⇒ *изображение в расширенном алфавите sc.s-коннекторов*:*

- $![_ \not\exists]!$
- $![\notin _]!$

⇒ *изображение в SCg-коде*:*

- []

- []

метаперименная постоянная нечеткая sc.s-дуга принадлежности

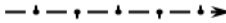
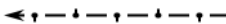
⇒ изображение в базовом алфавите sc.s-коннекторов*:

- $![_{-/}>]!$
- $![_{-/}<]!$

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

- $![_{/ \ni]!$
- $![_{/ \in]!$

⇒ изображение в SCg-коде*:

- []
- []

метаперименная временная позитивная sc.s-дуга принадлежности



⇒ изображение в базовом алфавите sc.s-коннекторов*:

- $![_{\sim}>]!$
- $![_{\sim}<]!$

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

- $![_{\sim \ni]!$
- $![_{\sim \in]!$

⇒ изображение в SCg-коде*:

- []
- []

метаперименная временная негативная sc.s-дуга принадлежности



⇒ изображение в базовом алфавите sc.s-коннекторов*:

- $![_{\sim}>]!$
- $![_{\sim}<]!$

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

- $![_{\sim \not\in]!$
- $![_{\sim \notin]!$

⇒ изображение в SCg-коде*:

- []
- []

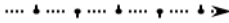
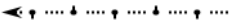
метаперименная временная нечеткая sc.s-дуга принадлежности

⇒ изображение в базовом алфавите sc.s-коннекторов*:

- $![_{\sim}>]!$

• $! [\langle / \sim _] !$
 \Rightarrow изображение в расширенном алфавите *sc.s*-коннекторов*:

• $! [_ \sim / \ni] !$
 • $! [\in / \sim _] !$
 \Rightarrow изображение в *SCg*-коде*:

• []
 • []]

sc.s-ребро общего вида


\Rightarrow изображение в базовом алфавите *sc.s*-коннекторов*:

$! [\langle \rangle] !$

\Rightarrow изображение в расширенном алфавите *sc.s*-коннекторов*:

$! [\leftrightarrow] !$

\Rightarrow изображение в *SCg*-коде*:

[]

sc.s-дуга общего вида

\Rightarrow изображение в базовом алфавите *sc.s*-коннекторов*:

• $! [\gg] !$



• $! [\ll] !$

\Rightarrow изображение в расширенном алфавите *sc.s*-коннекторов*:

• $! [\rightarrow] !$

• $! [\leftarrow] !$

\Rightarrow изображение в *SCg*-коде*:

• []
 • []]

константное постоянное *sc.s*-ребро общего вида


\Rightarrow изображение в базовом алфавите *sc.s*-коннекторов*:

$! [\langle \Rightarrow \rangle] !$

\Rightarrow изображение в расширенном алфавите *sc.s*-коннекторов*:

$! [\Leftrightarrow] !$

\Rightarrow изображение в *SCg*-коде*:

[]

константная постоянная *sc.s*-дуга общего вида

\Rightarrow изображение в базовом алфавите *sc.s*-коннекторов*:

• $! [\Rightarrow] !$

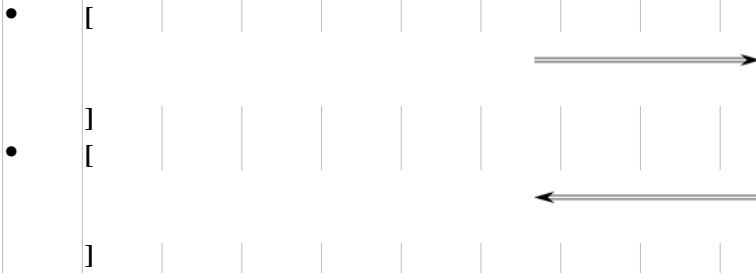
• $! [\Leftarrow] !$

\Rightarrow изображение в расширенном алфавите *sc.s*-коннекторов*:

• $! [\Rightarrow] !$

• $! [\Leftarrow] !$

⇒ изображение в SCg-коде*:



константное временное sc.s-ребро общего вида

⇒ изображение в базовом алфавите sc.s-коннекторов*:

!![~<=>]!

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

!![~⇔]!

⇒ изображение в SCg-коде*:



константная временная sc.s-дуга общего вида

⇒ изображение в базовом алфавите sc.s-коннекторов*:

- !![~=>]!

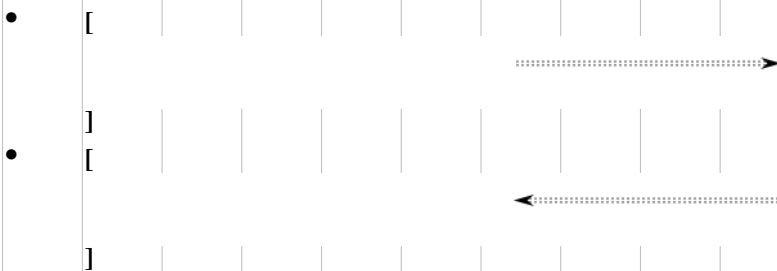
- !![<=~]!

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

- !![~⇒]!

- !![⇐~]!

⇒ изображение в SCg-коде*:



переменное постоянное sc.s-ребро общего вида

⇒ изображение в базовом алфавите sc.s-коннекторов*:

!![_<=>]!

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

!![_⇔]!

⇒ изображение в SCg-коде*:



переменная постоянная sc.s-дуга общего вида

⇒ изображение в базовом алфавите sc.s-коннекторов*:

- !![_=>]!

- !![<=_]!

⇒ изображение в расширенном алфавите sc.s-коннекторов*:

- !![_⇒]!

- !![⇐_]!

⇒ *изображение в SCg-коде*:*

- [=====▶]
- [◀=====]

переменное временное sc.s-ребро общего вида

⇒ *изображение в базовом алфавите sc.s-коннекторов*:*

![_ ~<=>]!

⇒ *изображение в расширенном алфавите sc.s-коннекторов*:*

![_ ~⇔]!

⇒ *изображение в SCg-коде*:*

- [.....]

переменная временная sc.s-дуга общего вида

⇒ *изображение в базовом алфавите sc.s-коннекторов*:*

- ![_ ~=>]!

- ![<=~_]!

⇒ *изображение в расширенном алфавите sc.s-коннекторов*:*

- ![_ ~⇒]!

- ![⇐~_]!

⇒ *изображение в SCg-коде*:*

- [.....▶]
- [◀.....]

метапеременное постоянное sc.s-ребро общего вида

⇒ *изображение в базовом алфавите sc.s-коннекторов*:*

![_ <=>]!

⇒ *изображение в расширенном алфавите sc.s-коннекторов*:*

![_ ⇔]!

⇒ *изображение в SCg-коде*:*

- [=.....=]

метапеременная постоянная sc.s-дуга общего вида

⇒ *изображение в базовом алфавите sc.s-коннекторов*:*

- ![_ =>]!

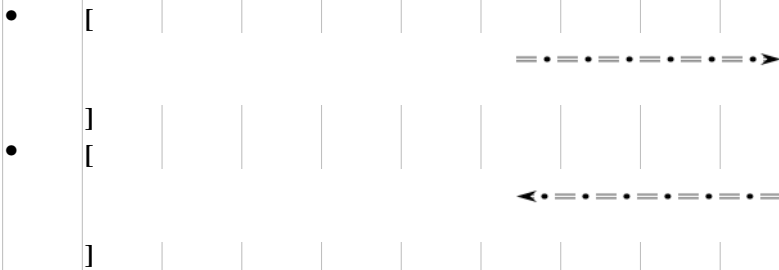
- ![<=_]!

⇒ *изображение в расширенном алфавите sc.s-коннекторов*:*

- ![_ ⇒]!

- ![⇐_]!

⇒ *изображение в SCg-коде*:*



метаемременное временное sc.s-ребро общего вида

⇒ *изображение в базовом алфавите sc.s-коннекторов*:*

![$_ \sim <=>$]!

⇒ *изображение в расширенном алфавите sc.s-коннекторов*:*

![$_ \sim \Leftrightarrow$]!

⇒ *изображение в SCg-коде*:*



метаемременная временная sc.s-дуга общего вида

⇒ *изображение в базовом алфавите sc.s-коннекторов*:*

- ![$_ \sim =>$]!

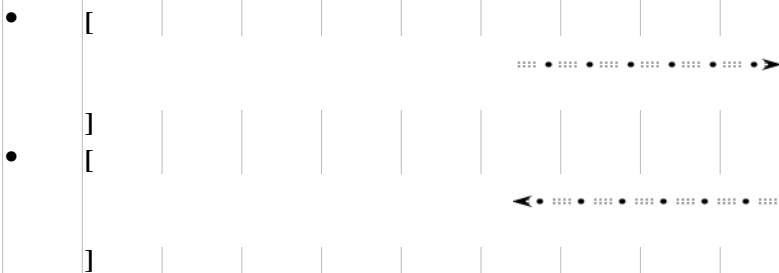
- ![$<=&sim _$]!

⇒ *изображение в расширенном алфавите sc.s-коннекторов*:*

- ![$_ \sim \Rightarrow$]!

- ![$\Leftarrow \sim _$]!

⇒ *изображение в SCg-коде*:*



}
⇒ *примечание*:*

[Аналогичным образом может быть описано изображение всех классов sc.s-коннекторов, представленных в Таблица. Алфавит sc.s-коннекторов, соответствующих sc.g-дугам принадлежности и Таблица. Алфавит sc.s-коннекторов, соответствующих sc.g-дугам принадлежности]

Примеры синтаксической трансформации sc.s-предложений с использованием Расширенного алфавита sc.s-коннекторов

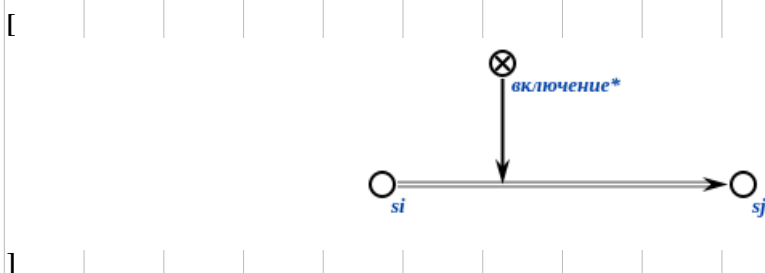
⊃=
{

[$s_i \Rightarrow$ включение* $: s_j$]

⇒ *синтаксическая трансформация*:*

[$s_i \supseteq s_j$]

⇔ *семантическая эквивалентность*:*



$[si _ \Rightarrow \text{включение}^*:: sj]$

\Rightarrow синтаксическая трансформация*:

$[si _ \supseteq sj]$

\Leftrightarrow семантическая эквивалентность*:

[



]

$[si _ \Rightarrow \text{включение}^*::: sj]$

\Rightarrow синтаксическая трансформация*:

$[si _ \supseteq sj]$

\Leftrightarrow семантическая эквивалентность*:

[



]

$[si \Rightarrow \text{строгое включение}^*:: sj]$

\Rightarrow синтаксическая трансформация*:

$[si \supseteq sj]$

\Leftrightarrow семантическая эквивалентность*:

[



]

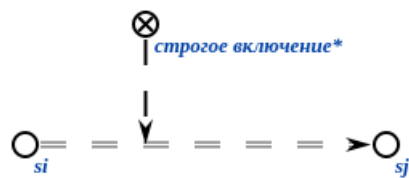
$[si _ \Rightarrow \text{строгое включение}^*::: sj]$

\Rightarrow синтаксическая трансформация*:

$[si _ \supseteq sj]$

\Leftrightarrow семантическая эквивалентность*:

[



]

$[si _ \Rightarrow \text{строгое включение}^*::: sj]$

\Rightarrow синтаксическая трансформация*:

$[si _ \supseteq sj]$

⇔ семантическая эквивалентность*:

[



]

[$si \Rightarrow$ порядок величин*: sj]

\Rightarrow синтаксическая трансформация*:

[$si \geq sj$]

⇔ семантическая эквивалентность*:

[



]

[$si _ \Rightarrow$ порядок величин*:: sj]

\Rightarrow синтаксическая трансформация*:

[$si _ \geq sj$]

⇔ семантическая эквивалентность*:

[



]

[$si _ _ \Rightarrow$ порядок величин*::: sj]

\Rightarrow синтаксическая трансформация*:

[$si _ _ \geq sj$]

⇔ семантическая эквивалентность*:

[



]

[$si \Rightarrow$ строгий порядок величин*: sj]

\Rightarrow синтаксическая трансформация*:

[$si > sj$]

⇔ семантическая эквивалентность*:

[



]

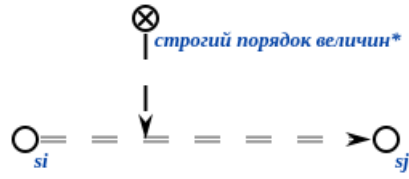
$[si _ \Rightarrow \text{строгий порядок величин}^*:: sj]$

\Rightarrow синтаксическая трансформация*:

$[si _ > sj]$

\Leftrightarrow семантическая эквивалентность*:

[



]

$[si _ _ \Rightarrow \text{строгий порядок величин}^*::: sj]$

\Rightarrow синтаксическая трансформация*:

$[si _ _ > sj]$

\Leftrightarrow семантическая эквивалентность*:

[



]

$[si \Rightarrow \text{внешний идентификатор}^*:: sj]$

\Rightarrow синтаксическая трансформация*:

$[si := sj]$

\Leftrightarrow семантическая эквивалентность*:

[



]

$[si _ _ \Rightarrow \text{внешний идентификатор}^*::: sj]$

\Rightarrow синтаксическая трансформация*:

$[si _ _ := sj]$

\Leftrightarrow семантическая эквивалентность*:

[



]

$[si _ _ _ \Rightarrow \text{внешний идентификатор}^*::: sj]$

\Rightarrow синтаксическая трансформация*:

$[si _ _ _ := sj]$

⇔ семантическая эквивалентность*:

[



]

[$si \Leftrightarrow$ синонимия*: sj]

⇒ синтаксическая трансформация*:

[$si = sj$]

⇔ семантическая эквивалентность*:

[



]

[$si \Rightarrow$ погружение*: sj]

⇒ синтаксическая трансформация*:

[$si \supseteq sj$]

⇔ семантическая эквивалентность*:

[



]

} ⇒ примечание*:

[Аналогичным образом может быть описана трансформация предложений, содержащих любые классы $sc.s$ -коннекторов, за исключением тех классов $sc.s$ -коннекторов, которые соответствуют классам sc -коннекторов, входящим в Ядро SC-кода.]

⇒ примечание*:

[В общем случае sc -элементы, инцидентные sc -коннекторам, классы которых описаны в данном примере, могут быть как sc -константами, так и sc -переменными (в том числе sc -метапеременными). При этом как переменному sc -коннектору может соответствовать константный sc -узел, так и константному sc -коннектору может соответствовать переменный sc -узел (например, если возникает необходимость переменному sc -узлу приписать внешний идентификатор*). Последняя ситуация встречается не очень часто и возникает в случае, когда область определения соответствующего отношения имеет непустое пересечение с классом sc -переменных.]

знак равенства

= ![=]!

:= [связь синонимии между sc -идентификаторами, по крайней мере один из которого является sc -выражением]

⇒ примечание*:

[знак равенства является $sc.s$ -разделителем двух sc -идентификаторов, которые идентифицируют (именуют) одну и ту же сущность и, соответственно, являются sc -идентификаторами* (внешними уникальными изображениями) одного и того же sc -элемента. При этом из указанных двух sc -идентификаторов чаще всего один является простым sc -идентификатором, а второй – sc -выражением. Реже оба эти sc -идентификатора являются sc -выражениями. И совсем редко оба они являются простыми sc -идентификаторами. Последнее обозначает то, что оба эти sc -идентификатора являются основными sc -идентификаторами* одного и того же sc -элемента. Пример:

SC-код = *sc.s-текст*;;

Здесь первый *sc-идентификатор* является *именем собственным*, а второй – *именем нарицательным*.

При трансляции *sc.s-текста* в *SC-код* знаку равенства на некотором этапе может быть поставлено в соответствие *sc-ребро*, принадлежащее отношению *синонимии** *sc-элементов*, идентифицируемых *sc-идентификаторами*, связанными знаком равенства. Но на последующем этапе указанное *sc-ребро* удаляется, а связанные им *sc-элементы* склеиваются. Таким образом *sc-ребро*, принадлежащее отношению *синонимии** *sc-элементов*, имеет не только денотационную, но и операционную семантику.]

знак равенства с включением

= (![\supset =]! \cup ![= \subset]!)

:= [изображение *sc-дуги*, принадлежащей отношению *погружения**, связывающей два *sc-узла*, обозначающих *sc-тексты*, первый из которых является погружающим, а второй (в который указанная *sc-дуга* входит) является погружаемым, вводимым в состав первого *sc-текста*]

⇒ *примечание**:

[*sc-дуга*, принадлежащая отношению *погружения**, интерпретируется как команда погружения одного *sc-текста* в состав другого. При выполнении этой команды (1) все *sc-элементы* погружаемого *sc-текста* становятся элементами, принадлежащими погружающему *sc-тексту*, (2) все синонимичные *sc-элементы*, оказавшиеся в составе погружающего *sc-текста*, склеиваются, (3) *sc-узел*, обозначающий погружаемый *sc-текст*, а так же спецификация этого *sc-текста* (включая перечень всех его *sc-элементов*) погружается в историю эволюции *базы знаний* вместе со спецификацией события погружения рассматриваемого *sc-текста* в состав *базы знаний*.]

(*знак равенства* \cup *знак равенства с включением*)

⇒ *примечание**:

[указанные *sc.s-коннекторы* отличаются от остальных *sc.s-коннекторов* тем, что они и соответствующие им *sc-коннекторы* (*sc-ребра*, принадлежащих отношению *синонимии* *sc-элементов* и *sc-дуги*, принадлежащие отношению погружения одного *sc-текста* в состав другого) имеют не только денотационную, но и операционную семантику, т.к. являются командами склеивания и командами погружения.]

sc.s-коннектор, соответствующий sc.g-дуге принадлежности

⇒ *таблица**:

Таблица. Алфавит sc.s-коннекторов, соответствующих sc.g-дугам принадлежности

sc.s-коннектор, соответствующий sc.g-коннектору, который не является sc.g-дугой принадлежности

⇒ *таблица**:

Таблица. Алфавит sc.s-коннекторов, соответствующих sc.g-коннекторам, которые не являются sc-дугами принадлежности

Таблица. Алфавит sc.s-коннекторов, соответствующих sc.g-дугам принадлежности

=
[

Класс <i>sc-элементов</i>	Изображение <i>sc.g-коннектора</i>	Изображение <i>sc.s-коннектора</i> в Расширенном алфавите	Изображение <i>sc.s-коннектора</i> в Базовом алфавите
константная постоянная позитивная <i>sc-дуга</i> принадлежности	• —————> •	∋	∈
константная постоянная негативная <i>sc-дуга</i> принадлежности	• ———+———> •	∋̄	∉
константная постоянная нечеткая <i>sc-дуга</i> принадлежности	• ———+———> •	∋̇	∈̇
константная временная позитивная <i>sc-дуга</i> принадлежности	•> •	∋̃	∈̃

Класс <i>sc</i> -элементов	Изображение <i>sc.g</i> -коннектора	Изображение <i>sc.s</i> -коннектора в Расширенном алфавите		Изображение <i>sc.s</i> -коннектора в Базовом алфавите	
константная временная негативная <i>sc</i> -дуга принадлежности		$\sim \cancel{\exists}$	$\notin \sim$	$\sim >$	$< \sim$
константная временная нечеткая <i>sc</i> -дуга принадлежности		\sim / \exists	\in / \sim	$\sim / >$	$< / \sim$
переменная постоянная позитивная <i>sc</i> -дуга принадлежности		$_ \exists$	$_ \in$	$_ - >$	$< - _$
переменная постоянная негативная <i>sc</i> -дуга принадлежности		$_ \cancel{\exists}$	$\notin _$	$_ - >$	$< - _$
переменная постоянная нечеткая <i>sc</i> -дуга принадлежности		$_ / \exists$	$\in / _$	$_ - / >$	$< / - _$
переменная временная позитивная <i>sc</i> -дуга принадлежности		$_ \sim \exists$	$\in \sim _$	$_ \sim >$	$< \sim _$
переменная временная негативная <i>sc</i> -дуга принадлежности		$_ \sim \cancel{\exists}$	$\notin \sim _$	$_ \sim >$	$< \sim _$
переменная временная нечеткая <i>sc</i> -дуга принадлежности		$_ \sim / \exists$	$\in / \sim _$	$_ \sim / >$	$< / \sim _$
метапеременная постоянная позитивная <i>sc</i> -дуга принадлежности		$_ _ \exists$	$_ _ \in$	$_ _ - >$	$< _ _$
метапеременная постоянная негативная <i>sc</i> -дуга принадлежности		$_ _ \cancel{\exists}$	$\notin _ _$	$_ _ - >$	$< _ _$
метапеременная постоянная нечеткая <i>sc</i> -дуга принадлежности		$_ _ / \exists$	$\in / _ _$	$_ _ - / >$	$< / _ _$
метапеременная временная позитивная <i>sc</i> -дуга принадлежности		$_ _ \sim \exists$	$\in \sim _ _$	$_ _ \sim >$	$< \sim _ _$
метапеременная временная негативная <i>sc</i> -дуга принадлежности		$_ _ \sim \cancel{\exists}$	$\notin \sim _ _$	$_ _ \sim >$	$< \sim _ _$
метапеременная временная нечеткая <i>sc</i> -дуга принадлежности		$_ _ \sim / \exists$	$\in / \sim _ _$	$_ _ \sim / >$	$< / \sim _ _$

Таблица. Алфавит *sc.s*-коннекторов, соответствующих *sc.g*-коннекторам, которые не являются *sc.g*-дугами принадлежности

Изображение <i>sc</i> -коннектора в SCg	Изображение <i>sc.s</i> -коннектора в Расширенном алфавите		Изображение <i>sc.s</i> -коннектора в Базовом алфавите	
	\leftrightarrow		$\langle \rangle$	
	\rightarrow	\leftarrow	\gg	\ll
	\Leftrightarrow		\Leftrightarrow	
	\Rightarrow	\Leftarrow	\Rightarrow	\Leftarrow
	\rightsquigarrow		\rightsquigarrow	
	\rightsquigarrow	$\leftarrow \rightsquigarrow$	\rightsquigarrow	$\leftarrow \rightsquigarrow$
	$_ \leftrightarrow$		$_ \langle \rangle$	
	$_ \Rightarrow$	$\Leftarrow _$	$_ \Rightarrow$	$\Leftarrow _$
	$_ \rightsquigarrow$		$_ \rightsquigarrow$	
	$_ \rightsquigarrow$	$\leftarrow \rightsquigarrow _$	$_ \rightsquigarrow$	$\leftarrow \rightsquigarrow _$
	$_ \leftrightarrow$		$_ \langle \rangle$	
	$_ \Rightarrow$	$\Leftarrow _$	$_ \Rightarrow$	$\Leftarrow _$
	$_ \rightsquigarrow$		$_ \rightsquigarrow$	
	$_ \rightsquigarrow$	$\leftarrow \rightsquigarrow _$	$_ \rightsquigarrow$	$\leftarrow \rightsquigarrow _$
	\supseteq	\subseteq		
	$_ \supseteq$	$_ \subseteq$		
	\supset	\subset		
	$_ \supset$	$_ \subset$		

Изображение <i>sc</i> -коннектора в SCg	Изображение <i>sc.s</i> -коннектора в Расширенном алфавите		Изображение <i>sc.s</i> -коннектора в Базовом алфавите
	\geq	\leq	
	$_ \geq$	$_ \leq$	
	$>$	$<$	
	$_ >$	$< _$	
	$:=$		
	$_ :=$		
	$=$		
	$\supset =$	$= \subset$	

]

}

/* Завершили Сегмент “Описание *sc.s*-разделителей и *sc.s*-ограничителей” */

/* Сегмент ***** */

Описание *sc.s*-предложений

$\supset =$
{

***sc.s*-предложение**

- $:=$ [минимальный семантически целостный фрагмент *sc.s*-текста]
- $:=$ [минимальный *sc.s*-текст]
- \subset *sc.s*-текст
- \supset простое *sc.s*-предложение

:= [минимальное sc.s-предложение]

⇒ *пояснение**:

[*sc.s-предложение*, (1) состоящее или из двух *sc-идентификаторов*, соединенных между собой *sc.s-коннектором*, или из трех *sc-идентификаторов*, разделенных *sc.s-разделителями*, изображающими связь инцидентности sc-элементов, и (2) завершающееся двойной точкой с запятой]

⇒ *примечание**:

[Нетрудно заметить, что простые sc.s-предложения по сути аналогичны триплетам языка RDF (RDF-триплетам), за тем исключением, что *простое sc.s-предложение* можно "развернуть" при помощи *Операции конверсии sc.s-предложений** не меняя при этом его смысл, а RDF-триплет нельзя. Это является одной из причин, по которой, в отличие от RDF-триплетов, в простых sc.s-предложениях *sc.s-коннекторы* и *sc.s-разделители*, изображающие связь инцидентности sc-элементов не могут быть опущены, поскольку они в том числе показывают направление изображаемой ими связи между sc-элементами.]

⇒ *пример**:

• [многоугольник \supset треугольник]

⇔ *семантическая эквивалентность**:

[

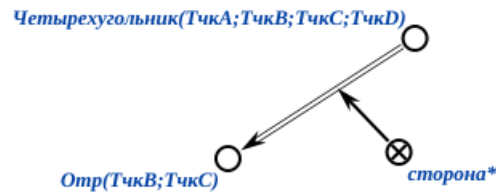


]

• [*сторона** \ni (*Четырехугол(ТчкА;ТчкВ;ТчкС;ТчкD)* \Rightarrow *Отр(ТчкВ;ТчкС)*);;]

⇔ *семантическая эквивалентность**:

[



]

• [*Si* | *ai* > | *ei*]

⇔ *семантическая эквивалентность**:

[



]

⇒ *примечание**:

[Признаком завершения любого *sc.s-предложения*, т.е. последними его символами является двойная точка с запятой, которую, следовательно, можно считать разделителем *sc.s-предложений*.]

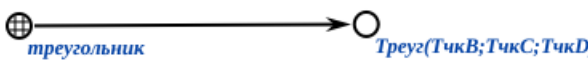
⇒ *заданная операция**:

• *Операция конверсии sc.s-предложения**

С *синтаксическая трансформация**

⇒ *пояснение**:

[Каждое *sc.s-предложение* (в том числе, и *простое sc.s-предложение*) можно преобразовать в семантически эквивалентное ему *sc.s-предложение* путем конверсии ("разворота") цепочки компонентов *sc.s-предложения*. Так, например, при конверсии ("развороте") простого *sc.s-предложения* (1) первый его *sc-идентификатор* (первый компонент этого *sc.s-предложения*) становится третьим компонентом конвертированного *sc.s-предложения*, (2) второй его *sc-идентификатор* (третий компонент исходного *sc.s-предложения*) становится первым компонентом "конвертированного" *sc.s-предложения* и (3) второй компонент исходного *sc.s-предложения* (*sc.s-коннектор* или *sc.s-разделитель*, изображающий связь инцидентности sc-элементов, соединяющий указанные выше компоненты) остается вторым компонентом конвертированного *sc.s-предложения*, но меняет направленность (" \ni " заменяется на " \in " и наоборот, " \supset " на " \subset " и наоборот, " \Rightarrow " на " \Leftarrow " и наоборот и т.д.)]

- ⇒ *примечание**:
[Можно говорить не только о конверсии *sc.s*-предложения, но и о конверсии *sc.s*-коннектора, о конверсии *sc.s*-разделителя, изображающего связь инцидентности *sc.s*-элементов.]
- ⇒ *sc.s-текст до трансформации**:
[*треугольник* \ni *Треуг*(ТчкВ;ТчкС;ТчкD);;]
- ⇒ *sc.s-текст после трансформации**:
[*Треуг*(ТчкВ;ТчкС;ТчкD) \in *треугольник*;;]
- ⇔ *семантическая эквивалентность**:
[
- 

треугольник \longrightarrow Треуг(ТчкВ;ТчкС;ТчкD)
-]

- *Операция присоединения sc.s-предложения**
С *синтаксическая трансформация**
:= [Операция соединения двух *sc.s*-предложений при совпадении последнего компонента первого предложения с первым компонентом второго*]
⇒ *пояснение**:
[В результате выполнения данной операции:
 - первый компонент второго *sc.s*-предложения удаляется;
 - оставшаяся часть второго предложения окружается *sc.s*-ограничителем присоединенных предложений ("*" и "*"). Разделитель *sc.s*-предложений (";;") также попадает внутрь указанного ограничителя;
 - полученная конструкция помещается между последним компонентом первого предложения и разделителем *sc.s*-предложений, которым заканчивалось первое предложение;
 - второе предложение, таким образом, становится присоединенным *sc.s*-предложением.
 Аналогичным образом к любому присоединенному *sc.s*-предложению могут "пристыковываться" другие присоединенные *sc.s*-предложения, в общем случае уровень такой вложенности не ограничен.]
- *Операция слияния sc.s-предложений**
С *синтаксическая трансформация**
:= [Операция присоединения простого *sc.s*-предложения к *sc.s*-предложению, у которого последний *sc.s*-коннектор совпадает с *sc.s*-коннектором простого *sc.s*-предложения, а предшествующий указанному *sc.s*-коннектору *sc*-идентификатор совпадает с первым *sc*-идентификатором простого *sc.s*-предложения*]
⇒ *пояснение**:
[В результате выполнения этой операции совпадающие *sc*-идентификаторы и *sc.s*-коннекторы соединяемых *sc.s*-предложений "склеиваются", а последние *sc*-идентификаторы соединяемых *sc.s*-предложений становятся последними компонентами объединенного *sc.s*-предложения, разделенными точкой с запятой. Аналогичным образом можно присоединять сколько угодно простых *sc.s*-предложений.]
- *Операция разложения sc.s-предложений на простые sc.s-предложения**
С *синтаксическая трансформация**
⇒ *пояснение**:
[Каждое *sc.s*-предложение можно разложить на множество простых *sc.s*-предложений, т.е. представить в виде последовательности простых *sc.s*-предложений.]
- *Операция разложения sc.s-предложений на простые sc.s-предложения с sc.s-разделителем, изображающим связь инцидентности sc-элементов**
С *синтаксическая трансформация**
⇒ *пояснение**:
[Каждое *sc.s*-предложение (в том числе и простое *sc.s*-предложение с *sc.s*-коннектором) можно представить в виде семантически эквивалентной последовательности простых *sc.s*-предложений с *sc.s*-разделителем, изображающим связь инцидентности *sc*-элементов.]
⇒ *примечание**:
[Данная операция осуществляет однозначное (!) формирование множества простых *sc.s*-предложений указанного вида.]

sc.s-предложение

⇒ *примечание**:

[Операции, заданные на множестве *sc.s-предложений* можно разделить на три группы:

- группа операций конверсии *sc.s-предложений*, состоящая из одной операции;
- группа операций соединения *sc.s-предложений*;
- группа операций декомпозиции *sc.s-предложений* и, в частности, операций разложения *sc.s-предложений*.

Очевидно, что операции соединения *sc.s-предложений* и операции декомпозиции *sc.s-предложений* являются обратными друг другу операциями.]

Описание примеров выполнения операций, заданных на множестве sc.s-предложений

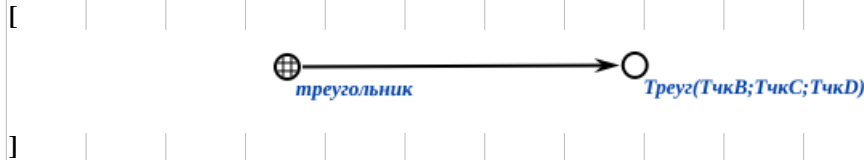
⊇=
{

[треугольник \ni Треугк(ТчкВ;ТчкС;ТчкD);;]

⇒ *Операция конверсии sc.s-предложения**:

[Треугк(ТчкВ;ТчкС;ТчкD) \in *треугольник*;;]

\Leftrightarrow *семантическая эквивалентность**:



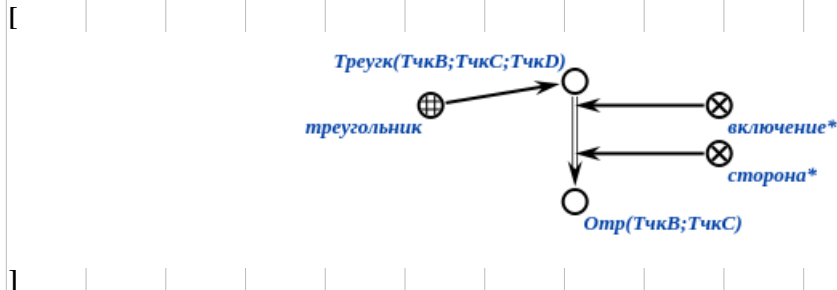
[*треугольник* \ni Треугк(ТчкВ;ТчкС;ТчкD);;

Треугк(ТчкВ;ТчкС;ТчкD) \Rightarrow *сторона**:*включение**: Отр(ТчкВ;ТчкС);;]

⇒ *Операция присоединения sc.s-предложения**:

[*треугольник* \ni Треугк(ТчкВ;ТчкС;ТчкD) (* \Rightarrow *сторона**:*включение**:Отр(ТчкВ;ТчкС);; *) ;;]

\Leftrightarrow *семантическая эквивалентность**:



[*сторона** \ni (Треугк(ТчкВ;ТчкС;ТчкD) \Rightarrow Отр(ТчкВ;ТчкС));;

*сторона** \ni (Треугк(ТчкВ;ТчкС;ТчкD) \Rightarrow Отр(ТчкС;ТчкD));;]

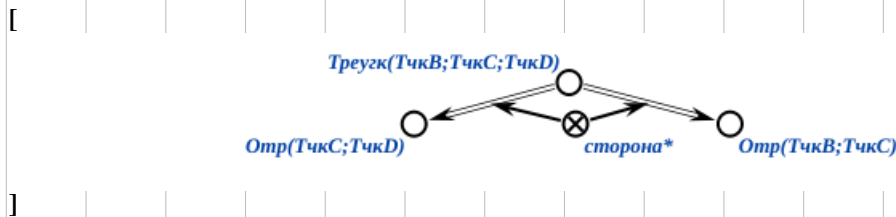
⇒ *Операция слияния sc.s-предложений**:

[*сторона** \ni ((Треугк(ТчкВ;ТчкС;ТчкD) \Rightarrow Отр(ТчкВ;ТчкС));(Треугк(ТчкВ;ТчкС;ТчкD) \Rightarrow Отр(ТчкС;ТчкD)));;]

⇒ *синтаксическая трансформация**:

[Треугк(ТчкВ;ТчкС;ТчкD) \Rightarrow *сторона**: Отр(ТчкВ;ТчкС);Отр(ТчкС;ТчкD);;]

\Leftrightarrow *семантическая эквивалентность**:



[Треугк(ТчкВ;ТчкС;ТчкD) \Rightarrow сторона*:включение*:Отр(ТчкВ;ТчкС);;]

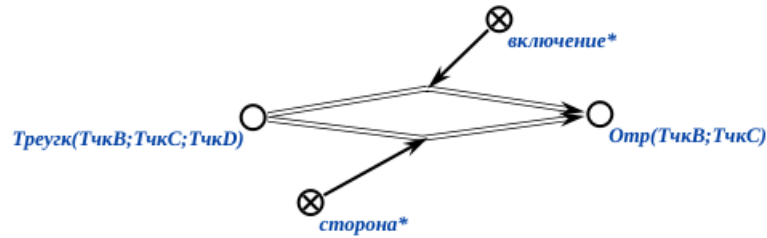
\Rightarrow Операция разложения sc.s-предложений на простые sc.s-предложения*:

[сторона* \ni (Треугк(ТчкВ;ТчкС;ТчкD) \Rightarrow Отр(ТчкВ;ТчкС));;

включение* \ni (Треугк(ТчкВ;ТчкС;ТчкD) \Rightarrow Отр(ТчкВ;ТчкС));;]

\Leftrightarrow семантическая эквивалентность*:

[



]

[треугольник \ni Треугк(ТчкВ;ТчкС;ТчкD)]

\Rightarrow Операция разложения sc.s-предложений на простые sc.s-предложения с sc.s-разделителем, изображающим связь инцидентности sc-элементов*:

[треугольник $| - a_i > |$ Треугк(ТчкВ;ТчкС;ТчкD);;

константный постоянный sc-узел, обозначающий класс \ni треугольник;;

константная постоянная позитивная sc-дуга принадлежности $\ni a_i$;;

константный постоянный sc-узел общего вида \ni Треугк(ТчкВ;ТчкС;ТчкD);;]

\Leftrightarrow семантическая эквивалентность*:

[



]

}

присоединенное sc.s-предложение

:= [встроенное sc.s-предложение]

\Rightarrow пояснение*:

[Присоединенные sc.s-предложения используются для того, чтобы продолжить спецификацию какого-либо sc-элемента, sc-идентификатор которого является последним компонентом в рамках какого-либо sc.s-предложения, не начиная при этом нового sc.s-предложения и, таким образом, не дублируя указанный sc-идентификатор. Внутри присоединенных sc.s-предложений также могут встраиваться другие присоединенные sc.s-предложения, в общем случае уровень вложенности таких предложений не ограничен. Таким образом присоединенные sc.s-предложения описывают "ветвление" sc.s-предложений, при этом точками такого "ветвления" выступают sc-идентификаторы, входящие в состав этих sc.s-предложений.

Благодаря введению присоединенных sc.s-предложений появляется возможность любой sc-текст изобразить в виде одного sc.s-предложения, содержащего необходимое количество присоединенных sc.s-предложений. Таким образом, SCs-код по выразительной мощности становится эквивалентным SCn-коду.]

sc.s-предложение

\Rightarrow денотационная семантика*:

[С семантической точки зрения sc.s-предложение представляет собой описание некоторого маршрута в соответствующем sc-тексте, который является графовой структурой специального вида и структура которого описывается (изображается) с помощью sc.s-предложений. Указанный маршрут "проводится" по sc-коннекторам и по связям инцидентности sc-элементов, если маршрут проходит через инцидентные sc-коннекторы. В описании указанного маршрута могут дополнительно указываться множества (чаще всего отношения), которым принадлежат sc-коннекторы, входящие в описываемый маршрут. Кроме того, указанный маршрут в начале и/или в конце может иметь разветвления, когда какой-либо sc-элемент

одинаково инцидентен нескольким однотипным *sc*-коннекторам, соединяющим указанный *sc*-элемент с некоторыми другими *sc*-элементами.

Таким образом каждое указанное разветвление состоит из неограниченного числа ветвей, каждая из которых состоит из одного *sc*-коннектора и одного связываемого им *sc*-элемента.]

компонент *sc.s*-предложения*

⇒ *пояснение**:

[Каждое *sc.s*-предложение представляет собой последовательность (1) *sc*-идентификаторов, (2) *sc.s*-коннекторов или *sc.s*-разделителей, изображающих связь инцидентности *sc*-элементов, (3) точек с запятыми, (4) ограничителей присоединенных *sc.s*-предложений, завершаемая двойной точкой с запятой. При этом непосредственно соседствовать друг с другом не могут ни *sc*-идентификаторы, ни *sc.s*-коннекторы, ни, очевидно, точки с запятыми и ограничители присоединенных *sc.s*-предложений.

Между *sc*-идентификаторами в рамках *sc.s*-предложения может находиться либо точка с запятой, либо *sc.s*-коннектор, либо *sc.s*-разделитель, изображающий связь инцидентности *sc*-элементов. Слева и справа от *sc.s*-коннектора и от *sc.s*-разделителя, изображающего связь инцидентности *sc*-элементов, должны находиться *sc*-идентификаторы.

Указанные *sc*-идентификаторы, *sc.s*-коннекторы и *sc.s*-разделители, изображающие связь инцидентности *sc*-элементов, считаются компонентами соответствующего *sc.s*-предложения. Понятие "быть компонентом *sc.s*-предложения" является относительным понятием (отношением), т.к. в состав некоторых компонентов *sc.s*-предложения (в состав *sc*-идентификаторов, являющихся *sc.s*-выражениями, ограничиваемыми фигурными или квадратными скобками) могут входить других *sc.s*-предложения, состоящие из своих компонентов.]

⇒ *первый домен**:

sc.s-предложение

⇒ *второй домен**:

(*sc*-идентификатор \cup *sc.s*-разделитель \cup *sc.s*-ограничитель)

sc.s*-модификатор

С *компонент *sc.s*-предложения**

⇒ *пояснение**:

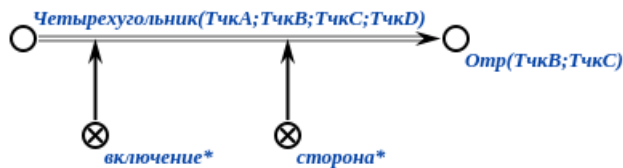
[Это дополнительный вид компонентов *sc.s*-предложений. Каждый *sc.s*-модификатор, являющийся компонентом некоторого *sc.s*-предложения, представляет собой *sc*-идентификатор, обозначающий множество (чаще всего, отношение), которому принадлежит *sc*-коннектор, изображенный *sc.s*-коннектором, который предшествует указанному *sc*-идентификатору. Признаком *sc.s*-модификатора является *двоеточие* (или *двойное двоеточие*), которое ставится после *sc.s*-модификатора и отделяет его либо от следующего за ним другого *sc.s*-модификатора для этого же *sc.s*-коннектора, либо от следующего за ним *sc*-идентификатора, соответствующего *sc*-элементу, который инцидентен *sc*-коннектору, изображенному *sc.s*-коннектором, находящимся левее рассматриваемого *sc*-идентификатора после одного или нескольких *sc.s*-модификаторов. Обычное ("одинарное") двоеточие обозначает, что *sc*-элемент, изображенный соответствующим *sc.s*-модификатором, связан с *sc*-коннектором, изображенным левее этого *sc.s*-модификатора, базовой *sc*-дугой (константной постоянной позитивной *sc*-дугой принадлежности), двойное двоеточие обозначает, что указанные элементы связаны переменной постоянной позитивной *sc*-дугой принадлежности.]

⇒ *пример**:

• [Четырехугольник(*ТчкА*; *ТчкВ*; *ТчкС*; *ТчкD*) ⇒ сторона* : включение* : Отр(*ТчкВ*; *ТчкС*); ;]

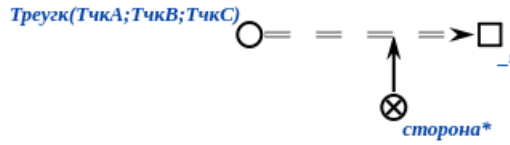
⇔ *семантическая эквивалентность**:

[



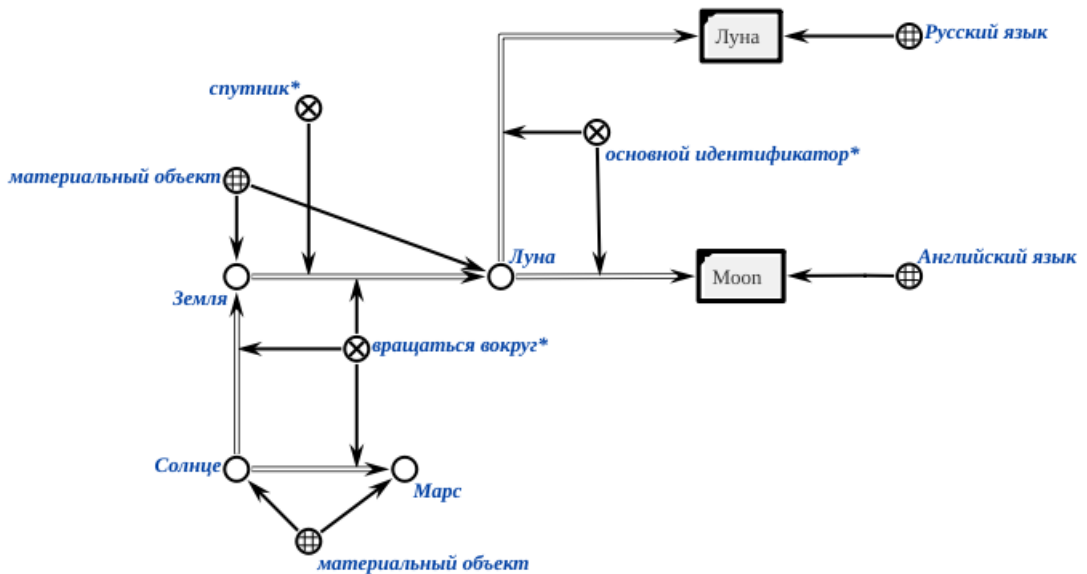
]

- [Треугк(ТчкА;ТчкВ;ТчкС) _ => сторона* :: _s;;]
 ⇔ семантическая эквивалентность*:
 [



sc.s-текст

- := [конкатенация sc.s-предложений]
- := [последовательность sc.s-предложений, разделяемых двойными точками с запятой]
- ⊃ максимальный исходный sc.s-текст
- := [конкатенция sc.s-предложений, слева и справа от которой отсутствуют какие-либо символы SCs-кода]
- ⊃ максимальный sc.s-текст, включенный в структуру
- := [конкатенция всех sc.s-предложений, входящих в состав sc.s-выражения структуры]
- ⊃ sc.s-текст, включенный в структуру
- := [часть цепочки sc.s-предложений, входящих в состав максимального sc.s-текста, включенного в структуру]
- ⊃ sc.s-предложение, включенное в структуру
- ⇒ примечание*:
 [sc.s-предложение является минимальным sc.s-текстом.]
- ⇒ свойство*:
 [Смысл sc.s-текста (а также sc.s-текста, включенного в структуру не зависит от порядка sc.s-предложений в этих sc.s-текстах. Т.е. перестановка sc.s-предложений в рамках таких sc.s-текстов смысла этих sc.s-текстов не меняет (т.е. приводит к семантически эквивалентным sc.s-текстам), но сильно влияет на трудоемкость человеческого восприятия (на "читабельность") этих текстов.]
- ⇒ пример*:
 [материальный объект ∃ Земля (* => вращаться вокруг*: спутник*: Луна;;*);;
 материальный объект ∃ Луна(* => основной идентификатор*: [Moon] (* <- Английский язык;; *); [Луна] (* <- Русский язык;; *); *);;
 материальный объект ∃ Солнце (* => вращаться вокруг*: Земля; Марс;; *);;
 материальный объект ∃ Марс;;]
- ⇒ семантическая эквивалентность*:
 [



} /* Завершили Сегмент “Завершили Описание sc.s-предложений” */

```

/* Сегмент *****/
Описание Ядра SCs-кода и различных направлений его расширения
⊃=
{
Ядро SCs-кода
:= [Подъязык SCs-кода, который использует минимальный набор синтаксических средств, но при этом имеет семантическую мощност, эквивалентную мощност SCs-кода в целом]
⇒ принципы, лежащие в основе*:
[В Ядре SCs-кода:


- используются только простые sc-идентификаторы, в том числе sc-идентификаторы внешних файлов ostis-систем (sc-выражения не используются);
- используются только sc.s-разделители, изображающие связь инцидентности sc-элементов, а также sc.s-коннектор, изображающий константную постоянную позитивную пару принадлежности ("∈" и "∋" в Расширенном алфавите и "->" и "<-" в Базовом алфавите). Другие sc.s-коннекторы не используются;
- не используются sc.s-модификаторы и, соответственно, двоеточия, являющиеся признаком завершения sc.s-модификаторов;
- используются только простые sc.s-предложения, которые, как следует из вышеуказанных свойств Ядра SCs-кода, либо состоят из двух простых sc-идентификаторов, соединяемых sc.s-коннектором, изображающим константную постоянную позитивную пару принадлежности, либо трех простых sc-идентификаторов, разделенных sc.s-разделителями, изображающими связь инцидентности sc-элементов.


Из перечисленных свойств Ядра SCs-кода следует, что для представления (изображения) любого sc-текста средствами Ядра SCs-кода необходимо для всех (!) sc-элементов этого sc-текста (кроме константных постоянных позитивных пар принадлежности) построить соответствующие им простые sc-идентификаторы, т.е. необходимо проименовать все указанные sc-элементы. В свою очередь, тип каждого используемого sc-элемента (кроме константных постоянных позитивных пар принадлежности) задается явно путем указания принадлежности этих элементов соответствующим классам sc-элементов, в том числе классам, входящим в Ядро SC-кода.
Как видно из приведенного описания, Ядро SCs-кода соответствует Ядру SCg-кода, за исключением того, что в Ядре SCg-кода нет необходимости именовать все изображаемые sc-элементы, а также в Ядре SCg-кода присутствуют графические изображения для sc-элементов, принадлежащих соответствующим классам Ядра SC-кода и эту принадлежность нет необходимости указывать явно. ]
⇒ примечание*:
[Очевидно, что широко практически применять Ядро SCs-кода для записи больших фрагментов баз знаний неудобно и неэффективно. Тем не менее, с практической точки зрения Ядро SCs-кода может использоваться, например, для обмена информацией со сторонними средствами представления графовых конструкций, рассчитанными на представление информации в виде триплетов (например, RDF-хранилищ). Для обеспечения возможности более широкого практического использования необходимы синтаксические расширения Ядра SCs-кода в целях:

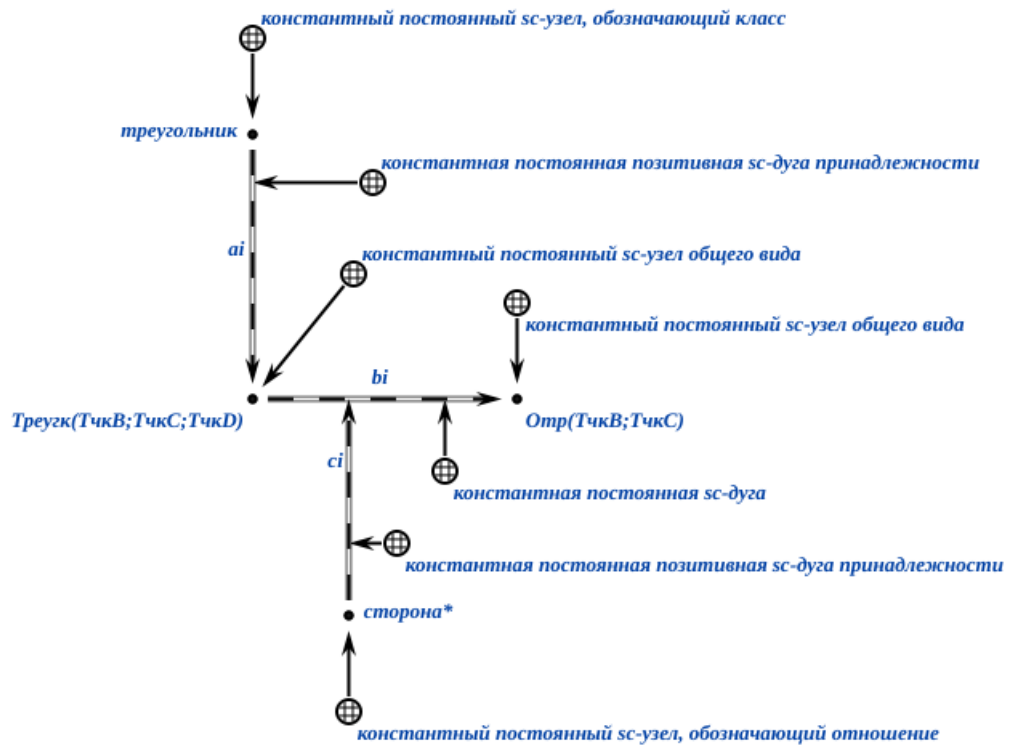

- минимизации числа идентифицируемых (именуемых) sc-элементов путем использования sc-выражений и ликвидации необходимости идентифицировать (именовать) все (!) sc-элементы;
- сокращения текста путем минимизации числа повторений одного и того же sc-идентификатора путем соединения sc.s-предложений;
- повышение уровня наглядности, "читабельности" sc.s-текстов.


]
⊃
⇒ пример':
[треугольник |- ai >| Треугк(ТчкВ;ТчкС;ТчкD);;
Треугк(ТчкВ;ТчкС;ТчкD) |- bi >| Отр(ТчкВ;ТчкС);;
сторона* |- ci >| bi;;
константнй постояннй sc-узел, обозначающй класс ⊃ треугольник;;
константнй постояннй sc-узел, обозначающй отношение ⊃ сторона*;;
константная постоянная позитивная sc-дуга принадлежности ⊃ ai;;
константная постоянная sc-дуга ⊃ bi;;
константная постоянная позитивная sc-дуга принадлежности ⊃ ci;;
константнй постояннй sc-узел общего вида ⊃ Отр(ТчкВ;ТчкС);;
константнй постояннй sc-узел общего вида ⊃ Треугк(ТчкВ;ТчкС;ТчкD);;]

```

⇔ семантическая эквивалентность*:

[



]

Первое направление расширения Ядра SCs-кода

:= [Первое направление расширения Ядра SCs-кода и всех иных его расширений]
 => принципы*:

[По сравнению с Ядром SCs-кода в Первом направлении расширения Ядра SCs-кода вместо *sc-идентификаторов*, являющихся идентификаторами (именами), которые взаимно однозначно соответствуют синонимичным им (представляемым ими) *sc-коннекторам*, вводятся *sc.s-коннекторы*, каждый из которых соответствует не одному конкретному *sc-коннектору*, а некоторому классу однотипных *sc-коннекторов*. Очевидно, что это ликвидирует необходимость каждому *sc-коннектору* приписывать уникальный *sc-идентификатор*. Кроме того, *Алфавит sc.s-коннекторов* включает в себя элементы этого Алфавита (классы синтаксически эквивалентных *sc.s-коннекторов*), которые соответствуют всем (!) элементам Алфавита *sc-коннекторов*, но при этом дополнительно включают в себя и другие элементы Алфавита *sc.s-коннекторов*, которые соответствуют часто используемым семантически явно выделяемым классам *sc-коннекторов*. К таким дополнительно вводимым классам *sc.s-коннекторов* относятся *константные sc.s-коннекторы* включения множеств ("⊃" или "⊂"), *переменные sc.s-коннекторы* включения множеств ("⊃_" или "_⊃"), *sc.s-коннектор* синонимии ("="), *sc.s-коннектор* погружения ("⊂=" или "⊃=") и др.

Заметим, что указанное расширение Алфавита *sc.s-коннекторов* аналогично расширенному Алфавиту *sc.g-коннекторов* в SCg-коде и ликвидирует необходимость (как и в SCs-коде) явно специфицировать (средствами SCs-кода) синтаксически выделяемые классы *sc.s-коннекторов*.]

Второе направление расширения Ядра SCs-кода

=> принципы*:

[Во Втором направлении расширения Ядра SCs-кода вводятся модификаторы *sc.s-коннекторов* (*sc.s-модификаторы*), которые позволяют достаточно компактно дополнительно специфицировать *sc-коннекторы*, изображаемые (представляемые) соответствующими *sc.s-коннекторами*. Речь идет о такой часто востребованной форме спецификации *sc-коннекторов*, как указание множества (возможно, нескольких множеств), которому принадлежит специфицируемый *sc-коннектор* (чаще всего, таким множеством является *бинарное отношение* (в частности, *ролевое отношение*) или *квазибинарное отношение*).]

sc.s-модификатор*

∈ отношение

:= [относительное понятие]

:= [модификатор *sc.s-коннектора**]

⇒ *пояснение**:

[*sc-идентификатор*, который (1) находится либо между *sc.s-коннектором* и *двоеточием*, либо между *двоеточиями* и (2) обозначает множество (чаще всего, отношение), которому принадлежит *sc-коннектор*, изображаемый ближайшим предшествующим *sc.s-коннектором*. Два подряд идущих двоеточия (":") обозначают, что указанное множество связано с указанным *sc-коннектором* *переменной позитивной постоянной sc-дугой принадлежности*.

Очевидно, что, если не использовать *sc.s-модификаторы*, указанного вида спецификация *sc-коннекторов* средствами *SCs-кода* будет выглядеть значительно более громоздкой.]

Третье направление расширения Ядра SCs-кода

⇒ *принципы**:

[В Третьем направлении расширения Ядра SCs-кода осуществляется переход от использования только простых *sc-идентификаторов* к использованию как простых *sc-идентификаторов*, так и *sc-выражений*, а также к использованию *sc.s-представлений* некоторых *неидентифицируемых sc-узлов*. Это существенно сокращает число придумываемых простых *sc-идентификаторов*, т.к. каждое *sc-выражение* в конечном счете — это комбинация простых *sc-идентификаторов*, построенная по правилам, которые достаточно легко семантически интерпретируются. Если проводить аналогию с SCg-кодом, то очевидно, что *sc-выражение*, ограничиваемое фигурными скобками, есть не что иное, как информационная конструкция, ограничиваемая *sc.g-контуром*, а *sc-выражение*, ограничиваемое квадратными скобками есть не что иное, как информационная конструкция, ограничиваемая *sc.g-рамкой*. Отличие здесь заключается в том, что круглыми и квадратными скобками можно ограничивать только линейные информационные конструкции (цепочки символов).]

sc.s-представление неидентифицируемого sc-узла

:= [изображение (представление) неидентифицируемого (неименоваемого) *sc-узла* в *sc.s-тексте*]

:= [*sc.s-обозначение* неименоваемой сущности, не являющейся парой, обозначаемой *sc-коннектором*]

:= [*sc.s-представление* *sc-узла*, не являющееся *sc-идентификатором* (именем этого *sc-узла*)]

← *разбиение**:

- {
 - *sc.s-обозначение неименоваемой структуры*
:= [конкатенация левой фигурной скобки и правой фигурной скобки]
 - *sc.s-обозначение неименоваемой неориентированной связи*
:= [конкатенация левой фигурной скобки, дефиса и правой фигурной скобки]
 - *sc.s-обозначение неименоваемого кортежа*
:= [конкатенация левой угловой скобки, дефиса и правой угловой скобки]
 - *sc.s-обозначение неименоваемого файла-экземпляра*
:= [конкатенация левой квадратной скобки и правой квадратной скобки]
 - *sc.s-обозначение неименоваемого файла-класса*
:= [конкатенация восклицательного знака, левой квадратной скобки и правой квадратной скобки]
 - *sc.s-обозначение неименоваемой терминальной сущности*
:= [конкатенация левой круглой скобки, буквы "о" и правой круглой скобки]

⇒ *примечание**:

[Если одно и то же обозначение неименоваемой сущности встречается в разных *sc.s-предложениях*, то считается, что это обозначения разных сущностей, т.е. изображения разных *sc-узлов*.]

Четвертое направление расширения Ядра SCs-кода

⇒ *принципы**:

[В Четвертом направлении расширения Ядра SCs-кода осуществляется переход от использования только простых *sc.s-предложений* к использованию также *sc.s-предложений*, построенных с помощью *Операции присоединения sc.s-предложения**. В результате этого, благодаря "склеиванию" одинаковых *sc-идентификаторов*, а также "склеиванию" синтаксически эквивалентных *sc.s-коннекторов* с одинаковыми *sc.s-модификаторами* (несмотря на то, что эти "склеиваемые" *sc.s-коннекторы* соответствуют разным *sc-коннекторам*), существенно сокращается число копий используемых *sc-идентификаторов* и *sc.s-коннекторов* с их *sc.s-модификаторами*.]

Пятое направление расширения Ядра SCs-кода

⇒ *принципы**:

[В Пятом направлении расширения Ядра SCs-кода разрешается использование *присоединенных sc.s-предложений*. В результате этого *sc.s-тексты* становятся более компактными и удобными для восприятия за счет снижения числа дублируемых *sc-идентификаторов* и более широких возможностей их структуризации.]

следует отличать*

- ⊃ {
 - *sc.s-представление неидентифицируемого sc-узла*
 - *sc.s-коннектор*
 := [sc.s-представление неидентифицируемого sc-коннектора]
- ⊃ {
 - *sc-коннектор*
 - *sc.s-коннектор*
- ⊃ {
 - *sc.s-коннектор*
 - *sc.s-модификатор**
 := [модификатор sc.s-коннектора*]
 ∈ отношение
- ⊃ {
 - *sc.s-коннектор*
 - *Правила построения sc.s-коннекторов*
- ⊃ {
 - *sc.s-предложение*
 - *Правила построения sc.s-предложений*
- ⊃ {
 - *sc.s-коннектор*
 - *sc.g-коннектор*
- ⊃ {
 - *sc.s-текст*
 - *sc.g-текст*

Примеры sc.s-текстов, трансформируемых по различным направлениям расширений SCs-кода

⊃=

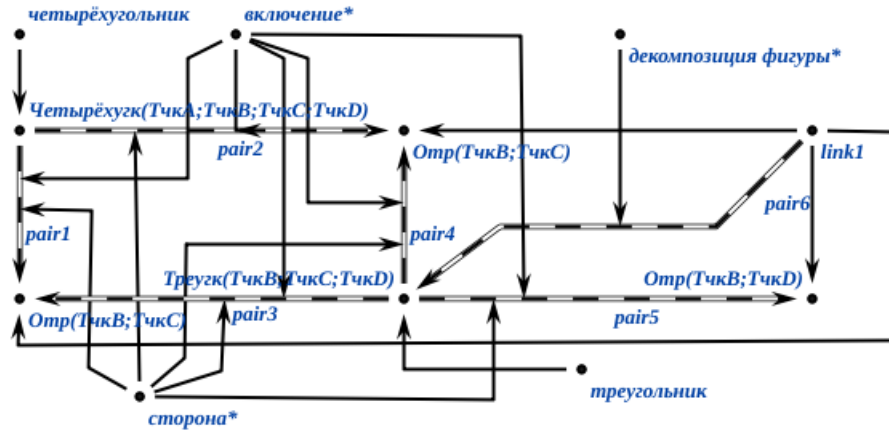
- [*включение** ⊃ *pair1*;;
*включение** ⊃ *pair2*;;
*включение** ⊃ *pair3*;;
*включение** ⊃ *pair4*;;
*включение** ⊃ *pair5*;;
*сторона** ⊃ *pair1*;;
*сторона** ⊃ *pair2*;;
*сторона** ⊃ *pair3*;;
*сторона** ⊃ *pair4*;;
*сторона** ⊃ *pair5*;;
Четырехугол(ТчкА;ТчкВ;ТчкС;ТчкD) ⊃ *pair1* >| *Отр(ТчкВ;ТчкС)*;;
Четырехугол(ТчкА;ТчкВ;ТчкС;ТчкD) ⊃ *pair2* >| *Отр(ТчкС;ТчкD)*;;
Треугол(ТчкВ;ТчкС;ТчкD) ⊃ *pair3* >| *Отр(ТчкВ;ТчкС)*;;
Треугол(ТчкВ;ТчкС;ТчкD) ⊃ *pair4* >| *Отр(ТчкС;ТчкD)*;;
Треугол(ТчкВ;ТчкС;ТчкD) ⊃ *pair5* >| *Отр(ТчкВ;ТчкD)*;;
четырёхугольник ⊃ *Четырехугол(ТчкА;ТчкВ;ТчкС;ТчкD)*;;
треугольник ⊃ *Треугол(ТчкВ;ТчкС;ТчкD)*;;
link1 ⊃ *pair6* >| *Треугол(ТчкВ;ТчкС;ТчкD)*;;
*декомпозиция фигуры** ⊃ *pair6*;;
link1 ⊃ *Отр(ТчкВ;ТчкС)*;;

$link1 \ni Отр(ТчкС;ТчкD);;$

$link1 \ni Отр(ТчкВ;ТчкD);;$]

\Leftrightarrow семантическая эквивалентность*:

[



]

\Rightarrow синтаксическая трансформация*:

[сторона* \ni (Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD) \Rightarrow Отр(ТчкВ;ТчкС));;

сторона* \ni (Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD) \supseteq Отр(ТчкС;ТчкD));;

сторона* \ni (Треугк(ТчкВ;ТчкС;ТчкD) \supseteq Отр(ТчкВ;ТчкС));;

сторона* \ni (Треугк(ТчкВ;ТчкС;ТчкD) \supseteq Отр(ТчкС;ТчкD));;

сторона* \ni (Треугк(ТчкВ;ТчкС;ТчкD) \supseteq Отр(ТчкВ;ТчкD));;

Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD) \supseteq Отр(ТчкВ;ТчкС);;

Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD) \supseteq Отр(ТчкС;ТчкD);;

Треугк(ТчкВ;ТчкС;ТчкD) \supseteq Отр(ТчкВ;ТчкС);;

Треугк(ТчкВ;ТчкС;ТчкD) \supseteq Отр(ТчкС;ТчкD);;

Треугк(ТчкВ;ТчкС;ТчкD) \supseteq Отр(ТчкВ;ТчкD);;

четырёхугольник \ni Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD);;

треугольник \ni Треугк(ТчкВ;ТчкС;ТчкD);;

декомпозиция фигуры* \ni (link1 \Rightarrow Треугк(ТчкВ;ТчкС;ТчкD));;

link1 \ni Отр(ТчкВ;ТчкС);;

link1 \ni Отр(ТчкС;ТчкD);;

link1 \ni Отр(ТчкВ;ТчкD);;

\Rightarrow синтаксическая трансформация*:

[Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD) \supseteq сторона*: Отр(ТчкВ;ТчкС);;

Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD) \supseteq сторона*: Отр(ТчкС;ТчкD);;

Треугк(ТчкВ;ТчкС;ТчкD) \supseteq сторона*: Отр(ТчкВ;ТчкС);;

Треугк(ТчкВ;ТчкС;ТчкD) \supseteq сторона*: Отр(ТчкС;ТчкD);;

Треугк(ТчкВ;ТчкС;ТчкD) \supseteq сторона*: Отр(ТчкВ;ТчкD);;

четырёхугольник \ni Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD);;

треугольник \ni Треугк(ТчкВ;ТчкС;ТчкD);;

link1 \Rightarrow декомпозиция фигуры*: Треугк(ТчкВ;ТчкС;ТчкD);;

link1 \ni Отр(ТчкВ;ТчкС);;

link1 \ni Отр(ТчкС;ТчкD);;

link1 \ni Отр(ТчкВ;ТчкD);;

\Rightarrow синтаксическая трансформация*:

[Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD) \supseteq сторона*: Отр(ТчкВ;ТчкС); Отр(ТчкС;ТчкD);;

Треугк(ТчкВ;ТчкС;ТчкD) \supseteq сторона*: Отр(ТчкВ;ТчкС); Отр(ТчкС;ТчкD); Отр(ТчкВ;ТчкD);;

четырёхугольник \ni Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD);;

треугольник \ni Треугк(ТчкВ;ТчкС;ТчкD);;

link1 \Rightarrow декомпозиция фигуры*: Треугк(ТчкВ;ТчкС;ТчкD);;

link1 \ni Отр(ТчкВ;ТчкС); Отр(ТчкС;ТчкD); Отр(ТчкВ;ТчкD);;

\Rightarrow синтаксическая трансформация*:

[четырёхугольник \ni Четырёхугк(ТчкА;ТчкВ;ТчкС;ТчкD)(* \supseteq сторона*:

Отр(ТчкВ;ТчкС); Отр(ТчкС;ТчкD);; *);;

треугольник \ni Треугк(ТчкВ;ТчкС;ТчкD)(* \supseteq сторона*: Отр(ТчкВ;ТчкС);

Отр(ТчкС;ТчкD); Отр(ТчкВ;ТчкD);; *);;

Треугк(ТчкВ;ТчкС;ТчкD) ← декомпозиция фигуры: link1(*∃ Отр(ТчкВ;ТчкС);
Отр(ТчкС;ТчкD); Отр(ТчкВ;ТчкD);; *);;]*

}

}

/ Завершили Сегмент “Описание Ядра SCs-кода и различных направлений его расширения” */*

}

/ Завершили Раздел “Описание языка линейного представления информационных конструкций в ostis-системах” */*

```

/* Раздел *****/
Описание языка структурированного представления
информационных конструкций в ostis-системах
⊃=
{
SCn-код
:= [Язык структурированного представления знаний ostis-систем]
⇒ пояснение*:
[SCn-код является языком структурированного внешнего представления текстов SC-кода и представляет собой синтаксическое расширение SCs-кода, направленное на повышение наглядности и компактности текстов SCs-кода.

SCn-код позволяет перейти от линейных текстов SCs-кода к форматированным и фактически двухмерным текстам, в которых появляется декомпозиция исходного линейного текста SCs-кода на строчки, размещенные "по вертикали". При этом начало всех строчек текста фиксировано и определяется известным и ограниченным набором правил, что дает возможность использовать это при форматировании sc.n-текста (текста, принадлежащего SCn-коду.)]
∈ язык двухмерных текстов
:= [язык, каждый текст которого задается (1) множеством входящих в него символов, (2) отношением порядка (последовательности) символов по "горизонтали", (3) отношением порядка (последовательности) символов по "вертикали".]
⇒ пояснение*:
[Символ, входящий в состав двухмерного текста, в общем случае может иметь четыре "соседних" символа: (1) символ, находящийся от него слева в рамках той же строчки, (2) символ, находящийся от него справа в рамках этой же строчки, (3) символ, находящийся строго над ним в предыдущей строчке и (4) символ, находящийся строго под ним в следующей строчке текста.]
⇒ сравнительный анализ*:
[Благодаря тому, что в состав sc.n-текстов могут входить и sc.s-тексты, и sc.g-тексты (ограниченные sc.n-контуром), SCn-код можно считать интегратором различных внешних языков представления знаний. Это дает возможность при визуализации и разработке базы знаний ostis-системы недостатки одного из предлагаемых вариантов внешнего представления sc-текстов (SCg-кода, SCs-кода, SCn-кода) компенсировать достоинствами других вариантов.]
{• SCn-код
• SCs-код
}
⇒ описание связи*:
{
SCn-код
⇒ синтаксическое расширение*: синтаксическое ядро языка*:
SCs-код
}
⇒ отличие*:
[Переход от линейности sc.s-текстов к двухмерности sc.n-текстов.]
⇒ уточнение*:
[Важной особенностью SCn-кода является "двухмерный" характер его текстов. Это проявляется в том, что для каждого фрагмента текста SCn-кода важное значение имеет величина отступа от левого края строчки.

В тексте SCn-кода в отличие от текста SCs-кода для каждого фрагмента текста важное значение имеет не только то, как этот фрагмент связан с другими фрагментами "по горизонтали" (какой фрагмент находится левее и какой правее на одной и той же строчке), но и то, как он связан с другими фрагментами "по вертикали" (какой фрагмент находится выше на предыдущей строчке и какой находится ниже на следующей строчке). Так, например, если в тексте SCn-кода некоторый sc-идентификатор (внешний идентификатор sc-элемента) размещен сразу после вертикальной табуляционной линии и точно под ним размещен некоторый sc.s-коннектор, то это означает, что указанный sc-элемент инцидентен sc-коннектору, изображенному указанным sc.s-коннектором.

```

Для того, чтобы обеспечить точное задание(формулировку) правил двухмерной инцидентности элементов(элементарных фрагментов) sc.n-текстов, вводится понятие **страницы sc.n-текста**, понятие **строчки sc.n-текста**, а также используется специальная разметка, представляющая собой вертикальные табуляционные линии, расстояние между которыми примерно равняется максимальной длине sc.s-коннектора (обычно это расстояние равно ширине 4-5 символов).]

sc.n-текст

:= [текст SCn-кода]

:= [последовательность предложений SCn-кода]

:= [последовательность предложений SCn-кода, каждое из которых не является частью какого-либо другого предложения из этой последовательности]

страница sc.n-текста

:= [страница, на которой размещается sc.n-текст]

⇒ *примечание**:

[если sc.n-текст является частью какого-либо другого файла, разделяемого на страницы, например, публикации какой-либо части базы знаний, то sc.n-страницей считается только часть страницы, на которой изображен sc.n-текст, в то время как страница указанного файла может быть больше за счет, например, белых полей по краям страницы, необходимых для последующей распечатки.]

строчка sc.n-текста

⇒ *примечание**:

[Максимальное количество символов в строчках sc.n-текста для каждого sc.n-текста фиксировано и определяется конкретным вариантом размещения sc.n-текста. При этом, в зависимости от отступов в рамках конкретного sc.n-предложения, строчка sc.n-текста может начинаться не с левого края sc.n-текста (но всегда с какой-то из вертикальных линий разметки) и иметь произвольную длину, ограничиваемую правой границей sc.n-страницы.]

линия разметки sc.n-текста

:= [табуляционная линия sc.n-текста]

:= [вертикальная линия разметки sc.n-текста]

:= [вертикальная табуляционная линия]

:= [вертикальная линия, используемая для упрощения восприятия sc.n-текстов и показывающая уровень отступа для компонентов sc.n-предложений]

⇒ *пояснение**:

[1-я линия разметки ограничивает левый край sc.n-страницы, 2-я линия разметки располагается примерно между 5 и 6 символами строчки и т.д. Расстояние между линиями разметки может меняться в зависимости от размера шрифта, однако в рамках одного sc.n-текста всегда остается одинаковым. Общее количество линий разметки ограничивается максимально возможной шириной sc.n-страницы в конкретном файле otis-системы, содержащем данный sc.n-текст.]

следует отличать*

∋ {

- страница sc.n-текста
- строчка sc.n-текста
- строка

}

{

- SCn-код
- SCs-код

}

⇒ *сходство**:

{
Алфавит SCs-кода

← *алфавит**:

- SCs-код
- SCn-код

}

⇔ *семантически эквивалентная информационная конструкция**:

[Алфавит символов *SCs*-кода является также алфавитом символов и *SCn*-кода, т.е. *алфавиты** этих языков совпадают.]

⇒ *сходство**:

[Все компоненты *sc.s*-текстов используются также и в *sc.n*-текстах:

- *sc*-идентификаторы
- *sc.s*-коннекторы
- модификаторы *sc.s*-коннекторов с соответствующими разделителями (двоеточиями)
- разделители, используемые в *sc*-выражениях, обозначающих *sc*-множества, заданные перечислением элементов с соответствующими разделителями (*точкой с запятой* или *круглым маркером*)
- *круглые маркеры* в перечислениях идентификаторов *sc*-элементов, связанных однотипными *sc*-коннекторами с однотипными модификаторами с заданным *sc*-элементом
- разделители предложений (двойные точки с запятой) (опускаются при преобразовании *sc.s*-предложений в *sc.n*-предложения)
- ограничители присоединенных *sc.s*-предложений (опускаются при преобразовании *sc.s*-предложений в *sc.n*-предложения)

]

⇒ *отличие**:

[В отличие от *sc.s*-текстов в *sc.n*-текстах:

- добавляются новые виды *sc*-выражений (а именно – *sc*-выражений, имеющих двухмерный характер);
- добавляется новый вид разделителей предложений – пустая строка;
- меняется размещение предложений с учетом двухмерного характера такого размещения.

]

⇒ *отличие**:

[В *SCn*-коде по сравнению с *SCs*-кодом добавляются новые виды *sc*-выражений:

- *sc*-выражение, представляющее собой двухмерный *sc.n*-текст, ограниченный *sc.n*-контуром или *sc.n*-рамкой. Каждый *sc.n*-контур изображается условно в виде *открывающей фигурной скобки* и расположенной строго под ней через несколько строчек *закрывающей фигурной скобки*. Внутри указанных скобок (начиная от линии вертикальной разметки, на которой расположены сами скобки, и до правого края *страницы*) размещается *sc.n*-текст. Полученный *sc.n*-контур является изображением структуры, являющейся результатом трансляции указанного *sc.n*-текста в *SC*-код. Каждая *sc.n*-рамка изображается аналогичным образом, только вместо *фигурных скобок* в ней используются *квадратные скобки*, либо *квадратные скобки с восклицательным знаком* (в случае файла-образца);
- *sc*-выражение, представляющее собой двухмерный *sc.g*-текст, ограниченный *sc.n*-контуром или *sc.n*-рамкой.
- *sc*-выражение, представляющее собой ограниченное *sc.n*-рамкой двухмерное графическое изображение *информационной конструкции*, закодированной в некотором *файле ostis-системы*. Такой *информационной конструкцией* может быть таблица, рисунок, фотография, диаграмма, график и многое другое.

]

⇒ *примечание**:

[Нетрудно заметить, что *sc.n*-контур является, по сути, двухмерным эквивалентом *sc*-выражения *структуры*, а *sc.n*-рамка – двухмерным эквивалентом *sc*-выражения *внутреннего файла ostis-системы* или *sc*-выражения, обозначающего *файл-образец ostis-системы*.]

***sc.n*-рамка**

:= [ограничитель изображения файла ostis-системы, используемый в *sc.n*-предложениях]

⇒ *пример'*:

*/** начало *sc.n*-рамки **/*

[

]

*/** конец *sc.n*-рамки **/*

⇒ *примечание**:

[С формальной точки зрения *sc.n-рамка* всегда представляет собой одну строку *sc.n-текста*. Это означает, что *sc.n-рамка* не может быть синтаксически разделена на части в рамках того *sc.n-текста*, в котором она используется, и внутри нее не могут вставляться, например, *присоединенные sc.n-предложения* или какой-либо другой текст (за исключением случаев, когда *sc.n-рамка* содержит *sc.n-текст*, но в этом случае указанный *sc.n-текст* все равно будет рассматриваться как целостный внешний файл, а не как фрагмент окружающего его *sc.n-текста*).]

sc.n-контур

:= [используемый в *sc.n-предложениях* ограничитель, являющийся изображением структуры]

⇒ *пример'*:

```
/* начало sc.n-контура */
{
}
/* конец sc.n-контура */
```

- {• *sc.s-предложение*
- *sc.n-предложение*
- }

⇒ *сходство**:

[Понятие *sc.n-предложения* является естественным обобщением понятия *sc.s-предложения*. Более того, аналогичным для *sc.s-предложений* образом вводятся понятия:

- простого sc.n-предложения*
- сложного sc.n-предложения*
- sc.n-предложения, содержащего присоединенные sc.n-предложения*
- sc.n-предложения, не содержащего присоединенные sc.n-предложения*
- присоединенного sc.n-предложения*
- неприсоединенного sc.n-предложения*

]

⇒ *отличие**:

- [Если каждое неприсоединенное *sc.s-предложение* либо является первым предложением *sc.s-текста*, либо начинается после *разделителя sc.s-предложений* (двойной точки с запятой), то каждое *неприсоединенное sc.n-предложение* начинается с начала новой строки]
- [Если каждое присоединенное *sc.s-предложение* начинается либо после открывающего ограничителя присоединенных *sc.s-предложений* (открывающей круглой скобки со звездочкой), либо после *разделителя sc.s-предложений*, то каждое *присоединенное sc.n-предложение* начинается с новой строки под *sc-идентификатором*, которым завершается то *sc.n-предложение* (и соответственно, *sc.s-предложение*), в которое встраивается данное *присоединенное sc.n-предложение*]
- [Первый *sc-идентификатор*, входящий в состав *sc.n-предложения* до *sc.s-коннектора* выделяется жирным курсивом]
- [В *sc.n-предложениях* *двойная точка с запятой* не используется в качестве признака завершения этих предложений и, соответственно, не используется в качестве *разделителя sc.n-предложений*. Таким разделителем является *пустая строка*.]

⇒ *отличие**:

[Благодаря двухмерности *SCn-кода* появляются более широкие возможности (степени свободы) для наглядного и компактного размещения *sc.n-предложений*.]

⇒ *уточнение**:

- [При оформлении *sc.n-предложения* осуществляется четкая табуляция всех присоединенных к нему *sc.n-предложений*, присоединяемых к исходному "по вертикали". Вертикальная линия табуляции задает левую границу исходного (максимального) *sc.n-предложения* или левую границу присоединенного *sc.n-предложения*, присоединяемого "по вертикали". Левая граница *sc.n-предложения* задает начало первого *sc-идентификатора*, входящего в состав этого *sc.n-предложения*, а также начало *sc.s-коннектора*, инцидентного указанному *sc-идентификатору* и размещаемого строго под этим *sc-идентификатором*. Расстояние между вертикальными табуляционными линиями фиксировано и примерно равно максимальной длине *sc.s-коннектора*]

- [В отличие от sc.s-текстов: в sc.n-текстах sc.s-коннектор может быть инцидентен предшествующему sc-идентификатору (как простому, так и sc-выражению) не только "по горизонтали", но и "по вертикали". Для этого sc.s-коннектор размещается строго под предшествующим ему sc-идентификатором]
- [Кроме того "по вертикали" sc-идентификатор может быть инцидентен не одному, а нескольким sc.s-коннекторам, которые последовательно "по вертикали" размещаются под указанным sc-идентификатором. Это позволяет в рамках одного sc.n-предложения представлять произвольное число "ответвлений" от каждого sc-идентификатора, т.е. произвольное число sc.s-коннекторов, инцидентных этому sc-идентификатору]
- [Каждый sc-идентификатор, включая sc-выражение, ограничиваемого фигурными или квадратными скобками, должен размещаться сразу правее вертикальной разметочной линии, если под ним размещается sc.s-коннектор]
- [Каждый sc.s-коннектор выделяется жирным некурсивным шрифтом и, если он находится под инцидентным ему sc-идентификатором, размещается строго между двумя вертикальными разметочными линиями, прижимаясь при этом к левой из этих двух разметочных линий.]

SCn-код

⇒ *правило синтаксической трансформации**:

[Поскольку по отношению к SCn-коду SCs-код является *синтаксическим ядром языка**, SCn-код можно рассматривать как результат интеграции нескольких направлений расширения SCs-кода, в основе которых лежат правила синтаксической трансформации sc.s-текстов и sc.n-текстов, ориентированные на повышение эффективности использования тех возможностей обеспечения наглядности и компактности sc.n-текстов, которые открываются при переходе от линейности sc.s-текстов к двухмерности sc.g-текстов]

sc.n-предложение

⇒ *заданная операция**:

- *Операция преобразования sc.s-предложения в sc.n-предложение**

С *синтаксическая трансформация**

⇒ *пояснение**:

[Каждое *sc.s-предложение*, записываемое линейно ("горизонтально") может быть преобразовано в соответствующее двухмерное *sc.n-предложение*. Перечислим основные правила трансформации sc.s-предложений в sc.n-предложения:

- sc.s-коннектор может размещаться на следующей строчке под предшествующим sc-идентификатором, начиная с того же символа следующей строчки, что и указанный sc-идентификатор;
- если sc-идентификатор переносится на следующую строчку, то его продолжение на следующей строчке осуществляется с таким же отступом от начала строчки, с каким указанный sc-идентификатор начинается;
- перечисление sc-идентификаторов, разделенных точкой с запятой, может осуществляться не "в строчку" , а "в столбик" при размещении каждого следующего sc-идентификатора строго под предшествующим. При этом, разделительная точка с запятой может быть заменена *круглым маркером*, который помещается перед каждым перечисляемым sc-идентификатором;
- закрывающая фигурная или квадратная скобка может быть размещена строго под соответствующей открывающей скобкой;
- sc-идентификатор в sc.n-предложении может быть связан с другими sc-идентификаторами через несколько разных sc.s-коннекторов. При этом, каждый из этих sc.s-коннекторов размещается строго под предшествующим, но только после того, когда будет завершена запись всей, в общем случае разветвленной, цепочки sc.s-коннекторов и sc-идентификаторов, которая начинается с предшествующего sc.s-коннектора. В SCs-коде аналога таким предложениям с неограниченной возможностью описания "разветвленных" связей sc-идентификаторов нет. Следовательно, если в sc.s-тексте sc-идентификатор может быть инцидентен не более, чем двум sc.s-коннекторам (слева и справа от него), то в sc.n-тексте sc-идентификатор может быть дополнительно инцидентен неограниченному числу (причем, не обязательно одинаковых) sc.s-коннекторов, которые размещаются "по вертикали" строго под ним.

]

- *Операция присоединения sc.n-предложения**

С *синтаксическая трансформация**

⇒ *пояснение**:

[Некоторое sc.n-предложение может быть присоединено к другому sc.n-предложению, если в этом другом sc.n-предложении есть sc-идентификатор (но не sc.s-модификатор), с которого начинается первое (присоединяемое) sc.n-предложение. Присоединение происходит следующим образом:

- начальный sc-идентификатор присоединяемого предложения опускается;
- оставшаяся часть sc.n-предложения, начиная от sc.s-коннектора, записывается под таким же sc-идентификатором, но входящим в состав того sc.n-предложения, к которому присоединяется данное sc.n-предложение. С учетом этого смещаются все отступы в присоединяемом sc.n-предложении.

Таким образом может формироваться произвольное число любых разветвлений.]

⇒ *примечание**:

[По сути, семантика sc.n-предложения – множество маршрутов в sc-тексте, возможно пересекающихся и исходящих из заданного sc-элемента]

Описание примеров выполнения операций, заданных на множестве sc.n-предложений

⊃=

{

[Треугк(ТчкВ;ТчкС;ТчкD) ⇒ сторона*: Отр(ТчкВ;ТчкС) (* ∈ отрезок;; *);;]

⇒ Операция преобразования sc.s-предложения в sc.n-предложение*:

[

Треугк(ТчкВ;ТчкС;ТчкD)

⇒ сторона*:

Отр(ТчкВ;ТчкС)

∈ отрезок

]

[

Треугк(ТчкВ;ТчкС;ТчкD)

⇒ сторона*:

Отр(ТчкВ;ТчкС)

Отр(ТчкВ;ТчкС)

∈ отрезок

]

⇒ Операция присоединения sc.n-предложения*:

[

Треугк(ТчкВ;ТчкС;ТчкD)

⇒ сторона*:

Отр(ТчкВ;ТчкС)

∈ отрезок

]

}

}

/* Завершили Раздел “Описание языка структурированного представления информационных конструкций в ostis-системах” */

/ Раздел *****/*

Описание общих принципов оформления внутреннего и внешнего представления информационных конструкций в ostis-системах

⊃=

{

←

конкатенация сегментов:*

- *Формальная типология информационных конструкций, хранимых в памяти ostis-системы*
- *Смысловая типология и спецификация знаний, представленных в памяти ostis-систем*
- *Структуризация баз знаний ostis-систем*
- *Описание правил оформления внешнего представления знаний ostis-систем*

}

/ Сегмент *****/*

Формальная типология информационных конструкций, хранимых в памяти ostis-системы

⊃=

{

информационная конструкция

⇒

разбиение:*

{

- *знак*
 - := [семантически и структурно элементарный фрагмент информационной конструкции]*
 - := [элементарные атомарная информационная конструкция]*
 - := [информационная конструкция, состоящая из одного знака]*
- *знаковая конструкция*
 - := [информационная конструкция, состоящая из нескольких знаков]*

}

⇒

покрытие:*

язык

- := [множество (класс) информационных конструкций,*
 - которому ставится в соответствие*
 - семейство синтаксических правил построения информационных конструкций, принадлежащих этому множеству, а также*
 - семейство семантических правил построения информационных конструкций этого множества*
 - и которому принадлежат не только синтаксически и семантически правильно построенные (корректные) информационные конструкции, но и также неправильно построенные (некорректные, ошибочные) информационные конструкции, которые, тем не менее, дают возможность их семантически интерпретировать (понимать) и, по крайней мере, локализовать допущенные в них ошибки.*

]

⇒ *примечание*:*

[*Очень важно обеспечить эффективный обмен информацией не только с грамотными носителями языков, но также и с не очень грамотными, которые допускают большое количество языковых ошибок.*]

⇒

примечание:*

[*Поскольку теоретически некоторые информационные конструкции могут принадлежать одновременно нескольким языкам и поскольку понятие синтаксической и семантической корректности информационных конструкций определяются правилами соответствующих языков (их синтаксисом и денотационной семантикой), для уточнения понятий структурного уровня и корректности (правильности) информационных конструкций вводится ряд ролевых отношений, связывающих различные языки с принадлежащими им информационными конструкциями.*]

знак'

:= [быть знаком, входящим в состав *информационных конструкций заданного языка'*]

синтаксически выделяемый класс знаков'

:= [быть синтаксически выделяемым *классом знаков* в рамках заданного языка']

∃ *пример'*:

- (*SC-код* ∈ *sc-узел*)
- (*SC-код* ∈ *sc-коннектор*)

⊃ *класс синтаксически эквивалентных знаков'*

класс синонимичных знаков'

:= [быть множеством *знаков заданного языка*, обозначающих одно это уже описываемую *сущность'*]

знаковая конструкция'

:= [быть *знаковой конструкцией* заданного языка']

текст'

:= [быть синтаксически корректной *информационной конструкцией* заданного языка']

синтаксически некорректная информационная конструкция'

:= [быть синтаксически ошибочной *информационной конструкцией* заданного языка']

семантически корректная информационная конструкция'

:= [*информационная конструкция* заданного языка, не имеющая семантических ошибок']

семантически некорректная информационная конструкция'

:= [*информационная конструкция* заданного языка, имеющая семантические ошибки']

знание'

:= [*текст* заданного языка, не имеющий семантических ошибок и обладающий целостностью и ценностью']

следует отличать*

∃ {

- *текст'*
 ∈ *ролевое отношение*
- *текст*
 := [второй домен ролевого отношения *текст'*]
 := [текст некоторого языка]
- *знание'*
 ∈ *ролевое отношение*
- *знание*
 := [второй домен ролевого отношения *знание'*]
- *информационная конструкция*
 ⊃ *текст*
 ⊃ *знание*

}

структура

:= [*sc-конструкция*]

:= [*информационная конструкция SC-кода*]

:= [Множество всевозможных структур]

⊂ *информационная конструкция*

⊃ *sc-текст*

:= [Множество *sc-текстов*]

:= [*текст SC-кода*]

:= [*информационная конструкция*, удовлетворяющая синтаксическим правилам *SC-кода*]

:= [синтаксически корректная для *SC-кода* *информационная конструкция*]

:= [связное множество *sc-элементов*, удовлетворяющее синтаксическим правилам *SC-кода*]

:=	[синтаксически правильно построенная <i>информационная конструкция SC-кода</i>]
C	<i>текст</i>
⊃	<i>sc-знание</i>
:=	[знание, представленное в <i>SC-коде</i>]
:=	[<i>sc-текст</i> , удовлетворяющий определенным семантическим правилам <i>SC-кода</i> и, в частности, определенным правилам семантической полноты]
:=	[семантически корректный и целостный <i>sc-текст</i>]
C	<i>знание</i>

sc-знание

:=	[формализованное знание, хранимое в <i>памяти ostis-системы</i> и непосредственно используемое (понимаемое) её <i>решателем задач</i>]
:=	[<i>sc-текст</i> , имеющий <u>однозначную</u> семантическую интерпретацию в рамках <i>SC-пространства</i> (формализованного смыслового пространства)]
⇒	<i>примечание*</i> : [Не каждый <i>sc-текст</i> является <i>знанием</i> . В отличие от <i>sc-текстов</i> в <i>знаниях</i> важна <u>семантическая целостность</u> (полнота) <i>информационных конструкций</i> . Так, например, <i>логическая формула</i> со свободными переменными не является <i>знанием</i> . Но полный текст <i>высказывания</i> , включающий в себя все компоненты всех логических связок (вплоть до <i>атомарных логических формул</i>) <i>знанием</i> является. Второй пример: <i>sc-текст</i> , в состав которого входит <i>sc-коннектор</i> или <i>связка</i> , но не входят все компоненты этого <i>sc-коннектора</i> или <i>связки</i> , <i>знанием</i> не является.]
⇒	<i>примечание*</i> : [Семантическая целостность <i>знания ostis-системы</i> означает, во-первых, то, что такое <i>знание</i> представляет собой <i>информационную конструкцию</i> , являющуюся высказыванием, то есть информационную конструкцию, имеющую истинностное значение, которое может быть подтверждено или опровергнуто, например, экспертом (рецензентом) <i>базы знаний ostis-системы</i> . Во-вторых, <i>знание ostis-системы</i> должно содержать достаточно полную и <u>однозначную</u> спецификацию по возможности всех входящих в него неидентифицированных (неименованных) <i>sc-элементов</i> . Это необходимо для того, чтобы внешнее представление <i>знания ostis-системы</i> можно было по возможности однозначно погрузить ("вставить") в <i>базу знаний ostis-системы</i> .]

информация, представленная в памяти ostis-системы

:=	[<i>информационная конструкция</i> , хранимая в <i>памяти ostis-системы</i>]
⇒	<i>разбиение*</i> : { <ul style="list-style-type: none"> • <i>структура</i> := [информация, представленная в виде конструкции <i>SC-кода</i>] • <i>файл ostis-системы</i> }
⇒	<i>пояснение*</i> : [В <i>ostis-системе</i> используются две формы представления информации в <i>памяти системы</i> – <ul style="list-style-type: none"> □ полностью формализованное представление, понятное для решателя задач <i>ostis-системы</i> (конструкции <i>SC-кода</i>); □ инородное для <i>SC-кода</i> представление, используемое исключительно для коммуникации <i>ostis-систем</i> с внешними субъектами (файлы <i>ostis-систем</i>).]

файл ostis-системы

:=	[файл, хранимый в <i>sc-памяти</i>]
:=	[инородное для <i>SC-кода</i> <i>информационная конструкция</i> , хранимая в <i>памяти ostis-системы</i> в виде содержимого <i>sc-узла</i> , обозначающего эту <i>информационную конструкцию</i>]
:=	[хранимая в <i>sc-памяти</i> "электронная" форма представления <i>информационной конструкции</i>]
⊃	<i>фрагмент знака, представленный файлом ostis-системы</i>
⊃	<i>синтаксически элементарный фрагмент</i> <i>информационной конструкции</i> , представленный файлом <i>ostis-системы</i>
⊃	<i>знак, представленный файлом ostis-системы</i>
⊃	<i>sc-идентификатор</i> , представленный файлом <i>ostis-системы</i>
⊃	<i>знаковая конструкция, представленная файлом ostis-системы</i>
⊃	<i>текст</i> , представленный файлом <i>ostis-системы</i>
⊃	<i>знание</i> , представленное файлом <i>ostis-системы</i>

знаковая конструкция, представленная файлом ostis-системы

⊃	<i>sc.s-файл ostis-системы</i>	:= [конструкция SCs-кода, хранимая в памяти ostis-системы в виде содержимого некоторого SC-узла]
		:= [файл ostis-системы, являющийся sc.s-конструкцией]
⊃	<i>sc-идентификатор, представленный файлом ostis-системы</i>	:= [файл ostis-системы, являющийся sc-идентификатором]
		:= [(sc-идентификатор ∩ файл ostis-системы)]
	← <i>пересечение*</i> :	{
		• <i>sc-идентификатор</i>
		• <i>файл ostis-системы</i>
		}
⊃	<i>sc.g-файл ostis-системы</i>	← <i>пересечение*</i> :
		{
		• <i>sc.g-конструкция</i>
		• <i>файл ostis-системы</i>
		}
⊃	<i>sc.n-файл ostis-системы</i>	← <i>пересечение*</i> :
		{
		• <i>sc.n-конструкция</i>
		• <i>файл ostis-системы</i>
		}
⊃	<i>ея-файл ostis-системы</i>	← <i>пересечение*</i> :
		{
		• <i>ея-конструкция</i>
		:= [естественно-языковая конструкция]
		:= [конструкция естественного языка]
		⊃ <i>ея-текст</i>
		• <i>файл ostis-системы</i>
		}
		:= [конструкция естественного языка, представленная в виде файла ostis-системы]
	⊃ <i>Русский язык</i>	:= [конструкция Русского языка]
	⊃ <i>Английский язык</i>	
		:= [естественно-языковая конструкция, являющаяся содержимым sc-узла, обозначающего эту конструкцию]

файл ostis-системы

⇒	<i>разбиение*</i> :	{
		• <i>файл ostis-системы, предполагающий одномерную визуализацию хранимой информации</i>
		• <i>файл ostis-системы, предполагающий двухмерную визуализацию хранимой информации</i>
		• <i>файл ostis-системы, предполагающий трехмерную визуализацию хранимой информации</i>
		}
⇒	<i>разбиение*</i> :	{
		• <i>файл ostis-системы, представляющий статическую информацию</i>
		• <i>файл ostis-системы, представляющий динамическую информацию</i>
		⊃ <i>видео-файл ostis-системы</i>
		⊃ <i>аудио-файл ostis-системы</i>
		}
⇒	<i>разбиение*</i> :	{
		• <i>файл-экземпляр ostis-системы</i>
		:= [sc-узел, обозначающий конкретное вхождение информационной конструкции, структура которой представлена содержимым этого sc-узла]
		• <i>файл-образец ostis-системы</i>
		:= [класс синтаксически эквивалентных файлов-экземпляров ostis-системы]
		:= [множество всевозможных файлов-экземпляров ostis-системы, которые являются синтаксически эквивалентными копиями содержимого заданного sc-узла]
		}
⇒	<i>разбиение*</i> :	{
		• <i>сформированный файл ostis-системы</i>

- := [файл *ostis-системы*, представленный *sc-узлом*, имеющим сформированное (полностью построенное) содержимое]
- *несформированный файл ostis-системы*
- := [файл *ostis-системы*, представленный *sc-узлом*, содержимое которого либо полностью отсутствует, либо сформировано частично]

}

⇒ *примечание**:

[файл *ostis-системы* может быть электронной копией и знаком (!) внешней *информационной конструкции*, которая может быть:

- общедоступный в сети Internet информационным ресурсом;
- документом, опубликованным на бумажном носителе в виде какой-либо книги или статьи.

Кроме того, *файл ostis-системы* может быть просто обозначением указанной внешней *информационной конструкции* (т.е. может быть *sc-узлом*, обозначающим *внешнюю информационную конструкцию*, но не имеющим содержимого). Такой *sc-узел* используется формальной спецификации (средствами *SC-кода*) соответствующего обозначаемого им информационного ресурса.]

ея-файл ostis-системы

:= [естественно-языковой *файл ostis-системы*]:= *пояснение**:

[файл *ostis-системы*, представляющий собой *sc-узел*, содержимым которого является "электронная" форма *информационной конструкции* (чаще всего, *текста*) одного из *естественных языков*]

:= [структурно выделенный *ея-текст*, хранимый в памяти *ostis-системы* (в *sc-памяти*)]⇒ *правила оформления**:

- [При оформлении текстов в естественно-языковых файлах (*ея-файлах*) *ostis-систем* используются обычные разделители (точки в аббревиатурах и между предложениями; круглые скобки, запятые, пробелы;), а также специальный символ □, используемый в *ея-файлах ostis-систем* как разделитель и целый ряд следующих ограничителей, позволяющих выделять некоторые фрагменты *ея-текстов*:
 - подчеркивание выделяет логически важные фрагменты в предложениях;
 - цитатные кавычки являются ограничителем цитат;
 - прямые кавычки ограничивают иносказательные термины, метафоры;
 - жирным курсивом стандартного размера выделяются идентификаторы *sc-элементов* базы знаний, являющиеся ключевыми для заданного контекста;
 - жирным курсивом стандартного размера выделяются также условные внешние идентификаторы (обозначения) условно вводимых *sc-элементов*, например, условные обозначения *sc-элементов* произвольного структурного типа (e_i, e_j, \dots), условные обозначения *sc-узлов* (v_i, v_j, \dots), условные обозначения *sc-коннекторов* (c_i, c_j, \dots), *sc-дуг, связок*, не являющихся *sc-коннекторами, структур* и так далее;
 - нежирным курсивом стандартного размера выделяются *sc-идентификаторы sc-элементов базы знаний*, не являющиеся ключевыми для данного *ея-текста*.

]

- [Если в *ея-тексте* идентификаторы *sc-элементов* базы знаний (чаще всего – *простые sc-идентификаторы*), выделенные курсивом одинаковой жирности, следуют друг за другом, то число пробелов между разными идентификаторами должно быть увеличено. Если при этом выделенный *sc-идентификатор* включает в себя другие *sc-идентификаторы*, на которые желательно отдельно сослаться, то такая ссылка оформляется в виде дополнительной фразы типа "Смотрите также ..."]
- [Текст *ея-файла ostis-системы* может иметь любые вставки не являющиеся естественно-языковыми текстами, в том числе, и фрагменты, являющиеся формальными внешними текстами представления знаний для *ostis-систем* (текстами *SCg-кода*, *SCs-кода*, *SCn-кода*). При этом указанные формальные фрагменты (вставки) могут быть как транслируемыми на внутренний язык *ostis-системы* (*SC-код*) и погружаемыми в состав ее базы знаний (т.е. фактически являться *sc-текстами*), так и нетранслируемыми формальными фрагментами, которые входят в состав базы знаний *ostis-системы* в виде содержимого соответствующих файлов. Все указанные выше "вставки" в *ея-файл ostis-системы* оформляются как ссылки на соответствующие *sc-тексты* или файлы *ostis-системы*. Каждая такая ссылка представляет собой *sc-идентификатор* соответствующего *sc-текста* или файла *ostis-системы* и выделяется в *ея-файле* жирным курсивом стандартного размера со стандартным расстоянием между символами. Таким образом, если в естественно-языковой *файл ostis-системы*, необходимо "вставить" *информационную конструкцию* иного рода (*sc.g-текст*, рисунок, таблицу, изображение), то (1) указанная конструкция оформляется как отдельный файл (2) которому приписывается имя

(название), построенное по установленным правилам, и (3) на который в указанном естественно-языковом файле делается ссылка. В естественно-языковых *файлах ostis-систем* можно делать ссылки не только на другие *файлы ostis-системы*, но и на именуемые (!) фрагменты базы знаний, которые во внешнем представлении базы знаний оформляются в виде именуемых (идентифицируемых) *sc.n-контуров*. Файлы и *sc-тексты*, на которые делаются ссылки из *ея-файла*, во внешнем представлении (при визуализации) базы знаний размещаются после указанного *ея-файла* в порядке первого их упоминания в этом *ея-файле*, если, конечно, на эти файлы или *sc-тексты* не было ссылок из ранее представленных *ея-файлов*.]

- [В состав *ея-файла ostis-системы* могут входить ссылки на любые идентифицированные (именованные) *информационные конструкции* (Internet-ресурсы, библиографические источники, различные документы). Для этого достаточно указывать соответствующие *sc-идентификаторы*.]
- [В *ея-текстах* все основные *sc-идентификаторы* описываемых в базе знаний сущностей должны быть выделены жирным или нежирным курсивом и могут быть представлены в любом склонении и спряжении.]
- [В *ея-текстах* используются только основные идентификаторы (термины). Используемые синонимы явно указываются как неосновные идентификаторы.]
- [Основная часть (содержимого текста) *ея-файла ostis-системы* оформляется стандартным печатным шрифтом.]

}

/* Завершили перечень правил оформления содержимого *ея-файлов ostis-систем* */

ея-файл ostis-системы

⇒ *примечание**:

[Выделенные курсивом в *ея-файле ostis-системы sc-идентификаторы sc-элементов* могут являться аргументами различного вида запросов к базе знаний ostis-системы и, в первую очередь запросов типа "Что это такое", предполагающих выделение из текущего состояния базы знаний ostis-системы семантической окрестности (спецификации) указываемого *sc-элемента*, содержащей основную информацию о сущности, обозначаемой этим sc-элементом.]

}

/* Завершили Сегмент "Формальная типология информационных конструкций, хранимых в памяти *ostis-системы*" */

/* Сегмент *****/

Смысловая типология и спецификация знаний, представленных в памяти ostis-систем

▷=

{

знание, представленное в памяти ostis-системы

▷ *предметная область*

:= [sc-модель предметной области]

▷ *sc-метазнание*

:= [sc-знание, являющееся спецификацией некоторого другого знания, представленного в памяти *ostis-системы*]

▷ *онтология предметной области*

:= [семантическая спецификация предметной области]

▷ *sc-знание "дидактического" назначения*

⇒ *покрытие**:

- {• спецификация "дидактического" файла *ostis-системы*

⇒ *примечание**:

[Такая спецификация описывает семантические связи специфицируемого "дидактического" файла *ostis-системы* с базой знаний этой системы]

- *sc-знание, описывающее различные сущности в сравнительном аспекте*

}

⇒ *эпиграф**:

[Сущности познаются в связях между собой и в сравнении друг с другом]

▷ "дидактический" файл *ostis-системы*

⇒ *примечание**:

[Для знания, представленного *файлом ostis-системы*, в *базе знаний* этой *ostis-системы* может присутствовать *семантически эквивалентное** этому файлу знание, представленное *текстом SC-кода*. Это является вариантом описания семантической интерпретации некоторых фрагментов *базы знаний*. Кроме того, *знание*, представленное *файлом ostis-системы*, может быть предварительным этапом формализации этого *знания*, предполагающим последующую трансляцию этого *знания* на *SC-код*. Использование таких файлов является важнейшим механизмом коллективной разработки *баз знаний ostis-систем*. Следует также заметить, что некоторые *знания*, представленные *файлами ostis-систем*, не требуют трансляции на *SC-код*, а носят вспомогательный характер, позволяющий разработчикам и конечным пользователям *ostis-систем* ускорить процесс установления семантического контакта (взаимопонимания) с *ostis-системами*.]

▷ *пояснение, представленное файлом ostis-системы*

▷ *примечание, представленное файлом ostis-системы*

▷ *сравнение, представленное файлом ostis-системы*

предметная область

⇒ *пояснение**:

[Для эффективной коллективной разработки и эксплуатации *базы знаний ostis-системы* важна не просто её структуризация, а такая структуризация, которая носит максимально объективный характер, имеющая четкую семантическую интерпретацию (!) и позволяющая на основе семантических (!) связей между структурно выделяемыми фрагментами *базы знаний* легко определять (локализовывать) "местоположение" либо искомого *знания*, либо новых *знаний*, вводимых в состав *базы знаний*. Такая семантическая структуризация *базы знаний*, формирование системы семантически связанных между собой "семантических полочек", на которых размещаются конкретные *знания*, удовлетворяющие четко заданным требованиям, существенно упрощает навигацию по *базе знаний* и четко локализует эволюцию *знаний*, находящихся на каждой "семантической полочке". В основе указанной семантической структуризации *базы знаний* лежит иерархическая система *предметных областей*. С содержательной точки зрения *предметная область* представляет собой совокупность фактографических *высказываний*, описывающих все (!) элементы заданного множества объектов исследования¹ с помощью заданного набора *отношений* и *параметров* (характеристик). Максимальный класс указанных *объектов исследования*, некоторые специально выделяемые подклассы этого класса, а также указанные *отношения* и *параметры* задают систему *понятий*, лежащих в основе *предметной области*, которую будем называть *схемой предметной области*. Подчеркнем то, что *схема предметной области*, определяющая семантическую "систему координат" в рамках соответствующей *предметной области*, в известной степени имеет субъективный характер и является предметом соглашения (!) (консенсуса) между специалистами в этой области. Подчеркнем также, что *схема предметной области* может эволюционировать (может меняться набор понятий, может уточняться их денотационная семантика). Понятие *предметной области* можно считать обобщением понятия *алгебраической системы*, ориентированным на решение проблемы семантической структуризации *базы знаний*. В *памяти ostis-системы* предметная область представляется обычно бесконечной *sc-структурой*, которая задается (1) неким множеством исследуемых *сущностей*, (2) семейством *отношений*, *алгебраических операций* и *параметров* (свойств), заданных на множестве исследуемых *сущностей*, каждое из которых либо рассматривается в рамках данной *предметной области*, либо "наследуется" из другой *предметной области* более высокого уровня. Каждая *предметная область* в *базе знаний* может быть представлена своим *разделом*. При этом на множестве таких *разделов* могут быть заданы не только рассмотренные выше "синтаксические" *отношения*, но и целый ряд семантически интерпретируемых *отношений*, например, отношение "быть *частной предметной областью**". Примером *связки* этого *отношения* является *связка* между *Предметной областью геометрических фигур* в Евклидовом пространстве и *Предметной областью планарных фигур* Евклидова пространства. Таких *отношений*, заданных на множестве *предметных областей* и уточняющих характер соотношения между множествами исследуемых *сущностей*, а также рассматриваемыми или "наследуемыми" *отношениями*, *алгебраическими операциями* и *параметрами*, существует достаточно много. Но, если к этому добавить анализ соотношения не только между самими *предметными областями*, но и соотношения между представляющими их *sc-структурами* в рамках *базы знаний* конкретной *ostis-системы*, а также в рамках глобального смыслового пространства (*SC-пространства*), то число семантически интерпретируемых *отношений*, заданных на множестве *предметных областей* существенно расширится. К числу важных *отношений*, заданных на множестве *предметных областей* относятся также *отношения*, связывающие *предметные области* с различными структурно выделяемыми фрагментами *базы знаний*, которые *предметными областями* не являются. Ключевым *отношением* такого вида является *Отношение, связывающее предметные области с соответствующими им онтологиями* (Отношение "быть *онтологией**").]

⇒ *правила построения**:

- { • [Понятия и соответствующие им термины в *базе знаний ostis-системы* группируются по четко выделенным *предметным областям*. При этом связи между *предметными областями* явно (!) указываются.]
- [Изложение материала в *базе знаний ostis-системы* построено по принципу "сверху вниз", переходя от общих *предметных областей и онтологий* к частным для явного указания направления наследования свойств.]
- [Во всех представляемых и описываемых *предметных областях* необходимо явно указывать все (!) *понятия*, используемые в *предметных областях* с указанием каждой роли этих *понятий* в рамках *предметной области*. К таким ролям относятся: *максимальный класс объектов исследования'*; *класс объектов исследования'*; *исследуемое отношение'*, заданное на объектах исследования; *вспомогательное понятие'*, используемое в описываемой предметной области, но исследуемое в других предметных областях).]

онтология предметной области

⇒ *пояснение**:

[Все рассматриваемые в *базе знаний ostis-системы* *предметные области* должны быть формально специфицированы, т.е. для каждой из них должна быть построена и указана соответствующая ей *онтология* и, в частности, должны быть указаны все известные связи с другими *предметными областями* и с иными фрагментами *базы знаний ostis-системы*. Можно выделить целый ряд частных онтологий, описывающих свойства соответствующих *предметных областей* с разных "ракурсов".]

⊃ *схема предметной области*

:= [спецификация внутренней структуры *предметной области*]

:= [перечень всех *понятий* и некоторых *ключевых объектов исследования*, лежащих в основе специфицируемой *предметной области*, с указанием их роли в рамках этой *предметной области*]

⊃ *спецификация внешних связей предметной области*

:= [описание связей *предметной области* с *онтологиями* и с другими *предметными областями*]

⊃ *онтология определений исследуемых понятий*

⊃ *логическая онтология предметной области*

:= [онтология аксиом, теорем и текстов доказательств теорем]

⊃ *онтология классов задач предметной области и методов их решения*

:= [онтология задач, решаемых в рамках предметной области, объединенной с соответствующей онтологией]

⊃ *терминологическая онтология предметной области*

⊃ *онтология эволюции предметной области и соответствующей ей онтологии*

спецификация "дидактического" файла

⇒ *используемое понятие**:

- *иллюстрация*
- *иллюстрация**
- *пояснение*
- *пояснение**
- *примечание*
- *примечание**
- *закономерность**

:= [закономерность, описывающая некоторое свойство экземпляров заданного понятия*]

:= [утверждение*]

- *определение**

- *однозначная спецификация**

- *упражнение**

:= [упражнение, относящееся к заданной сущности или к заданному фрагменту базы знаний*]

- *вопрос для самопроверки**

:= [вопрос, относящийся к заданной сущности или к заданному фрагменту базы знаний*]

- *сравнение*

- *сравнение**

:= [быть сравнительным анализом двух заданных сущностей*]

- *сходство**

:= [быть описанием сходства двух заданных сущностей*]

- *отличие**
:= [быть описанием одного из различий двух заданных сущностей*]
- *спецификация типичного примера**
:= [быть спецификации типичного примера (экземпляра) заданного класса*]
- *спецификация типичного подкласса заданного класса**
- *пример использования экземпляра заданного класса**
- *используемое понятие**
:= [понятие, используемое в заданной информационной конструкции или в заданном классе информационных конструкций*]
- *принцип, лежащий в основе**
:= [основное свойство (ключевое положение), характеризующее заданную сущность*]
- *обоснование целесообразности создания**
- *требование**
:= [требование, предъявляемое к заданной сущности*]
- *правило построения**
- *достоинство**
- *недостаток**
- *сравнительный анализ**
- *логическое следствие**
:= [следовательно*]
:= [быть ориентированной парой двух знаний ostis-системы, второе из которых логически следует из первого*]
:= [что из этого логически следует*]
- *логическое обоснование*
- *последствие**
:= [быть ориентированной парой двух знаний ostis-системы, первое из которых является описанием ситуации, являющейся причиной (предпосылкой, достаточным условием) возникновения ситуации, которая описывается вторым из указанных знаний ostis-системы*]
:= [причинно-следственная связь*]
⇔ *обратное отношение*:*
*причина**
:= [предпосылка*]
:= [ситуация, являющаяся достаточным условием*]
- *семантическая эквивалентность**
∈ *бинарное неориентированное отношение*
:= [бинарное неориентированное отношение, каждая пара которого связывает знаки двух семантически эквивалентных *информационных конструкций*, которые могут *принадлежать разным языкам*]
⇒ *пояснение*:*
[*семантическая эквивалентность** двух *информационных конструкций* означает то, что эти *информационные конструкции*:
□ оперируют (явно или неявно) знаками одних и тех же сущностей и
□ описывают одни и те же связи между этими *сущностями*
]
- *синтаксическая эквивалентность**
- *семантическое включение**

sc-знание, описывающее различные сущности в сравнительном аспекте

- ⇒ *используемое понятие*:*
- *следует отличать**
 - *сходство**
 - *отличие**
 - *пример'*
 - *ключевой знак'*
 - *спецификация**
:= [быть спецификацией (описанием) заданной сущности*]
 - *аналог**
:= [быть аналогом заданной сущности*]
 - *изоморфность**


```

•   изоморфизм*
•   гомоморфность*
•   гомоморфизм*
}   /* Завершили Сегмент “Смысловая типология и спецификация знаний, представленных в памяти ostis-
    систем” */

/* Сегмент *****/
Структуризация баз знаний ostis-систем
⊃=
{
С   сегмент базы знаний
:=  [Структурная типология знаний ostis-системы]
⇒  введение*:
    [База знаний ostis-системы имеет достаточно развитую иерархическую структуру. База знаний делится
    на разделы. Разделы бывают атомарными и неатомарными. Неатомарный раздел состоит из сегментов.
    Атомарные разделы не имеют сегментов. Разделы базы знаний ostis-системы могут иметь самое различное
    назначение. Так, например, База знаний Метасистемы IMS.ostis включает в себя:
    □ раздел, содержащий текущее состояние постоянно пополняемого и совершенствуемого Стандарта
    Технологии OSTIS;
    □ раздел, посвящённый описанию конечных пользователей и разработчиков Метасистемы IMS.ostis;
    □ раздел, посвящённый описанию истории эксплуатации Метасистемы IMS.ostis;
    □ раздел, посвящённый описанию истории эволюции Метасистемы IMS.ostis (в т.ч. истории эволюции и
    её базы знаний);
    □ раздел посвящённый описанию интеллектуальных компьютерных систем, разработанных (порождённых)
    с помощью Метасистемы IMS.ostis.
    ]
выделенный фрагмент базы знаний
:=  [фрагмент базы знаний, для которого в базе знаний вводится знак, обозначающий этот фрагмент, т.е.
    являющийся знаком множества всех знаков, входящих в состав этого фрагмента. При представлении
    фрагмента базы знаний на внешних языках (SCg-коде, SCn-коде) указанный знак выделенного фрагмента
    базы знаний представляется либо в виде sc.g-контура, либо в виде пары фигурных скобок, ограничивающих
    текст обозначаемого фрагмента базы знаний]
:=  [выделенный фрагмент базы знаний]
∈   сокращённые sc-идентификатор
:=  [явно структурно выделенный фрагмент базы знаний]
⇒  примечание*:
    [явное выделение фрагмента базы знаний осуществляется:
    □ в SC-коде путем введения знака, обозначающего множество всех знаков, входящих в состав выделенного
    фрагмента базы знаний;
    □ в SCg-коде с помощью sc.g-контура, ограничивающего sc.g-представление выделенного фрагмента
    базы знаний;
    □ в SCn-коде с помощью пары фигурных скобок, ограничивающих sc.n-представление выделенного
    фрагмента базы знаний.
    ]
⇒  разбиение*:
    {
    •   семейство разделов базы знаний
        :=  [семантически целостное множество разделов базы знаний, имеющих достаточно сильные
            семантические связи между собой]
    •   раздел базы знаний
        :=  [Основной (по семантической значимости) вид выделенных фрагментов баз знаний]
        :=  [раздел базы знаний ostis-системы]
        :=  [модуль (блок) базы знаний]
    }

```

- ⇒ *примечание**:
[В общем случае многим *разделам базы знаний* ставятся в соответствие такие тексты, как предисловие, введение, заключение, аннотация, оглавление, упражнения.
Некоторые из этих текстов могут иметь статус разделов.]
- ⇒ *разбиение**:
{
 - *неатомарный раздел базы знаний*
:= [раздел базы знаний, состоящий из сегментов, декомпозируемый на сегменты]
⇒ *примечание**:
[В общем случае *неатомарный раздел базы знаний* может иметь неограниченное число *сегментов*. *Сегменты базы знаний* не могут состоять из других сегментов (подсегментов). В этом смысле *сегменты базы знаний* имеют атомарный характер.]
 - *атомарный раздел базы знаний*
:= [раздел базы знаний, не содержащий сегментов]
}
- *сегмент базы знаний*
:= [сегмент раздела базы знаний]
:= [сегмент базы знаний *ostis-системы*]
:= [структурно выделяемое *sc-знание ostis-системы*, структурный уровень которого ниже уровня разделов базы знаний]
⇒ *примечание**:
[Сегменты базы знаний не могут иметь иерархической структуры, т.е. не могут состоять из сегментов более низкого структурного уровня.]
⇒ *примечание**:
[Сегменты базы знаний входят в состав неатомарных разделов базы знаний.]
 - *выделенный фрагмент сегмента или атомарного раздела базы знаний*
⇒ *разбиение**:
{
 - *выделенный фрагмент атомарного раздела базы знаний*
 - *выделенный фрагмент сегмента базы знаний*
}
- C *sc-знание*
⇒ *примечание**:
[Каждый *выделенный фрагмент базы знаний* представляет собой структурно оформленное (структурно выделенное) *знание*, хранимое в *базе знаний интеллектуальной компьютерной системы*, (точнее, в *базе знаний ostis-системы*) и представленное в формализованном виде на *внутреннем языке представления знаний* (в *SC-коде*) Представленное таким образом *знание* будем называть *sc-знанием*.]
:= *пояснение**:
[*информационная конструкция*, которая:
 - принадлежит *SC-коду*;
 - является синтаксически корректный;
 - обладает семантической целостностью – отсутствием *информационных дыр* ("недомолвок"), препятствующих её пониманию;
 - имеет нетривиальный *объём информации* – количество описываемых сущностей (в том числе, *связей* и *классов*);
 - имеет достаточно высокое качество по другим характеристикам, в частности, достаточно высокую ценность.
]
- ⇒ *примечание**:
[Типология *sc-знаний* по объему представленной информации определённым образом коррелирует с типологией *выделенных фрагментов баз знаний* – объём информации, содержащейся в *разделах баз знаний*, должен быть приблизительно одинаковым, объём, содержащейся в разделе базы знаний, должен быть ниже объёма информации, содержащегося в любом семействе разделов базы знаний, и должен быть выше объёма информации, содержащегося в любом *сегменте базы знаний*. При этом в процессе эволюции базы знаний *раздел базы знаний* может преобразоваться в семейства разделов, а *сегмент базы знаний* может преобразоваться в *раздел базы знаний*.]

раздел базы знаний⇒ *примечание**:

[Различные *предметные области*, различные *онтологии*, а также предметные области, объединённые с соответствующими им онтологиями, являются важнейшими видами *знаний ostis-систем*, обеспечивающими логически стройную систематизацию *sc-знаний ostis-систем* и, соответственно семантическую структуризацию *баз знаний*. При этом указанные типы знаний обычно представляются в виде *разделов базы знаний*, иерархия которых, задаваемая Отношением *частная предметная область** или Отношением *частная предметная область и онтология**, соответствует логико-семантической иерархии *предметных областей*. Но, кроме *разделов базы знаний*, являющихся *предметными областями, онтологиями, предметными областями*, объединёнными с соответствующими им онтологиями, вводится и целый ряд других семантических типов *разделов базы знаний*, определяемых характером соотношения разделов базы знаний с используемыми (рассматриваемыми) предметными областями и онтологиями.]

⊃ *предметная область*⊃ *фрагмент предметной области*⊃ *интегрированная онтология*⊃ *частная онтология*⊃ *предметная область и онтология*

:= [предметная область, объединенная (интегрированная) с соответствующей ей онтологией]

:= [предметная область вместе с онтологией, которая её специфицирует]

⇒ *примечание**:

[Если каждому *разделу базы знаний ostis-системы* будет четко соответствовать его семантический тип, то к "синтаксическим" связям между *разделами базы знаний* добавится большое количество "осмысленных" (семантические интерпретируемых) связей, определяющих "семантическое местоположение" ("семантические координаты") каждого *раздела базы знаний* во множестве всех разделов, входящих в состав базы знаний *ostis-системы*.]

⇒ *примечание**:

[В основе представления *базы знаний ostis-системы* лежат развитые средства семантической структуризации *баз знаний* и семантической систематизации *баз знаний ostis-систем*. Можно выделить следующие уровни систематизации элементов и фрагментов смыслового пространства, построенного на основе *SC-кода*:

- уровень знаков всевозможных сущностей (уровень *sc-элементов*);
- уровень вводимых *понятий*, обозначающих ключевые (исследуемые) в предметных областях классы сущностей;
- уровень *высказываний*, описывающих закономерности (свойства) экземпляров исследуемых классов сущностей (исследуемых понятий);
- уровень *предметных областей, онтологий* и разделов, семантический тип которых известен.

]

:= [раздел б.з.]

∈ *сокращенный sc-идентификатор*⇒ *пояснение**:[Множество *разделов баз знаний* имеет:

- богатую семантическую типологию;
- богатый набор отношений, описывающих семантические связи между разделами.

Синтаксически *разделы баз знаний* могут пересекаться (иметь общие элементы), но никогда один раздел не может включаться (полностью входить в состав) другого раздела. В этом смысле понятие подраздела (точнее, *частного раздела**) имеет не "синтаксический" смысл, а семантический – глубокое наследование свойств при достаточно большой степени "независимости" друг от друга]

⇒ *примечание**:

[Важным свойством *раздела базы знаний ostis-системы* является его семантическая целостность – наличие достаточно стабильного набора классов исследуемых сущностей (*объектов исследования*) и достаточно стабильного семейства *отношений* и семейства *параметров*, заданных на различных классах объектов исследования, а также семейства *классов структур* использующих указанные выше понятия (введенные *классы объектов исследования, введенные отношения и классы структур*). Такая целостность дает возможность развивать *раздел базы знаний*, не выходя "за рамки" используемой системы *понятий*. Это позволяет развивать каждый *раздел базы знаний* в известной мере независимо от других разделов, что существенно повышает гибкость и стратифицированность *базы знаний*.]

- ⇒ *пояснение**:
 [Основными свойствами *раздела базы знаний* как структурно выделяемого фрагмента базы знаний являются следующие:
- семантическая целостность – наличие четкого критерия, позволяющего установить для каждого конкретного знания то, включать или не включать это знание в состав данного раздела;
 - потенциальная возможность эволюционировать в достаточной степени независимо от других разделов, но при условии соблюдения всех требований, обеспечивающих постоянную поддержку семантической совместимости данного раздела со всеми остальными семантически смежными разделами.
-]

⇒ *следовательно**:
 [Грамотная декомпозиция *базы знаний* на разделы, основанная на четкой стратификации процесса эволюции накапливаемой человечеством общечеловеческой объединенной *базы знаний*, сутью которой является минимизация трудоемкости усилий по согласованию и обеспечению с авторами других разделов *семантической совместимости* со смежными разделами, создает предпосылки высоких темпов эволюции *базы знаний* в целом.]
- ⇒ *примечание**:
 [Любой *раздел базы знаний* не является структурной частью другого раздела. Каждый раздел самодостаточен и целостен. Это обеспечивается тем, что в состав *титульной спецификации* каждого раздела входит *семантическая окрестность*, описывающая связи специфицируемого раздела со всеми семантически близкими ему разделами.
 При этом разные разделы могут иметь разный семантический тип:
- раздел может быть предметной областью, интегрированной со всеми ее онтологиями;
 - раздел может быть просто предметной областью;
 - раздел может быть какой-либо онтологией чего угодно (не обязательно предметной области);
 - и т.д.
-]

⇒ *примечание**:
 [Если в *титульную спецификацию* каждого раздела будет входить семантическая спецификация каждого раздела, включающая его семантические связи со всеми семантически близкими разделами, то последовательность (порядок) разделов в "линейном" исходном тексте, публикуемом в качестве очередной версии Стандарта OSTIS, может быть в достаточной степени произвольной.]

семейство разделов базы знаний

:= [множество семантически связанных друг с другом разделов базы знаний]

:= [кластер разделов базы знаний]

⇒ *примечание**:

[Семантические связи между разделами, входящими в состав семейства разделов, представляются в рамках *титульных спецификаций разделов*, каждая из которых является специальной частью соответствующего (специфицируемого) раздела, входящего в состав семейства разделов.

Для каждого *раздела базы знаний* в рамках его титульной спецификации формируется *семантическая окрестность* его связей со всеми семантически близкими ему разделами (и, прежде всего, с теми разделами, которые входят в состав тех семейств разделов, в которые входит заданный раздел). При этом акцентируется внимание именно на семантических связях между разделами. Так, например, вместо структурной ("синтаксической") связи "быть подразделом*" (т.е. быть частью заданного раздела) вводится связь "быть дочерним разделом*".

Данная связь указывает направление наследования свойств исследуемых объектов заданного раздела от разделов, исследующих более общие классы объектов. Порядок (последовательность) разделов в рамках *семейства разделов базы знаний* при наличии явно представленных семантических связей между разделами, входящими в семейство разделов, может быть достаточно произвольным, что очень важно, например, при формировании оглавления очередной издаваемой версии *Стандарта OSTIS*. Таким образом, трактовка *Стандарта OSTIS*, а также всех издаваемых версий этого Стандарта как *семейства разделов базы знаний Метасистемы IMS.ostis* обеспечивает высокий уровень гибкости *Стандарта OSTIS*, а также легкость "переиздаваемости" его версий.]

сегмент или атомарный раздел базы знаний

⇒ *примечание**:

[Простейшей формой *сегмента* или *атомарного раздела* базы знаний является просто последовательность *файлов ostis-системы*. Некоторые из этих файлов могут быть идентифицированными (именованными), если

на них ссылаются другие файлы, а некоторые из них могут быть связаны с другими файлами различными отношениями (в частности, один файл может быть пояснением другого). Кроме того, некоторые из этих файлов могут быть формально специфицированы (например, указаны соответствующие им ключевые *sc-элементы*).

В самом простом случае *сегмент* или *атомарный раздел базы знаний* может быть *структурой*, состоящей из одного (!) *sc-узла*, обозначающего *файл ostis-системы* (чаще всего, *ея-файл ostis-системы*). Т.е. сам *файл ostis-системы* может быть *знанием ostis-системы*, но не может быть структурно *выделяемым фрагментом базы знаний ostis-системы*. При этом *sc-узел*, обозначающий *файл ostis-системы*, являющийся *знанием*, может быть единственным *sc-элементом* структурно выделяемого *знания ostis-системы*.]

⇒ *примечание**:

[Для наглядного отображения (визуализации) *сегмента* или *атомарного раздела базы знаний ostis-системы* целесообразно представить указанное *sc-знание* в виде конкатенации (последовательности) таких *sc-знаний*, которые, во-первых, были бы достаточно крупными и логико-семантически значимыми для соответствующего *сегмента* или *атомарного раздела базы знаний ostis-системы* и, во-вторых, для которых существовал бы алгоритм однозначного (!) размещения (на экране) внешнего представления этих *sc-знаний* (в *SCg-коде* или в *SCn-коде*).

Однозначность здесь означает наличие легко усваиваемого пользователями стандартного стиля визуализации sc-знаний и заключается в том, что многократная визуализация одного и того же *sc-знания* с помощью указанного алгоритма должна приводить к синтаксически эквивалентным, а в случае *SCg-кода* и к геометрически конгруэнтным текстам. Очевидно, что для произвольных *sc-знаний* большого объема такого алгоритма не существует, но для *sc-знаний*, содержащих описание собственной структуры и семантической типологии собственных фрагментов, разработка такого алгоритма вполне реальна при наличии достаточного количества указанных *метазнаний* о структуре отображаемых (визуализируемых) *sc-знаний*.]

sc-идентификатор выделенного фрагмента базы знаний

:= [название (имя) выделенного фрагмента базы знаний]

⇒ *пояснение**:

[Не следует путать объект описания (спецификации) и само описание. Поэтому в *sc-идентификаторе* фрагмента базы знаний должны присутствовать слова, указывающие на семантический или структурный тип именуемого фрагмента базы знаний (описание, спецификация, анализ, сравнительный анализ, сравнение, определение, раздел, предметная область, онтология и т.п.).

Таким образом, *sc-идентификатор выделенного фрагмента базы знаний ostis-системы* должен иметь явное (!) указание на то, что он является обозначением именно фрагмента базы знаний, а не того, что описывается в этом фрагменте.]

⇒ *примечание**:

[Мы не будем использовать такой изменчивый для нас способ идентификации разделов *Стандарта OSTIS*, как нумерацию этих разделов, поскольку, например, в разных издаваемых официальных версиях *Стандарта OSTIS* одному и тому же разделу *Стандарта OSTIS* могут соответствовать разные номера.]

выделенный фрагмент базы знаний

⇒ *основной sc-идентификатор**:

[выделенный фрагмент базы знаний]

⇒ *используемая аббревиатура**:

[выделенный фр-нт б.з.]

⇒ *разбиение**:

{ • *именованный фрагмент базы знаний*

:= [выделенный фрагмент базы знаний, имеющий *sc-идентификатор* (имя, название)]

⇒ *примечание**:

[Именованными фрагментами баз знаний могут быть только структурно выделенные фрагменты баз знаний]

⇒ *примечание**:

[Все семейства разделов баз знаний, все разделы баз знаний и все сегменты баз знаний должны быть именованными]

⊃ *семейство разделов базы знаний*

⊃ *раздел базы знаний*

⊃ *сегмент базы знаний*

• *неименованный фрагмент базы знаний*

:= [выделенный фрагмент базы знаний, не имеющий *sc-идентификатора* (имени, названия)]

⇒ *примечание**:
 [неименованными фрагментами баз знаний могут быть только выделенные фрагменты сегментов баз знаний либо выделенные фрагменты таких разделов баз знаний, которые не состоят из сегментов]

}

титульная спецификация выделенного фрагмента базы знаний

⇒ *пояснение**:

[Титульная спецификация выделенного фрагмента базы знаний *ostis*-системы представляет собой структуру, описывающую свойства специфицируемого знания и включающую в себя:

- связи принадлежности специфицируемого знания соответствующим классам знаний *ostis*-систем;
- связи, указывающие логически предшествующее и логически следующее знание *ostis*-системы;
- связь, описывающую декомпозицию специфицируемого знания на последовательность знаний более низкого структурного уровня (декомпозицию разделов на сегменты);
- различного вида связи с другими знаниями *ostis*-систем, которые сами "целиком" входят в состав спецификации специфицируемого знания (такими знаниями могут быть аннотации, предисловия, введения, оглавления, заключения);
- различного вида связи с другими знаниями *ostis*-систем, которые сами не входят в состав спецификации специфицируемого знания (такого рода связями могут быть связи семантической близости специфицируемого знания с другими знаниями, связи семантической эквивалентности, связисемантического включения, связи противоречивости знаний);
- связи, указывающие различного вида ключевые *sc*-элементы (ключевые знаки), соответствующие специфицируемому знанию;
- связи специфицируемого знания с авторским коллективом, коллективом рецензентов, с датой его последнего обновления;
- для каждого нового целостного фрагмента, вводимого в состав базы знаний, в истории эволюции этой базы знаний указываются:
 - автор** или *авторы** первой версии этого фрагмента;
 - отметка времени появления (дата-час-минута) всех версий этого фрагмента (в том числе и окончательно утверждённой, согласованной версии, которая, собственно, и становится фрагментом, включенным в согласованную часть базы знаний);
 - рецензии** (замечания к доработке) всех предварительных версий разрабатываемого фрагмента базы знаний;
 - авторы** всех указанных рецензий;
 - отметка времени появления всех указанных рецензий;
 - события по одобрению, утверждению различных предварительных версий разрабатываемого фрагмента базы знаний различными рецензентами и экспертами с указанием отметки времени появления этих событий;
 - темпоральная последовательность предварительных версий.

]

⇒ *примечание**:

[Знак такой спецификации явно не вводится, а сама эта спецификация непосредственно входит в состав специфицируемого фрагмента и включает в себя аннотацию, предисловие, авторов, ключевые знаки, декомпозицию специфицируемого фрагмента базы знаний и прочее]

⇒ *разбиение**:

- {
 - титульная спецификация раздела базы знаний
 - титульная спецификация семейства разделов базы знаний
 - титульная спецификация сегмента базы знаний
 - титульная спецификация выделенного фрагмента сегмента или атомарного раздела базы знаний

⇒ *пояснение**:

[Титульная спецификация выделенного фрагмента базы знаний содержит общую информацию об этом фрагменте, является непосредственно частью специфицируемого фрагмента базы знаний и при этом сама не является явно выделенным фрагментом базы знаний]

∈ спецификация

:= [основная метаинформация (основное метазнание) о выделенном фрагменте базы знаний – о его структуре, авторах', ключевых знаках' и т.д.]

- ⇒ *пояснение**:
 [неявно выделяемый фрагмент базы знаний, который:
- не имеет "собственного" ограничителя ("собственного" контура или "собственных" ограничивающих фигурных скобок);
 - является семантической спецификацией соответствующего явно выделяемого фрагмента базы знаний;
 - является непосредственной частью специфицируемого фрагмента базы знаний

-]]
 ⇒ *примечание**:
 [В *с.п-тексте* титульная спецификация фрагмента базы знаний размещается сразу после фигурной скобки, открывающей этот фрагмент]

титульная спецификация раздела базы знаний

- ⇒ *примечание**:
 [титульная спецификация раздела базы знаний должна включать в себя достаточно подробное описание семантических свойств этого раздела и, в частности, подробное описание его связей с другими семантически близкими разделами. Это необходимо для обеспечения автономности разделов баз знаний.]

титульная спецификация семейства разделов базы знаний

- ⇒ *примечание**:
 [Если разделы базы знаний являются семантически ключевыми выделенными фрагментами баз знаний, определяющими спецификацию систем используемых понятий и направления наследования свойств, то семейства разделов баз знаний являются ключевыми для структуризации виртуальной Базы знаний Экосистемы OSTIS, для обмена знаниями между различными субъектами Экосистемы OSTIS. Поэтому типология семейств разделов баз знаний и качество титульной спецификации семейств разделов баз знаний имеют большое значение.]

титульная спецификация выделенного фрагмента базы знаний

- ⇒ *множество используемых понятий**:
 Множество понятий используемых в титульных спецификациях выделенных фрагментов баз знаний
- ⊃ *класс выделенных фрагментов*
 - Э *семейство разделов базы знаний*
 - Э *раздел базы знаний*
 - Э *неатомарный раздел базы знаний*
 - Э *атомарный раздел базы знаний*
 - Э *сегмент базы знаний*
 - Э *выделенный фрагмент сегмента или атомарного раздела базы знаний*
 - Э *выделенный фрагмент атомарного раздела базы знаний*
 - Э *выделенный фрагмент сегмента базы знаний*
 - ⊃ *отношение, связывающее выделенные фрагменты баз знаний с персонами*
 - Э *автор**
 - Э *рецензент**
 - Э *эксперт**
 - Э *технический редактор**
 - Э *консультант**
 - := [активный участник обсуждения вопросов, рассматриваемых в специфицируемом фрагменте базы знаний*]
 - ⊃ *отношение, связывающее выделенные фрагменты баз знаний с ея-файлами*
 - Э *аннотация**
 - Э *предисловие**
 - := [Бинарное ориентированное отношение, каждая пара которого связывает:
 - знак некоторого информационного ресурса (в частности, раздела базы знаний или раздела опубликованного документа);
 - знак информационной конструкции, описывающей цели создания указанного информационного ресурса, предысторию его создания, планируемые направления дальнейшего развития, состав авторов и др.
-]]
- Э *введение**
 - Э *эпиграф**
 - Э *заключение**

```

Э рассматриваемый вопрос*
Э основные положения*
Э вопрос для самопроверки*
Э упражнение*
:= [задача*]
:= [самостоятельная (индивидуальная) работа*]
Э коллективный проект*
Э неосновной sc-идентификатор*
⇒ примечание*:
    [неосновным sc-идентификатором, в частности, может быть альтернативное название
    выделенного (специфицируемого) фрагмента базы знаний]
Э часто используемый sc-идентификатор*
Э сокращенный sc-идентификатор*
Э библиографический источник, отражающий аналогичную точку зрения*
Э библиографический источник, отражающий альтернативную точку зрения*
Э библиографический источник, дополняющий данную точку зрения*
Э сокращение*
:= [используемое сокращение*]
:= [сокращение, используемое в специфицируемом фрагменте базы знаний при построении
sc-идентификаторов, а также при оформлении ея-файлов*]
:= [Бинарное ориентированное отношение, каждая пара которого связывает естественно-
языковую фразу с ее сокращенной записью*]
⊃ отношение, описывающее структурные или семантические связи и между выделенными
фрагментами баз знаний
Э конкатенация сегментов*
Э предыдущий сегмент*
:= [предыдущий сегмент в рамках соответствующего раздела*]
Э следующий сегмент*
Э дочерний фрагмент базы знаний*
⊃ дочерний раздел базы знаний*
⊃ дочерняя предметная область*
⊃ дочерняя предметная область и онтология*
⇒ примечание*:
    [Для фрагмента базы знаний важно указать не только дочерние по отношению к нему
фрагменты базы знаний, но и те фрагменты базы знаний, по отношению к которым
данный фрагмент базы знаний является дочерним]
⊃ отношение, описывающее ролевой статус знаков, входящих в состав выделенных фрагментов баз
знаний
Э ключевой знак*
Э ключевой знак первого плана*
Э ключевой знак второго плана*
Э ключевой объект исследования*
Э ключевое понятие*
Э ключевой класс объектов исследования*
Э исследуемое отношение*
Э исследуемый параметр*
Э исследуемый класс структур*
} /* Завершили Сегмент "Структуризация баз знаний ostis-систем" */

```

```

/* Сегмент *****/

```

```

Описание правил оформления внешнего представления знаний ostis-систем

```

```

⊃=

```

```

{

```

```

внешнее представление знаний ostis-системы

```

```

⇒ пояснение*:

```

```

[внешнее представление некоторого фрагмента базы знаний ostis-системы, используемое для ввода новой
информации в состав базы знаний ostis-системы или для вывода (отображения) запрашиваемого фрагмента
базы знаний]

```


⇒ *примечание**:

[Способ представления исходных текстов баз знаний ostis-систем должен быть максимально возможным образом использован и для вывода (отображения) запрашиваемых пользователем фрагментов баз знаний, особенно, если запрашиваются достаточно большие фрагменты баз знаний, которые необходимо не только представлять, но и структурировать унифицированным образом. Очевидно, что для пользователей желательно, чтобы и для ввода информации в ostis-систему, и для ее вывода использовались одни и те же языковые средства и правила оформления]

⇒ *примечание**:

[Требования, предъявляемые к оформлению внешних текстов знаний ostis-систем (sc-знаний) носят достаточно противоречивый характер – с одной стороны, речь идет о формальных текстах, легко воспринимаемых (понимаемых, транслируемых) ostis-системами, а, с другой стороны, желательно, чтобы эти же формальные тексты легко воспринимались (понимались) широким кругом людей и не требовали для этого от них длительной подготовки. Отметим при этом, что работа с формальными текстами требует от человека достаточно высокой культуры точного мышления (математической культуры).

Отметим также, что использование формальных языков является важнейшим и необходимым этапом эволюции человеческой деятельности в любой области (в математике, в физике, в технике).

Тем не менее, проблема создания универсального языка представления исходных текстов различного вида знаний, который был бы достаточно удобен как для интеллектуальных компьютерных систем, так и для широкого круга разработчиков баз знаний и экспертов, требует конкретного решения.]

⇐ *требования**:

- [Стиль и характер оформления внешнего представления sc-знаний должен обеспечить возможность интуитивного понимания смысла текста при отсутствии понимания различного рода синтаксических деталей. Для этого:

 - формальный текст должен максимально возможным образом использовать привычную для широкого круга специалистов терминологию;
 - структуризация, форматирование формальных текстов также должны опираться на сформировавшиеся традиции;
 - внешнее представление (внешний текст) sc-знания должен включать в себя такое количество отображаемых ея-файлов, прочтения которых было бы достаточно для понимания смысла представляемого sc-знания, а также для понимания формальных средств его представления
-]
- [Формальный текст (как внутреннего, так и внешнего представления sc-знаний) должен включать в себя средства для уточнения смысла используемых знаков и соответствующих им терминов, а также смысла некоторых фрагментов формального текста. Для этого в формальный язык вводятся естественно-языковые файлы, отображаемые в исходных текстах и поясняющие используемые термины, а также комментирующие или даже полностью переводящие на естественный язык различные фрагменты формального представления *базы знаний*.]
- [Все используемые в базе знаний ostis-системы (в том числе, и в её ея-файлах) внешние идентификаторы *sc-элементов* (термины, имена, условные обозначения) должны быть формально специфицированы средствами *SC-кода*. Подчеркнем, что здесь речь идет о спецификации не самих sc-элементов, а их внешних идентификаторов (в первую очередь, простых sc.s-идентификаторов) – их происхождение, использование, авторство и т.д.]
- [Все отношения, параметры и другие понятия, используемые в формальных текстах должны быть пояснены в соответствующих формальных онтологиях. Первое упоминание во внешнем тексте каждого такого понятия должно быть кратко пояснено с помощью поясняющего ея-файла, а также сделана ссылка на раздел базы знаний, в которых приведена подробная и формальная спецификация указанного понятия с дополнительным указанием номера этого раздела с помощью нетранслируемого комментария.]
- [Аналогичным образом в отображаемом внешнем представлении sc-знания поясняются и комментируются все первые (в рамках этого внешнего представления) использования средств формального представления знаний со ссылками на разделы базы знаний, где указанные языковые средства подробно описываются. С самого начала внешнего представления большого структурированного фрагмента базы знаний (каковым, в частности, является Стандарт OSTIS) с помощью нетранслируемых комментариев, не входящих в состав базы знаний, либо с помощью ея-файлов ostis-систем необходимо пояснять все нюансы формализации со ссылкой на ближайший раздел и сегмент, где это будет подробнее рассмотрено.]

- [При описании формальных средств должны быть приведены конкретные примеры со ссылкой на раздел или сегмент, где этот пример будет рассмотрен подробнее (например, на соответствующую предметную область и онтологию)]
- [Все комментарии и примечания, которые можно представить средствами *SCn-кода* или *SCg-кода*, нужно оформлять именно так. Нетранслируемыми комментариями не стоит увлекаться.]
- [В формальных текстах и в естественно-языковых файлах, входящих в состав *базы знаний ostis-системы*, для идентификации (именования) *sc-элементов* должны использоваться только те термины, которые являются основными(!) внешними идентификаторами соответствующих *sc-элементов*, выделяемыми жирным и нежирным курсивом. При этом, если идентификаторы (названия, имена) разделов, сегментов базы знаний находятся в позиции заголовков указанных фрагментов базы знаний, то они оформляются жирным курсивом с увеличенным расстоянием между символами, а заголовки разделов дополнительно выделяются увеличенным размером символов.]
- [В согласованной (общепризнанной) части *базы знаний* противоречия трактуются как выявленные ошибки в *базе знаний*, подлежащие устранению. Но в истории эволюции базы знаний противоречия могут присутствовать как противоречия разных точек зрения разных авторов. Заметим при этом, что разные точки зрения далеко не всегда являются противоречивыми (взаимоисключающими). Они могут просто дополнять друг друга, описывать исследуемые сущности с разных "ракурсов". Умение видеть противоречия только там, где, они действительно есть, и умение локализовать эти противоречия (выделить их суть) – это необходимые навыки для разработки практически полезных баз знаний.]

)

sc.n-представление знаний ostis-системы

⇒ *правила оформления**:

⟨

- [В исходных текстах баз знаний *интеллектуальных компьютерных систем*, построенных по *Технологии OSTIS* (т.е. *ostis-систем*) используются следующие языки:
 - Различные *естественные языки* (***Русский язык, Английский язык*** и др.)
 - ***SCg-код*** (Semantic Code graphical), являющийся *универсальным формальным языком* графического представления *семантических сетей*
 - ***SCs-код*** (Semantic Code string), являющийся *универсальным формальным языком* линейного представления *семантических сетей* в виде *строк* (цепочек символов)
 - ***SCn-код*** (Semantic Code Natural) являющийся *универсальным формальным языком* представления *семантических сетей* в виде структурированных форматированных *текстов* (строк, размещенных на плоскости.

Синтаксис и денотационная семантика указанных *формальных языков* подробно рассмотрены в Разделах *Описание языка графического представления информационных конструкций в ostis-системах, Описание языка линейного представления информационных конструкций в ostis-системах, Описание языка структурированного представления информационных конструкций в ostis-системах.*]
- [Основным языком внешнего представления *баз знаний ostis-систем* является *SCn-код*, рассмотренный в разделе *Описание языка структурированного представления информационных конструкций в ostis-системах*. Но в состав текста *SCn-кода* могут входить тексты и других языков (тексты *SCg-кода*, тексты *SCs-кода*, тексты естественных языков, тексты различных искусственных языков), а также различного рода нетекстовые информационные конструкции (рисунки, таблицы, чертежи, графики, фотографии). Указанные "иностранцы" для *SCn-кода* информационные конструкции, а также описываемые тексты самого *SCn-кода* оформляются во внешнем представлении базы знаний либо как нетранслируемые, но специфицируемые файлы, либо как транслируемые иностранцы для *SCn-кода* информационные конструкции, ограниченные, соответственно, либо *sc.n-рамками* (квадратными скобками), либо *sc.n-контурами* (фигурными скобками).]
- [Структуризация внешнего представления баз знаний *ostis-систем* является полным отражением структуризации внутреннего представления *баз знаний ostis-систем.*]
- [В случае, если осуществляется внешнее представление полного текста указываемого сложноструктурированного фрагмента базы знаний, как, например, внешнее представление семейства разделов под названием "*Стандарт OSTIS*" вместе со всеми его разделами, то последовательность и иерархическая структура отображения разделов и сегментов указанного сложноструктурированного фрагмента базы знаний в точности соответствует иерархической структуре представляемого (отображаемого, визуализируемого) семейства разделов базы знаний.]
- [База знаний каждой *ostis-системы* представляет собой иерархическую систему разделов, к которым должны "привязываться" исходные тексты каждой новой информации, вводимой в *базу знаний*, и,

прежде всего, исходные тексты достаточно крупных фрагментов *баз знаний*. К таким исходным текстам, в частности, относится и данная публикуемая версия Стандарта OSTIS. Очевидно при этом, что нумерация разделов баз знаний не может быть стабильна. Кроме того, при оформлении исходного текста крупного фрагмента базы знаний, обладающего достаточной целостностью и по научно-технической значимости достигшего уровня монографии или диссертации, желательно иметь собственную (свою, локальную) нумерацию разделов при сохранении их иерархической структуры. Это означает, что номера разделов имеет смысл использовать только в рамках каждого исходного вводимого текста и не должны использоваться в самой базе знаний. Таким образом, ссылаться на разделы базы знаний следует по названию разделов.]

- [Внешнее представление каждого структурно *выделяемого фрагмента* базы знаний ostis-системы за исключением sc-знаний нижнего уровня начинается с заголовка (имени, названия) этого фрагмента. Указанный заголовок есть не что иное, как простой sc-идентификатор sc-узла, обозначающего представляемый фрагмент базы знаний (представляемое sc-знание). Рассматриваемый заголовок оформляется жирным курсивом с увеличенным расстоянием между символами. При этом заголовки внешнего представления разделов базы знаний имеют дополнительно увеличенный размер шрифта. Заголовок внешнего представления sc-знания размещается с первого символа строки. После заголовка представляемого фрагмента базы знаний с новой строки размещается (1) sc.s-коннектор вида "▷=" (2) следом за ним на следующей строке левая фигурная скобка (открывающая фигурная скобка).]
- [Внешнее представление каждого раздела базы знаний начинается с новой страницы. Соответственно, сегмент базы знаний может начинаться не с новой страницы.]
- [Внешнее представление каждого неструктурируемого выделяемого фрагмента базы знаний (атомарного раздела базы знаний, сегмента базы знаний) оформляется в виде *sc.n-предложения*, связывающего sc-идентификатор (имя, название) представляемого фрагмента базы знаний, оформленный в виде заголовка внешнего текста этого фрагмента (признаком чего является жирный курсив с увеличенным расстоянием между символами), с *sc.n-контуром*, который фигурными скобками ограничивает sc.n-изображение (sc.n-визуализацию) представляемого фрагмента базы знаний. В самом простом случае представляемым неструктурируемым выделяемым фрагментом базы знаний является один ея-файл ostis-системы.]
- [Внешний текст *базы знаний* может иметь самый разный объем и может касаться только одного раздела или сегмента, а может включать в себя материалы нескольких разделов или сегментов. Если представляемый (отображаемый) фрагмент базы знаний является sc-знанием нижнего уровня иерархии, т.е. частью неструктурированного фрагмента базы знаний (например, частью сегмента базы знаний), то при его представлении необходимо указать, частью какого неструктурированного фрагмента базы знаний представляемый фрагмент является. Для исходного текста sc-знания нижнего уровня эта информация необходима для того, чтобы знать, в какой фрагмент базы знаний требуется включить, "погрузить" данное вводимое *sc-знание*.]
- [Виды выделений во внешнем представлении знаний ostis-системы:
 - с помощью шрифта
 - вид шрифта (печатный, курсив)
 - размер шрифта (стандартный, увеличенный)
 - расстояние между символами (стандартное, увеличенное)
 - подчеркиванием
 - с помощью символьных ограничителей (скобок различного вида)
]
- [Выделяемые объекты:
 - основные термины (имена, идентификаторы)
 - специфицируемые файлы
 - нетранслируемые в базу знаний файлы
 - транслируемые в базу знаний файлы
 - нетранслируемые комментарии к внешнему тексту
 - цитаты (и короткие, и длинные)
 - ключевые фрагменты ея-текста
 - метафорические термины
]

- [На любой странице внешнего текста при распечатке делается разметка тонкими вертикальными табуляционными линиями для четкой визуализации длины отступа от левого края страницы. Особенно это важно при переходе на новую страницу.]

}

/* Завершили перечень правил оформления внешнего представления знаний ostis-системы */

sc.n-представление выделенного фрагмента базы знаний

← обобщенная конкатенация*:

- **признак начала sc.n-представления выделенного фрагмента базы знаний**
⇒ **пояснение***:

[Данным признаком является *нетранслируемый комментарий*, размещенный по всей длине строки и состоящий

- либо из слова "Раздел" (изображенного жирным печатным шрифтом такого же размера и с таким же расстоянием между символами, что и в заголовке *sc.n-представления раздела базы знаний*) и последующих "звездочек" до конца строки;
- либо из слова "Сегмент" (изображенного жирным печатным шрифтом такого же размера и с таким же расстоянием между символами, что и в заголовке *sc.n-представления сегмента базы знаний*) и последующих "звездочек" до конца строки;
- либо из одних "звездочек" до конца строки.

]

- **заголовок sc.n-представления выделенного фрагмента базы знаний**

- **связка "⊃="**

:= [связка, связывающая заголовок *sc.n-представления выделенного фрагмента базы знаний* с самим *sc.n-текстом* этого фрагмента]

⇒ **размещение***:

с начала новой строки

- **фигурная скобка, открывающая sc.n-текст выделенного фрагмента базы знаний**

⇒ **размещение***:

первый символ новой строки

- **собственно sc.n-текст выделенного фрагмента базы знаний**

⇒ **примечание***:

[Данный текст может занимать большое количество строчек]

- **фигурная скобка, закрывающая sc.n-текст выделенного фрагмента базы знаний**

⇒ **размещение***:

первый символ новой строки

:= [первый символ строки, которая следует после последней строки *sc.n-текста выделенного фрагмента базы знаний*]

- **дополнительный признак завершения sc.n-представления выделенного фрагмента базы знаний**

⇒ **пояснение***:

[Данный признак используется только при представлении выделенных фрагментов сегментов баз знаний и выделенных фрагментов неструктурированных разделов баз знаний, которые не декомпозируются на сегменты. При этом рассматриваемый признак изображается как нетранслируемый комментарий, имеющий длину в половину строки и состоящий из "звездочек". В случае, если в *sc.n-представлении* сразу за *выделенным фрагментом базы знаний* следует другой *выделенный фрагмент базы знаний*, допускается не указывать упомянутый признак завершения *sc.n-представления выделенного фрагмента базы знаний*.]

}

⇒ **примечание***:

[Отличия в *sc.n-представлении* различного вида *выделенных фрагментов баз знаний* могут заключаться в следующем:

- размер шрифта и расстояние между символами в заголовке *sc.n-представления выделенного фрагмента базы знаний* могут быть различны;
- для выделенных фрагментов сегментов и фрагментов, не содержащих сегменты, заголовок *sc.n-представления фрагмента базы знаний* и, соответственно, связка "⊃=" могут отсутствовать, т.е. такие фрагменты баз знаний могут быть неименованными.

]

заголовок sc.n-представления выделенного фрагмента базы знаний

C sc-идентификатор выделенного фрагмента базы знаний

C жирный курсив возможно увеличенного размера и с увеличенным расстоянием между символами

Є шрифт[^]

⇒ размещение*:

с начала новой строки

:= [с первого символа новой строки]

:= [от первой табуляционной линии размещения sc.n-текста]

титульная спецификация публикации семейства разделов базы знаний

⇒ примечание*:

[Имеется в виду публикация (издание) семейства разделов базы знаний в виде sc.n-представления семейства разделов базы знаний]

⇒ примечание*:

[В простейшем случае публикуемое семейство разделов базы знаний может состоять из одного раздела. Например, это может быть публикация раздела базы знаний в виде статьи.]

нетранслируемый комментарий к внешнему тексту

:= [нетранслируемый комментарий к внешнему тексту отображаемого фрагмента базы знаний ostis-системы]

⇒ пояснение*:

[естественно-языковой текст, который ограничен слева наклонной чертой и звездочкой “/*” и справа – звездочкой и наклонной чертой “*/” и который может находиться в любом месте внешнего текста]

⇒ примечание*:

[При загрузке и трансляции исходного текста базы знаний ostis-системы все входящие в него нетранслируемые комментарии игнорируются – и в том случае, если эти комментарии входят в состав изображения какого-либо файла ostis-системы (в частности, естественно-языкового файла), и в том случае, если эти комментарии входят в состав формального текста, транслируемого в SC-код.

При трансляции внутреннего текста (sc-текста) базы знаний ostis-системы во внешнее представление (например, в sc.n-текст) некоторые нетранслируемые комментарии могут автоматически генерироваться. Например, нетранслируемые комментарии, предшествующие заголовкам внешнего представления структурно выделяемых фрагментов баз знаний ostis-систем (разделов, сегментов), нетранслируемые комментарии к правым (закрывающим) фигурным и квадратным скобкам, если ограничиваемые этими скобками выражения размещаются на нескольких страницах.

Приведем несколько конкретных примеров:

/* представление данного файла будет продолжено на следующей странице */

/* продолжение представления файла */

/* представление данного sc-текста будет продолжено на следующей странице */

/* продолжение представления sc-текста */

/* sc-текст (файл), обозначаемый данным sc-узлом, представлен на следующей странице (на следующих страницах) */

Примеры нетранслируемых комментариев к закрывающим скобкам:

/* Завершили Раздел “Описание общих принципов оформления внутреннего и внешнего представления информационных конструкций в ostis-системах” */

/* Завершили Сегмент “Описание правил оформления внешнего представления знаний ostis-систем” */.]

⇒ примечание*:

[Нетранслируемые комментарии к внешнему тексту базы знаний непосредственно в состав базы не входят. Указанными комментариями не следует злоупотреблять, они должны касаться оформления внешнего текста, но не смысла представляемой базы знаний.

Содержательные комментарии к базе знаний должны оформляться в виде файлов, входящих в состав базы знаний.]

} /* Завершили Сегмент “Описание правил оформления внешнего представления знаний ostis-систем” */

} /* Завершили Раздел “Описание общих принципов оформления внутреннего и внешнего представления информационных конструкций в ostis-системах” */

/* Раздел *****/

Предметная область и онтология кибернетических систем

⊃=
{

Предметная область кибернетических систем

- ∈ предметная область
- ⊃ максимальный класс объектов исследования¹:
 - кибернетическая система
- ⊃ класс объектов исследования¹:
 - искусственная сущность
 - компьютерная система
 - простая кибернетическая система
 - индивидуальная кибернетическая система
 - кибернетическая система, встроенная в индивидуальную кибернетическую систему
 - многоагентная система
 - одноуровневая многоагентная система
 - коллектив индивидуальных кибернетических систем
 - иерархический коллектив индивидуальных кибернетических систем
 - информация, хранимая в памяти кибернетической системы
 - абстрактная память кибернетической системы
 - решатель задач кибернетической системы
 - действие кибернетической системы
 - задача
 - задача, решаемая кибернетической системой
 - навык
 - интерфейс кибернетической системы
 - физическая оболочка кибернетической системы
 - память кибернетической системы
 - процессор кибернетической системы
 - компьютер
 - качество кибернетической системы
 - гибридная кибернетическая система
 - приспособленность кибернетической системы к её совершенствованию
 - гибкость кибернетической системы
 - производительность кибернетической системы
 - надежность кибернетической системы
 - качество физической оболочки кибернетической системы
 - качество памяти кибернетической системы
 - интеллект
 - образованность кибернетической системы
 - интеллектуальная система
 - кибернетическая система, основанная на знаниях
 - кибернетическая система, управляемая знаниями
 - целенаправленная кибернетическая система
 - обучаемая кибернетическая система
 - социально ориентированная кибернетическая система
 - интеллектуальная компьютерная система
 - информация
 - сенсорная информация
 - качество решателя задач кибернетической системы
 - обучаемость кибернетической системы
 - стратифицированность кибернетической системы
 - рефлексивность кибернетической системы
 - синергетическая кибернетическая система
 - социализация кибернетической системы
- ⊃ исследуемое отношение¹:
 - информация, хранимая в памяти кибернетической системы*
 - задача, решаемая кибернетической системой*

- *внешняя среда кибернетической системы**
- *среда кибернетической системы**
- *агент**

:= [Иерархическая система свойств (характеристик) кибернетических систем, определяющих общий (интегральный) уровень их качества]

:= [Эволюционный подход к определению качества и, в частности, уровня интеллекта кибернетической системы]

⇒ *аннотация**:
[Рассмотрена иерархическая система свойств (в т.ч. способностей) кибернетических систем, определяющих их качество и позволяющих сформулировать требования, которым должна удовлетворять высокоинтеллектуальная система (кибернетическая система с сильным интеллектом).

Уровень качества кибернетических систем определяется достаточно большим набором свойств (параметров, характеристик) кибернетических систем, каждое из которых определяет уровень качества кибернетической системы в соответствующем аспекте (ракурсе), указывая (задавая) уровень развития конкретных способностей и возможностей кибернетической системы. При этом важно подчеркнуть следующее:

- существенное значение имеет не столько сам набор свойств, а иерархия этих свойств, позволяющая уточнять (детализировать) направления проявления (реализации) каждого свойства;
- существенное значение также имеет баланс уровней развития различных свойств – вклад разных свойств, обеспечивающих (определяющих) значение одного и того же свойства более высокого уровня иерархии, а значение этого свойства более высокого уровня может быть разным. Из этого следует, что не всегда следует акцентировать внимание на развитие некоторых свойств (характеристик). Нужен целостный, коллективный подход;
- рассмотренная иерархия свойств кибернетических систем является общей как для естественных, так и для искусственных кибернетических систем;
- приведенная иерархическая детализация свойств кибернетических систем (с помощью отношения “частное свойство*” и отношения “свойство-предпосылка**”), определяющих качество таких систем, (1) дает возможность четко определить направления совершенствования (развития) кибернетических систем и (2) дает ориентир (систему критериев) для обоснования конкретных предложений по совершенствованию компьютерных систем, а также для сравнения различных альтернативных предположений;
- особое значение для развития кибернетических систем имеют такие их свойства, как стратифицированность, рефлексивность и социализация;
- важное значение имеет не только совершенствование кибернетических систем в соответствии с иерархической системой их свойств, но и совершенствование (в том числе, детализация) самой этой иерархической системы свойств.

⇒] *предисловие**:

[Свойства (способности), которым должны удовлетворять *интеллектуальные системы*, рассматриваются в целом ряде публикаций. Тем не менее, для практической реализации *компьютерных систем*, обладающих указанными свойствами (способностями), т.е. *интеллектуальных компьютерных систем*, необходимо детализировать (уточнить) эти *свойства*, пытаясь свести их к более конструктивным, прозрачным и понятным для реализации свойствам.]

⇒ *рассматриваемые вопросы**:

- { • [По каким свойствам (параметрам, характеристикам, способностям) кибернетических систем можно оценивать уровень их качества.]
- [Можно ли считать уровень развития какого-либо свойства (способности) кибернетической системы, т.е. значение какого-либо ее параметра (характеристики) оценкой уровня качества кибернетической системы по соответствующему аспекту.]
- [Может ли какое-либо свойство кибернетических систем определять (влиять на) значение сразу нескольких свойств более высокого уровня иерархии.]
- [Какими отношениями свойства кибернетических систем связаны со свойствами более низкого и, соответственно, более высокого уровня иерархии.]
- [Зачем нужна такая иерархия свойств, определяющих качество кибернетических систем и позволяющих детализировать (уточнять) то, какими свойствами определяется уровень (степень) развития каждого свойства (значение каждого свойства) за исключением свойств, которые условно можно считать элементарными, не требующими детализации (по крайней мере, пока).]
- [Может ли иерархия свойств, определяющих качество кибернетических систем, быть критерием оценки и выбора того или иного подхода к построению интеллектуальных компьютерных систем.]
- [Какими свойствами (способностями) должна обладать кибернетическая система, имеющая высокий уровень интеллекта.]

- [Какими свойствами определяется уровень интеллекта многоагентной кибернетической системы.]
- [Как связан уровень интеллекта многоагентной системы с уровнем интеллекта агентов, входящих в ее состав.]
- [Почему, например, не каждый коллектив высокоинтеллектуальных людей демонстрирует высокий уровень интеллекта самого коллектива.]
- [Какими дополнительными свойствами кроме достаточно высокого уровня интеллекта должны обладать агенты многоагентных систем для обеспечения высокого уровня интеллекта самой многоагентной системы как самостоятельной целостной кибернетической системы.]
- [Как зависит уровень интеллекта многоагентной системы от организации взаимодействия между агентами, например, от использования централизованного или децентрализованного управления.]

}

⇒

ключевые знаки*:

- { • кибернетическая система
 - ⇒ разбиение*:
 - { • естественная кибернетическая система
 - компьютерная система
 - := [искусственная кибернетическая система]
 - естественно-искусственная кибернетическая система
 - := [кибернетическая система, являющаяся симбиозом компонентов как естественного, так и искусственного происхождения]
- качество кибернетической системы
- физическая оболочка кибернетической системы
- качество физической оболочки кибернетической системы
- интеллект
 - := [уровень интеллекта кибернетической системы]
 - := [интеллектуальность]
- интеллектуальная система
 - := [интеллектуальная кибернетическая система]
 - ⊃ интеллектуальная компьютерная система
- информация, хранимая в памяти кибернетической системы
- качество информации, хранимой в памяти кибернетической системы
- база знаний
- смысловое представление информации в памяти кибернетической системы
- решатель задач кибернетической системы
- качество решателя задач кибернетической системы
- память кибернетической системы
- качество памяти кибернетической системы
- обучаемость кибернетической системы
- гибкость кибернетической системы
- стратифицированность кибернетической системы
- рефлексивность кибернетической системы
 - := [уровень рефлексии кибернетической системы]
- многоагентная система
- качество многоагентной системы
- унифицированность агентов многоагентной системы
- семантическая совместимость агентов многоагентной системы
- социализация кибернетической системы
 - := [способность кибернетической системы своей внутренней и внешней деятельностью обеспечивать высокий уровень интеллекта тех многоагентных систем, членом (агентом) которых она является]

}

⇒

библиография*:

- { • Винер.Н.Кибернетика-1952кн
- Поспелов.Д.А.МоделированиеРОАМА-1989кн
- Финн.В.К.ПринципКонстрИнтелСистем-2008ст
- Ярушикина.Н.Г.НечетГС-2007кн
- Редько.В.Г.МоделКЭ-2019кн

}


```

← конкатенация сегментов*:
{
• Уточнение понятия кибернетической системы
• Комплекс свойств, определяющий общий уровень качества кибернетической системы
• Комплекс свойств, определяющих качество физической оболочки кибернетической системы
• Комплекс свойств, определяющих уровень интеллекта кибернетической системы
• Комплекс свойств, определяющий качество информации, хранимой в памяти кибернетической системы
• Комплекс свойств, определяющих качество решателя задач кибернетической системы
• Комплекс свойств, определяющих уровень обучаемости кибернетической системы
• Комплекс свойств, определяющих качество многоагентной системы
• Комплекс свойств, определяющих уровень социализации кибернетической системы как фактора существенного повышения уровня ее обучаемости, а также фактора существенного повышения качества всех тех многоагентных систем, в состав которых входит данная кибернетическая система
}

/* Сегмент *****/
Уточнение понятия кибернетической системы
⊃=
{
кибернетическая система
:= [система, которая способна управлять своими действиями, адаптируясь к изменениям состояния внешней среды (среды своего "обитания") в целях самосохранения (сохранения своей целостности и "комфортности" существования путем удержания своих "жизненно" важных параметров в определенных рамках "комфортности") и/или в целях формирования определенных реакций (воздействий на внешнюю среду) в ответ на определенные стимулы (на определенные ситуации или события во внешней среде), а также которая способна (при соответствующем уровне развития) эволюционировать в направлении:
□ изучения своей внешней среды как минимум для предсказания последствий своих воздействий на внешнюю среду, а также для предсказания изменений внешней среды, которые не зависят от собственных воздействий;
□ изучения самой себя и, в частности, своего взаимодействия с внешней средой;
□ создания технологий (методов и средств), обеспечивающих изменение своей внешней среды (условий своего существования) в собственных интересах.
]
:= [адаптивная система]
:= [целенаправленная (целеустремленная) система]
:= [активный субъект самостоятельной деятельности]
:= [материальная сущность, способная целенаправленно (в своих интересах) воздействовать на среду своего обитания как минимум для сохранения своей целостности, жизнеспособности, безопасности]
⇒ примечание*:
[Уровень (степень) адаптивности, целенаправленности, активности у систем, основанных на обработке информации может быть самым различным.]
:= [система, организация функционирования которой основано на обработке информации о той среде, в которой существует эта система]
:= [материальная сущность, способная к активной целенаправленной деятельности, которая на определенном уровне развития указанной сущности становится "осмысленной планируемой, преднамеренной деятельностью]
:= [субъект, способный на самостоятельное выполнение некоторых "внутренних" и "внешних" действий либо порученных извне, либо инициированных самим субъектом]
:= [сущность, способная выполнять роль субъекта деятельности]
:= [естественная или искусственно созданная система, способная мониторить и анализировать свое состояние и состояние окружающей среды, а также способная достаточно активно воздействовать на собственное на собственное состояние и на состояние окружающей среды]
:= [система, способная в достаточной степени самостоятельно взаимодействовать со своей средой , решая различные задачи]
:= [система, основанная на обработке информации]

```

← *ключевой знак**:
 Глушков.В.М.Кибернетика-1979ст
 € *статья*

/*****/

Типология кибернетических систем

⊃=

{

кибернетическая система

⇒ *разбиение**:

Признак естественности или искусственности кибернетических систем

= { • *естественная кибернетическая система*
 := [кибернетическая система естественного происхождения]
 ⊃ *человек*
 • *компьютерная система*
 := [искусственная кибернетическая система]
 := [кибернетическая система искусственного происхождения]
 := [технически реализованная кибернетическая система]
 • *симбиоз естественных и искусственных кибернетических систем*
 := [кибернетическая система, в состав которой входят компоненты как естественного, так
 и искусственного происхождения]
 ⊃ *сообщество компьютерных систем и людей*
 }

искусственная сущность

:= [артефакт]

:= [сущность, являющаяся либо результатом человеческой деятельности, либо частью самой этой деятельности]

:= [сущность искусственного происхождения]

:= [антропогенная сущность]

⊃ *научно-техническое знание*

:= [знание, приобретенное в результате научно-технической деятельности человеческого общества]

⊃ *материальная искусственная сущность*

⊃ *компьютерная система*

компьютерная система

:= [искусственная кибернетическая система]

⇒ *примечание**:

[Особенностью компьютерных систем является то, что они могут выполнять "роль" не только продуктов соответствующих действий по реализации этих систем, но и сами являются *субъектами**, способными выполнять (автоматизировать) широкий спектр действий. При этом интеллектуализация этих систем существенно расширяет этот спектр. См. *интеллектуальная компьютерная система.*]

:= [технически реализованная кибернетическая система]

:= [искусственная кибернетическая система]

⊃ *кибернетическая система*

⊃ *современная компьютерная система традиционного вида*

⊃ *современная интеллектуальная компьютерная система*

⊃ *интеллектуальная компьютерная система следующего поколения*

⊃ *ostis-система*

⇒ *примечание**:

[Основной тенденцией эволюции компьютерных систем является повышение уровня их интеллектуальности.]

⇒ *особенность**:

{ • [Ориентация на принципиально новые компьютеры]
 • [Существенное повышение уровня интеллекта]
 }

⇒ разбиение*:
Структурная классификация кибернетических систем
 = {

- *простая кибернетическая система*
- *индивидуальная кибернетическая система*
- *многоагентная система*

 ⇒ разбиение*:
 {

- *одноуровневый коллектив кибернетических систем*
 := [многоагентная система, агентами которой не могут быть многоагентные системы]
- *иерархический коллектив кибернетических систем*
 := [многоагентная система, по крайней мере одним агентом которой является многоагентная система]

 }
 ⇒ разбиение*:
 {

- *коллектив из простых кибернетических систем*
 ⇒ *примечание**:
 [Такой коллектив может быть либо одноуровневым, либо иерархическим коллективом]
- *коллектив из индивидуальных кибернетических систем*
- *коллектив из индивидуальных и простых кибернетических систем*

 }
 }

кибернетическая система

⇒ разбиение*:
Классификация кибернетических систем по признаку наличия надсистемы и роли в рамках этой надсистемы
 = {

- *кибернетическая система, не являющаяся частью никакой другой кибернетической системы*
 := [кибернетическая система, не имеющая надсистем]
- *кибернетическая система, встроенная в индивидуальную кибернетическую систему*
- *агент многоагентной системы*
 := [кибернетическая система, являющаяся агентом одной или нескольких многоагентных систем]

 }

простая кибернетическая система

:= [кибернетическая система, уровень развития которой находится ниже уровня *индивидуальных кибернетических систем* и которая является специализированным средством обработки информации специализированным решателем задач, реализующим (интерпретирующим) чаще всего один *метод* решения задач и, соответственно, решающим только *задачи* заданного класса задач]
 := [специализированный *решатель задач*]
 ⇒ *примечание**:
 [простая кибернетическая система может быть компонентом*, встроенным в индивидуальную кибернетическую систему, а также может быть агентом* многоагентной системы, являющейся коллективом из простых кибернетических систем]

индивидуальная кибернетическая система

:= [условно выделенный уровень развития *кибернетических систем*, в основе которого лежит переход от специализированного решателя задач к индивидуальному решателю, обеспечивающему интерпретацию произвольного (нефиксированного) набора *методов* (программ) решения задач при условии, если эти *методы* введены (загружены, записаны) в память кибернетической системы]
 := [кибернетическая система, способная быть самостоятельной]
 ⇒ *пояснение**:
 [Признаками индивидуальных кибернетических систем являются:

- наличие *памяти*, предназначенной для хранения как минимум интерпретируемых *методов* (программ) и обеспечивающей корректировку (редактирование) хранимых *методов*, а также их удаление из *памяти* и ввод (запись) в *память* новых *методов*;

]

- легкая возможность "перепрограммировать" *кибернетическую систему* на решение других задач, что обеспечивается наличием *универсальной модели решения задач* и, соответственно, *универсальным интерпретатором любых моделей*, представленных (записанных) на соответствующем языке;
- наличие пусть даже простых средств коммуникации (обмена информацией) с другими *кибернетическими системами* (например, с людьми);
- способность входить в различные *коллективы кибернетических систем*.

]

⇒ *примечание**:

[класс *индивидуальных кибернетических систем* — это определенный этап эволюции кибернетических систем, означающий переход к кибернетическим системам, которые способны самостоятельно "выживать"]

:= [самостоятельная автономная, целостная кибернетическая система]

:= [субъект деятельности]

⇒ *примечание**:

[*индивидуальная кибернетическая система* может быть агентом (членом) многоагентной системы (членом коллектива индивидуальных кибернетических систем), но некоторые многоагентные системы могут состоять из агентов, не являющихся *индивидуальными кибернетическими системами*, представляющих собой простые специализированные кибернетические системы, выполняющие достаточно простые действия (*Стефанюк.В.Л.ЛокалОргИС-2004кн, Д.фон.Нейман.ТеорСамовосАвтомат-1971кн*)]

:= [кибернетическая система, которая обладает достаточной самостоятельностью (целостностью), но не является коллективом таких самостоятельных кибернетических систем]

:= [минимальная самостоятельная (самодостаточная, в известной степени автономная) кибернетическая система]

:= [индивидуальный субъект]

кибернетическая система, встроенная в индивидуальную кибернетическую систему

⊃ *пример**:

sc-агент ostis-системы

⊃ *пример**:

решатель задач ostis-системы

многоагентная система

:= [коллектив взаимодействующих автономных кибернетических систем, имеющих общую среду обитания (жизнедеятельности)]

⇒ *разбиение**:

- *одноуровневая многоагентная система*
- *иерархическая многоагентная система*

одноуровневая многоагентная система

:= [специализированное средство решения задач, реализующее либо одну модель параллельного (распределенного) решения задач соответствующего класса, либо комбинацию фиксированного числа разных и параллельно реализованных моделей решения задач]

⇒ *разбиение**:

- *одноуровневая однородная многоагентная система*
- *одноуровневая неоднородная многоагентная система*

коллектив индивидуальных кибернетических систем

⊂ *многоагентная система*

:= [многоагентная система, агентами (членами) которой являются индивидуальные(!) кибернетические системы]

⇒ *разбиение**:

- *коллектив людей*
:= [человеческое сообщество]
- *сообщество компьютерных систем и людей*

иерархический коллектив индивидуальных кибернетических систем

:= [многоагентная система, агентами (членами) которой могут быть:

- индивидуальные кибернетические системы;

- коллективы индивидуальных кибернетических систем;
- коллективы, состоящие из индивидуальных кибернетических систем и коллективов индивидуальных кибернетических систем и т.д.

]

}

/*****

Структура кибернетической системы

⊃=

{

кибернетическая система

⇒ обобщенная декомпозиция*:

- { • информация, хранимая в памяти кибернетической системы
- абстрактная память кибернетической системы
- решатель задач кибернетической системы
- физическая оболочка кибернетической системы

}

информация, хранимая в памяти кибернетической системы

:= [информация, хранимая в памяти кибернетической системы и представляющая собой информационную модель среды, в которой действует (существует, функционирует) эта кибернетическая система]

:= [текущее состояние памяти кибернетической системы]

:= [текущее состояние внутренней (информационной) среды кибернетической системы]

← второй домен*:

информация, хранимая в памяти кибернетической системы*

∈ бинарное отношение

∈ ориентированное отношение

абстрактная память кибернетической системы

:= [внутренняя абстрактная информационная среда кибернетической системы, представляющая собой динамическую информационную конструкцию, каждое состояние которой есть не что иное, как информация, хранимая в памяти кибернетической системы в соответствующий момент времени]

:= [абстрактная динамическая модель памяти кибернетической системы]

С динамическая информационная конструкция

:= [процесс преобразования информационной конструкции]

решатель задач кибернетической системы

:= [совокупность всех навыков (умений), приобретенных кибернетической системой к рассматриваемому моменту]

:= [встроенный в кибернетическую систему субъект, способный выполнять целенаправленные ("осознанные") действия во внешней среде этой кибернетической системы, а также в её внутренней среде (в абстрактной памяти)]

действие кибернетической системы

С действие

:= [целенаправленное ("осознанное") действие, выполняемое кибернетической системой, а точнее, её решателем задач]

⇒ разбиение*:

- { • внешнее действие кибернетической системы

:= [действие, выполняемое кибернетической системой в её внешней среде]

:= [поведенческое действие]

- действие кибернетической системы, выполняемое в собственной физической оболочке

- действие кибернетической системы, выполняемое в собственной абстрактной памяти

:= [речь идёт о действиях, направленных на преобразование информации, хранимой в памяти, но никак не на преобразование физической памяти (физической оболочки абстрактной памяти)]

}

⇒	<i>примечание*</i> : [Каждое <u>сложное</u> действие, выполняемое кибернетической системой вне собственной абстрактной памяти, включает в себя поддействия, выполняемые в указанной абстрактной памяти. Это означает, что все внешние действия кибернетической системы <u>управляются</u> внутренними её действиями (действиями в абстрактной памяти).]
	задача
:=	[спецификация действия]
:=	[формулировка задачи с различной степенью детализации (уточнения) специфицируемого (описываемого) действия, в состав которой может входить: □ описание цели (целевой ситуации); □ указание объектов (аргументов) действия; □ указание типа действия (класса действий, которому принадлежит данное действие); □ указание субъекта действия; □ указание инструмента (средств) выполненного действия; □ и др.]
⇒	<i>примечание*</i> : [Процесс решения задачи и действие, специфицируемое этой задачей (точнее, процесс выполнения этого действия) суть одно и то же.]
	задача, решаемая кибернетической системой
:=	[задача, решаемая соответствующей кибернетической системой]
:=	[Второй домен отношения "быть задачей, решаемой заданной кибернетической системой*"]
⇔	<i>следует отличать*</i> : <i>задача, решаемая кибернетической системой*</i>
:=	[быть задачей, решаемой заданной кибернетической системой*]
⇒	<i>разбиение*</i> : { • <i>задача, решаемая кибернетической системой во внешней среде</i> := [внешняя задача кибернетической системы] := [задача, направленная на изменение состояния внешней среды соответствующей кибернетической системы, но включающая в себя (в качестве подзадач) задачи, решаемые в памяти кибернетической системы, например: □ интерфейсные задачи (анализ первичной информации о текущем состоянии внешней среды), □ сенсо-моторную координацию выполнения сложных действий во внешней среде, состоящих из большого количества частных (более простых) действий, находящихся на разных уровнях иерархии, □ задачи планирования целенаправленного поведения во внешней среде, □ задачи принятия решений.] • <i>задача, решаемая кибернетической системой в собственной физической оболочке</i> • <i>задача, решаемая кибернетической системой в абстрактной памяти</i> := [задача, полностью решаемая в памяти кибернетической системы и направленная на изменение состояния информации, хранимой в памяти кибернетической системы] := [внутренняя задача кибернетической системы] }
	навык
С	<i>знание</i>
⇒	<i>пояснение*</i> : [знание частного вида, содержащее (1) некоторый метод – знание о том, как можно решать задачи, принадлежащие соответствующему множеству задач, (2) полное знание о том, как указанный метод следует интерпретировать (реализовывать), декомпозируя исходные задачи на подзадачи и, в конечном счёте на элементарные действия, выполняемые <i>процессором кибернетической системы</i>]
:=	[умение]

- := [методы и средства, обеспечивающие способность *кибернетической системы* решать некоторое множество задач (выполнять некоторое множество действий)]
- интерфейс кибернетической системы**
- := [условно выделяемый компонент *решателя задач кибернетической системы*, обеспечивающий решение *интерфейсных задач*, направленных на непосредственную реализацию взаимодействия *кибернетической системы* с её *внешней средой*]
- := [решатель интерфейсных задач кибернетической системы]
- ⇐ *обобщенная часть**:
решатель задач кибернетической системы
- ⇔ *следует отличать**:
физическое обеспечение интерфейса кибернетической системы
- ⇐ *обобщенная часть**:
физическая оболочка кибернетической системы
- физическая оболочка кибернетической системы**
- ⇒ *обобщенная декомпозиция**:
- *память кибернетической системы*
 - *процессор кибернетической системы*
 - *физическое обеспечение интерфейса кибернетической системы*
- := [аппаратное обеспечение интерфейса кибернетической системы с её внешней средой]
- ⇒ *обобщенная декомпозиция**:
- *сенсорная подсистема физической оболочки кибернетической системы*
 - *эффлекторная подсистема физической оболочки кибернетической системы*
- *корпус кибернетической системы*
- }
- физическая оболочка кибернетической системы**
- := [часть кибернетической системы, являющаяся "посредником" между её внутренней средой (памятью, в которой хранится и обрабатывается информация кибернетической системы) и её внешней средой]
- ⇐ *второй домен**:
*физическая оболочка кибернетической системы**
- ∈ *бинарное отношение*
- ∈ *ориентированное отношение*
- память кибернетической системы**
- := [физическая оболочка (реализация) абстрактной *памяти кибернетической системы* – внутренней среды *кибернетической системы*, в рамках которой *кибернетическая система* формирует и использует (обрабатывает) информационную модель своей внешней среды]
- ⇒ *примечание**:
[Не каждая *кибернетическая система* имеет *память*. В *кибернетических системах*, которые не имеют *памяти*, обработка информации сводится к обмену сигналами между компонентами этих систем. Появление в *кибернетических системах* памяти как среды для "централизованного" хранения и обработки *информации* является важнейшим этапом их эволюции. Дальнейшая эволюция *кибернетических систем* во многом определяется:
- *качеством памяти* как среды для хранения и обработки информации;
 - *качеством информации* (информационной модели), хранимой в памяти кибернетической системы;
-]
- := [компонент кибернетической системы, в рамках которого *кибернетическая система* осуществляет отображение (формирование информационной модели) среды своего существования, а также использование этой информационной модели для управления собственным поведением в указанной среде]
- := [физическая оболочка для хранения информации, которую кибернетическая система приобретает и обрабатывает (т.е. меняет состояния этой информации)]
- := [физическая (аппаратная) реализация внутренней среды кибернетической системы, каковой является среда "существования" информации, накапливаемой и непосредственно используемой решателем задач этой кибернетической системы]

- ⇒ *примечание**:
[Сам факт появления в кибернетической системе памяти, которая (1) обеспечивает представление различного вида информации о среде, в рамках которой кибернетическая система решает различные задачи (выполняет различные действия), (2) обеспечивает хранение достаточно полной информационной модели указанной среды (достаточно полной для реализации своей деятельности), (3) обеспечивает высокую степень гибкости указанной хранимой в памяти информационной модели среды жизнедеятельности (т.е. лёгкость внесения изменений в эту информационную модель), существенно повышает уровень адаптивности кибернетической системы к различным изменениям своей среды.]
- ⇒ *примечание**:
["появление" *памяти* в кибернетических системах является основным признаком перехода от "простых" автоматов к компьютерным системам, от роботов 1-го поколения к роботам следующих поколений]
- := [физическая реализация хранилища информации, которую приобрела (накопила) к текущему моменту соответствующая кибернетическая система]
- := [физическая оболочка внутренней абстрактной информационной среды кибернетической системы]
- := [среда хранения и обработки информации]
- := [запоминающая среда]
- := [среда хранения и обработки информационных конструкций]
- ⇒ *примечание**:
[Принципы организации памяти кибернетической системы могут быть разными (ассоциативная, адресная, структурно фиксированная/структурно перестраиваемая, нелинейная/линейная). От организации памяти во многом зависит её качество.]

кибернетическая система

- ⇒ *уровни эволюции**:
Уровни структурной эволюции кибернетических систем
- = {
- *простая кибернетическая система, не имеющая памяти*
 - *простая кибернетическая система, имеющая память*
 - *одноуровневый коллектив, не имеющий общей памяти и состоящий из простых кибернетических систем, не имеющих памяти*
 - *одноуровневый коллектив, не имеющий общей памяти и состоящий из простых кибернетических систем, имеющих память*
 - *иерархический коллектив, имеющий общую память и состоящий из простых кибернетических систем*
 - *индивидуальная кибернетическая система*
⇒ *примечание**:
[Каждая *индивидуальная кибернетическая система* содержит *память*, имеющую достаточно высокий уровень качества]
 - *одноуровневый коллектив индивидуальных кибернетических систем, не имеющий общей памяти*
 - *одноуровневый коллектив индивидуальных кибернетических систем, имеющий общую память*
 - *иерархический коллектив из индивидуальных кибернетических систем, не имеющий общей памяти*
 - *иерархический коллектив из индивидуальных кибернетических систем, имеющий общую память*
- }

процессор кибернетической системы

- := [физически (аппаратно реализованный) интерпретатор хранимых в памяти кибернетической системы методов (программ), соответствующих базовой (для данной кибернетической системы) модели решения задач, т.е. такой модели решения задач, которая для данной кибернетической системы является моделью решения задач самого нижнего уровня и, следовательно, не может быть интерпретирована с помощью другой модели решения задач, используемой этой же кибернетической системой, а может быть проинтерпретирована либо путем аппаратной реализации такого интерпретатора, либо путём его программной реализации, например, на современных компьютерах, но в последнем случае, кроме собственного интерпретатора, необходимо также построить модель памяти реализуемой кибернетической системы]
- := ["физически" реализованные средства, обеспечивающие выполнение "элементарных" действий, направленных на изменение состояния памяти кибернетической системы (на изменение информации, хранимой в этой памяти)]
- := ["движок" ("мотор") кибернетической системы]

\Leftarrow *второй домен**:
*процессор кибернетической системы**
 $:=$ *пояснение**:
 [бинарное ориентированное отношения, каждая пара которого связывает знак кибернетической системы со знаком её процессора]
 \in *бинарное отношение*
 \in *ориентированное отношение*

компьютер

\subset *физическая оболочка кибернетической системы*
 $:=$ [физическая оболочка искусственной кибернетической системы]
 $:=$ [аппаратное обеспечение компьютерной системы]
 $:=$ [hardware of computer system]
 \supset *компьютер для интеллектуальных систем*
 $:=$ [компьютер, ориентированный на реализацию интеллектуальных компьютерных систем]
 \Rightarrow *примечание**:
 [Развитие рынка интеллектуальных компьютерных систем существенно сдерживается неприспособленностью современного поколения компьютеров к реализации на их основе интеллектуальных компьютерных систем. Попытки создания компьютеров, приспособленных к реализации интеллектуальных компьютерных систем, не привели к успеху, т.к. эти проекты были направлены на выполнение отдельных (частных) требований, предъявляемых к физическому (аппаратному) уровню интеллектуальных систем, что неминуемо приводило к приспособленности создаваемых компьютеров к реализации не всего многообразия интеллектуальных компьютерных систем, а только некоторых подмножеств таких систем. Указанные подмножества интеллектуальных компьютерных систем в основном определялись ориентацией на конкретные используемые модели решения интеллектуальных задач, тогда, как важнейшим фактором, определяющим уровень интеллекта кибернетических систем (в том числе, и компьютерных систем), является их универсальность в плане многообразия используемых моделей решения задач. Следовательно, компьютер для интеллектуальных компьютерных систем должен быть эффективным аппаратным интерпретатором любых моделей решения задач (как интеллектуальных задач, так и достаточно простых задач, т.к. интеллектуальная система должна уметь решать любые задачи).]
 $:=$ [компьютер, приспособленный к реализации интеллектуальных компьютерных систем]
 $:=$ [универсальный компьютер для интеллектуальных систем]
 $:=$ [компьютер, обеспечивающий интерпретацию любых моделей решения задач]

}

/*****

Семейство отношений, заданных на множестве кибернетических систем

$\supset =$
 {

отношение, заданное на множестве кибернетических систем

\ni *память кибернетической системы**
 \ni *процессор кибернетической системы**
 \ni *член коллектива**
 \ni *внешняя среда кибернетической системы**
 \ni *сенсор кибернетической системы**
 \ni *эффектор кибернетической системы**
 \ni *физическая оболочка кибернетической системы**
 \ni *информация, хранимая в памяти кибернетической системы**
 \ni *абстрактная память кибернетической системы**
 \ni *часть**
 \supset *встроенная кибернетическая система**

информация, хранимая в памяти кибернетической системы*

$:=$ [информационная модель среды*, в которой существует (осуществляет деятельность) соответствующая кибернетическая система*]

⇒	<i>примечание*</i> : [От того, насколько полна, адекватна (корректна) и систематизирована (структурирована) внутренняя среда кибернетической системы, зависит уровень интеллектуальности и эффективность соответствующей кибернетической системы.]
	<i>следует отличать*</i>
⊃	{ <ul style="list-style-type: none"> • <i>решатель задач кибернетической системы*</i> • <i>решатель задач кибернетической системы</i> := [иерархическая система моделей решения задач] ⇒ <i>обобщённая часть*</i> : <i>процессор кибернетической системы</i> ⇒ <i>пояснение*</i> : [Это реализация модели решения задач, обеспечивающей интерпретацию всех используемых моделей решения задач верхнего уровня]
	}
	<i>задача, решаемая кибернетической системой*</i>
:=	[быть задачей, решаемой заданной кибернетической системой*]
⊃	<i>задача, решаемая в памяти кибернетической системы*</i>
:=	[внутренняя задача кибернетической системы*]
⊃	<i>задача, решаемая во внешней среде кибернетической системы*</i>
	<i>внешняя среда кибернетической системы*</i>
:=	[внешняя среда*]
⇒	<i>примечание*</i> : [Понятие <i>внешней среды кибернетической системы*</i> является понятием относительным, т.к. (1) разные кибернетические системы имеют в общем случае разную внешнюю среду и (2) одна кибернетическая система может входить в состав внешней среды другой кибернетической системы]
:=	[быть внешней средой для заданной кибернетической системы*]
∈	<i>бинарное отношение</i>
∈	<i>ориентированное отношение</i>
⇒	<i>первый домен*</i> : [кибернетическая система]
⊃	<i>внешняя информационная среда кибернетической системы*</i>
:=	[совокупность всевозможных информационных конструкций, к которым данная кибернетическая система имеет доступ и которые представлены самым различным образом (в том числе, и в памяти тех кибернетических систем (субъектов), с которыми данная система взаимодействует)*]
	<i>среда кибернетической системы*</i>
:=	[быть средой существования (жизнедеятельности) заданной (указанной, соответствующей) кибернетической системы*]
⇒	<i>примечание*</i> : [В общем случае среда жизнедеятельности <i>кибернетической системы</i> включает в себя (1) <i>внешнюю среду*</i> этой системы, (2) <i>физическую оболочку*</i> этой системы и (3) её <i>абстрактную память</i> , т.е. внутреннюю среду*, которая является хранилищем информационной модели всей среды]
⇒	<i>разбиение*</i> : { <ul style="list-style-type: none"> • <i>внешняя среда*</i> • <i>физическая оболочка*</i> • <i>абстрактная память*</i> }
}	
}	
}	/* Завершили Сегмент “Уточнение понятия кибернетической системы” */

```

/* Сегмент *****/
Комплекс свойств, определяющий общий уровень качества кибернетической
системы
⊃=
{
качество кибернетической системы
:= [интегральный уровень качества кибернетической системы в заданный момент]
:= [комплексная оценка (характеристика) уровня качества кибернетической системы]
⇒ пояснение*:
[Для того, чтобы уточнить (детализировать) понятие качества кибернетической системы, необходимо
□ задать метрику качества кибернетических систем и
□ построить иерархическую систему свойств (параметров, признаков), определяющих качество киберне-
тической системы.
]
∈ упорядоченное свойство
:= [эволюционный уровень кибернетической системы]
:= [интегральная (комплексная) оценка уровня развития (совершенства) кибернетической системы]
⇒ пояснение*:
[Качество кибернетической системы – это такое свойство (характеристика) кибернетических систем, такой
признак их классификации, который позволяет разместить эти системы по "ступенькам" некоторой условной
"эволюционной лестницы". На каждую такую "ступеньку" попадают кибернетические системы, имеющие
одинаковый уровень развития, каждому из которых соответствует свой набор значений дополнительных
свойств кибернетических систем, которые уточняют (детализируют, специализируют) соответствующий
уровень развития кибернетических систем. Такой эволюционный подход к рассмотрению кибернетических
систем даёт возможность, во-первых, детализировать направления эволюции кибернетических систем
и, во-вторых, уточнить то место этой эволюции, где и благодаря чему осуществляется переход от неин-
теллектуальных кибернетических систем к интеллектуальным. Фактически речь идёт об эволюционной
теории качества кибернетических систем, рассматривающей эволюцию кибернетических систем как
в рамках жизненного цикла каждой из них, так и в рамках эволюции целой "популяции" при переходе
от одного поколения кибернетических систем к другому поколению (в частности, от одного поколения
компьютерных систем к другому).
В основе эволюционного подхода к рассмотрению многообразия кибернетических систем лежит положение
о том, что идеальных кибернетических систем не существует, но существует постоянное стремление к
идеалу, к большему совершенству. При этом важно уточнить, что конкретно в каждой кибернетической
системе следует изменить, чтобы привести эту систему к более совершенному виду.
Эволюционный подход к рассмотрению кибернетических систем имеет важное практическое значение
для развития (совершенствования) каждой конкретной компьютерной системы (искусственной кибернети-
ческой системы), а также для развития технологий разработки компьютерных систем. Так, например,
развитие технологий разработки компьютерных систем должно быть направлено на переход к таким новым
архитектурным и функциональным принципам, лежащим в основе компьютерных систем, которые
□ обеспечивают существенное снижение трудоемкости их разработки и сокращение сроков разработки, а
также
□ обеспечивают существенное повышение уровня интеллекта и, в частности, уровня обучаемости
разрабатываемых компьютерных систем, например, путём перехода от поддержки обучения с учителем
к реализации эффективного самообучения (к автоматизации организации самостоятельного обучения).
]
⇒ примечание*:
[В эволюции кибернетических систем (и, в частности, компьютерных систем) можно выделить целый ряд
этапов:
□ переход от стимульно-реактивного поведения к поведению, предполагающему учёт постоянно накап-
ливаемого собственного опыта, означает переход от протопамяти, которая просто фиксирует связи
между стимулами и соответствующими реакциями и которая не предполагает изменения этих связей,
к памяти, которая становится средой "существования" информации, отражающей собственный опыт
кибернетической системы (а в перспективе и многое другое) и которая обеспечивает высокую степень
гибкости хранимой информации, т.е. широкие возможности изменения (корректировки) этой информа-
ции в процессе функционирования кибернетической системы. Таким образом, память кибернетической
системы вместе с хранимой в ней информацией становится управляемым самой этой кибернетической

```

системой гибким "коммутатором" между её стимулами и реакциями, учитывающим не только накапливаемый собственный опыт, но и контекст (дополнительные обстоятельства) выполняемых *действий* (реакций), рассматривающий выполняемые *действия* с самых разных аспектов;

- включение в состав *информации, хранимой в памяти компьютерной системы, программ*, описывающих различные *методы* обработки этой *информации* и интерпретируемых *процессором* указанной *компьютерной системы*;
- переход от указанной выше "коммутационной" трактовки *информации, хранимой в памяти кибернетической системы* к её трактовке как мощной и постоянно совершенствуемой информационной модели внешней среды, в которой существует указанная *кибернетическая система*. Это означает
 - переход *информации, хранимой в памяти кибернетической системы* на уровень *базы знаний*, которой ставится в *соответствие* достаточно чёткая *денотационная семантика*, и
 - переход *программ, хранимых в памяти кибернетической системы*, на уровень *программ*, которые ориентированы на обработку *базы знаний* и которые сами являются частью обрабатываемой *базы знаний*;
- существенное расширение *семантической мощности баз знаний* и многообразия используемых *моделей решения задач*, в том числе, *моделей*, способных работать в условиях неполноты (недостаточности), нечеткости и недостоверности обрабатываемых *знаний*.

]

⇒ *примечание**:

[Повышение качества искусственных *кибернетических систем (компьютерных систем)* потребует формирования таких свойств (характеристик, способностей) *компьютерных систем*, которые аналогичны психическим свойствам людей. Таким образом, дальнейшее развитие *Искусственного интеллекта* (теории и практики создания *интеллектуальных компьютерных систем* – интеллектуальных искусственных *кибернетических систем*) настоятельно потребует обобщения современной психологии (психологии биологических индивидов и их коллективов – *психологии естественных кибернетических систем*) и создания *общей психологии кибернетических систем* (как естественных, так и искусственных) основанной на высоком уровне формализации.]

качество кибернетической системы

⇒ *свойство-предпосылка**:

- *качество физической оболочки кибернетической системы* мы *качество решателя задач кибернетической системы*
- *качество решателя задач кибернетической системы*
⇒ *свойство-предпосылка**:
качество информации, хранимой в памяти кибернетической системы
- *качество информации, хранимой в памяти кибернетической системы*
- *гибридность кибернетической системы*
:= [степень многообразия (1) видов знаний, хранимых в памяти кибернетической системы, (2) используемых моделей решения задач, (3) видов сенсоров и эффекторов]
⇒ *частное свойство**:
 - *многообразие видов знаний, хранимых в памяти кибернетической системы*
 - *многообразие моделей решения задач*
 - *многообразие видов сенсоров и эффекторов*
- *приспособленность кибернетической системы к её совершенствованию*
- *производительность кибернетической системы*
:= [скорость решения задач кибернетической системы]
- *надёжность кибернетической системы*
- *социализация кибернетической системы*

гибридность кибернетической системы

⇒ *частное свойство**:

- *многообразие видов знаний, хранимых в памяти кибернетической системы*
- *многообразие моделей решения задач*
- *многообразие видов сенсоров и эффекторов*

гибридная кибернетическая система

:= [кибернетическая система, использующая многообразие рецепторных и/или эффекторных подсистем, и/или многообразие видов обрабатываемой информации, и/или многообразие способов решения задач]

⊃ *гибридная компьютерная система*
 := [компьютерная система, способная решать комплексные задачи, требующие использования многообразия различных видов обрабатываемой информации и различных моделей решения задач]

приспособленность кибернетической системы к её совершенствованию

:= [приспособленность кибернетической системы к эволюции, к повышению уровня своего качества]

⇒ комплекс свойств-предпосылок*:

{ • *обучаемость кибернетической системы*

:= [способность кибернетической системы самостоятельно повышать уровень своего качества]

:= [способность кибернетической системы к самоэволюции, саморазвитию, устранению своих недостатков]

• *приспособленность кибернетической системы к её совершенствованию, осуществляемому извне*

:= [приспособленность кибернетической системы к её совершенствованию, осуществляемому внешними субъектами]

:= [удобство совершенствования кибернетической системы для её создателей]

⇒ *примечание**:

[Важнейшим фактором качества каждой технологии разработки кибернетических систем является гибкость и стратифицированность разрабатываемых кибернетических систем при их совершенствовании, осуществляемом руками разработчиков]

}

⇒ комплекс свойств-предпосылок*:

{ • *гибкость кибернетической системы*

• *стратифицированность кибернетической системы*

:= [уровень стратифицированности кибернетической системы]

:= [качество разделения (декомпозиции) кибернетической системы на в достаточной степени независимые части (компоненты), определенные виды изменений которых не предполагают внесения изменений в другие части системы]

}

гибкость кибернетической системы

:= [реконфигурируемость кибернетической системы]

:= [модифицируемость кибернетической системы]

:= [реформируемость кибернетической системы]

:= [трансформируемость кибернетической системы]

:= [пластичность кибернетической системы]

:= [легкость реализации различного вида изменений в кибернетической системе]

:= [степень трансформенности кибернетической системы]

:= [простота внесения изменений в кибернетическую систему и многообразие видов возможных таких изменений]

:= [модифицируемость кибернетической системы]

:= [трансформируемость кибернетической системы]

:= [реконфигурируемость кибернетической системы]

:= [приспособленность к реинжинирингу кибернетической системы]

:= [мягкость]

:= [softness]

:= [приспособленность к внесению изменений]

:= [легкость внесения изменений]

⇒ *примечание**:

[Чем легче вносить изменения в кибернетическую систему, тем выше скорость ее эволюции]

⇒ *примечание**:

[изменения могут вноситься (1) полностью самостоятельно (без учителя) (2) с помощью учителя-тренера ("терапевта") путем создания определенных условий для совершенствования системы (3) "хирургически" – путем непосредственного вмешательства извне (например, вмешательства разработчика)]

- ⇒ *примечание**:
[Чем выше *гибкость кибернетической системы* – тем ниже трудоемкость и меньше сроки внесения различных изменений в систему в направлении ее совершенствования (приближения к идеалу)]
- ⇒ *комплекс свойств-предпосылок**:
{
 - *простота внесения изменений в кибернетическую систему*
⇒ *свойство-предпосылка**:
стратифицированность кибернетической системы
 - *многообразии, возможных изменений, вносимых в кибернетическую систему*
}
- ⇒ *комплекс частных свойств**:
{
 - *гибкость информации, хранимой в памяти кибернетической системы*
 - *гибкость решателя задач кибернетической системы*
 - *гибкость физической оболочки кибернетической системы*
⇒ *частное свойство**:
гибкость памяти кибернетической системы
 - *гибкость интерфейса кибернетической системы*
}
- ⇒ *комплекс частных свойств**:
{
 - *гибкость кибернетической системы при ее совершенствовании, осуществляемом извне*
 - *гибкость возможных самоизменений кибернетической системы*
← *свойство-предпосылка**:
обучаемость кибернетической системы
}

приспособленность кибернетической системы к её совершенствованию, осуществляемому извне

- := [приспособленность кибернетической системы к "хирургическим" методам её совершенствования, реализуемым разработчиками]
- := [насколько легко осуществлять обновление, перепроектирование, тестирование, ремонт (исправление ошибок) кибернетической системы]
- ⇒ *свойство-предпосылка**:
 - *простота внесения изменений в кибернетическую систему, реализуемых извне*
⇒ *свойство-предпосылка**:
стратифицированность кибернетической системы
 - *многообразии возможных изменений кибернетической системы, реализуемых извне*

производительность кибернетической системы

- := [быстродействие кибернетической системы]
- := [интегральная оценка скорости решения задач, время реакции кибернетической системы на задачные ситуации]
- ⇒ *частное свойство**:
 - *производительность базового интерпретатора логико-семантической модели кибернетической системы*
 - *качество используемых кибернетической системой методов и моделей решения задач*

надежность кибернетической системы

- := [способность кибернетической системы при соответствующих условиях ее функционирования сохранять (и, точнее, не снижать) уровень всех свойств и способностей, определяющих общее (комплексное) качество кибернетической системы]
- ⇒ *свойство-предпосылка**:
 - *безотказность кибернетической системы*
 - *долговечность кибернетической системы*
 - *"ремонтпригодность" кибернетической системы*
⇒ *основной sc-идентификатор**:
"ремонтпригодность" кибернетических систем
⇒ *примечание**:
[Здесь слово "ремонтпригодность" взято в кавычки, т.к. речь идет не только об искусственных (технических) кибернетических системах]
}

```

/* Сегмент *****/
Комплекс свойств, определяющих качество физической оболочки
кибернетической системы
⊃=
{
качество физической оболочки кибернетической системы
:= [интегральное качество "аппаратной" (физической) основы кибернетической системы]
:= ["hardware" кибернетической системы]
⇒ свойство-предпосылка*:


- качество памяти кибернетической системы
- качество процессора кибернетической системы
- качество сенсоров кибернетической системы
- качество эффекторов кибернетической системы
- приспособленность физической оболочки кибернетической системы к ее совершенствованию
- удобство транспортировки кибернетической системы
- надежность физической оболочки кибернетической системы

качество памяти кибернетической системы
⇐ свойство-предпосылка*:


- качество информации, хранимой в памяти кибернетической системы
- качество решателя задач кибернетической системы


⇒ свойство-предпосылка*:


- способность памяти кибернетической системы обеспечить хранение высококачественной информации
- способность памяти кибернетической системы обеспечить функционирование высококачественного решателя задач
- объем памяти

память кибернетической системы
:= [компонент кибернетической системы, представляющий собой "внутреннюю" среду кибернетической системы, в которой она хранит (запоминает) и преобразует информационную модель своей внешней среды. При этом важно, чтобы память обеспечивала высокий уровень гибкости указанной информационной модели. Важно также, чтобы эта информационная модель была моделью не только внешней среды кибернетической системы, но также и моделью самой этой информационной модели – описанием её текущей ситуации, предыстории, закономерностей. Таким образом, кибернетическая система, имеющая память, функционирует в двух средах – во внешней, в которой существуют и преобразуются внешние(материальные) сущности, и во внутренней, в которой существуют и преобразуются(обрабатываются) внутренние информационные конструкции.]
⇒ примечание*:
[Кибернетические системы, находящиеся на низком уровне развития(качества) памяти не имеют. Адаптационные механизмы такой кибернетической системы "жестко запаяны" в связях между блоками обработчика сигналов при переходе от сигналов, вырабатываемых сенсорами к сигналам, которые управляют эффекторами.]
:= [внутренняя среда кибернетической системы, обеспечивающая хранение и преобразование(обработку) информационной модели внешней среды кибернетической системы]
⇒ примечание*:
[Сам факт возникновения памяти в кибернетической системе является важнейшим этапом её эволюции. Дальнейшее развитие памяти кибернетической системы, обеспечивающее:


- хранение все более качественной информации, хранимой в памяти
- все более качественную организацию обработки этой информации, т.е. переход на поддержку(обеспечение) все более качественных моделей обработки информации

является важнейшим фактором эволюции кибернетических систем.]
способность памяти кибернетической системы обеспечить хранение высококачественной информации
⇒ свойство-предпосылка*:


- способность системы обеспечить компактное хранение сложноструктурированных баз знаний

```

- ⇒ *примечание**:
[Здесь имеется в виду необходимость перехода от линейной организации, памяти на физическом уровне (как последовательности ячеек памяти) к нелинейной, графодинамической памяти.]
- *способность памяти кибернетической системы обеспечить хранение широкого многообразия знаний*
- ⇒ *примечание**:
[имеется в виду хранение гибридных баз знаний]

способность памяти кибернетической системы обеспечить функционирование высококачественного решателя задач

- ⇒ *свойство-предпосылка**:
 - *качество доступа к информации, хранимой памяти кибернетической системы*
 - ⇒ *примечание**:
[Здесь имеется в виду необходимость перехода от адресного к ассоциативному доступу, причем, с расширением многообразия видов реализуемых запросов, в частности, к реализации запросов фрагментов баз знаний по заданному образцу произвольного размера и произвольной конфигурации.]
 - *логико-семантическая гибкость памяти кибернетической системы*
 - *способность памяти кибернетической системы обеспечить интерпретацию широкого многообразия моделей решения задач*

логико-семантическая гибкость памяти кибернетической системы

- := [степень близости физической организации памяти кибернетической системы к реализуемым ею базовым семантически целостным действиям над информацией, хранимой в памяти]
- := [простота реализации базовых семантически целостных действий над информацией, хранимой в памяти кибернетической системы]
- ⇒ *примечание**:
[Важен переход от "мелких" действий, к элементарным действиям, имеющим логико-семантический смысл (целостность, законченность)]

качество процессора кибернетической системы

- ⇐ *свойство-предпосылка**:
качество решателя задач кибернетической системы
- ⇒ *свойство-предпосылка**:
 - *способность процессора кибернетической системы обеспечить функционирования высококачественного решателя задач*
 - ⇒ *свойство-предпосылка**:
 - *многообразие моделей решения задач, интерпретируемых процессором кибернетической системы*
 - *простота и качество интерпретации процессором системы широкого многообразия моделей решения задач*
 - ⇒ *примечание**:
[Указанная простота определяется степенью близости интерпретируемых моделей решения задач к "физическому" уровню организации процессора кибернетической системы.]
 - *обеспечение процессором кибернетической системы качественного управления информационными процессами в памяти*
 - ⇒ *примечание**:
[Речь идет о грамотном сочетании таких аспектов управление процессами, как централизация и децентрализация, синхронность и асинхронность, последовательность и параллельность.]
 - ⇒ *свойство-предпосылка**:
уровень параллелизма обработки информации в памяти кибернетической системы
 - := [максимальное количество одновременно выполняемых информационных процессов в памяти кибернетической системы]
 - *быстродействие процессора кибернетической системы*

многообразие моделей решения задач, интерпретируемых процессором кибернетической системы⇒ *примечание**:

[Максимальным уровнем качества процессора кибернетической системы по данному параметру является его универсальность, т.е. его принципиальная возможность интерпретировать любую модель решения как интеллектуальных, так и неинтеллектуальных задач (алгоритмизацию, процедурную параллельную синхронную, процедурную параллельную асинхронную, продукционную, нейросетевую, генетическую, функциональную, целое семейство моделей).]

качество сенсоров кибернетической системы⇒ *свойство-предпосылка**:*многообразие видов сенсоров кибернетической системы*

:= [многообразие средств восприятия (отображения) информации о текущем состоянии внешней среды кибернетической системы и её собственной физической оболочки]

качество эффекторов кибернетической системы⇒ *свойство-предпосылка**:*многообразие видов эффекторов кибернетической системы*

:= [многообразие средств воздействия на собственную физическую оболочку кибернетической системы и через нее на внешнюю среду этой системы]

⇒ *примечание**:

[Эффекторы кибернетической системы являются инструментами воздействия кибернетической системы на свою внешнюю среду.]

приспособленность физической оболочки кибернетической системы к её совершенствованию

:= [приспособленность кибернетической системы к повышению качества её физической оболочки]

:= [простота ремонта и совершенствования таких компонентов кибернетической системы как память, процессор, сенсоры, эффекторы]

⇒ *частное свойство**:*ремонтпригодность физической оболочки кибернетической системы*⇒ *группа свойств-предпосылок**:

- *гибкость физической оболочки кибернетической системы*
- *стратифицированность физической оболочки кибернетической системы*

:= [мобильность физической оболочки кибернетической системы]

:= [легкость сохранения целостности физической оболочки кибернетической системы при внесении различных изменений (локализация области учета последствий внесения изменений, предсказуемость последствий)]

}

} /* Завершили Сегмент "Комплекс свойств, определяющих качество физической оболочки кибернетической системы" */

/* Сегмент *****/

Комплекс свойств, определяющих уровень интеллекта кибернетической системы

⊃=

{

интеллект∈ *свойство*∈ *упорядоченное свойство*

:= [уровень (степень, величина) интеллекта кибернетической системы]

:= [Семейство классов кибернетических систем, обладающих эквивалентным (одинаковым) уровнем интеллекта – от низкого до высокого уровня интеллекта]

:= [свойство кибернетических систем, характеризующее эффективность их взаимодействия со своей средой (средой их "жизнедеятельности")]

⇒ *область определения**:*кибернетическая система*

- ⇒ *пояснение**:
 [С формальной точки зрения интеллектуальность – это семейство классов кибернетических систем, в каждый из которых входят кибернетические системы, эквивалентные по уровню и характеру проявления интеллектуальных свойств (в том числе способностей).
 Таким образом, характер (вид) интеллектуальных свойств кибернетических систем и уровень их развития для разных кибернетических систем может быть разным. В соответствии с этим кибернетические системы можно сравнивать между собой.]
- ⇒ *примечание**:
 [Основным свойством (характеристикой, качеством, параметром) кибернетической системы является уровень (степень) ее интеллекта, который является интегральной характеристикой, определяющей уровень эффективности взаимодействия кибернетической системы со средой своего существования.]
- := [комплексное свойство (качество) кибернетической системы, определяющее уровень ее "выживаемости" во внешней среде и предполагающее возможность воздействия на эту среду и даже возможность ее преобразования]
- := [интеллектуальный потенциал кибернетической системы]
- := [спектр знаний, навыков и способностей к обучению кибернетической системы]
- := [интеллектуальность кибернетической системы]
- ⇒ *примечание**:
 [Процесс эволюции *кибернетических систем* следует рассматривать как процесс повышения уровня их качества по целому ряду свойств (характеристик) и, в первую очередь, как процесс повышения уровня их *интеллекта*. При этом можно говорить об эволюции каждой конкретной *кибернетической системы* в процессе своей "жизнедеятельности", а также об эволюции целого класса *кибернетических систем*, когда новые экземпляры этого класса являются более интеллектуальными, чем их предшественники. В таком аспекте, в частности, можно рассматривать эволюцию *компьютерных систем* (искусственных кибернетических систем).]
- ⇒ *примечание**:
 [Очень важно уточнить, какими иными свойствами *кибернетических систем* определяется уровень и характер их интеллектуальности. Подчеркнем, что любая кибернетическая система обладает соответствующим уровнем интеллектуальности. Пусть даже и достаточно низким. Существенным является уточнение того, за счет чего уровень интеллектуальности *кибернетической системы* может быть повышен. Нет смысла проводить четкую границу между *интеллектуальными кибернетическими системами* и неинтеллектуальными. Но есть смысл уточнять направления повышения уровня интеллектуальности *кибернетических систем*.]
- ⇒ *эпиграф**:
 [Никто не может провести линию, отделяющую атмосферу от космоса, или черту, за которой начинается жизнь, или границу электронного облака. Все дело в степени проявления свойства.]
- ⇒ *автор**:
 Барт Коско
- ⇒ *примечание**:
 [Прежде, чем говорить о требованиях, предъявляемых к *технологии проектирования и производства интеллектуальных компьютерных систем (искусственных кибернетических систем)*, обладающих высоким уровнем *интеллекта*, необходимо уточнить (детализировать) *свойства*, присущие указанным системам и являющиеся предпосылками, обеспечивающими высокий уровень *интеллекта*. Подчеркнем, что указанные *свойства*, уточняющие (детализирующие, обеспечивающие, определяющие) *свойства интеллектуальных систем (свойства, определяющие уровень интеллекта* этих систем) должны быть общими как для искусственных кибернетических систем (*компьютерных систем*), так и для *естественных кибернетических систем*.]
- := [интегральное качество информационного обеспечения и информационных процессов в кибернетической системе]
- := [интегральное качество кибернетической системы, определяемое:
 □ уровнем ее образованности – качеством накопленных к заданному моменту знаний и умений (навыков);
 □ уровнем ее обучаемости – способностью самостоятельно повышать уровень своей образованности.
]
- ⇒ *свойство-предпосылка**:
 • образованность кибернетической системы
 • обучаемость кибернетической системы
 • социализация кибернетической системы
- ⇒ *примечание**:
 [Интеллект *кибернетической системы*, как и лежащий в его основе познавательный процесс, выполняемый кибернетической системой, имеет социальный характер, поскольку наиболее

эффективно формируется и развивается в форме взаимодействия *кибернетической* системы с другими *кибернетическими системами*.]

образованность кибернетической системы

:= [уровень навыков (умений), а также иных знаний, приобретенных *кибернетической системой* к заданному моменту]

⇒ *свойство-предпосылка**:

- **качество навыков, приобретенных *кибернетической системой***

:= [качество умений, которыми владеет кибернетическая система в текущий момент]

⇒ *свойство-предпосылка**:

качество информации, хранимой в памяти кибернетической системы

:= [качество знаний, приобретенных кибернетической системой к заданному моменту]

- **качество информации, хранимой в памяти кибернетической системы**

⇒ *примечание**:

[Следует обратить внимание на то, что *качество информации, хранимой в памяти кибернетической системы*, является фактором, обеспечивающим не только *качество навыков, приобретенных кибернетической системой*, но и общий *уровень качества кибернетической системы*.]

кибернетическая система

⇐ *объединение**:

Признак интеллектуальности кибернетических систем

- =** {
- *неинтеллектуальная кибернетическая система*
 - *интеллектуальная система*

:= [интеллектуальная кибернетическая система]

⇐ *объединение**:

- {
- *слабоинтеллектуальная система*

:= [кибернетическая система со слабым интеллектом]

:= [кибернетическая система с низким уровнем интеллекта]

:= [кибернетическая система с элементами интеллекта]

- *высокоинтеллектуальная система*

:= [идеальная интеллектуальная система]

:= [кибернетическая система с сильным интеллектом]

:= [кибернетическая система с высоким уровнем интеллекта]

:= [действительно интеллектуальная система]

}

Признак интеллектуальности кибернетических систем

⇒ *примечание**:

[Данный признак классификации кибернетических систем формально является не разбиением, а покрытием множества *кибернетических систем*, так как отсутствует четкая грань между неинтеллектуальными и интеллектуальными кибернетическими системами, а также между слабоинтеллектуальными и высокоинтеллектуальными кибернетическими системами.]

интеллектуальная система

:= [интеллектуальная кибернетическая система]

⇒ *примечание**:

[В этом термине слово "кибернетическая" можно опустить, так как интеллектуальными могут быть только *кибернетические системы*]

⇒ *примечание**:

[Интеллектуальные кибернетические системы могут быть *естественными интеллектуальными системами*, искусственными интеллектуальными системами (которые будем называть *интеллектуальными компьютерными системами*), а также естественно-искусственными интеллектуальными системами, состоящими из компонентов как естественного, так и искусственного происхождения. Важнейшим примером естественно-искусственных интеллектуальных систем являются человеко-машинные системы, представляющие собой коллективы (многоагентные системы), состоящие из *интеллектуальных компьютерных систем* и людей (конечных пользователей и разработчиков этих компьютерных систем).]

⇒ *примечание**:
 [Вводя понятие *интеллектуальной системы*, важно, во-первых, уточнить понятие *кибернетической системы* и определить те свойства, которые присущи всем кибернетическим системам, и, во-вторых, локализовать ту условную грань перехода от неинтеллектуальных *кибернетических систем* к интеллектуальным, а также грань перехода от слабоинтеллектуальных к высокоинтеллектуальным кибернетическим системам. В этом и заключается уточнение феномена *интеллекта* (интеллектуальности) кибернетических систем.]

⇒ *примечание**:
 [Все *свойства* (в том числе способности и активности), присущие *кибернетическим системам*, в различных *кибернетических системах* могут иметь самый различный уровень (уровень развития). Более того, в некоторых *кибернетических системах* некоторые из этих свойств могут вообще отсутствовать. При этом в кибернетических системах, которые условно будем называть ***интеллектуальными системами***, все указанные выше свойства должны быть представлены в достаточно развитом виде. Заметим также, что мы называем *интеллектуальными системами*, иногда называют кибернетическими системами с сильным интеллектом (с высоким уровнем интеллекта), противопоставляя их кибернетическим системам со слабым интеллектом (с низким уровнем интеллекта).]

С *образованная кибернетическая система*

:= [кибернетическая система, имеющая высокий уровень образованности]

:= [кибернетическая система, обладающая высоким уровнем знаний и навыков]

С *кибернетическая система, основанная на знаниях*

С *кибернетическая система, управляемая знаниями*

С *целенаправленная кибернетическая система*

С *гибридная кибернетическая система*

С *потенциально универсальная кибернетическая система*

С *обучаемая кибернетическая система*

:= [когнитивная кибернетическая система]

:= [кибернетическая система, имеющая высокий уровень обучаемости]

С *кибернетическая система с высоким уровнем стратифицированности своих знаний и навыков*

С *рефлексивная кибернетическая система*

С *самообучаемая кибернетическая система*

С *кибернетическая система с высоким уровнем познавательной активности*

С *социально ориентированная кибернетическая система*

:= [кибернетическая система, имеющая высокий уровень социализации]

С *кибернетическая система, способная устанавливать и поддерживать высокий уровень семантической совместимости и взаимопонимания с другими системами*

С *договороспособная кибернетическая система*

:= [кибернетическая система, способная координировать (согласовывать) свою деятельность с другими системами]

кибернетическая система, основанная на знаниях

:= [кибернетическая система, в основе которой лежит формируемая в ее памяти, постоянно совершенствуемая и структурированная информационная модель той среды, в рамках которой она существует и решает соответствующие задачи]

:= [кибернетическая система, в основе которой лежит ее база знаний – систематизированная совокупность всех используемых ею знаний]

:= [кибернетическая система, формирующая в своей памяти систематизированную информационную модель среды своего обитания и использующая эту модель для организации своего целенаправленного поведения]

кибернетическая система, управляемая знаниями

:= [кибернетическая система, в которой выполняемые ею действия инициируются соответствующими ситуациями и/или событиями, возникающими в ее базе знаний]

целенаправленная кибернетическая система

:= [субъект, "осознанно" и целенаправленно осуществляющий свою деятельность, "ведущий" то, что он творит]

обучаемая кибернетическая система

:= [когнитивная система]

:= [кибернетическая система, способная познавать (изучать) среду своего обитания, то есть строить и постоянно уточнять в своей памяти информационную модель (описание) этой среды, а также использовать эту модель для решения различных задач (для организации своей деятельности (поведения)) в указанной среде]

:= [кибернетическая система, способная к самосовершенствованию]

социально ориентированная кибернетическая система

:= [кибернетическая система, имеющая достаточно высокий уровень интеллекта, чтобы быть полезным членом различных, в том числе, и человеко-машинных сообществ]

⇒ *примечание**:

[Определенный уровень социально значимых качеств является необходимым условием интеллектуальности кибернетической системы. Это, своего рода, модификация теста Тьюринга – важна не имитация, не иллюзия "человекоподобия", а реальная польза в процессе коллективного решения сложных задач.]

интеллектуальная компьютерная система

:= [искусственная интеллектуальная система]

:= [искусственная кибернетическая система, обладающая высоким уровнем интеллекта (высоким уровнем знаний и умений), а также высоким уровнем обучаемости]

С *компьютерная система*

С *кибернетическая система*

:= *основной sc-идентификатор**:

[интеллектуальная компьютерная система]

⇒ *сокращение**:

[и.к.с.]

:= [система искусственного интеллекта]

:= [искусственная интеллектуальная система]

С *интеллектуальная система*

С *кибернетическая система*

} /* Завершили Сегмент “Комплекс свойств, определяющих уровень интеллекта кибернетической системы” */

/* Сегмент *****/

Комплекс свойств, определяющих качество информации, хранимой в памяти кибернетической системы

⊃=

{

информация

:= [информационная конструкция]

:= [информационная модель, состоящая из некоторого множества различных *знаков*, обозначающих моделируемые (описываемые) *сущности* любого вида и, в частности, *знаков*, обозначающих различного вида *связи* между *знаками* описываемых *сущностей* (такие *связи* чаще всего являются отражениями (моделями) *связей* между *сущностями*, которые обозначаются связываемыми *знаками*)]

⇒ *примечание**:

[Подчеркнем, что *связи* между *знаками* описываемых *сущностей* сами также могут быть описываемыми *сущностями*, но для этого указанные *связи* в рамках информационной модели должны быть представлены своими *знаками*. Не все *связи* между *знаками* являются описываемыми *сущностями*. Такими неопределяемыми связями являются связи инцидентности знаков.]

:= [конфигурация знаков]

:= [знаковая конструкция]

:= [текст]

:= [описание (отражение) некоторого множества (1) первичных сущностей, (2) понятий, (3) связей между ними, (4) связей между связями, (5) фрагментов данного описания, (6) связей между этими фрагментами]

⊃ *дискретная информационная конструкция*

:= [информационная конструкция, у которой все входящие в неё знаки имеют чёткие границы]

⊃ *дискретная информационная конструкция, у которой входящие в неё знаки имеют условную структуру*

⇒ *разбиение**:

{• *внутренняя информационная конструкция*

- := [информационная конструкция, хранимая в памяти некоторой кибернетической *системы*, и непосредственно интерпретируемая (понимаемая) решателем задач этой системы]
- *внешняя информационная конструкция*
- := [информационная конструкция, представленная на каком-либо внешнем носителе или в памяти другой кибернетической системы]
- *файл*
- := [первичный электронный образ некоторой внешней информационной конструкции]

}
:= [информационная модель]

:= [информационная модель (отражение, описание) некоторого множества связей между некоторым описываемыми (рассматриваемыми, исследуемыми, изучаемыми) сущностями]

⇒ *определение**:

[Множество всевозможных информационных конструкций (понятие информационной конструкции) представляет собой множество, на котором задано

- Отношение синтаксической эквивалентности и, соответственно, семейство классов синтаксической эквивалентности информационных конструкций
- Отношение семантической эквивалентности и, соответственно, семейство классов семантической эквивалентности информационных конструкций
- Отношение логической эквивалентности и, соответственно, семейство классов логической эквивалентности информационных конструкций.

При этом можно говорить об инварианте каждого класса синтаксически эквивалентных информационных конструкций, об инварианте каждого класса семантически эквивалентных информационных конструкций и об инварианте каждого класса логически эквивалентных информационных конструкций синтаксически эквивалентные информационные конструкции могут отличаться вариантами изображения букв (различным почерком, разными шрифтами), вариантами "разрезания" текста на страницы и на строчки. Семантически эквивалентные информационные конструкции могут отличаться разными именами, обозначающими одни и те же сущности, разным порядком размещения этих имён.]

денотационная семантика информационной конструкции

⇒ *пояснение**:

[Каждая информационная конструкция имеет денотационную семантику, описывающую то, как связаны входящие в информационную конструкцию знаки с соответствующими им денотатами (т.е. сущностями, обозначаемыми этими знаками.)]

сенсорная информация

С *информация*

:= [первичная информация, приобретаемая кибернетической системы с помощью её сенсоров (рецепторов)]

:= [первичная информация]

⇒ *примечание**:

[Подчеркнем, что *сенсорная информация кибернетической системы* с точки зрения её *денотационной семантики* является простейшим видом *знаковой конструкции*, в которой *внешняя среда кибернетической системы* описывается

- путём задания параметрического пространства (множество параметров, признаков, *свойства*, характеристик), с помощью которого описываются состояние элементарных (атомарных) фрагментов *внешней среды*, которые непосредственно являются смежными (соприкасаются с) чувствительными поверхностями *сенсоров кибернетической системы*;
- путём пространственной декомпозиции наблюдаемой *внешней среды* с выделением указанных выше элементарных фрагментов этой среды (элементарных с "точки зрения" *сенсоров кибернетической системы*) и с явным описанием пространственных связей между указанными элементарными фрагментами (эти связи соответствует пространственным связям между сенсорами);
- путём темпоральной декомпозиции наблюдаемой *внешней среды*, которая предполагает фиксацию моментов времени для каждого события по изменению состояния измеряемого параметра каждого элементарного фрагмента наблюдаемой *внешней среды*

]

⇒ *примечание**:

[Качество (в частности, информативность) *сенсорной информации* обеспечивается:

- качеством используемого параметрического пространства

- многообразием видов *сенсоров*, т.е. многообразием параметров (свойств), с помощью которых описывается внешняя среда
- информативностью каждого из указанных параметров
- целостностью (полнотой, достаточностью) всего набора рассматриваемых параметров
- отсутствием избыточности в наборе этих параметров
- общим количеством сенсоров и количеством сенсоров, соответствующих каждому параметру
- способностью кибернетической системы перемещать сенсоры в пространстве

]

⇒ *примечание**:

[*сенсорная информация* обеспечивает формирование первичного описания состояния и динамики изменения не только *внешней среды кибернетической системы*, но также и её физической оболочки, которую можно рассматривать как часть всей **физической среды кибернетической системы**, противопоставляя такую *физическую среду кибернетической системы* её внутренней (информационной, абстрактной) среде, в которой хранится и обрабатывается *информация*, используемая *кибернетической системой*. Указанную абстрактную внутреннюю среду кибернетической системы будем называть **абстрактной памятью кибернетической системы**.]

язык

:= [множество информационных конструкции, построенных по общим синтаксическим и семантическим правилам]

⊃ *внутренний язык кибернетической системы*

:= [язык, используемый кибернетической системой для представления информации, хранимой в её памяти]

информация, хранимая в памяти кибернетической системы

:= [совокупность всей информации, хранимой в памяти кибернетической системы]

⊂ *информация*

качество информации, хранимой в памяти кибернетической системы

:= [качество знаний, приобретенных кибернетической системой к текущему моменту]

:= [уровень качества хранимой информации]

:= [качество информационной модели среды кибернетической системы, хранимой в её памяти]

:= [уровень качества хранимых в памяти кибернетической системы внутренней информационной модели среды существования (жизнедеятельности) этой кибернетической системы]

:= [интегральное качество знаний, накопленных кибернетической системой к текущему моменту]

:= [степень приближения информации, хранимой в памяти кибернетической системы к качественной информационной модели той среды, в которой существует кибернетическая система, к систематизированной базе знаний, описывающей все свойства этой среды, необходимые для функционирования этой кибернетической системы]

:= [качество хранимой в памяти кибернетической системы информационной модели среды жизнедеятельности этой системы]

⇒ *примечание**:

[Качество информационной модели среды "обитания" кибернетической системы, в частности, определяется

- корректностью этой модели (отсутствием в ней ошибок);
- адекватностью этой модели;
- полнотой – достаточностью находящейся в ней информации для эффективного функционирования кибернетической системы;
- структурированностью, систематизированностью.

Важнейшим этапом эволюции информационной модели среды кибернетической системы является переход от недостаточно полной и несистематизированной информационной модели среды к *базе знаний*. Именно поэтому важнейшим этапом повышения уровня интеллектуальности компьютерных систем является переход от традиционных компьютерных систем к компьютерным системам, основанным на знаниях.]

⇒ *комплекс свойств-предпосылок**:

не-фактор

⇒ *свойство-предпосылка**:

- *семантическая мощьность языка представления информации в памяти кибернетической системы*
- *объём информации, в память кибернетической системы*

- *степень конвергенции и интеграции различного вида знаний, хранимых в памяти кибернетической системы*
- *стратифицированность информации, хранимой в памяти кибернетической системы*
- *простота и локальность выполнения семантически целостных операций над информацией, хранимой в памяти кибернетической системы*

не-фактор

:= [группа семантических свойств, определяющих качество информации, хранимой в памяти кибернетической системы]

= {

- *корректность/некорректность информации, хранимой в памяти кибернетической системы*
- *однозначность/неоднозначность информации, хранимой в памяти кибернетической системы*
- *целостность/нецелостность информации, хранимой в памяти кибернетической системы*
- *чистота/загрязненность информации, хранимой в памяти кибернетической системы*
- *достоверность/недостоверность информации, хранимой в памяти кибернетической системы*
- *точность/неточность информации, хранимой в памяти кибернетической системы*
- *четкость/нечеткость информации, хранимой в памяти кибернетической системы*
- *определенность/неопределенность информации, хранимой в памяти кибернетической системы*

}

⇒ *пояснение**:

[Ярушкина.Н.Г.НечетГС-2007кн.-стр.10-28]

← *цитата**:

Ярушкина.Н.Г.НечетГС-2007кн

корректность/некорректность информации, хранимой в памяти кибернетической системы

:= [уровень адекватности хранимой информации той среде, в которой существует кибернетическая система и информационной моделью которой эта хранимая информация является]

непротиворечивость/противоречивость информации, хранимой в памяти кибернетической системы

:= [уровень присутствия в хранимой информации различного вида противоречий и, в частности, ошибок]

противоречие*

:= [пара противоречащих друг другу фрагментов информации, хранимой в памяти кибернетической системы*]

⇒ *примечание**:

[Чаще всего противоречащими друг другу информационными фрагментами являются:

- явно представленная в памяти некоторая закономерность (некоторое правило)
- информационный фрагмент, не соответствующий (противоречащий) указанной закономерности

В этом случае некорректность может присутствовать:

- либо в информационном фрагменте, который противоречит указанной закономерности;
- либо в самой этой закономерности;
- либо и там и там.

]

информационная ошибка

:= *определение**:

[противоречие, заключающееся в нарушении некоторой закономерности (некоторого правила), которая не подвергается сомнению]

информационная ошибка

⇒ *примечание**:

[Ошибки (ошибочные фрагменты) в хранимой информации могут быть синтаксическими и семантическими, противоречащими некоторым правилам (закономерностям), которые явно в памяти могут быть не представлены и считаются априори истинными.]

полнота/неполнота информации, хранимой в памяти кибернетической системы

:= [уровень того, насколько информация, хранимая в памяти кибернетической системы, описывает среду существования этой системы и используемые ею методы решения задач достаточно полно (достаточно детально) для того, чтобы кибернетическая система могла действительно решать все множество соответствующих ей задач]

- := [уровень соответствия хранимой информации объёму задач (действий), которые соответствующая кибернетическая система желает уметь решать (выполнять)]
- := [степень достаточности информации, хранимой в памяти кибернетической системы, для достижения целей этой системы, для выполнения своих "обязанностей"]
- ⇒ *примечание**:
[Чем полнее информация, хранимая в памяти кибернетической системы, чем полнее информационное обеспечение деятельности этой системы, тем эффективнее (качественнее) сама эта деятельность.]
- ⇒ *свойство-предпосылка**:
 - *многообразие видов знаний, хранимых в памяти кибернетической системы*
 - *структурированность информации, хранимой в памяти кибернетической системы*
- однозначность/неоднозначность информации, хранимой в кибернетической системе***
- ⇒ *свойство-предпосылка**:
 - *многообразие форм дублирования информации, хранимой в памяти кибернетической системы*
 - *частота дублирования информации, хранимой в памяти кибернетической системы*
- целостность/нецелостность информации, хранимой в памяти кибернетической системы***
- := [уровень содержательной информативности информации, хранимой в памяти кибернетической системы]
- := [уровень того, насколько содержательно (семантически) связной является информация, хранимая в памяти кибернетической системы, насколько полно специфицированы все описываемые в памяти сущности (путём описания необходимого набора связей этих сущностей с другими описываемыми сущностями), насколько редко или часто в рамках хранимой информации встречаются *информационные дыры*, соответствующие явной недостаточности некоторых спецификаций]
- := [известность/неизвестность информации, хранимой в памяти кибернетической системы]
- := [многообразие форм и частота присутствия *информационных дыр* в информации, хранимой в памяти кибернетической системы]
- информационная дыра в информации, хранимой в памяти кибернетической системы***
- := [информация, отсутствие которой в памяти кибернетической системы существенно усложняет деятельность этой системы]
- ⇒ *примечание**:
[Примерами информационных дыр являются:
- отсутствующий метод решения часто встречающихся задач;
 - отсутствующее определение используемого определяемого понятия;
 - недостаточно подробная спецификация часто рассматриваемых сущностей
-]
- чистота/загрязненность информации, хранимой в памяти кибернетической системы***
- := [многообразие форм и общее количество информационного мусора, входящего в состав информации, хранимой в памяти кибернетической системы]
- информационный мусор, входящий в состав информации, хранимой в памяти кибернетической системы***
- := [информационный фрагмент, входящий в состав информации, хранимой в памяти кибернетической системы, удаление которого существенно не усложнит деятельность кибернетической системы]
- ⇒ *примечание**:
[Примерами информационного мусора являются:
- информация, которая нечасто востребована, но при необходимости может быть легко логически выведена
 - информация, актуальность которой истекла
-]
- семантическая мощьность языка представления информации в памяти кибернетической системы***
- := [семантическая мощьность внутреннего языка кибернетической системы]
- ⇒ *свойство-предпосылка**:
гибридность информации, хранимой в памяти кибернетической системы

⇒ *примечание**:
 [Универсальность внутреннего языка кибернетической системы является важнейшим фактором её интеллектуальности]

универсальный язык

:= [язык, информационные конструкции которого могут представить (описать) любую конфигурацию любых связей между любыми сущностями]

гибридность информации, хранимой в памяти кибернетической системы

⇒ *свойство-предпосылка**:

- *многообразие видов знаний, хранимых в памяти кибернетической системы*
- *степень конвергенции и интеграции различного вида знаний, хранимых в памяти кибернетической системы*

многообразие видов знаний, хранимых в памяти кибернетической системы

⇒ *частное свойство**:

- *рефлексивность информации, хранимой в памяти кибернетической системы*
 := [многообразие видов метаинформации (метазнаний), хранимых в памяти кибернетической системы]
- *многообразие моделей решения задач, используемых кибернетической системой*
- *многообразие видов целей, анализируемых или синтезируемых кибернетической системой*
- *многообразие планов решения задач, решаемых кибернетической системой*
- *многообразие протоколов решения задач, решаемых кибернетической системой*

объем информации, хранимой в памяти кибернетической системы

:= [объем знаний, приобретенных кибернетической системой к текущему моменту]

:= [содержательная совокупность всех знаний, хранимых в текущий момент в памяти кибернетической системы]

⇒ *примечание**:

[Чем больше кибернетическая система знает, тем при прочих равных условиях выше уровень её качества]

/******

степень конвергенции и интеграции различного вида знаний, хранимых в памяти кибернетической системы

:= [уровень "бесшовной" интеграции различного вида знаний кибернетической системы]

⇒ *примечание**:

[Максимальный уровень конвергенции и интеграции знаний (в том числе, и знаний различного вида) предполагает:

- использование универсального базового языка, по отношению к которому всем используемым видам знаний соответствуют специализированные языки, являющиеся подъязыками указанного базового языка
- построение четкой иерархии указанных специализированных языков по принципу "язык-подъязык"
- явное введение семейства отношений, заданных на множестве различных знаний и, в том числе, связывающих знания различного вида

]

⇒ *свойство-предпосылка**:

уровень формализованности информации, хранимой в памяти кибернетической системы

уровень формализованности информации, хранимой в памяти кибернетической системы

:= [степень приближения информации, хранимой в памяти кибернетической системы, к максимально простой и компактной форме представления информационной модели некоторого множества описываемых сущностей, которая является отражением определенной конфигурации связей между указанными сущностями]

⇒ *примечание**:

[Высшим уровнем формализации информации, хранимой в памяти кибернетической системы, является смысловое представление информации в форме семантических сетей. Смотрите Раздел "Предметная область и онтология семантических сетей, семантических языков и семантических моделей баз знаний".]

⇔ *следует отличать**:

*формализация**

:= [Бинарное ориентированное отношение, каждая пара которого связывает некоторую информационную конструкцию с другой информационной конструкцией, которая семантически эквивалентна первой, но имеет более высокий уровень формализованности]

⇒ *примечание**:

[Приобретение навыков формального представления информации не является простой проблемой даже для человека. По сути совокупность таких навыков – это основа математической культуры, культуры точного изложения своих соображений. Некоторые примеры, иллюстрирующие нетривиальность проблемы смотрите в *Арнольд.В.И. ЧтоТМ-2012кн*]

⇔ *следует отличать**:
формализация

:= [деятельность, направленная на повышение уровня формализованности представление информации]

⇒ *метафора**:

[сближение синтаксиса с семантикой – сближение синтаксической структуры информационной конструкции с её смысловой структурой]

:= [уровень способности кибернетической системы к формальному представлению знаний и используемых понятий, к рационализации идей]

:= [степень близости языка внутреннего представления (способа внутреннего кодирования) информации в памяти кибернетической системы к смысловому представлению информации]

:= [степень близости к изоморфизму соответствия между: (1) синтаксической структурой внутреннего представления информации в памяти кибернетической системы и (2) конфигурацией связей описываемых сущностей]

⇒ *свойства-предпосылка**:

- *многообразие форм дублирования информации, хранимой в памяти кибернетической системы*
- *относительный объём дублирования информации, хранимой в памяти кибернетической системы*
- *многообразие фрагментов хранимой информации, не являющихся ни знаками, ни конфигурациями знаков*
- *компактность представления представление информации, хранимой в памяти кибернетической системы*

смысловое представление информации

:= *пояснение**:

[способ представления информации, в котором минимизируются "чисто синтаксические" аспекты представления информационных конструкций, не имеющие непосредственной семантической интерпретации]

⇒ *примечание**:

[Примерами "чисто синтаксических" аспектов представления информационных конструкций являются:

- буквы, которые входят в состав слов и которые, следовательно, не являются знаками описываемых сущностей;
- алфавиты букв различных языков;
- знаки препинания (разделители и ограничители);
- инцидентность (порядок, последовательность) букв и других символов, входящих в состав информационной конструкции.

]

⇒ *следовательно**:

[Информационная конструкция, представленная на каком-либо привычном для нас языке, является достаточно громоздкой информационной конструкцией, смысл которой (т.е. знаки описываемых сущностей и семантически интерпретируемые связи между знаками, отражающие соответствующие связи между обозначаемыми сущностями) сильно закамуфлирован. Это существенно усложняет обработку информации. если пытаться реализовывать "осмысленные" модели решения задач, для которых "смысловые" аспекты обрабатываемой информации являются ключевыми.]

⇒ *примечание**:

[Существенно подчеркнуть, что приближение внутреннего представления информации в памяти кибернетической системы к смысловому представлению информации является важнейшим фактором упрощения решателя задач кибернетической системы при реализации сложных моделей решения задач, требующих глубокого анализа смысла обрабатываемой информации. А это, в свою очередь, является важнейшим фактором качества решателя задач кибернетической системы.]

многообразие форм дублирования информации, хранимой в памяти кибернетической системы

:= [многообразие видов семантической эквивалентности фрагментов информации, хранимой в памяти кибернетической системы]

⇒ *примечание**:

[Простейшим видом семантической эквивалентности является синонимия знаков, когда два разных фрагмента хранимой информации являются знаками, имеющими один и тот же денотат (т. е. обозначающими одну и ту же сущность).]

относительный объем дублирования информации, хранимой в памяти кибернетической системы

:= [частота присутствия в хранимой информации семантически эквивалентных информационных фрагментов и, в частности, синонимичных знаков]

многообразие фрагментов хранимой информации, не являющихся ни знаками, ни конфигурациями знаков

⇒ *примечание**:

[Примерами фрагментов хранимой информации, не являющихся знаками или конфигурациями знаков, являются:

- буквы, входящие в состав слов
- слова, входящие в состав словосочетаний
- различного вида разделители, знаки препинания
- различного вида ограничители.

]

компактность представления информации, хранимой в памяти кибернетической системы

⇒ *примечание**:

[Должно уменьшаться число элементов памяти, используемых для представления информации, т.е. необходим переход к более компактным, но семантически эквивалентным информационным конструкциям.]

стратифицированность информации, хранимой в памяти кибернетической системы

⇒ *свойство-предпосылка**:

структурированность информации, хранимой в памяти кибернетической системы

:= [способность кибернетической системы выделять такие разделы информации, хранимой в памяти этой системы, которые бы ограничивали области действия агентов решателя задач кибернетической системы, являющиеся достаточными для решения заданных задач]

⇒ *примечание**:

[Существует правило, позволяющее каждой заданной задаче поставить в соответствие априори известный (выделенный) раздел хранимой информации, являющийся областью действия агентов решателя, осуществляющих решение заданной задачи. Основными видами такого рода разделов хранимой информации являются *предметные области* и *онтологии*.]

⇒ *свойство-предпосылка**:

рефлексивность информации, хранимой в памяти кибернетической системы

:= [уровень систематизации знаний, хранимых в памяти кибернетической системы]

:= [уровень перехода от неструктурированных или слабоструктурированных данных к хорошо структурированным базам знаний]

:= [уровень перехода от первичной информации к метаинформации, метаметаинформации и т.д.]

рефлексивность информации, хранимой в памяти кибернетической системы

:= [уровень применения средств самоописания (метаязыковых средств) в информации, хранимой в памяти кибернетической системы]

:= [относительный, объем и многообразие метаинформации, хранимой в памяти кибернетической системы]

⇒ *примечание**:

[рефлексивность информации, хранимой в памяти кибернетической системы, т.е. наличие метаязыковых средств, является фактором, обеспечивающим не только структуризацию хранимой информации, но возможность описания синтаксиса и семантики самых различных языков, используемых кибернетической системой.]

простота и локальность выполнения семантически целостных операций над информацией, хранимой в памяти кибернетической системы

⇒ *примечание**:

[Данное свойство касается не самой информации, хранимой в памяти, а язык кодирования (представления) информации в памяти кибернетической системы]

:= [гибкость выполнения семантически целостных операций над информацией, хранимой в памяти кибернетической системы]

база знаний

:= [база знаний кибернетической системы]

C информация, хранимая в памяти кибернетической системы

:= *пояснение**:

[информация, хранимая в памяти кибернетической системы и имеющая высокий уровень качества по всем показателям и, в частности, высокий уровень:

- семантической мощности языка представления информации хранимой в памяти кибернетической системы (в базе знаний указанный язык должен быть универсальным);
- гибридности информации, хранимой в памяти кибернетической системы;
- многообразия видов знаний, хранимых в памяти кибернетической системы;
- формализованности информации, хранимой в памяти кибернетической системы;
- структурированности информации, хранимой в памяти кибернетической системы

]

⇒ *примечание**:

[Переход информации, хранимой в памяти кибернетической системы на уровень качества, соответствующий базам знаний, является важнейшим этапом эволюции кибернетических систем. Подчеркнем при этом, что базы знаний по уровню своего качества могут сильно отличаться друг от друга.]

} /* Завершили Сегмент “Комплекс свойств, определяющих качество информации, хранимой в памяти кибернетической системы” */

/* Сегмент *****/

Комплекс свойств, определяющих качество решателя задач кибернетической системы

⊃=

{

качество решателя задач кибернетической системы

:= [интегральная качественная оценка множества задач (действий), которые кибернетическая система способна выполнять в заданный момент]

:= [качество навыков, приобретенных кибернетической системой]

⇒ *примечание**:

[Основным свойством и назначением *решателя задач кибернетической системы* является способность решать задачи на основе накапливаемых (приобретаемых) *кибернетической системой* различного вида навыков с использованием *процессора кибернетической системы*, являющегося универсальным интерпретатором всевозможных накопленных *навыков*. При этом качество (уровень развития, уровень совершенства) указанной способности определяется целым рядом дополнительных факторов (свойств).]

:= [интеллектуальный уровень качества решателя задач кибернетической системы]

:= [интегральное качество умений (навыков), приобретенных *кибернетической системой* к текущему моменту]

⇒ *свойство-предпосылка**:

- *общая характеристика решателя задач кибернетической системы*
- *качество логико-семантической организации памяти кибернетической системы*
- *качество решения интерфейсных задач в кибернетической системе*

/*-*/

общая характеристика решателя задач кибернетической системы

⇒ *свойство-предпосылка**:

- *общий объем задач, решаемых кибернетической системой*
- *многообразие видов задач, решаемых кибернетической системой*
- *способность кибернетической системы к анализу решаемых задач*
- *способность кибернетической системы к решению задач, методы решения которых в текущий момент известны*
- *способность кибернетической системы к решению задач, методы решения которых ей в текущий момент не известны*
- *множество навыков, используемых кибернетической системой*
- *степень конвергенции и интеграции различного вида моделей решения задач, используемых кибернетической системой*
- *качество организации взаимодействия процессов решения задач в кибернетической системе*
- *быстродействие решателя задач кибернетической системы*
- *способность кибернетической системы решать задачи, предполагающие использование информации, обладающей различного рода не-факторами*
- *многообразие и качество решения задач информационного поиска*
- *способность кибернетической системы генерировать ответы на вопросы различного вида в случае, если они целиком или частично отсутствуют в текущем состоянии информации, хранимой в памяти*
- *способность кибернетической системы к рассуждениям различного вида*
- *качество целеполагания*
- *качество реализации планов собственных действий*
- *способность кибернетической системы к локализации такой области информации, хранимой в ее памяти, которой достаточно для обеспечения решения заданной задачи*
- *способность кибернетической системы к выявлению существенного в информации, хранимой в ее памяти*
- *активность кибернетической системы*

общий объем задач, решаемых кибернетической системой

:= [общий объем задач, которые кибернетическая система способна решать]

:= [общий объем (множество), задач (действий), которые кибернетическая система способна (может, умеет) решать (выполнять) в заданный (в том числе, в текущий) момент]

⇒ *свойство-предпосылка**:

мощность языка представления задач, решаемых кибернетической системой

мощность языка представления задач, решаемых кибернетической системой

:= [мощность языка спецификации (описания) различного вида действий, выполняемых кибернетической системой]

⇒ *примечание**:

[*мощность языка представления задач прежде всего определяется многообразием видов представляемых задач (многообразием видов описываемых действий).*]

⇐ *свойство-предпосылка**:

многообразие видов задач, решаемых кибернетической системой

⇒ *частное свойство**:

- *мощность языка представления задач, решаемых в памяти кибернетической системы*
⇐ *свойство-предпосылка**:
многообразие видов задач, решаемых в памяти кибернетической системы
- *мощность языка представления задач, решаемых во внешней среде кибернетической системы*
⇐ *свойство-предпосылка**:
многообразие видов задач, решаемых во внешней среде кибернетической системы
- *мощность языка представления задач, решаемых в рамках физической оболочки кибернетической системы*
⇐ *свойство-предпосылка**:
многообразие видов задач, решаемых в рамках физической оболочки кибернетической системы

многообразие видов задач, решаемых кибернетической системой

:= [многообразие видов действий, которые кибернетическая система способна выполнять]

⇒ *примечание**:
 [Подчеркнем, что каждая задача есть спецификация соответствующего (описываемого) действия. Поэтому рассмотрение многообразия видов задач, решаемых кибернетической системой, полностью соответствует многообразию видов деятельности, осуществляемой этой системой. Важно заметить, что есть виды деятельности кибернетической системы, которые определяют качество и, в частности, уровень интеллекта кибернетической системы.]

⇒ *свойство-предпосылка**:
мощность языка представления задач в памяти кибернетической системы

⇒ *комплекс частных свойств**:

- *многообразие видов задач, решаемых в памяти кибернетической системы*
- *многообразие видов задач, решаемых во внешней среде кибернетической системы*
- *многообразие видов задач, решаемых в рамках физической оболочки кибернетической системы*

способность кибернетической системы к анализу решаемых задач

:= [способность кибернетической системы осмысливать (ведать) то, что она творит]

:= [способность анализировать свои цели и, соответственно, решаемые задачи на предмет:

- сложности достижения;
- целесообразности достижения (нужности, важности, приоритетности);
- соответствия цели существующим нормам (правилам) соответствующей деятельности

]

способность кибернетической системы к решению задач, методы решения которых ей в текущий момент известны

⇒ *примечание**:

[Указанными методами могут быть не только алгоритмы, но также и функциональные программы, продукционные системы, логические исчисления, генетические алгоритмы, искусственные нейронные сети различного вида.]

⇒ *свойство-предпосылка**:

- *способность кибернетической системы к поиску хранимых в своей памяти методов решения иницированных задач*
- *способность кибернетической системы к интерпретации хранимых в своей памяти методов решения задач*

способность кибернетической системы к решению задач, методы решения которых ей в текущий момент неизвестны

:= [способность кибернетической системы к решению задач, для которых не найдены соответствующие (релевантные) им методы их решения]

:= [способность кибернетической системы строить цепочку "цель-план достижения цели-система действий"]

⇒ *примечание**:

[Задачи, для которых не находятся соответствующие им методы, решаются с помощью метаметодов (стратегий) решения задач, направленных:

- на генерацию нужных исходных данных (нужного контекста), необходимых для решения каждой задачи;
- на генерацию плана решения задачи, описывающего сведение исходной задачи к подзадачам (до тех подзадач, методы решения которых системы известны);
- на сужение области решения задачи (на сужения контекста задачи, достаточного для ее решения).

]

множество навыков, используемых кибернетической системой

:= [объем и многообразие навыков, приобретенных кибернетической системой к текущему моменту (с помощью учителей-разработчиков или полностью самостоятельно)]

:= [возможности, навыки, приобретенные кибернетической системой]

:= [опыт, приобретенный кибернетической системой]

⇒ *примечание**:

[Новые навыки могут приобретаться кибернетической системой либо полностью самостоятельно, либо с помощью учителей, которые в простейшем случае просто сообщают обучаемой системе полностью сформулированные навыки. Для компьютерных систем учителями является их разработчики.]

- ⇒ *частное свойство**:
- множество методов решения задач, используемых кибернетической системой
 - множество моделей решения задач, используемых кибернетической системой
 - мощность языка представления в памяти кибернетической системы методов и моделей решения задач

множество методов решения задач, используемых кибернетической системой

:= [множество методов решения задач, используемых кибернетической системой и хранимых в ее памяти]

- ⇐ *частное свойство**:
 многообразие видов знаний, хранимых в памяти кибернетической системы

метод решения задач

- ⇒ *пояснение**:
 [метод решения задач – это вид знаний, хранимых в памяти кибернетической системы и содержащих информацию, которой достаточно либо для сведения каждой задачи из соответствующего класса задач к полной системе подзадач*, решение которых гарантирует решение исходной задачи, либо для окончательного решения этой задачи из указанного класса задач]

множество моделей решений задач, используемых кибернетической системой

:= [способность кибернетической системы к использованию различных видов методов решения задач, соответствующих различным моделям решения задач]

:= [многообразие методов решения задач, используемых кибернетической системой]

- ⇒ *свойство-предпосылка**:
 мощность языка представления в памяти кибернетической системы методов и моделей решения задач

множество моделей решения задач, используемых кибернетической системой

- ⇒ *примечание**:
 {
 следует отличать*
 ∃ { • вид задач
 • модель решения задач
 ⇒ *пояснение**:
 [каждая модель решения задач задается
 □ языком, обеспечивающим представление в памяти кибернетической системы некоторого класса методов решения задач
 □ интерпретатором указанных методов, определяющим операционную семантику указанного языка
]
 • метод решения задач
 • класс задач
 := [множество всех тех и только тех задач, которые решаются с помощью соответствующего метода]
 }
 }

Степень конвергенции и интеграции различного вида моделей решения задач, используемых кибернетической системой

- ⇒ *примечание**:
 [Необходим переход от эклектики никак не связанных друг с другом моделей решения задач к их конвергенции, это предполагает:
 □ разработку общего (базового) для всех моделей решения задач языка описания операционной семантики языков описания методов, соответствующих различным моделям решения задач;
 □ включение всех языков описания методов решения задач в общую систему языков, связанных между собой отношением “язык-подязык*”.
]

качество организации взаимодействия процессов решения задач в кибернетической системе

- ⇒ *частное свойство**:
 • качество управления информационным процессом в памяти кибернетической системы

- ⇒ *свойство-предпосылка**:
обеспечение процессором кибернетической системы качественного управления информационными процессами в памяти
- *качество организации взаимодействия процессов решения задач во внешней среде или в физической оболочке кибернетической системы*
- ⇒ *свойство-предпосылка**:
 - *последовательность/параллельность процессов решения задач в кибернетической системе*
 - *синхронность/асинхронность процессов решения задач в кибернетической системе*
 - *централизованной/децентрализованность управления процессами решения задач в кибернетической системе*

⇒ *примечание**:

[Качество решения каждой задачи определяется:

- временем её решения (чем быстрее задача решается, тем выше качество её решения);
- полнотой и корректностью результата решения задачи;
- затраченными для решения задачи ресурсами памяти (объемом фрагмента хранимой информации, используемой для решения задачи);
- затраченным для решения задачи ресурсами решателя задач (количеством используемых внутренних агентов).

Таким образом, повышение качества процесса решения каждой конкретной задачи, а также каждого класса задач (путем совершенствования соответствующего метода, в частности, алгоритма) является важным фактором повышения качества решателя задач в целом.]

агентно-ориентированная модель обработки информации в памяти

:= [агентно-ориентированная модель управления действиями кибернетической системы, выполняемыми ею в своей памяти]

⇒ *пояснение**:

[Перспективным вариантом построения решателя задач кибернетической системы является реализация агентно-ориентированной модели обработки информации, т.е. построение решателя задач в виде многоагентной системы, агенты которой осуществляют обработку информации, хранимой в памяти кибернетической системы, и управляются этой информацией (точнее, её текущим состоянием). Особое место среди этих агентов занимают сенсорные (рецепторные) и эффекторные агенты, которые, соответственно, воспринимают информацию о текущем состоянии внешней среды и воздействуют на внешнюю среду, в частности, путем изменения состояния физической оболочки кибернетической системы.

Подчеркнем, что указанная агентно-ориентированная модель организации взаимодействия процессов решения задач в кибернетической системе по сути есть не что иное, как модель ситуационного управления процессами решения задач, решаемых кибернетической системой как в своей внешней среде, так и в своей памяти.]

модель инициирования действий кибернетической системы

:= [модель управления поведением кибернетической системы]

⇒ *разбиение**:

- {
 - *стимульно-реактивная модель инициирования действий*
 - ⇒ *пояснение**:
[от комбинации исходных сигналов, формируемых, например, априори известным набором сенсоров (рецепторов) к комбинации выходных сигналов, управляющих, например, априори известным набором эффекторов]
 - *ситуационная модель инициирования действий без учета предыстории ситуаций и событий*
 - ⇒ *пояснение**:
[действие инициируется возникновением в памяти ситуации априори известной конфигурации или априори известного события]
 - *ситуационная модель инициирования действий с учетом предыстории ситуаций и событий*
 - ⇒ *пояснение**:
[действие инициируется не только текущей ситуацией но и предшествующими ситуациями, т.е. событиями перехода от одних ситуаций к другим]

⇒ *примечание**:

[Речь идет о действиях, выполняемых кибернетической системой как во внешней среде, так и в своей внутренней среде (в своей памяти).]

последовательность/параллельность процессов решения задач в кибернетической системе

:= [способность одновременно решать несколько разных задач, некоторые из которых могут быть подзадачами одной и той же задачи]

:= [способность одновременно решать несколько разных задач, некоторые из которых могут быть подзадачами одной и той же задачи]

⇒ *свойство-предпосылка**:

- *максимально возможное количество действий, одновременно выполняемых кибернетической системой*
- *способность кибернетической системы к одновременному выполнению взаимосвязанных действий*
 - :=** [способность кибернетической системы к одновременному выполнению действий, выполнение каждого из которых может помешать выполнению другого]
 - :=** [способность кибернетической системы к “эквилибристике”]

⇒ *комплекс частных свойств**:

- { • *физическая последовательность/параллельность процессов решения задач в кибернетической системе*
- *логическая последовательность/параллельность процессов решения задач в кибернетической системе*

⇒ *пояснение**:

[Логическая параллельность выполняемых процессов (действий) предполагает возможность существования выполняемых процессов в двух режимах:

- в активном режиме – в режиме непосредственного выполнения;
- в режиме прерывания – в режиме ожидания условий (событий и/или ситуаций) при возникновении которых прерванный процесс переходит в режим активного процесса.

]

}

⇒ *комплекс частных свойств**:

- { • *последовательность/параллельность информационных процессов в памяти кибернетической системы*
- *последовательность/параллельность процессов решения задач во внешней среде или в физической оболочке кибернетической системы*

}

⇒ *примечание**:

[Подчеркнем, что есть целый ряд задач, решаемых кибернетической системой, процессы решения которых носят перманентный (постоянный) характер. К таким задачам относятся:

- поддержка высокого качества базы знаний (устранение противоречий, информационного мусора);
- поддержка семантической совместимости с другими компьютерными системами;
- мониторинг и анализ состояния внешней среды;
- обеспечение собственной безопасности;
- самообучение.

]

быстродействие решателя задач кибернетической системы

:= [скорость решения задач в кибернетической системе]

:= [быстродействие решателя задач кибернетической системы]

:= [скорость реакции кибернетической системы на различные задачные ситуации]

⇒ *свойство-предпосылка**:

быстродействие процессора кибернетической системы

способность кибернетической системы решать задачи, предполагающие использование информации, обладающей различного рода не-факторами

:= [способность кибернетических систем решать задачи, которые:

- либо нечетко сформулированы ("делай то, не знаю что");
- либо решаются в условиях неполноты, неточности, противоречивости исходных данных;
- либо являются задачами, принадлежащими классам задач, для которых практически невозможно построить соответствующие алгоритмы.

]

- := [способность кибернетической системы решать труднорешаемые, трудноформализуемые задачи]
- := [способность решать интеллектуальные (трудноформализуемые) задачи, для которых характерна:
 - неточность и недостоверность исходных данных;
 - отсутствие критерия качества результата;
 - невозможность или высокая трудоемкость разработки алгоритма;
 - необходимость учета контекста задачи.
-]

задача, предполагающая использование информации, обладающей различного рода не-факторами

:= [трудноформализуемая задача]

- ⊃ задача проектирования
- ⊃ задача распознавания
- ⊃ задача прогнозирования
- ⊃ задача целеполагания
- ⊃ задача планирования

многообразие и качество решения задач информационного поиска

- ⇒ свойство-предпосылка*: семантический уровень доступа к информации, хранимой в памяти кибернетической системы
- ← частное свойство*: многообразии видов задач, решаемых кибернетической системой
- := [способность кибернетической системы качественно решать широкое многообразие задач информационного поиска в рамках текущего состояния хранимой информации]
- := [способность кибернетической системы находить в текущем состоянии хранимой информации релевантные ответы на запросы (вопросы) самого различного вида]

вопрос

- := [запрос]
- ⊃ запрос изоморфных или гомоморфных фрагментов хранимой информации по заданному образцу с указанием знаков известных сущностей
 - ⇒ класс частных вопросов*: запрос всех связей различных отношений, обязывающих заданную сущность с другими
 - ⇒ класс частных вопросов*: запрос всех связей заданных отношений, связывающих заданную сущность с другими
- ⊃ вопрос типа "как связаны между собой заданные две сущности"
 - ⇒ пояснение*: [Две сущности будем считать связанными в том и только в том случае, если существует маршрут, соединяющий указанные две сущности, в состав которого входят связи, принадлежащие в общем случае разным отношениям]
 - ⇒ примечание*: [Здесь принципиально важным является учет семантической силы связей между сущностями, которая определяется семантической силой отношений, которым принадлежат связи, входящие в состав связей (маршрутов) между сущностями.]
 - ← класс частных вопросов*: вопрос типа "как связаны между собой заданные сущности"
 - ⇒ примечание*: [Здесь имеется в виду произвольное количество связываемых сущностей, а это предполагает, что ответом на данный запрос является связный граф, вершинами которого являются знаки заданных сущностей.]
- ⊃ вопрос типа "что это такое"
 - := [запрос спецификации (описания) заданной сущности]
 - ⇒ класс частных вопросов*: запрос определения
 - := [запрос определения заданного понятия]
 - ⇒ класс частных вопросов*: запрос документации заданного объекта
- ⊃ почему-вопрос
 - ⊃ запрос причины возникновения заданной ситуации или события
 - ⊃ запрос логического обоснования заданного высказывания

	:=	[запрос объяснения корректности заданного высказывания, которое, в частности, может быть порождено (сгенерировано) в процессе решения некоторой задачи с помощью некоторого метода (алгоритма, искусственной нейронной сети логического исчисления и т.п.)]
	⊃	<i>запрос доказательства заданной теоремы</i>
⊃		<i>запрос возможных последствий заданной ситуации или события</i>
⊃		<i>запрос того, что логически следует из заданного высказывания</i>
⊃		<i>запрос метода решения данной задачи</i>
⊃		<i>запрос плана решения данной задачи</i>
	:=	[запрос декомпозиции данной задачи на систему и/или подзадач]
⊃		<i>зачем-вопрос</i>
	:=	[каково назначение заданной сущности]
	:=	[для решения какой задачи (для чего, достижения какой цели) нужна данная сущность]
⊃		<i>запрос аналогов заданной сущности</i>
⊃		<i>запрос антиподов заданной сущности</i>
⊃		<i>запрос сходств и отличий двух связанных сущностей</i>
⊃		<i>запрос сравнительного анализа заданной сущности</i>
	⊃	<i>запрос достоинств заданной сущности</i>
	⊃	<i>запрос недостатков заданной сущности</i>
⊃		<i>где-вопрос</i>
	:=	[запрос информации о местоположении заданной пространственной сущности примечание]
	⇒	<i>примечание*:</i> [Здесь запрашивается любая информация о пространственных связях заданной сущности]
⊃		<i>когда-вопрос</i>
	:=	[запрос информации о темпоральных свойствах и связях заданной временной сущности (о моменте начала, о моменте завершения, о длительности)]

способность кибернетической системы генерировать ответы на вопросы различного вида в случае, если они целиком или частично отсутствуют в текущем состоянии информации, хранимой в памяти

	:=	[способность кибернетической системы генерировать (порождать, строить, синтезировать, выводить) ответы на самые различные вопросы и, в частности, на вопросы типа “что это такое”, на почему-вопросы, это означает способность кибернетической системы <u>объяснять</u> (обосновывать корректность) своих действий]
	⇒	<i>свойство-предпосылка*:</i>
	•	<i>семантическая гибкость информации, хранимой в памяти кибернетической системы</i>
	•	<i>способность кибернетической системы к рассуждениям различного вида</i>

способность кибернетической системы к рассуждениям различного вида

	:=	[способность кибернетической системы к целенаправленному порождению (генерации) новых истинных или правдоподобных знаний (следствий) на основе имеющихся знаний (посылок)]
	⇒	<i>частное свойство*:</i>
	•	<i>способность кибернетической системы к дедуктивному выводу</i>
	•	<i>способность кибернетической системы к индуктивному выводу</i>
	•	<i>способность кибернетической системы к абдуктивному выводу</i>

качество целеполагания

	:=	[качество реализации первого этапа решения сложных задач – этапа генерации (построения) планов решения сложных задач]
	:=	[качество генерации планов выполнения сложных действий:
	□	как внутренних действий (в памяти кибернетической системы), так и внешних действий (во внешней среде)
	□	как собственных действий, так и действий других субъектов
]	
	:=	[качество генерации планов действий кибернетической системы и, в частности, трудоемкость процесса генерации этих планов]
	:=	[качество организации целенаправленной деятельности кибернетической системы]
	:=	[качество построения цепочек "цель-план-действие"]
	:=	[качество генерации, анализа и инициирования собственных целей (собственных задач)]
	:=	[способность кибернетической системы к целеполаганию]
	⇒	<i>свойство-предпосылка*:</i>
	•	<i>самостоятельность целеполагания</i>

- := [самостоятельность генерации и инициирования целей (задач), направленных на создание условий достижения соответствующих стратегических целей (сверхзадач)]
- *целенаправленность целеполагания*
- := [степень соответствия (степень полезности) генерируемых целей (задач) для достижения соответствующих стратегических целей (сверхзадач)]
- *сбалансированность целеполагания*
- := [качество расстановки приоритетов у сгенерированных и инициированных целей (задач) для обеспечения баланса между тактическими и стратегическими целями]

самостоятельность целеполагания

- := [способность кибернетической системы генерировать, инициировать и решать задачи, которые не являются подзадачами, инициированными внешними (другими) субъектами, а также способность на основе анализа своих возможностей отказаться от выполнения задачи, инициированной извне, переадресовав её другой кибернетической системе, либо на основе анализа самой этой задачи обосновать её нецелесообразность или некорректность]
- := [способность к самостоятельному целеполаганию (генерации идей) и к инициированию процессов их достижения (т.е. к принятию решений), способность свободно (в определенных рамках) выбирать (ставить перед собой цели)]
- := [уровень самостоятельности]
- := [способность решать задачи в комплексе, включая создание всех необходимых условий для их решения с учетом конкретных обстоятельств]
- := [умение решать задачи в условиях сильных помех (в осложненных обстоятельствах)]
- ⇒ *примечание**:
[Повышение уровня самостоятельности существенно расширяет возможности кибернетической системы, т.е. объем тех задач, которые она может решать не только в "идеальных" условиях, но и в реальных (осложненных) обстоятельствах.]
- := [степень свободы выбора целей, подлежащих достижению, а также свободы генерации целей, не являющихся подцелями извне поставленных целей]

целенаправленность целеполагания

- := [целеустремленность]
- := [целенаправленность]
- := [степень целостности деятельности]
- := [степень соответствия между тактическими и стратегическими уровнями деятельности]
- := [общее соотношение между временем, затраченным на "лишние" (ненужные, нецелесообразные, нецеленаправленные) действия и полезные действия]
- := [целесообразность деятельности]
- := [способность адекватно расставлять приоритеты своим целям и не "распыляться" на достижение неприоритетных (несущественных) целей]

качество реализации планов собственных действий

- := [качество реализации целенаправленной деятельности на основе построенных планов]
- := [качество реализации построенных в памяти кибернетической системы планов выполнения сложных собственных действий, которые могут предполагать участие других субъектов]

способность кибернетической системы к локализации такой области информации, хранимой в ее памяти, которой достаточно для обеспечения решения заданной задачи

- := [способность кибернетической системы к сужению области решения каждой решаемой ею задачи, что существенно минимизирует затраты кибернетической системы на учет и анализ факторов, априори незначимых (несущественных) для решения каждой решаемой задачи]
- ⇒ *примечание**:
[Для реализации данной способности важное значение имеет качественная стратификация базы знаний кибернетической системы на предметные области и соответствующие им онтологии.]

способность кибернетической системы к выявлению существенного в информации, хранимой в ее памяти

:= [способность к выявлению (обнаружению, выделению) таких фрагментов информации, хранимой в памяти кибернетической системы, которые существенны (важны) для достижения соответствующих целей]

⇒ *примечание**:

[Понятие существенного (важного) фрагмента информации, хранимой в памяти кибернетической системы, относительно и определяется соответствующей задачей. Тем не менее, есть важные перманентно (постоянно) решаемые задачи, в частности задачи анализа качества информации, хранимой в памяти кибернетической системы. Существенные фрагменты хранимой информации, выделяемые в процессе решения этих задач, являются относительными не столько по отношению к решаемой задаче, сколько по отношению к текущему состоянию хранимой информации. Примерами таких фрагментов являются:

- обнаруженные противоречия (ошибки) с явным указанием того, что чему противоречит;
- обнаруженные информационные дыры, точнее точная спецификация этих дыр;
- обнаруженные мусорные фрагменты, которые либо носят вспомогательный характер, либо могут быть легко восстановлены (воспроизведены).

]

следует отличать*

- Э** { • *способность кибернетической системы к выявлению существенного в информации, хранимой в ее памяти*
- ⇒** *примечание**:
[Здесь кибернетическая система выделяет информацию, которая необходима, но не обязательно достаточна для решения соответствующей задачи.]
- *способность кибернетической системы к локализации такой области информации, хранимой в ее памяти, которой достаточно для обеспечения решения заданной задачи*
- ⇒** *примечание**:
[Здесь кибернетическая система "отбрасывает" (исключает) информацию, которая априори несущественна для решения соответствующей (заданной) задачи.]
- }

активность кибернетической системы

:= [уровень активности кибернетической системы]

:= [уровень мотивации к деятельности в различных направлениях]

:= [уровень "желания" действовать]

:= [активность/пассивность кибернетической системы]

:= [уровень инициативности, пассионарности, мотивированности]

⇒ *примечание**:

[Уровень активности кибернетической системы может быть разным для разных решаемых задач, для разных классов выполняемых действий, для разных видов деятельности.]

⇒ *примечание**:

[Следует отличать уровень активности (мотивации, желания) и направленность этой активности.]

⇒ *примечание**:

[Чем выше активность кибернетической системы, тем (при прочих равных условиях) она больше успевает сделать, следовательно, тем выше ее качество (эффективность).]

⇔ *обратное свойство**:

пассивность

:= [уровень бездеятельности, медлительности, вялости, ленивости]

⇒ *частное свойство**:

- *познавательная активность*
- *социальная активность*

/*****/

качество логико-семантической организации памяти кибернетической системы

:= [качество базовых семантически целостных действий в памяти кибернетической системы]

:= [качество семантически элементарных (законченных, целостных) информационных процессов, выполняемых кибернетической системой в своей памяти]

- := [интегральная оценка того, насколько способствует (насколько близка) организация памяти кибернетической системы реализации "осмысленных" преобразований, хранимых в памяти знаний]
- := [степень приспособленности решателя задач кибернетической системы к обработке сложноструктурированных баз знаний]
- := [степень приспособленности решателя задач кибернетической системы к обработке хранимой в её памяти информации, имеющий высокий уровень качества как по форме представления информации, так и по её содержанию – по многообразию представляемых знаний и по уровню их конвергенции и интеграции]
- ⇒ *свойство-предпосылка**:
- *семантический уровень доступа к информации, хранимой в памяти кибернетической системы*
 - *семантическая гибкость информации, хранимой в памяти кибернетической системы*
 - *степень конвергенции и интеграции представления навыков, хранимых в памяти кибернетической системы, с представлением обрабатываемой информации*
- семантический уровень доступа к информации, хранимой в памяти кибернетической системы**
- := [степень ассоциативности доступа к информации, хранимой в памяти кибернетической системы]
- := [способность кибернетической системы локализовывать (находить) требуемый (запрашиваемый) фрагмент информации, хранимой в её памяти, не на основании известного адреса запрашиваемой информации (её местоположения в памяти), а на основании:
- известного типа запрашиваемой информации;
 - известных сущностей, знаки которых входят в состав запрашиваемой информации;
 - полностью или частично известной конфигурации запрашиваемой информации (т.е. конфигурации связей между известными и искомыми сущностями)
-]]
- ⇒ *пояснение**:
- [*уровень доступа к информации, хранимой в памяти кибернетической системы* определяется тем, что нам достаточно знать об искомой в памяти кибернетической системы информации (в частности, об искомом знаке некоторой интересующей нас сущности). Мы можем знать место в памяти (ячейку памяти, область памяти), где находится интересующая нас информация. Такой доступ называется адресным. Мы можем знать имя интересующей нас сущности, но не знать, где находится информация, описывающая эту сущность. Мы можем не знать имени интересующей нас сущности, но знать, как эта сущность связана с другими известными нам сущностями.]
- ⇒ *пояснение**:
- [Пусть нам необходимо локализовать (выделить) хранимую в памяти информацию, описывающую известные нам сущности, связанные известными нам отношениями, но местонахождение этой информации в памяти нам не известно. Если организация памяти нам представляет такую возможность, то такую память будем называть ассоциативной, т.е. памятью, обеспечивающей семантический доступ к хранимой в ней информации.]
- ⇒ *примечание**:
- [Для того, чтобы построить информационную модель среды, в которой действует (функционирует) кибернетическая система, необходимо, с одной стороны, "разложить" эту информационную модель "по полочкам", превратить её в некую систему из компонентов этой информационной модели, а, с другой стороны, обеспечить быстрый поиск нужного фрагмента указанной информационной модели, не зная, на каких "полочках" находятся компоненты этого искомого фрагмента, который при этом может иметь произвольную конфигурацию и произвольный размер. Это и есть высший уровень ассоциативности доступа к информации, хранимой в памяти кибернетической системы.]
- ⇒ *пояснение**:
- [Данное свойство, данная характеристика организации информации, хранимой в памяти кибернетической системы, является важнейшей характеристикой внутреннего языка представления информации в памяти кибернетической системы. Указанная характеристика внутреннего языка определяется простотой процедур поиска востребованных (запрашиваемых) фрагментов хранимой информации – например, процедуры поиска знаков всех сущностей, каждая из которых связана с заданными (известными) сущностями связями заданных (известных) типов, процедуры поиска (выделения) знаков всех сущностей, которые связаны с заданной (известной) сущностью связью неважно какого типа, процедуры поиска информационного фрагмента заданному образцу (шаблону) произвольного размера и конфигурации, в котором выделены знаки известных сущностей и условные обозначения искомым сущностей.]
- ⇒ *свойство-предпосылка**:
- степень близости языка внутреннего представления информации в памяти кибернетической системы к смысловому представлению информации*

семантическая гибкость информации, хранимой в памяти кибернетической системы

- ⇒ *свойство-предпосылка**:
степень близости языка внутреннего представления информации в памяти кибернетической системы к смысловому представлению информации
- := [простота реализации базовых (элементарных), но семантически целостных (семантически значимых, осмысленных) действий (операций) преобразования (обработки) информации, хранимой в памяти кибернетической системы]

базовое семантически целостное действие над информацией, хранимой в памяти кибернетической системы

- := [элементарная семантически значимая (осмысленная) операция над информацией, хранимой в памяти кибернетической системы]
- ⇒ *примечание**:
 [Здесь принципиальной является семантическая целостность (осмысленность) действия над хранимой информацией. Так, например, операция адресного доступа к требуемому фрагменту хранимой информации не является семантически целостной, так как смысл искомого (запрашиваемого) фрагмента хранимой информации не уточняется.]
- ⇒ *примечание**:
 [Разные кибернетические системы могут использовать разные наборы классов базовых семантически целостных действий над информацией, хранимой в их памяти.]
- ⇒ *примечание**:
 [Примерами базовых семантически целостных действий над информацией, хранимой в памяти кибернетической системы, в частности, являются:
- операции поиска, генерации, удаления или замены связей между знаками известных сущностей;
 - операции поиска, генерации, удаления или замены имен, приписываемых знакам известных сущностей.
- Существенно подчеркнуть, что простота реализации такого рода операций (т.е. гибкость хранимой в памяти информации) во многом обеспечивается стремлением к локальности выполнения этих операций. Такая локальность означает то, что при выполнении каждой из указанных операций меняется только обрабатываемый фрагмент хранимой информации и не требуется никакого перемещения в памяти остальной части хранимой информации.]

степень конвергенции и интеграции представления навыков, хранимых в памяти кибернетической системы, с представлением обрабатываемой информации

- ⇐ *частное свойство**:
степень конвергенции и интеграции различного вида знаний, хранимых в памяти кибернетической системы
- ⇒ *свойство-предпосылка**:
степень близости языка внутреннего представления информации в памяти кибернетической системы к смысловому представлению информации
- ⇒ *примечание**:
 [Навыки кибернетической системы являются частным видом знаний, хранимых в её памяти, поэтому степень конвергенции навыков и обрабатываемых знаний определяется "глубиной" и "объемом" общих (одинаковых) принципов, лежащих в основе как представления навыков, так представления обрабатываемых знаний.]

/*****

качество решения интерфейсных задач в кибернетической системе

- ⇒ *частное свойство**:
- способность кибернетической системы к пониманию сенсорной информации
 - способность кибернетической системы к пониманию принимаемых сообщений
 - способность кибернетической системы к самостоятельной деятельности во внешней среде
- := [способность кибернетической системы к воздействию на внешнюю среду и к управлению своим поведением во внешней среде]

интерфейсная задача

- ⊃ *задача анализа введенной информации*
- ⊃ *задача анализа сенсорной информации*
- := [задача анализа информации, порождаемой (генерируемой) непосредственно сенсорами кибернетической системы]

- ⊃ *задача синтаксического анализа сенсорной информации*
- ⊃ *задача семантического анализа сенсорной информации*
- := [задача анализа сенсорной информации, направленного на понимание этой информации – на выявление (распознавание) в этой информации отображения (сенсорного описания) объектов, важных для кибернетической системы (т.е. объектов, описанных в базе знаний этой системы и, соответственно, представленных в этой базе знаний своими знаками либо знаками классов, которым эти объекты принадлежат), а также важных для кибернетических связей между указанными объектами]
- := [задача генерации фрагмента базы знаний кибернетической системы, являющегося логическим следствием заданной сенсорной информации и представляющего собой важную для кибернетической системы информацию]
- := [задача извлечения из сенсорной информации (первичной информации) важной для кибернетической системы вторичной информации]
- ⊃ *задача анализа принимаемого вербального сообщения*
- := [задача анализа введенных знаковых конструкций]
- := [задача анализа сообщений, введенных в кибернетическую систему]
- := [задача анализа внешних знаковых конструкций]
- ⊃ *задача синтаксического анализа принимаемого вербального сообщения*
- ⊃ *задача трансляции принимаемого вербального сообщения на внутренний язык кибернетической системы*
- ⊃ *задача погружения нового фрагмента в состав согласованной части базы знаний*
- := [задача интеграции (встраивания) нового фрагмента базы знаний в состав базы знаний]
- := [задача понимания нового фрагмента базы знаний в контексте её текущего состояния, что, прежде всего, требует обеспечения семантической совместимости (согласования понятий) между базой знаний и интегрируемым новым фрагментом]
- ⊃ *задача управления эффекторами кибернетической системы при выполнении сложных воздействий на внешнюю среду и/или физическую оболочку этой кибернетической системы*
- := [задача целенаправленной сенсорно-эффекторной (в частности, сенсомоторной) координации]

сенсорная информация

- := [информация, генерируемая непосредственно некоторой группой (конфигурацией) сенсоров (рецепторов) кибернетической системы]
- := [рецепторная информация]
- := [первичная информация, получаемая (приобретаемая) кибернетической системой]
- := [первичная знаковая конструкция, которая описывает те или иные свойства текущего состояния физической окружающей среды (внешней среды и физической оболочки) кибернетической системы]

сенсор кибернетической системы

- := [рецептор кибернетической системы]
- ⇒ *пояснение**:
[Компонент кибернетической системы, генерирующий в памяти этой системы информацию о текущем значении соответствующего этому компоненту свойства (характеристики, параметра) того фрагмента физической окружающей среды кибернетической системы, который непосредственно смежен (пограничен) указанному компоненту.]

эффектор кибернетической системы

- := [компонент кибернетической системы, который способен менять своё состояние в целях непосредственного воздействия на свою физическую оболочку и на внешнюю среду]

способность кибернетической системы к пониманию сенсорной информации

- := [способность к синтаксическому и семантическому анализу информации, формируемой сенсорами кибернетической системы, а также к "погружению" этой информации в состав общей информационной модели внешней среды кибернетической системы (в состав общей картины внешнего мира)]
- := [способность кибернетической системы к переходу от первичной (сенсорной) информации ко вторичной информации, которая описывает связи между вторичными объектами, каждый из которых представлен (описан) в первичной информации конфигурацией знаков своих частей с дополнительным описанием свойств каждой из этих частей]

способность кибернетической системы к самостоятельной деятельности во внешней среде

⇒ свойство-предпосылка*:

- уровень развития эффекторов, обеспечивающих самостоятельное перемещение кибернетической системы
 - ⇒ частное свойство*:
 - уровень развития эффекторов, обеспечивающих локальное перемещение сенсоров кибернетической системы
 - уровень развития эффекторов, обеспечивающих функционирование манипуляторов кибернетической системы
 - уровень развития эффекторов, обеспечивающих перемещение всей физической оболочки кибернетической системы
- качество управления поведением кибернетической системы во внешней среде
 - := [качество сенсорно-эффекторной координации действий кибернетической системы при выполнении сложных действий во внешней среде]

} /* Завершили Сегмент “Комплекс свойств, определяющих качество решателя задач кибернетической системы” */

/* Сегмент *****/

Комплекс свойств, определяющих уровень обучаемости кибернетической системы

⊃=

{

обучаемость кибернетической системы

:= [способность кибернетической системы повышать своё качество, адаптируясь к решению новых задач, качество внутренней информации модели своей среды, качество своего решателя задач и даже качество своей физической оболочки.]

:= [способность кибернетической системы к самосовершенствованию с различной степенью самостоятельности (с учителем, с экспертом, с внешними источниками информации, только на собственном опыте)]

⇔ следует отличать*:

приспособленность кибернетической системы к её совершенствованию, осуществляемому извне

:= [способность кибернетической системы к самостоятельному повышению уровня (качества) своих знаний, навыков, а также уровня своей обучаемости]

:= [способность кибернетической системы к самостоятельному самосовершенствованию]

:= [скорость эволюции кибернетической системы]

:= [уровень (степень) обучаемости кибернетической системы]

:= [способность кибернетической системы к совершенствованию (к эволюции, к повышению уровня своего качества)]

⇒ примечание*:

[Максимальный уровень обучаемости кибернетической системы – это её способность эволюционировать (повышать уровень своего качества) максимально быстро и в любом(!) направлении, т.е. способность быстро и без каких-либо ограничений приобретать любые(!) новые знания и навыки.]

:= [способность кибернетической системы к повышению своего качества (в том числе, путем устранения своих недостатков, выявленных в результате самоанализа (рефлексии), в частности, в результате работы над своими ошибками, разбора собственных "полетов")]

:= [способность кибернетической системы к обучению]

:= [умение кибернетической системы учиться]

:= [способность кибернетической системы обучаться]

⇒ примечание*:

[Реализация способности кибернетической системы обучаться, т.е. решать перманентно инициированную сверхзадачу самообучения, накладывает дополнительные требования, предъявляемые к информации, хранимой в памяти кибернетической системы, к решателю задач кибернетической системы, а в перспективе также и к физической оболочке кибернетической системы.]

:= [способность кибернетической системы повышать уровень своего интеллекта – (1) общий (интегральный) уровень качества информации, хранимый в собственной памяти, (2) общий уровень качества своих приобретаемых навыков, (3) уровень своей обучаемости.]

:= [способность кибернетической системы к максимально возможной самостоятельной эволюции, в процессе которой кибернетическая система сама постоянно заботится о своей эволюции и о повышении темпов этой эволюции]

⇒ *примечание**:

[Важнейшей характеристикой кибернетической системы является не только то, какой уровень интеллекта (интеллектуальных возможностей) кибернетическая система имеет в текущий момент, какое множество действий (задач) она способна выполнять, но и то, насколько быстро этот уровень может повышаться.]

следует отличать*

⊃ {

- *образованность кибернетической системы*
:= [навыки и другие знания, которые кибернетическая приобрела (с учителем, экспертом или самостоятельно) к заданному моменту]
:= [результат, который кибернетическая система достигла в процессе своей эволюции к заданному моменту]
- *обучаемость кибернетической системы*
:= [скорость повышения уровня образованности кибернетической системы]
:= [скорость эволюции кибернетической системы]
- *скорость повышения уровня обучаемости кибернетической системы*
:= [ускорение повышения уровня образованности кибернетической системы]
⇒ *примечание**:
 [с увеличением объема и качества приобретаемых кибернетической системой новых навыков и знаний и, в первую очередь, при грамотной их систематизации скорость обучения кибернетической системы существенно возрастает.]

}

обучаемость кибернетической системы

⇒ *комплекс свойств-предпосылок**:

Комплекс свойств, определяющих обучаемость кибернетических систем по уровню обучаемости различных их компонентов

⇒ *комплекс частных свойств**:

Комплекс свойств кибернетических систем, определяющих их обучаемость по различным формам обучения

/*****/

Комплекс свойств, определяющих обучаемость кибернетических систем по уровню их гибкости, стратифицированности, рефлексивности, активности

= {

- *гибкость кибернетической системы*
- *стратифицированность кибернетической системы*
- *рефлексивность кибернетической системы*
- *ограниченность обучения кибернетической системы*
- *познавательная активность кибернетической системы*
- *способность кибернетической системы к самосохранению*

}

гибкость возможных самоизменений кибернетической системы

:= [гибкость кибернетической системы при выполнении ею изменений над самой этой системой]

⇒ *комплекс свойств-предпосылок**:

{

- *простота возможных самоизменений кибернетической системы*
- *многообразие возможных самоизменений кибернетической системы*

}

⇒ *комплекс частных свойств**:

{

- *семантическая гибкость обработки информации, хранимой в памяти кибернетической системы*
- *семантическая гибкость возможных самоизменений решателя задач кибернетической системы*
- *гибкость возможных изменений физической оболочки кибернетической системы, осуществляемых самой системой*

}

гибкость возможных самоизменений кибернетической системы

:= [гибкость кибернетической системы при её самосовершенствовании]

- ← *частное свойство**:
гибкость кибернетической системы
- ⇒ *примечание**:
[[Поскольку обучение всегда сводится к внесению тех или иных изменений в обучаемую кибернетическую систему, без высокого уровня гибкости этой системы не может быть высокого уровня её обучаемости.]
- ← *свойство-предпосылка**:
обучаемость кибернетической системы

простота возможных самоизменений кибернетической системы

- := [легкость (трудоемкость) внесения различных изменений в кибернетическую систему, осуществляемых самой этой кибернетической системой]
- := [приспособленность кибернетической системы к самостоятельному внесению различных изменений в саму себя]

стратифицированность кибернетической системы

- := [иерархическая декомпозиция кибернетической системы на такие подсистемы, структура и функционирование которых минимально возможным образом связаны друг с другом, что существенным образом сужает область учета последствий различных изменений вносимых в систему, а также область поиска причин всевозможных ошибок]
- := [модульность кибернетической системы]
- := [возможность разделить кибернетическую систему на такие части (страты), эволюция (изменения) которых может осуществляться независимо друг от друга.]
- ⇒ *примечание**:
[Уровень стратифицированности определяется
- количеством страт;
 - степенью зависимости страт друг от друга.
-]
- ⇒ *примечание**:
[[При наличии стратифицированности кибернетической системы появляется возможность четкого определения области действия различных изменений, вносимых в кибернетическую систему, т.е. возможность четкого ограничения тех частей кибернетической системы, за пределы которых нет необходимости выходить для учета последствий внесенных в систему первичных изменений, т.е. осуществлять дополнительные изменения, являющиеся последствиями первичных изменений.]
- ⇒ *примечание**:
[Стратификация кибернетической системы – это не просто её структуризация (прежде всего, структуризация информации, хранимой в памяти кибернетической системы), а такая её структуризация, которая четко определяет границы учета возможных последствий вносимых в систему изменений различного вида.]
- ⇒ *частное свойство**:
- *стратифицированность информации, хранимой в памяти кибернетической системы*
 - *стратифицированность решателя задач кибернетической системы*
 - *стратифицированность физической оболочки кибернетической системы*

рефлексивность кибернетической системы

- := [уровень (степень) рефлексивности кибернетической системы]
- := [способность кибернетической системы к самоанализу (к анализу интегрального уровня своего качества и, в том числе, уровня своего интеллекта)]
- := [способность кибернетической системы самостоятельно анализировать (оценивать) свое качество]
- := [уровень рефлексии кибернетической системы]
- := [способность кибернетической системы к самоанализу – к анализу своих знаний, навыков, своих действий во внутренней и внешней среде]
- := [способность кибернетической системы к самонаблюдению и самоанализу]
- := [способность кибернетической системы к рефлексии]
- := [способность кибернетической системы к анализу своего качества]
- := [Способность кибернетической системы к самоанализу (к анализу самой себя во всевозможных аспектах).]

⇒ *примечание**:
 [Конструктивным результатом рефлексии кибернетической системы является генерация в её памяти спецификации различных негативных или подозрительных особенностей, которые следует учитывать для повышения качества кибернетической системы. Такими особенностями (недостатками) могут быть выявленные противоречия (ошибки), выявленные пары синонимичных знаков, омонимичные знаки, информационные дыры и многое другое.]

⇒ *частное свойство**:

- *способность кибернетической системы к анализу качества информации, хранимой в её памяти*
- *способность кибернетической системы к анализу качества своего решателя задач*
 ⇒ *частное свойство**:
способность кибернетической системы к анализу качества своего поведения во внешней среде
- *способность кибернетической системы к анализу качества своей физической оболочки*
 ⇒ *частное свойство**:
способность кибернетической системы к анализу качества физического обеспечения своего интерфейса с внешней средой

ограниченность обучения кибернетической системы

⇒ *пояснение**:
 [Данное свойство определяет границу между теми знаниями и навыками, которые соответствующая *кибернетическая система* принципиально может приобрести, и теми знаниями и навыками, которые указанная кибернетическая система не сможет приобрести никогда. Данное свойство определяет максимальный уровень потенциальных возможностей соответствующей кибернетической системы. Очевидно, что максимальная степень отсутствия ограничений в приобретении новых знаний и навыков – это полное отсутствие ограничений, т.е. полная универсальность возможностей соответствующих кибернетических систем, которые всё могут познать и всё могут сотворить.]

:= [максимум того, чему кибернетическая система может обучиться]

:= [максимальная перспектива обучения кибернетической системы]

:= [максимальный уровень качества, который кибернетическая система может достичь в процессе обучения]

⇒ *частное свойство**:

- *максимальный объём знаний, которые кибернетическая система может приобрести в процессе обучения*
- *максимальный объём навыков, которые кибернетическая система может приобрести в процессе обучения*

максимальный объём знаний, которые кибернетическая система может приобрести в процессе обучения

:= [граница приобретаемых знаний, за пределы которой кибернетическая система принципиально не может перейти в процессе своего обучения]

:= [максимум того, чему можно научить соответствующую кибернетическую систему]

:= [максимальный объём знаний, которые кибернетическая система принципиально может приобрести]

⇐ *свойство-предпосылка**:

обучаемость

⇒ *примечание**:

[чем больше *максимальный объём знаний, которые кибернетическая система принципиально может приобрести*, тем выше уровень *обучаемости* кибернетической системы]

познавательная активность кибернетической системы

:= [познавательная мотивированность]

:= [познавательная пассионарность]

:= [любопытность]

:= [активность и самостоятельность в приобретении новых знаний и навыков]

:= [стремление, активная целевая установка к постоянному совершенствованию (повышению качества) и пополнению собственной базы знаний]

⇒ *примечание**:

[Следует отличать

- способность (возможность) приобретать новые знания и навыки и совершенствовать приобретенные знания и навыки

□ от желания (стремления) это делать.

]

⇒ *примечание**:

[желание (целевая установка) научиться решать те или иные задачи может быть сформулировано кибернетической системой либо самостоятельно, либо извне (некоторым учителем).]

⇒ *частное свойство**:

- *активность в изучении внешней среды*
- *активность в анализе качества информации, хранимой в собственной памяти*
- *активность в анализе собственных действий и действий других кибернетических систем*

⇒ *свойство-предпосылка**:

- *способность кибернетической системы к синтезу познавательных целей и процедур*
- *способность кибернетической системы к самоорганизации собственного обучения*
- *способность кибернетической системы к экспериментальным действиям*

способность кибернетической системы к синтезу познавательных целей и процедур

:= [способность планировать своё обучение и управлять процессом обучения]

:= [умение задавать вопросы или целенаправленные последовательности вопросов самому себе или другим субъектам как важнейший фактор обучаемости]

:= [способность генерировать (формулировать, задавать) вопросы, адресуемые либо самому себе, либо некоторому внешнему источнику знаний и направленные на повышение качества собственных знаний и навыков]

:= [способность генерировать четкую спецификацию своей информационной потребности]

:= [способность кибернетической системы четко формулировать то, что она не знает (в частности, не умеет), но хотела бы знать и уметь]

:= [способность к формированию спецификаций информационных баз в своих знаниях]

:= [способность кибернетической системы самостоятельно генерировать цели на приобретение знаний и навыков, обеспечивающих решение различных классов задач]

способность кибернетической системы к самоорганизации собственного обучения

:= [способность осуществлять управление своим обучением]

:= [способность кибернетической системы самой выполнять роль своего учителя, организующего процесс своего обучения]

способность кибернетической системы к экспериментальным действиям

:= [способность к отклонениям от составленных планов своих действий для повышения качества результата или сохранении целенаправленности этих действий]

:= [способность к экспромтам и импровизации]

способность кибернетической системы к самосохранению

:= [способность кибернетической системы к выявлению и устранению угроз, направленных на снижение её качества и даже на её уничтожение, что означает полную потерю необходимого качества]

:= [уровень самообеспечения безопасности (защищенности) кибернетической системы]

⇒ *пояснение**:

[Данное свойство кибернетических систем является необходимым фактором высокого уровня обучаемости кибернетических систем. Чем выше уровень безопасности кибернетической системы, тем выше её уровень обучаемости.]

:= [способность кибернетической системы к обеспечению собственной безопасности]

⇒ *свойство-предпосылка**:

- *способность кибернетической системы анализировать смысл задач, инициированных извне, и отказываться от решения вредных задач*

⇒ *эпиграф**:

[Прежде, чем выполнять приказ, подумай]

⇒ *пояснение**:

[Примером вредной задачи для *ostis*-системы является запрос всех хранимых в памяти *sc*-элементов]

⇒ *пояснение**:

[Подчеркнем, что в современных компьютерных системах и интеллектуальных компьютерных системах подходы к обеспечению их информационной безопасности имеют принципиальные отличия, связанные, прежде всего с тем интеллектуальные компьютерные системы обладают более мощными средствами семантического и контекстного анализа приобретаемой информации.]

/*****/

Комплекс свойств, определяющих обучаемость кибернетических систем по уровню обучаемости различных их компонентов

= { ● способность кибернетической системы к повышению качества информации хранимой в её памяти
● способность кибернетической системы к повышению качества своего решателя задач
● способность кибернетической системы к повышению качества своей физической оболочки
}

способность кибернетической системы к повышению качества информации, хранимой в её памяти

:= [способность кибернетической системы к постоянному пополнению и совершенствованию информации, хранимой в её памяти, по всевозможным направлениям и, в первую очередь, в направлении повышения уровня адекватности (корректности) и полноты описания своей внешней среды и своей физической оболочки]

⇒ *свойство-предпосылка**:

- семантическая гибкость информации, хранимой в памяти кибернетической системы
- стратифицированность информации, хранимой в памяти кибернетической системы
- способность кибернетической системы к повышению уровня структуризации информации, хранимой в памяти кибернетической системы
- способность кибернетической системы к анализу качества информации, хранимой в её памяти
- способность кибернетической системы к устранению противоречий, обнаруженных в информации, хранимой в её памяти
- способность кибернетической системы к устранению информационных дыр, обнаруженных в информации, хранимой в её памяти
- способность кибернетической системы к удалению информационного мусора, обнаруженного в информации, хранимой в её памяти
- способность кибернетической системы к погружению новых знаний в состав информации, хранимой в её памяти
- способность кибернетической системы к обнаружению сходств в знаниях, хранимых в её памяти
- способность кибернетической системы к конвергенции знаний, хранимых в её памяти
- способность кибернетической системы к интеграции знаний, хранимых в её памяти
- способность кибернетической системы к обобщениям и формированию новых понятий
- способность кибернетической системы к генерации гипотез и обнаружению закономерностей в информации, хранимой в её памяти
- способность кибернетической системы к обоснованию или опровержению знаний, хранимых в её памяти
- способность кибернетической системы к экспериментальному подтверждению или опровержению гипотез о свойствах динамических систем с помощью имитационных моделей этих систем
- способность кибернетической системы к коррекции теорий, хранимых в её памяти

семантическая гибкость информации, хранимой в памяти кибернетической системы

:= [гибкость информации, хранимой в памяти кибернетической системы, при её обработке на семантическом уровне]

:= [гибкость возможных действий (операций), выполняемых кибернетической системой над информацией, хранимой в её памяти, и осуществляемых на семантическом (осмысленном) уровне представления этой информации]

:= [трудоемкость содержательного редактирования информации, хранимой в памяти кибернетической системы (поиска, удаления, вставки, замены различных фрагментов информации), при соблюдении семантической целостности и корректности всей редактируемой информации]

⇒ *примечание**:

[Обработка информации на семантическом уровне предполагает такие операции над хранимой информацией, как:

- замена имени некоторой сущности
- поиск связи заданного вида между знаками заданных сущностей и корректировка этой связи
- поиск семантической окрестности знака заданной сущности, то есть поиск всех известных связей, инцидентных этому знаку и, соответственно, всех смежных ему знаков
- поиск фрагмента хранимой информации, релевантного заданному семантическому образцу – конфигурации знаков сущностей и связей между ними

□ удаление или генерация (порождение) связи между заданными знаками

]

⇒ *примечание**:

[Все операции семантического уровня обработки информации рассматривают обрабатываемую информацию на абстрактном уровне знаков описываемых сущностей и знаков связей между описываемыми сущностями. При этом указанные связи рассматриваются как частный вид описываемых (и, соответственно, обозначаемых) сущностей.]

:= [простота и многообразие редактирования информации, хранимой в памяти кибернетической системы]

:= [простота и многообразие внесения изменений в информацию, хранимую в памяти кибернетической системы]

⇒ *пояснение**:

[*Гибкость обработки информации, хранимой в памяти кибернетической системы*, определяется не столько трудоемкостью непосредственно самой операции редактирования, сколько тем дополнительными действиями, которые являются обязательными последствиями каждой такой операции редактирования. Так, например, изменение имени какой-либо описываемой сущности требует внесения этого изменения во всех местах, где это имя упоминается, удаление какой-либо связи между известными описываемыми сущностями требует внесения этого изменения везде, где удаляемая связь упоминается.]

стратифицированность информации, хранимой в памяти кибернетической системы

:= [логико-семантическая стратифицированность информации, хранимой в памяти кибернетической системы]

⇒ *свойство-предпосылка**:

структуризация информации, хранимой в памяти кибернетической системы

⇒ *свойство-предпосылка**:

качество метаязыковых средств представления информации, хранимой в памяти кибернетической системы

:= [уровень развития метаязыковых средств кодирования (внутреннего представления) информации, хранимой в памяти кибернетической системы]

способность кибернетической системы к повышению уровня структуризации информации, хранимой в памяти указанной системы

⇔ *следует отличать**:

структурированность информации, хранимой в памяти кибернетической системы

:= [уровень структуризации информации, хранимой в памяти кибернетической системы]

⇒ *примечание**:

[Качественная структуризация информации, хранимой в памяти кибернетической системы, то есть качественное "разложение" этой информации по семантическим "полочкам" существенно упрощает и, следовательно, ускоряет повышение качества самой этой информации.]

⇔ *следует отличать**:

структуризация информации, хранимой в памяти кибернетической системы

⊂ *действие, выполняемое кибернетической системой в своей памяти*

⊂ *процесс*

способность кибернетической системы к анализу качества информации, хранимой в её памяти

:= [способность кибернетической системы к анализу информации, хранимой в собственной памяти, для последующего повышения качества этой информации]

⇐ *частное свойство**:

способность кибернетической системы к рефлексии

⇒ *примечание**:

[Рефлексия кибернетической системы, то есть анализ собственного качества, включает в себя не только анализ качества информации, хранимой в её памяти, но и анализ собственной деятельности как во внешней среде, так и в собственной памяти. При этом анализ собственной деятельности сводится к анализу описания этой деятельности, представленного в собственной памяти.]

⇒ *свойство-предпосылка**:

- *качество метаязыковых средств описания в памяти кибернетической системы качества информации, хранимой в её памяти*

- *способность кибернетической системы к обнаружению противоречий в информации, хранимой в её памяти*

⇒ *частное свойство**:

- *способность кибернетической системы к обнаружению пар синонимичных знаков, входящих в состав информации, хранимой в её памяти*

- способность кибернетической системы к обнаружению семантически эквивалентных фрагментов, входящих в состав информации, хранимой в её памяти
- способность кибернетической системой к обнаружению омонимичных знаков в информации, хранимой в её памяти
- способность кибернетической системы к обнаружению информационных дыр в информации, хранимой в её памяти
- способность кибернетической системой к обнаружению информационного мусора в информации, хранимой в её памяти

способность кибернетической системы к устранению противоречий, обнаруженных в информации, хранимой в её памяти

⇒ частное свойство*:

- способность кибернетической системы к устранению синонимии знаков, входящих в состав информации, хранимой в памяти указанной системы
- способность кибернетической системы к устранению семантической эквивалентности фрагментов, входящих в состав информации, хранимой в памяти указанной системы
- способность кибернетической системы к устранению омонимичных знаков, входящих в состав информации, хранимой в памяти указанной системы
- способность кибернетической системы к устранению противоречий, обнаруженных в информации, хранимой в памяти указанной системы, и не являющихся обнаруженной синонимией, семантической эквивалентностью или омонимией

способность кибернетической системы к устранению семантической эквивалентности фрагментов, входящих в состав информации, хранимой в памяти указанной системы

:= [способность кибернетической системы к устранению дублирования информации в рамках памяти указанной системы]

следует отличать*

- Э {
- **семантическая эквивалентность***
:= [эквивалентность информационных конструкций по смыслу (содержанию)*]
 - **синтаксическая эквивалентность***
:= [эквивалентность информационных конструкций по форме*]
 - **логическая эквивалентность***
:= [пары информационных конструкций, первая из которых логически следует из второй и наоборот*]
⇒ **примечание***:
[Если с семантической эквивалентности в памяти кибернетической системы можно и нужно бороться, то без логической эквивалентности обойтись трудно (как минимум из-за необходимости вводить определяемые понятия и, соответственно, формулировать определения). Тем не менее, логической эквивалентностью и, в частности, расширением числа определяемых понятий увлекаться не следует. Так, например, если определение нового понятия не является громоздким (в частности, понятия, являющегося теоретико-множественным объединением или пересечением ранее введенных понятий), то явно вводить это новое понятие не следует.]
- }

способность кибернетической системы к устранению информационных дыр, обнаруженных в информации, хранимой в её памяти

⇒ **свойство-предпосылка***:

способность кибернетической системы генерировать ответы на вопросы различного вида в случае, если они целиком или частично отсутствуют в текущем состоянии информации, хранимой в памяти

⇒ **примечание***:

[Формальным результатом обнаружения информационной дыры является формулировка запроса на недостающую информацию, которую необходимо сгенерировать.]

способность кибернетической системы к удалению информационного мусора, обнаруженного в информации, хранимой в её памяти

:= [способность кибернетической системы к забыванию (стиранию, удалению) ненужной (лишней, "отработанной") информации, которая, например, играет роль информационных "лесов" при решении различных задач]

- ⇒ *примечание**:
 [Критериями информационного мусора может быть:
- завершение решения задачи, для которой данная информация является вспомогательной и востребованной только в рамках решения соответствующей задачи;
 - истечение срока давности хранения;
 - легкая воспроизводимость (при необходимости).
-]

способность кибернетической системы к семантическому погружению новых знаний в состав информации, хранимой в ее памяти

- ⇒ *примечание**:
 [Новая введенная в память информационная конструкция трактуется как конструкция, у которой входящие в нее знаки являются потенциальными синонимами тем знакам, которые уже присутствуют в хранимой информации. Поэтому для всех этих знаков надо проверить наличие их синонимов. После этого синонимичные знаки должны быть отождествлены. Отождествление знаков осуществляется либо путем приписывания им одинаковых идентификаторов (имен), либо путем "физического" склеивания этих знаков.]
- ⇒ *примечание**:
 [Новой информацией, погружаемой (вводимой) в состав информации, хранимой в памяти кибернетической системы, может быть:
- либо принятое сообщение, поступившее от другой кибернетической системы и переведенное на внутренний язык данной системы;
 - либо информация, сгенерированная в результате решения какой-либо задачи.
-]

способность кибернетической системы к обнаружению сходств в знаниях, хранимых в ее памяти

- ⇒ *примечание**:
 [Сходства в знаниях могут иметь самый разнообразный вид и далеко не всегда являются очевидными.]
- ⇒ *примечание**:
 [Умение "видеть" сходство в различном и различие в сходном является важнейшим признаком интеллекта.]

способность кибернетической системы к конвергенции знаний, хранимых в ее памяти

- := [способность кибернетической системы к увеличению сходств в знаниях хранимых в ее памяти]
- ⇒ *свойство-предпосылка**:
способность к увеличению числа общих понятий для различных фрагментов информации, хранимой в памяти кибернетической системы, без ущерба качеству этих фрагментов
- := [способность к "сближению" знаний путем:
- увеличения числа общих понятий, используемых в "сближаемых" знаниях;
 - преобразования исходных знаний к их логически эквивалентным вариантам в целях получения фрагментов как можно большего размера и как можно в большем количестве, которые были бы:
 - либо синтаксически изоморфными и содержащими как можно большее число общих понятий;
 - либо синтаксически изоморфными и одновременно семантически эквивалентными.
-]

способность кибернетической системы к интеграции знаний, хранимых в ее памяти

- := [способность объединять имеющиеся знания и формировать целостную картину различных исследуемых объектов, систем, процессов, явлений]
- ⇒ *свойство-предпосылка**:
способность кибернетической системы к конвергенции знаний, хранимых в ее памяти
- ⇒ *примечание**:
 [Качество (глубина) интеграции знаний определяется тем, насколько качественно до этого была проведена конвергенция интегрируемых знаний.]
- ⇒ *примечание**:
 [Качественная ("бесшовная", глубокая) интеграция различных знаний, хранимых в памяти кибернетической системы, дает возможность существенно снизить количество хранимых в памяти методов решения задач, так как позволяет некоторые ранее различные классы задач объединить в один класс задач. При этом очевидно, что такая интеграция знаний, хранимых в памяти кибернетической системы, требует разработки общих (базовых) синтаксических и семантических принципов представления знаний различного вида.]

конвергенция и интеграция знаний⇒ *примечание**:

[Мы вынуждены смотреть на окружающую нас внешнюю среду (внешний мир) через "замочную скважину" своих сенсоров (рецепторов), своих персональных точек зрения, мировоззрения различных научных дисциплин. Но необходимо помнить, что целостную картину внешней среды (картину мира) можно построить только путем сближения (конвергенции) и соединения (интеграции) самых различных точек зрения, самых различных научных дисциплин и направлений. Мир не делится на различные дисциплины – он един. Для эффективного решения задач конвергенции и интеграции знаний необходимо построить искусственную ("рукотворную") среду (память), в которой было бы удобно не только хранить самые различные знания, но и осуществлять конвергенцию и интеграцию этих знаний. При этом очень важно, чтобы формируемая таким образом информационная модель окружающей нас внешней среды (информационной картины мира) была общедоступна как для просмотра (ознакомления), причем, без каких бы то ни было "замочных скважин", так и для ввода новых знаний, представляющих (отражающих) точку зрения их авторов.]

⇒ *эпиграф**:

Древнеиндийская притча о слоне и слепцах

следует отличать*

- ∋ { • *конвергенция*[^]
:= [Свойство, определяющее степень близости (уровень конвергенции) между двумя заданными сущностями и, в частности, знаниями]
- *конвергенция**
:= [Множество пар близких (аналогичных, сходных) сущностей*]
 - *конвергенция*
:= [Множество процессов сближения различных пар сущностей]
- }
- ∋ { • *интеграция**
:= [Квазибинарное отношение, каждая пара которого связывает множество интегрируемых сущностей с результатом интеграции*]
- *интеграция*
:= [Множество процессов интеграции множества заданных сущностей]
- }

способность кибернетической системы к обобщениям и формированию новых понятий⇒ *примечание**:

[Важным примером обобщения является переход от задач к классам часто решаемых задач.]

способность кибернетической системы к генерации гипотез и обнаружению закономерностей в информации, хранимой в ее памяти⇒ *примечание**:

[Данная способность кибернетической системы является важнейшим фактором эволюции информации, хранимой в памяти кибернетической системы, в направлении перехода от данных (от фактографической информации) к знаниям.]

способность кибернетической системы к обоснованию или опровержению знаний, хранимых в ее памяти⇒ *примечание**:

[Примерами знаний, подлежащих обоснованию или опровержению, являются:

- любое введенное в кибернетическую систему сообщение (любая новая информация, поступающая от любого субъекта);
- формулировка какой-либо задачи, предлагаемой для решения;
- формулировка какого-либо гипотетического утверждения (теоремы), подлежащего доказательству.

]

:= [способность к объяснению (обоснованию, аргументации) корректности, важности и целесообразности использовать (обратить внимание на) указываемое знание]

:= [способность либо находить в текущем состоянии базы знаний, либо генерировать (строить) ответы на *почему-вопросы*]

способность кибернетической системы к экспериментальному подтверждению или опровержению гипотез о свойствах динамических систем с помощью имитационных моделей этих систем

⇒ *примечание**:

[Создание динамических информационных моделей сложных динамических систем и проведение различного рода "мысленных" экспериментов с такими моделями является весьма перспективным и мощным методом исследования сложных динамических систем.]

способность кибернетической системы к коррекции теорий, хранимых в ее памяти

:= [способность к адаптации накопленных знаний к различным изменениям условий и жизненных ситуаций]

⇒ *примечание**:

[В основе данного свойства кибернетической системы лежит:

- постоянная готовность кибернетической системы подвергнуть сомнению любое знание, хранимое в ее памяти;
- постоянное уточнение степени достоверности каждого знания, хранимого в памяти кибернетической системы.

]

/*****

способность кибернетической системы к повышению качества своего решателя задач

:= [способность кибернетической системы повышать качество своих приобретаемых навыков]

⇒ *свойство-предпосылка**:

- *способность кибернетической системы к повышению качества информации, хранимой в ее памяти*
- *семантическая гибкость возможных самоизменений решателя задач кибернетической системы*
- *стратифицированность решателя задач кибернетической системы*
- *способность кибернетической системы к анализу качества своего решателя задач*
- *способность кибернетической системы к целенаправленной коррекции своей деятельности*
- *способность кибернетической системы к оптимизации хранимых в памяти методов решения задач*
- *способность кибернетической системы к генерации новых методов решения задач*
- *способность кибернетической системы интегрировать у себя новые приобретаемые извне методы и модели решения задач*

семантическая гибкость возможных самоизменений решателя задач кибернетической системы

:= [простота реализации решателем задач кибернетической системы различного рода изменений самого себя]

⇒ *примечание**:

[Очевидно, что семантическая гибкость решателя задач кибернетической системы во многом определяется процессором кибернетической системы (прежде всего, его универсальностью и близостью реализуемой им модели обработки информации к смысловому уровню). Но, поскольку решатель задач кибернетической системы кроме процессора включает в себя хранимые в памяти кибернетической системы методы решения различного вида задач (в том числе, и методы интерпретации методов высокого уровня), семантическая гибкость решателя задач определяется также *семантической гибкостью информации, хранимой в памяти кибернетической системы.*]

⇒ *свойство-предпосылка**:

семантическая гибкость информации, хранимой в памяти кибернетической системы

стратифицированность решателя задач кибернетической системы

⇒ *частное свойство**:

- *стратифицированность методов и навыков решения задач, представленных в памяти кибернетической системы*
 - *стратифицированность технологий, соответствующих различным видам деятельности*
 - *стратифицированность различного вида действий, классов действий и видов деятельности*
- ⇒ *частное свойство**:
- стратифицированность различного вида информационных процессов, выполняемых в памяти кибернетической системы*

качество внутренних языковых средств кибернетической системы для описания качества собственного решателя задач

⇒ *свойство-предпосылка**:

качество внутренних языковых средств кибернетической системы для описания качества собственных действий

способность кибернетической системы к анализу качества своего решателя задач

:= [способность кибернетической системы к анализу (к оценке качества) своей деятельности в собственной внутренней среде (в своей памяти), а также в своей внешней среде]

⇒ *примечание**:

[Анализ качества решателя задач включает в себя:

- анализ качества используемых методов и технологий решения задач;
- анализ качества используемых моделей решения задач;
- анализ полноты набора постоянно иницированных целей (задач), направленных на эволюцию и на борьбу с деградацией (снижением качества) кибернетической системы;
- анализ качества выполняемых действий (процессов решения задач).

]

:= [способность кибернетической системы к описанию (к построению в своей памяти информационной модели) собственных действий, выполняемых в собственной памяти, а также к анализу и оценке этих действий]

:= [способность кибернетической системы к анализу своего поведения в своей внутренней среде (в своей памяти), а также в своей внешней среде и в своей физической оболочке]

⇒ *свойство-предпосылка**:

- *качество внутренних языковых средств кибернетической системы для описания качества собственного решателя задач*

- *способность кибернетической системы к анализу собственной деятельности*

⇒ *частное свойство**:

- *способность кибернетической системы к анализу качества информационных процессов, выполняемых в собственной памяти*

:= [способность кибернетической системы к анализу качества своего поведения (действий, информационных процессов) в собственной внутренней среде – своих действий, сводящихся к поиску, генерации, удалению и преобразованию информационных конструкций, хранимых в собственной памяти]

- *способность кибернетической системы к анализу качества своего поведения во внешней среде*

- *способность кибернетической системы к анализу качества методов, хранимых в собственной памяти*

⇒ *частное свойство**:

- *способность кибернетической системы к анализу качества методов и технологий, используемых ею для выполнения сложных действий в собственной памяти*

- *способность кибернетической системы к анализу качества методов и технологий, используемых ею для выполнения сложных действий во внешней среде*

качество внутренних языковых средств кибернетической системы для описания качества собственных действий

⇒ *примечание**:

[В данном свойстве кибернетической системы имеется в виду описание собственных действий, выполняемых кибернетической системой как в своей внутренней среде (в собственной памяти), так и в своей внешней среде.]

способность кибернетической системы к анализу качества своего поведения во внешней среде

:= [способность кибернетической системы к анализу соответствия между тем, что планировалось сделать во внешней среде и тем, что реально получилось]

⇒ *примечание**:

[Поведение кибернетической системы во внешней среде рассматривается ею как "эксперимент", подтверждающий или опровергающий ее представление о внешней среде.]

:= [способность кибернетической системы к анализу своего опыта взаимодействия с внешней средой и, в частности, к выявлению своих ошибок]

способность кибернетической системы к целенаправленной коррекции своей деятельности

:= [способность кибернетической системы к коррекции своего поведения в целях повышения его качества (эффективности)]

:= [способность кибернетической системы учиться на ошибках своей деятельности на основе анализа этих ошибок]

⇒ *свойство-предпосылка**:
способность кибернетической системы к анализу собственной деятельности

способность кибернетической системы к оптимизации хранимых в памяти методов решения задач

⇒ *примечание**:

[Хранимые в памяти *методы* решения задач разбиваются на следующие классы:

- методы верхнего уровня* – интерпретируемые методы;
- методы базового уровня*, представленные на базовом языке программирования, который интерпретируется непосредственно процессором кибернетической системы;
- метаметоды*, описывающие интерпретацию методов верхнего уровня.

]

способность кибернетической системы к генерации новых методов решения задач

⇒ *примечание**:

[Целесообразность генерации нового метода решения задач возникает, когда кибернетической системе приходится часто решать эквивалентные задачи некоторого класса. Генерация соответствующего метода и последующая его оптимизация позволяет существенно сократить время решения задач.]

:= [способность кибернетической системы расширять множество используемых ею методов решения задач]

⇒ *примечание**:

[Если добавляемые методы соответствуют используемым моделям решения задач, то, кроме добавления самих методов, желательно, чтобы в стратифицированной кибернетической системе никакие другие изменения не потребовались. Если добавляемый метод соответствует новой (ранее не известной) модели решения задач, то желательно, чтобы в стратифицированной кибернетической системе никакие другие изменения не потребовались, кроме добавления агентов, обеспечивающих интерпретацию (описание операционной семантики) методов нового класса.]

⇒ *примечание**:

[Речь идет о методах решения как внутренних задач, решаемых в памяти кибернетической системы, так и внешних задач, решаемых во внешней среде путем управления деятельностью эффекторов и рецепторов кибернетической системы.]

⇒ *свойство-предпосылка**:

способность расширять множество использованных моделей решения задач

способность кибернетической системы интегрировать у себя новые приобретаемые извне методы и модели решения задач

⇒ *примечание**:

[Для обеспечения такой способности необходима:

- разработка универсальной базовой модели решения задач, для которой соответствующие ей методы решения задач интерпретируются процессором кибернетической системы;
- разработка семейства классов методов верхнего уровня, что предполагает:
 - разработку языков представления методов для каждого класса методов верхнего уровня;
 - разработку интерпретаторов для каждого класса методов верхнего уровня на основе указанной выше базовой модели решения задач.

]

/*****/

способность кибернетической системы к повышению качества своей физической оболочки

:= [способность кибернетической системы к самостоятельному совершенствованию (эволюции) своей физической оболочки]

⇒ *примечание**:

[Данная способность кибернетической системы накладывает определенные требования к построению ее физической оболочки.]

- ⇒ *свойство-предпосылка**:
- *гибкость возможных изменений физической оболочки кибернетической системы*
 - *стратифицированность физической оболочки кибернетической системы*
 - *способность кибернетической системы к анализу качества своей физической оболочки*
- ⇒ *свойство-предпосылка**:
качество внутренних языковых средств кибернетической системы для описания качества собственной физической оболочки
- *способность кибернетической системы расширять и/или совершенствовать набор собственных сенсоров и эффекторов*

/*****

комплекс свойств кибернетических систем, определяющих их обучаемость по различным формам обучения

- = {
- *обучаемость с учителем*
 - *самообучаемость с экспертом*
 - *самообучаемость на основе внешних информационных источников*
 - *самообучаемость без внешних информационных источников*
- }

обучаемость с учителем

- := [уровень способности к обучению под управлением внешнего субъекта-учителя]
- := [способность кибернетической системы к эффективному обучению с помощью учителя, осуществляющего управление процессом обучения]
- := [способность заданной кибернетической системы эффективно обучаться с помощью внешней кибернетической системы (внешнего субъекта, внешнего активного учителя), осуществляющей организацию обучения заданной кибернетической системы на основе различных методик обучения, учитывающих особенности обучаемой системы и определяющих характер (в том числе последовательность) передачи знаний и навыков, а также тестирование качества их усвоения]

самообучаемость с экспертом

- := [способность кибернетической системы к самообучению в диалоге с экспертом-консультантом]
- := [способность кибернетической системы не просто задавать нужные для собственного обучения вопросы (информационные цели), но и вести вопросно-ответный диалог с другими субъектами (кибернетическими системами), которые являются экспертами в соответствующей области (указанные эксперты – это своего рода "пассивные учителя", которые много знают и умеют в соответствующей области, могут отвечать на вопросы, но не желают управлять процессом передачи этих знаний и умений другим кибернетическим системам)]
- := [эффективность самообразования кибернетической системы, в основе которого лежит диалог, управляемый этой обучаемой системой и осуществляемый с кибернетической системой, являющейся носителем востребованных знаний и навыков]
- := [эффективность самообучения, осуществляемого в форме консультации]
- := [способность управлять процессом самообучения путем формирования последовательности вопросов (познавательных целей), адресуемых внешним субъектам]
- ⇒ *свойство-предпосылка**:
способность кибернетической системы к синтезу познавательных целей и процедур

обучаемость на основе пассивных внешних информационных источников

- := [способность кибернетической системы к извлечению информации, содержащейся во внешних информационных источниках, к поиску нужных внешних источников и к построению на этой основе систематизированной картины мира]
- := [эффективность самообучения, основанного на анализе пассивных источников информации (документов различного вида, публикаций, текстов, которые необходимо находить в различного рода библиотеках, читать и понимать)]

самообучаемость без внешних информационных источников

- := [способность кибернетической системы формировать систематизированную модель (картину) окружающей среды, используя для ее непосредственного восприятия и изучения только собственные сенсоры и эффекторы, а также некоторые дополнительные средства, усиливающие возможности сенсоров и эффекторов]

```

:= [Эффективность самообучения кибернетической системы, основанного исключительно на собственном опыте,
на анализе собственной деятельности и собственных ошибок]
⇒ примечание*:
[Данная способность кибернетической системы является необходимым, но явно недостаточным фактором
ее высокого качества. Учиться только на собственном опыте – существенно понизить уровень своего
интеллекта. В этом смысле познавательный процесс социален.]
} /* Завершили Сегмент “Комплекс свойств, определяющих уровень обучаемости кибернетической систе-
мы” */

/* Сегмент *****/
Комплекс свойств, определяющих качество многоагентной системы
⊃=
{
многоагентная система
⇒ пояснение*:
[Переход от кибернетических систем к коллективам взаимодействующих между собой кибернетических
систем, т.е. к социальной организации кибернетических систем, является важнейшим фактором эволюции
кибернетических систем.]
⊃ кибернетическая система
⇒ разбиение*:
{ • моногенная многоагентная система
:= [однородная многоагентная система, состоящая из однотипных агентов]
• гетерогенная многоагентная система
:= [неоднородная многоагентная система, состоящая из агентов разного типа]
}
⇒ разбиение*:
{ • простая многоагентная система
:= [многоагентная система, агенты которой не являются многоагентными системами]
• иерархическая многоагентная система
:= [многоагентная система, некоторые или все агенты которой являются многоагентными
системами]
}
⇒ примечание*:
[Агенты многоагентной системы могут (но вовсе не обязательно должны) быть интеллектуальными
системами. Так, например, агенты интеллектуального решателя задач, имеющего агентно-ориентированную
архитектуру, не являются интеллектуальными системами.]

агент*
:= [быть агентом данной многоагентной системы*]
:= [быть кибернетической системой, входящей в состав данной многоагентной системы*]
⇒ примечание*:
[Агентом иерархической многоагентной системы может быть другая многоагентная система]
⊃ член многоагентной системы*
:= [агент многоагентной системы, не являющийся агентом другого агента этой системы*]
:= [непосредственный (ближайший) агент многоагентной системы*]

кибернетическая система
⇒ разбиение*:
{ • индивидуальная кибернетическая система
:= пояснение*:
[минимальная целостная кибернетическая система обладающая достаточно высоким уровнем
самостоятельности и способности “выживать” в своей внешней среде]
• кибернетическая система, являющаяся минимальным компонентом индивидуальной
кибернетической системы

```


- ⇒ *пояснение**:
[Это такой компонент, в состав которого не входят *кибернетические системы*]
- *кибернетическая система, являющаяся комплексом компонентов соответствующей индивидуальной кибернетической системы*
- *сообщество индивидуальных кибернетических систем*
- ⇒ *разбиение**:
 - *простое сообщество индивидуальных кибернетических систем*
 - *иерархическое сообщество индивидуальных кибернетических систем*

многоагентная система

- := [коллектив взаимодействующих кибернетических систем]
- ⇒ *разбиение**:
 - *сообщество индивидуальных кибернетических систем*
 - *индивидуальная кибернетическая система, реализованная в виде многоагентной системы*
 С *кибернетическая система, являющаяся комплексом компонентов соответствующей индивидуальной кибернетической системы*
 - ⇒ *пояснение**:
[Такая "внутренняя" *многоагентная система* в индивидуальной кибернетической системе появляется, когда на определенном этапе её эволюции *решатель задач индивидуальной кибернетической системы* "переходит" на *агентно-ориентированную модель обработки информации в памяти индивидуальной компьютерной системы*]
- := [кибернетическая система, представляющая собой коллектив взаимодействующих кибернетических систем, обладающих определенной степенью самостоятельности (самодостаточности, свободы выбора)]

многоагентная система с централизованным управлением

- := [многоагентная система, в которой специально выделяются агенты, которые принимают решения в определенной области деятельности многоагентной системы и обеспечивают выполнение этих решений путем управления деятельностью остальных агентов, входящих в состав этой системы]
- С *многоагентная система*

сообщество интеллектуальных систем с децентрализованным управлением

- := [многоагентная система с децентрализованным управлением, агентами которой являются интеллектуальные системы]
- := [многоагентная система, в которой решения принимаются коллегиально и "автоматически" (решения о признании новой кем-то предложенной информации – в том числе, об инициировании некоторой задачи, решения о коррекции (уточнении) уже ранее признанной (одобренной, согласованной) информации) на основе четко продуманной и постоянно совершенствуемой методики, а также на основе активного участия всех агентов в формировании новых предложений, подлежащих признанию (одобрению, согласованию)]
- С *многоагентная система*
- ⇒ *примечание**:
[В такой многоагентной системе все агенты участвуют в управлении этой системы]
- Э *Экосистема OSTIS*
- ⇒ *пояснение**:
[В такой многоагентной системе отсутствуют специально "назначенные" агенты, которые "обязаны" принимать решения о том, какую коллективно решаемую задачу надо инициировать, и о том, как распределить между агентами подзадачи указанной инициированной задачи.]
- С *многоагентная система с децентрализованным управлением*
- С *сообщество интеллектуальных систем*
- ⇒ *примечание**:
[Примером такой системы является оркестр, способный играть без дирижера. При этом подчеркнем, что каждый музыкант такого оркестра:
 - должен иметь квалификацию не только музыканта, но и дирижера и даже композитора
 - должен быть договороспособным – уметь согласовывать свои действия с действиями коллег
 Аналогичным примером децентрализованной многоагентной системы является строительная бригада, способная построить дом без бригадира, прораба, архитектора.]

синергетическая кибернетическая система

:= [эволюционная многоагентная система]

:= [многоагентная система, состоящая из когнитивных агентов]

:= [многоагентная система, обладающая высоким уровнем коллективного интеллекта, атомарными агентами которой являются индивидуальные интеллектуальные системы, имеющие высокий уровень социализации]

⇒ *пояснение**:

Ярушкіна.Н.Г.НечетГС-2007кн.-стр.88-101

← *цитата**:

Ярушкіна.Н.Г.НечетГС-2007кн

⇒ *пояснение**:

- *Тарасов.В.Б.ЭволюСНСвИИ-1997кн*
- *Тарасов.В.Б.ЭволюСнМСОТпкППО-1998ст*

⇒ *примечание**:

[Очевидным примером синергетической кибернетической системы является творческий коллектив, реализующий сложный наукоемкий проект. Огромная сложность создания таких коллективов является главной причиной медленного развития целого ряда весьма актуальных научно-технических проектов, таких как создание принципиально нового технологического уровня автоматизации человеческой деятельности на основе интеллектуальных семантически совместимых компьютерных систем, способных самостоятельно взаимодействовать друг с другом.]

многоагентная система

⇒ *примечание**:

[Переход к *многоагентным системам* является важнейшим фактором повышения *качества* (и, в частности, уровня *интеллекта*) *кибернетических систем*, т.к. уровень интеллекта *многоагентной системы* может быть значительно выше уровня интеллекта каждого входящего в неё агента. Но это бывает далеко не всегда, поскольку важнейшим фактором качества многоагентных систем является не только качество входящих в неё агентов, но и организация взаимодействия агентов и, в частности, переход от централизованного к децентрализованному управлению. Количество далеко не всегда переходит в новое качество.

Повышение уровня интеллекта многоагентной системы обеспечивается

- не только повышением уровня интеллекта и, в первую очередь, уровня *социализации* ее агентов;
- не только переходом от централизованного к децентрализованному управлению деятельностью агентов;
- но и качеством общей базы знаний всей многоагентной системы.

]

социализация кибернетической системы

⇒ *примечание**:

[Когда мы говорим о *социализации кибернетических систем*, речь идет только об *индивидуальных кибернетических системах*, т.е. о *кибернетических системах*, достигших некоторого уровня целостности и автономности и способных входить в состав различных коллективов. Соответственно этому, качество *индивидуальных кибернетических систем* определяется, кроме всего прочего тем, насколько большой вклад *индивидуальная кибернетическая система* вносит в повышение качества тех коллективов, в состав которых она входит. Указанное свойство *индивидуальных кибернетических систем* будем называть уровнем их *социализации*. Прежде, чем детализировать это свойство, целесообразно рассмотреть то, чем определяется качество коллектива кибернетических систем, например, качество творческого сообщества компьютерных систем и людей.]

качество сообщества компьютерных систем и людей

⇒ *пояснение**:

[Эффективность творческого коллектива (например в области научно-технической деятельности) определяется:

- согласованностью мотивации (целевой установки) всего коллектива и каждого его члена:
 - не должно быть синдрома "лебедя, рака и щуки";
 - не должно быть противоречий между целью коллектива и творческой самореализацией каждого его члена;
- эффективной организацией децентрализованного управления деятельностью членов сообщества;
- четкой, оперативной и доступной всем фиксацией документации текущего состояния содеянного и направлений его дальнейшего развития;

- уровнем трудоемкости оперативности фиксации индивидуальных результатов в рамках коллективно создаваемого общего результата;
- уровнем структурированности и, прежде всего, стратифицированности обобщенной документации (базы знаний);
- эффективностью ассоциативного доступа к фрагментам документации;
- гибкостью коллективно создаваемой базы;
- автоматизацией анализа содеянного и управления проектом.

]

качество многоагентной системы

⇒ *свойство-предпосылка**:

- *средний уровень интеллекта членов многоагентной системы*
- *средний уровень социализации членов многоагентной системы*
- *минимальный уровень социализации членов многоагентной системы*

⇒ *примечание**:

[Члены многоагентной системы, имеющие низкий уровень социализации, существенно снижают качество системы.]

- *качество организации взаимодействия членов многоагентной системы*

⇒ *примечание**:

[Высший уровень качества организации взаимодействия агентов многоагентной системы обеспечивается:

- введением дополнительного специального (корпоративного) агента, выполняющего функцию хранителя интегратора общих (корпоративных) знаний многоагентной системы;
- реализацией децентрализованного взаимодействия агентов, управляемого текущим состоянием информации, хранимой в памяти корпоративного агента.

]

{ ostis-система

С *многоагентная система, управляемая общей базой знаний*

}

⇒ *примечание**:

[Агенты *ostis-системы* (sc-системы) являются специализированными кибернетическими системами, действия каждой из которых (кроме *сенсорных sc-агентов*) инициализируются определенного вида ситуациями и/или событиями в памяти *ostis-системы* и закljučаются (за исключением *эффекторных sc-агентов*) в преобразовании текущего состояния информации, хранимой в этой памяти. Таким образом, sc-агенты не являются интеллектуальными системами.]

} /* Завершили Сегмент “Комплекс свойств, определяющих качество многоагентной системы” */

/* Сегмент *****

Комплекс свойств, определяющих уровень социализации кибернетической системы как фактора существенного повышения уровня ее обучаемости, а также фактора существенного повышения качества всех тех многоагентных систем, в состав которых входит данная кибернетическая система

⊃=

{

:= [Комплекс свойств *кибернетической системы*, определяющих необходимые требования к тем *кибернетическим системам*, которые могут входить в состав *синергетических кибернетических систем*]

социализация кибернетической системы

:= [способность кибернетической системы взаимодействовать с другими кибернетическими системами в целях создания коллектива кибернетических систем (*многоагентных систем*), уровень качества и, в частности, уровень *интеллекта* которого выше уровня качества каждой *кибернетической системы*, входящей в состав этого коллектива)]

- := [комплекс способностей кибернетической системы, которые определяют ее вклад в уровень коллективной (социальной) интеллектуальности, т.е. в уровень интеллектуальности того коллектива кибернетических систем, членом которого данная кибернетическая система является (в уровень интеллектуальности соответствующей многоагентной системы)]
- := [уровень вклада *кибернетической системы* в обеспечение *интеллекта* тех многоагентных систем, в состав которых эта *кибернетическая система* входит]
- := [уровень социализации кибернетической системы]
- := [социализация]
- ⇒ *примечание**:
[Уровень *интеллекта* коллектива кибернетических систем (*многоагентной системы*) может быть значительно ниже уровня *интеллекта* самого "глупого" члена этого коллектива, но может быть и значительно выше уровня *интеллекта* самого "умного" члена указанного коллектива. Для того, чтобы количество *интеллектуальных систем* переходило в существенно более интеллектуальное качество коллектива таких систем, все объединяемые в коллектив *интеллектуальные системы* должны иметь высокий уровень *социализации*, что накладывает дополнительные требования, предъявляемые к *информации, хранимой в памяти*, а также к *решателям задач интеллектуальных систем*, объединяемых в коллектив.]
- ⇒ *примечание**:
[Коллектив *кибернетических систем* может иметь значительно более высокий уровень качества, в том числе, уровень интеллекта, чем уровень качества *кибернетических систем*, являющихся членами этого коллектива. Но так бывает не всегда. Для того, чтобы количество членов коллектива *кибернетической системы* перешло в более высокое качество самого коллектива, члены коллектива должны обладать дополнительными способностями, которые будем называть свойствами *социализации*. Основными такими свойствами являются способность устанавливать и поддерживать достаточный уровень *семантической совместимости* (взаимопонимания) с другими кибернетическими системами и *договороспособность* (способность согласовывать свои действия с другими).]
- ⇒ *примечание**:
[Целенаправленный обмен информацией между *кибернетическими системами* существенно ускоряет процесс их обучения (процесс накопления знаний и навыков). Следовательно, способность эффективно использовать указанный канал накопления знаний и навыков существенно повышает уровень *обучаемости кибернетических систем*. В этом смысле можно сказать, что познавательный процесс социален.]
- := [уровень развития социально значимых качеств кибернетической системы]
- ⇒ *примечание**:
[Повышение уровня *социализации кибернетической системы* является, с одной стороны, дополнительным повышением уровня *интеллекта* самой этой *кибернетической системы*, а также фактором повышения уровня *интеллекта* тех коллективов, тех *многоагентных систем*, в состав которых эта *кибернетическая система* входит.]
- ⇒ *примечание**:
[Переход к *многоагентным системам* не только является важным фактором повышения качества *кибернетических систем*, но также имеет и обратную "сторону медали" – появление целого ряда угроз, связанного с возможными целенаправленными вредоносными воздействиями на *многоагентную систему* (со стороны некоторых ее *агентов*), существенно снижающими уровень ее качества. Наличие таких *вредоносных целей* у соответствующих *агентов* свидетельствует о низжайшем уровне *социализации* этих *агентов*.]
- := [умение согласовывать (синхронизировать) свою деятельность с деятельностью других кибернетических систем в процессе решения задач, требующих коллективных усилий]
- := [умение участвовать в децентрализованном процессе распределения подзадач некоторой коллективно (распределенно) решаемой задачи между членами заданного коллектива кибернетических систем и умение участвовать в управлении коллективного решения указанной задачи]
- ⇒ *примечание**:
[Речь идет о децентрализованном асинхронном управлении деятельностью коллектива кибернетических систем]
- := [способность и готовность кибернетической системы к координации своей деятельности в рамках коллектива кибернетических систем, в состав которого она входит в целях:
- эффективного решения тактических задач, решаемых указанным коллективом;
 - решения главной стратегической задачи этого коллектива – обеспечения как можно более высокой скорости роста уровня интеллекта указанного коллектива.
-]]
- ⇒ *примечание**:
[Подчеркнем, что повышение уровня интеллекта коллектива кибернетической системы (*многоагентной системы*) имеет свои особенности:

- во-первых, это забота о семантической совместимости кибернетических систем входящих в состав коллектива;
- во-вторых, это переход от виртуальной распределенной базы знаний коллектива к реально поддерживаемым базам знаний и к порталам корпоративных знаний, реализованных в виде индивидуальных кибернетических систем, через которые осуществляются все процессы координации и согласования деятельности соответствующих членов коллектива

]

⇒ *свойство-предпосылка**:

- *договороспособность кибернетической системы*
:= *часто используемый sc-идентификатор**:
[договороспособность]
- *социальная ответственность кибернетической системы*
:= *часто используемый sc-идентификатор**:
[социальная ответственность]
- *социальная активность кибернетической системы*
:= *часто используемый sc-идентификатор**:
[социальная активность]

/*****/

договороспособность кибернетической системы⇒ *свойство-предпосылка**:

- *способность кибернетической системы к пониманию принимаемых сообщений*
- *особенности кибернетической системы к формированию передаваемых сообщений, понятных адресатам*
- *семантическая совместимость кибернетической системы с партнёрами*
- *способность кибернетической системы к обеспечению семантической совместимости с партнёрами*
- *коммуникабельность кибернетической системы*
- *способность кибернетической системы к обсуждению и согласованию целей и планов коллективной деятельности*
- *способность кибернетической системы брать на себя выполнение актуальных задач в рамках согласованных планов коллективной деятельности*

способность кибернетической системы к пониманию принимаемых сообщений

:= [способность кибернетической системы к пониманию информации, поступающей извне от других кибернетических систем]

:= [способность кибернетической системы к отображению принимаемых сообщений в семантически эквивалентные фрагменты собственной базы знаний]

⇒ *комплекс частных свойств**:

- { • *способность кибернетической системы к пониманию принимаемых вербальных сообщений*
- *способность кибернетической системы к пониманию принимаемых невербальных сообщений*
- }

⇒ *свойство-предпосылка**:*способность кибернетической системы к обеспечению семантической совместимости с партнёрами***сообщение**

:= [информация, передаваемая (пересылаемая) от одной кибернетической системы к другой или к другим кибернетическим системам]

⇒ *примечание**:

[Каждому сообщению ставится в соответствие одна кибернетическая система, являющаяся **источником сообщения*** и одна или несколько кибернетических систем, являющихся **адресатами сообщения***. В соответствии с этим для каждой кибернетической системы те сообщения, **источником*** которых она является, будем называть **передаваемыми сообщениями***, а те сообщения, **адресатами*** которых она является, будем называть **принимаемыми сообщениями***.]

⇒ *разбиение**:

- { • *вербальное сообщение*
:= [передаваемая словесная информация]
- *невербальное сообщение*

	⇒	<i>примечание*</i> : [Примерами невербальных сообщений являются пересылаемые фото-документы, видеоматериалы]
	}	
⇒		<i>обобщённая декомпозиция*</i> :
	{	• <i>спецификация сообщения</i>
	⇒	<i>обобщённая декомпозиция*</i> :
	{	• <i>указание источника специфицируемого сообщения</i>
		• <i>указание множества адресатов специфицируемого сообщения</i>
		• <i>отметка момента времени отправления специфицируемого сообщения</i>
		• <i>указание прагматического типа специфицируемого сообщения</i>
		• <i>указания запроса, ответом на который является специфицируемое сообщение</i>
	⇒	<i>примечание*</i> : [Если специфицируемое сообщение является ответом на некоторый запрос]
		• <i>указание раздела баз знаний адресатов, которому соответствует специфицируемое сообщение</i>
		• <i>указание способа представления тела сообщения</i>
	⇒	<i>примечание*</i> : [Для вербальных сообщений это указание используемого внешнего языка]
	}	
	•	<i>тело сообщения</i>
	:=	[собственно само сообщение]
	}	
⇒		<i>разбиение*</i> :
		<i>прагматический тип сообщения</i>
	=	{
		• <i>повествовательное сообщение</i>
		⊃ <i>ответ на запрос</i>
		• <i>вопросительное сообщение</i>
		• <i>команда редактирования баз знаний адресатов</i>
		• <i>команда, инициирующая действие адресатов в их внешней среде</i>
		}
		<i>следует отличать*</i>
∃	{	• <i>вербальная информация</i>
		• <i>файл, содержащий вербальную информацию</i>
	:=	[вербальная информация, представленная в виде файла]
		• <i>вербальное сообщение</i>
	}	
		<i>вербальная информация</i>
:=		[знаковая конструкция, которая имеет в общем случае произвольную денотационную семантику и которая может либо поступать на вход кибернетической системы через соответствующие ее сенсоры (рецепторы), либо через соответствующие эффекторы передаваться (пересылаться) в качестве сообщения другим кибернетическим системам]
		<i>следует отличать*</i>
∃	{	• <i>вербальная информация</i>
		• <i>сенсорная информация</i>
	}	
	⇒	<i>примечание*</i> : [И <i>вербальная информация</i> и <i>сенсорная информация</i> являются знаковыми конструкциями, но, во-первых, <i>вербальная информация</i> может быть как внешней знаковой конструкцией, так и внутренней знаковой конструкцией, хранимой в памяти кибернетической системы, а <i>сенсорная информация</i> всегда является внутренней знаковой конструкцией кибернетической системы и, во-вторых, <i>сенсорная информация</i> описывает только "пограничную" для кибернетической системы физическую окружающую среду, тогда, как <i>вербальная информация</i> может описывать все, что угодно.]
		<i>следует отличать*</i>
∃	{	• <i>невербальная информация</i>
		• <i>файл, содержащий невербальную информацию</i>

- := [файл, содержимым которого является электронный образ некоторой невербальной информации]
- *невербальное сообщение*
- := [невербальная информация, представленная в виде файла и передаваемая (пересылаемая) от одной кибернетической системы к другой]
- *сенсорная информация*
- := [информация, формируемая сенсорами кибернетической системы]

}

невербальная информация

- ⊃ *музыкальное произведение*
- ⊃ *танец*
- ⊃ *произведение изобразительного искусства*
 - ⊃ *живопись*
 - ⊃ *скульптура*
 - ⊃ *графика*
- ⊃ *статическое изображение*
- ⊃ *динамическое изображение*

способность кибернетической системы к пониманию принимаемых вербальных сообщений

:= [способность кибернетической системы к пониманию вербальной информации, поступающей извне из разных источников]

⇒ *примечание**:

[Понимание информации, поступающей извне, включает в себя:

- перевод этой информации на внутренний язык кибернетической системы;
- локальную верификацию вводимой информации;
- погружение (конвергенцию, размещение) текста, являющегося результатом указанного перевода в состав хранимой информации (в частности, в состав базы знаний)

]

⇒ *примечание**:

[Погружение вводимой информации в состав базы знаний кибернетической системы сводится к выявлению и устранению противоречий, возникающих между погружаемым текстом и текущего состояния базы знаний. Первым уровнем таких противоречий являются появляющиеся при интеграции погружаемого текста с текущим состоянием базы знаний *омонимичные знаки* и пары *синонимичных знаков*. Омонимичные знаки появляются в результате ошибочного отождествления знака, входящего в состав погружаемого текста, со знаком, входящим в состав погружаемого текста, со знаком, входящим в состав текущего состояния базы знаний. Появление пар синонимичных знаков, один из которых входит в погружаемый текст, а второй – в текущее состояние базы знаний, при погружении вводимого текста является "штатным" противоречием, устранение которого осуществляется путем отождествления ("склеивания") синонимичных знаков.]

⇒ *примечание**:

[Сложность проблемы понимания вводимой вербальной информации заключается не только в сложности непротиворечивого погружения вводимой информации в текущее состояние базы знаний, но и в сложности трансляции этой информации с внешнего языка на внутренний язык кибернетической системы, т. е. в сложности генерации текста внутреннего языка, семантически эквивалентного вводимому тексту внешнего языка. Очевидно, что для естественных языков указанная трансляция является сложной задачей, так как в настоящее время проблема формализации синтаксиса и семантики естественных языков не решена.]

семантическая совместимость кибернетической системы с партнерами

:= [уровень взаимопонимания кибернетической системой со своими партнерами]

:= [степень конвергенции (близости) базы знаний кибернетической системы с базами знаний своих партнеров]

семантическая совместимость кибернетической системы с партнерами

⇐ *частное свойство**:

совместимость кибернетических систем

⇒ *пояснение**:

[*семантическая совместимость кибернетических систем* определяется

- количеством знаков, которые хранятся в памяти одной заданной кибернетической системы и денотационная семантика которых совпадает с денотационной семантикой знаков, хранимых в памяти другой

заданной кибернетической системы (другими словами, это количество сущностей, которые описывают как в памяти первой кибернетической системы, так и в памяти второй кибернетической системы),

- тем, согласованы ли между двумя заданными кибернетическими системами факт совпадения денотационной семантики указанных выше знаков сущностей, описываемых в памяти как первой, так и второй кибернетической системы (такое согласование осуществляется путем согласования уникальных внешних идентификаторов (имен), которые приписываются указанным знакам сущностей и которые используются указанными кибернетическими системами при обмене сообщениями между ними).

]

⇒ *примечание**:

[Прежде всего семантическая совместимость двух заданных кибернетических систем определяется согласованностью систем понятий, используемых обеими взаимодействующими кибернетическими системами, (т.е. совпадением семантической трактовки всех этих понятий) и включением в число таких общих понятий всех или почти всех неопределяемых понятий, а также тех определяемых понятий, которые обеими кибернетическими системами часто используются при определении остальных определяемых понятий.]

⇒ *примечание**:

[Высокий уровень семантической совместимости даже для кибернетических систем с высоким уровнем интеллекта (например, для людей) встречается значительно реже, чем хотелось бы. Очевидно, что проблема обеспечения перманентной поддержки семантической совместимости взаимодействующих кибернетических систем является необходимым условием обеспечения высокого уровня взаимопонимания кибернетических систем и, как следствие, эффективного их взаимодействия.]

способность кибернетической системы к обеспечению семантической совместимости с партнерами

:= [способность кибернетической системы к обеспечению взаимопонимания со своими партнерами.]

⇒ *комплекс частных свойств**:

- { • *способность кибернетической системы к обеспечению семантической совместимости собственной базы знаний с базами знаний своих партнеров*
- *способность кибернетической системы к обеспечению коммуникационной совместимости со своими партнерами*

⇒ *примечание**:

[Речь идет о согласовании внешних языков, используемых кибернетическими системами при их общении.]

}

⇒ *комплекс частных свойств**:

- { • *уровень предварительной семантической совместимости кибернетической системы с партнерами*
- ⇒ *примечание**:
[Речь идет об обеспечении начальной (стартовой) семантической совместимости.]
- *способность кибернетической системы к перманентной поддержке семантической совместимости с партнерами*

⇒ *примечание**:

[Речь идет о перманентном процессе поддержки необходимого уровня семантической совместимости(взаимопонимания) в условиях постоянной эволюции всех взаимодействующих кибернетических систем.]

}

уровень предварительной семантической совместимости кибернетической системы с партнерами

:= [унификация представления информации, хранимой в памяти всевозможных кибернетических систем]

:= [максимально возможная конвергенция, стандартизация, согласованность представления информации, хранимой в памяти всевозможных кибернетических систем]

⇒ *примечание**:

[Речь идет об использовании всеми кибернетическими системами общего универсального языка внутреннего представления знаний и о согласовании используемых ими понятий]

способность кибернетической системы к перманентной поддержке семантической совместимости с партнерами

:= [способность кибернетической системы к согласованию денотационной семантики знаков (и, в первую очередь, знаков понятий), используемых в собственной базе знаний с денотационной семантике тех знаков, которые входят в состав информации поступающей от других кибернетических систем-партнеров]

:= [способность кибернетической системы к повышению уровня семантической совместимости и взаимопонимания с другими системами (в том числе, с компьютерными системами, с людьми) в условиях перманентного

процесса собственной эволюции (следствием которой является появление новых знаковых понятий и других описываемых сущностей, а также уточнение денотационной семантики используемых знаков), перманентной эволюции партнерских кибернетических систем и перманентной эволюции коллективно согласованной картины мира]

⇒ *примечание**:

[Рассматриваемое свойство (способность) кибернетической системы заключается в самостоятельной реализации перманентного (постоянного) процесса обеспечения поддержки своей семантической совместимости со всеми(!) кибернетическими системами, с которыми данная кибернетическая система взаимодействует в текущий момент времени. Подчеркнем при этом, что условия поддержки семантической совместимости постоянно меняются – меняется состав "партнеров", меняются (эволюционируют) сами "партнеры", эволюционирует и сама данная кибернетическая система]

*следует отличать**

- ⇒ {
- *способность кибернетической системы к обеспечению семантической совместимости с партнерами*
 - ∈ *свойство*
 - ⇒ *область определения**:
кибернетическая система
 - *семантическая совместимость кибернетической системы с партнерами*
 - ∈ *свойство*
 - ⇒ *область определения**:
*множество всевозможных неориентированных пар кибернетических систем**
 - := [множество всевозможных сочетаний кибернетических систем по две*]
 - := [множество всевозможных двухмощных множеств кибернетических систем*]
 - := [степень (уровень) семантической совместимости различных пар кибернетических систем]
- }

коммуникабельность кибернетической системы

:= *часто используемый sc-идентификатор**:

[коммуникабельность]

:= [способность кибернетической системы к установлению взаимовыгодных контактов с другими кибернетическими системами (в том числе, с коллективами интеллектуальных систем) путем честного выявления взаимовыгодных общих целей (интересов).]

:= [способность кибернетической системы к формированию новых партнерских связей с другими кибернетическими системами]

способность кибернетической системы к обсуждению и согласованию целей и планов коллективной деятельности

:= [способность активно участвовать в коллективном (в согласовании каких-либо предложений) – т.е. в подтверждении (признании) этих предложений, либо в их отклонении с указанием причин или предлагаемых доработок]

способность кибернетической системы брать на себя выполнение актуальных задач в рамках согласованных планов коллективной деятельности

⇒ *примечание**:

[Данная способность кибернетической системы предполагает:

- учет приоритета актуальных задач;
- учет собственных возможностей;
- согласование распределения актуальных задач по исполнителям;
- публикацию момента начала и предполагаемого момента завершения выполнения указанной актуальной задачи

]

социальная ответственность кибернетической системы

⇒ *свойство-предпосылка**:

- *способность кибернетической системы выполнять качественно и в срок взятые на себя обязательства в рамках соответствующих коллективов*
- *способность кибернетической системы адекватно оценивать свои возможности при распределении коллективной деятельности*

- *альтруизм/эгоизм кибернетической системы*
 - *отсутствие/наличие действий, которые по безграмотности кибернетической системы снижают качество коллективов, в состав которых она входит*
 - *отсутствие/наличие "осознанных", мотивированных действий снижающих качество коллективов, в состав которых кибернетическая система входит*
- }

альтруизм/эгоизм кибернетической системы

⇒ *примечание**:

[уровень мотивации к повышению качества коллективов, в состав которых кибернетическая система входит]

⇒ *эпиграф**:

[Надо любить науку, а не себя в науке.]

⇒ *эпиграф**:

[Ты играешь и всем своим видом показываешь: "Смотрите, как я красиво играю", а надо играть и показывать красоту самой музыки.]

социальная активность кибернетической системы

:= *часто используемый sc-идентификатор***:

[социальная активность]

:= [пассионарность]

⇒ *свойство-предпосылка**:

- {
 - *способность кибернетической системы к генерации предлагаемых целей и планов коллективной деятельности*
 - *активность кибернетической системы в экспертизе результатов других участников коллективной деятельности*
 - *способность кибернетической системы к анализу качества всех коллективов, в состав которых она входит, а также всех член этих коллективов*
 - *способность кибернетической системы к участию в формировании новых коллективов*
 - *количество и качество тех коллективов, в состав которых кибернетическая система входит или входила*

}

способность кибернетической системы к участию в формировании коллективов

:= [уровень способности в создании таких коллективов кибернетических систем, в состав которых входит данная кибернетическая система и которые направлены на коллективное решение соответствующего актуального класса сложных комплексных задач, с каждой из которых не может справиться любая из имеющихся кибернетических систем.]

⇒ *примечание**:

[Формирование специализированного коллектива кибернетических систем сводится к тому, что в памяти каждой кибернетической системы, входящей в коллектив, генерируется спецификация этого коллектива, включающая в себя:

- перечень всех членов коллектива;
- способности (возможности) каждого из них;
- обязанности в рамках коллектива;
- спецификацию всего множества задач (вида деятельности), для решения (выполнения) которых сформирован данный коллектив кибернетических систем

]

⇒ *примечание**:

[Каждая кибернетическая система может входить в состав большого количества коллективов, выполняя при этом в разных коллективах в общем случае разные "должностные обязанности", разные "бизнес-процессы"...]

⇒ *примечание**:

[Рассмотренный принцип формирования специализированного коллектива, состоящего из компьютерных систем и людей, фактически означает автоматизацию системной интеграции компьютерных систем и децентрализованный ("горизонтальный") характер такой интеграции, это очевидно предполагает наличие достаточно высокого уровня интеллекта у интегрируемых компьютерных систем и людей.]

количество и качество тех коллективов, в состав которых кибернетическая система входит или входила

⇒ *пояснение**:

[Данная характеристика кибернетической системы уточняет спектр ее социальной активности]

⇒ *примечание**:

[Чем умнее (интеллектуальнее) многоагентные системы, членом которых является данная кибернетическая система, тем выше ее "социальный статус" и перспективы быть умнее – есть у кого учиться]

} */* Завершили Сегмент “Комплекс свойств, определяющих уровень социализации кибернетической системы как фактора существенного повышения уровня ее обучаемости, а также фактора существенного повышения качества всех тех многоагентных систем, в состав которых входит данная кибернетическая система” */*

} */* Завершили Раздел “Предметная область и онтология кибернетических систем” */*

/* Раздел *****/

Предметная область и онтология логико-семантических моделей компьютерных систем, основанных на смысловом представлении информации

⊃=

{

⇒

дочерний раздел*:

- Предметная область и онтология смыслового представления информации
- Предметная область и онтология многоагентных моделей решения задач, основанных на смысловом представлении информации
- Предметная область и онтология логико-семантических моделей интерфейсов компьютерных систем, основанных на смысловом представлении информации

⇒

аннотация*:

[Данный раздел и дочерние ему разделы являются уточнением и обоснованием наших предложений, направленных на построение компьютерных систем следующего поколения, основанных на смысловом представлении обрабатываемой информации.]

⇒

основной тезис*:

[Для любой компьютерной системы можно построить эквивалентную ей логико-семантическую модель, основанную на смысловом представлении обрабатываемой информации]

логико-семантическая модель компьютерной системы

⇒

пояснение*:

[Главным фактором обеспечения совместимости различных видов знаний, различных моделей решения задач и различных компьютерных систем в целом является

- унификация (стандартизация) представления информации в памяти компьютерных систем;
- унификация принципов организации обработки информации в памяти компьютерных систем.

Унификация представления информации, используемой в компьютерных системах, предполагает:

- синтаксическую унификацию используемой информации – унификацию формы представления (кодирования) этой информации. При этом следует отличать:
 - кодирование информации в памяти компьютерной системы (внутреннее представление информации);
 - внешнее представление информации, обеспечивающее однозначность интерпретации (понимания, трактовки) этой информации разными пользователями и разными компьютерными системами;
- семантическую унификацию используемой информации, в основе которой лежит согласование и точная спецификация всех (!) используемых понятий (концептов) с помощью иерархической системы формальных онтологий.

]

стандарт

Э

Стандарт OSTIS

:= [Предлагаемый нами стандарт логико-семантических моделей компьютерных систем, основанных на смысловом представлении информации, и технологии разработки таких моделей и соответствующих компьютерных систем]

:=

[знания о структуре и принципах функционирования искусственных систем соответствующего класса]

:=

[онтология искусственных систем некоторого класса]

:=

[теория искусственных систем некоторого класса]

⇒

пояснение*:

[Важно отметить, что грамотная унификация (стандартизация) должна не ограничивать творческую свободу разработчика, а гарантировать совместимость его результатов с результатами других разработчиков. Подчеркнем также, что текущая версия любого стандарта – это не догма, а только опора для дальнейшего его совершенствования.

Целью качественного стандарта является не только обеспечения совместимости технических решений, но и минимизация дублирования (повторения) таких решений. Один из важных критериев качества стандарта – ничего лишнего.

Стандарты, как и другие важные для человечества *знания*, должны быть формализованы и должны постоянно совершенствоваться с помощью специальных *интеллектуальных компьютерных систем*, поддерживающих процесс эволюции стандартов путем согласования различных точек зрения.]

семантическая совместимость компьютерных систем

⇒ *пояснение**:

[Уровень совместимости *компьютерных систем* определяется трудоемкостью реализации процедур интеграции (объединения, соединения знаний этих систем), а также трудоемкостью и глубиной интеграции входящих в эти системы *решателей задач* (интеграции навыков и интерпретаторов этих навыков). Подчеркнем при этом, что интеграция интеграции рознь – от эклектики до гибридности и синергетичности дистанция огромного размера.

Совместимые *компьютерные системы* – это компьютерные системы, для которых существует автоматически выполняемая процедура их интеграции, превращающая эти системы в единую *гибридную систему*, в рамках которой каждая интегрируемая компьютерная система в процессе своего функционирования может свободно использовать любые необходимые информационные ресурсы (знания и навыки), входящие в состав другой интегрируемой компьютерной системы.

Целостную *компьютерную систему* можно рассматривать как решатель задач, интегрировавший несколько моделей решения задач и обладающий средствами взаимодействия с внешней для себя средой (с другими компьютерными системами, с пользователями).

Таким образом, для того, чтобы повысить уровень совместимости *компьютерных систем*, необходимо преобразовать их к виду *многоагентных систем*, работающих над общей семантической памятью. Такие *компьютерные системы* не всегда целесообразно непосредственно объединять (интегрировать) в более крупные *компьютерные системы*. Иногда целесообразнее их объединять в *коллективы взаимодействующих компьютерных систем*. Но при создании таких коллективов компьютерных систем унификация и совместимость таких систем также очень важны, т.к. существенно упрощают обеспечение высокого уровня их взаимопонимания. Так, например, противоречия между компьютерными системами, входящими в коллектив, можно обнаруживать путем анализа на непротиворечивость *виртуальной объединенной базы знаний* этого коллектива. Более того, непротиворечивость указанной виртуальной базы знаний можно считать одним из критериев семантической совместимости систем, входящих в соответствующий коллектив.]

компьютерная система, основанная на смысловом представлении информации

⇒ *пояснение**:

[Предлагаемое нами устранение проблем современных информационных технологий путем перехода к *смысловому представлению информации* в памяти компьютерных систем фактически преобразует современные компьютерные системы (в том числе и современные интеллектуальные компьютерные системы) в *компьютерные системы, основанные на смысловом представлении информации*, которые являются не альтернативной ветвью развития *компьютерных систем*, а естественным этапом их эволюции, направленным на обеспечение высокого уровня их *обучаемости* и, в первую очередь, *совместимости*.

Архитектура *компьютерных систем, основанных на смысловом представлении информации* (см. *Рис. Архитектура компьютерных систем, основанных на смысловом представлении информации*) практически совпадает с архитектурой *интеллектуальных компьютерных систем, основанных на знаниях*. Отличие здесь заключается в том, что в *компьютерных системах, основанных на смысловом представлении информации*:

- база знаний имеет смысловое представление;
- интерпретатор знаний и навыков представляет собой коллектив *агентов*, осуществляющих обработку *базы знаний*.

Как следствие этого, *компьютерные системы, основанные на смысловом представлении информации*, обладают высоким уровнем *обучаемости*, т.е. способностью быстро приобретать новые и совершенствовать уже приобретенные знания и навыки и при этом не иметь никаких ограничений на вид приобретаемых и совершенствуемых ею знаний и навыков, а также на их совместное использование.

Более того, при согласовании соответствующих стандартов, а также при перманентном совершенствовании этих стандартов и при грамотной их поддержке в условиях интенсивной эволюции как самих стандартов, так и *компьютерных систем, основанных на смысловом представлении информации* (речь идет о постоянной поддержке соответствия между текущим состоянием компьютерных систем и текущим состоянием эволюционируемых стандартов), *компьютерные системы, основанные на смысловом представлении информации* и их компоненты обладают весьма высокой степенью *совместимости*.

Это, в свою очередь, практически исключает дублирование инженерных решений и дает возможность существенно ускорить разработку *компьютерных систем, основанных на смысловом представлении информации*.

информации с помощью постоянно расширяемой библиотеки многократно используемых и совместимых между собой компонентов.

Основным лейтмотивом перехода от современных компьютерных систем (в том числе интеллектуальных) к компьютерным системам, основанным на смысловом представлении информации, хранимой в ее памяти, является создание **общей семантической теории компьютерных систем**, включающей в себя:

- семантическую теорию знаний и баз знаний;
- семантическую теорию задач и моделей решения задач;
- семантическую теорию взаимодействия информационных процессов;
- семантическую теорию пользовательских и, в том числе, естественно-языковых интерфейсов;
- семантическую теорию невербальных (сенсорно-эффекторных) интерфейсов;
- теорию универсальных интерпретаторов *логико-семантических моделей компьютерных систем* и, в частности, теорию семантических компьютеров.

Эпицентром следующего этапа развития информационных технологий является решение проблемы обеспечения **семантической совместимости компьютерных систем** и их компонентов. Для решения этой проблемы необходимо

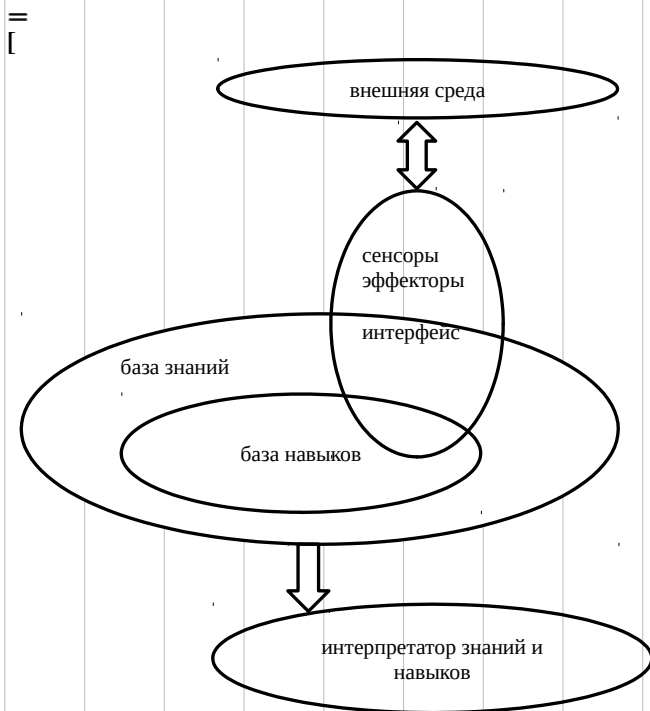
- переход от традиционных компьютерных систем и от современных интеллектуальных компьютерных систем к компьютерным системам, основанным на смысловом представлении информации;
- разработка стандарта компьютерных систем, основанных на смысловом представлении информации.

Очевидно, что компьютерные системы, основанные на смысловом представлении информации являются компьютерными системами нового поколения, устраняющие многие недостатки современных компьютерных систем. Но для массовой разработки таких систем необходима соответствующая технология, которая должна включать в себя

- теорию компьютерных систем, основанных на смысловом представлении информации и комплекс всех стандартов, обеспечивающих совместимость разрабатываемых систем;
- методы и средства проектирования компьютерных систем, основанных на смысловом представлении информации;
- методы и средства перманентного совершенствования самой технологии.

]

Рис. Архитектура компьютерных систем, основанных на смысловом представлении информации



]

}

/* Завершили Раздел “Предметная область и онтология логико-семантических моделей компьютерных систем, основанных на смысловом представлении информации” */

/* Раздел	*****/
Предметная область и онтология смыслового представления информации	
⊃=	
{	
←	частная предметная область и онтология*:
	Предметная область и онтология информационных конструкций
∋	максимальный класс объектов исследования ¹ :
	смысловое представление информации
∋	класс объектов исследования ¹ :
•	семантическая сеть
⇒	разбиение*:
{	• нерафинированная семантическая сеть
•	рафинированная семантическая сеть
}	
⇒	разбиение*:
{	• абстрактная семантическая сеть
:=	[семантическая сеть, абстрагирующаяся от того, как физически представлены ее элементарные (атомарные) фрагменты, а также связи инцидентности между этими фрагментами]
•	графически представленная семантическая сеть
:=	[нарисованная семантическая сеть]
•	семантическая сеть, хранимая в графодинамической памяти
⇔	следует отличать*:
	представление семантической сети в адресной памяти
∄	семантическая сеть
:=	[представление семантической сети в виде линейной информационной конструкции, которая хранится в адресной памяти и которая, строго говоря, уже не является семантической сетью, но является информационной конструкцией, семантически эквивалентной соответствующей (представляемой) семантической сети]
}	
•	язык семантических сетей
:=	[язык, все тексты которого являются семантическими сетями]
⇒	разбиение*:
{	• специализированный язык семантических сетей
•	универсальный язык семантических сетей
}	
⊃	язык рафинированных семантических сетей
⇒	рассматриваемые вопросы*:
{	• [Что такое семантические сети и в чем их принципиальное отличие от других вариантов представления информации]
•	[До какой степени можно минимизировать алфавит элементов семантических сетей]
•	[Можно ли все описываемые связи свести к бинарным связям и почему это целесообразно]
•	[Можно ли разработать универсальный язык семантических сетей]
•	[До какой степени можно упростить синтаксические структуры семантических сетей до, условно говоря, рафинированного вида]
•	[Какими достоинствами обладает семантические сети]
}	
⇒	ссылка*:
•	Понятие Технологии OSTIS
⇒	аннотация*:
	[В указанном сегменте Стандарта OSTIS рассматриваются принципы, лежащие в основе Технологии OSTIS, основным из которых является ориентация на использование универсального языка рафинированных семантических сетей в качестве внутреннего языка интеллектуальных компьютерных систем]
•	Описание внутреннего языка ostis-систем

- ⇒ *аннотация**:
[В указанном разделе *Стандарта OSTIS* рассматриваются принципы, лежащие в основе универсального языка *рафинированных семантических сетей*, используемого в качестве внутреннего языка *ostis-систем – интеллектуальных компьютерных систем* следующего поколения]
- *Описание языка графического представления знаний ostis-систем*
⇒ *аннотация**:
[В указанном разделе *Стандарта OSTIS* рассматриваются принципы, лежащие в основе универсального языка графически представленных семантических сетей, используемого в *пользовательском интерфейсе ostis-систем*]
 - *Бирюков.Б.В.ТеориСГФ-1960ст*
 - *Моррис.Ч.У.ОсновТЗ-2001ст*
 - *Пирс.Ч.С.ЧтоТЗ-2009ст*
 - *Степанов.Ю.С.Семиотика-1971ст*
 - *Мельчук.И.А.ОпытГЛМСТ-1999кн*

/*****/

знак

:= [фрагмент информационной конструкции, обладающий свойством, обозначать некоторую сущность (объект), которая наряду с другими сущностями описывается указанной информационной конструкцией]

⇒ *примечание**:

[**Форма** представления знаков в известной степени условна и является результатом соглашения между носителями соответствующего языка. Знак может быть, например, представлен:

- в виде фрагмента речевого сообщения (последовательностью фонем);
- в виде строки символов (последовательности букв) в заданном алфавите;
- в виде иероглифа, пиктограммы;
- в виде жеста.

]

€ *ключевой знак'*:

Предметная область и онтология информационных конструкций

Э *раздел Базы знаний IMS.ostis*

⇒ *характеристика элементов данного множества**:

[Знаки, используемые в различных языках, характеризуются:

- синтаксической структурой, по совпадению (изоморфизму) которых для разных знаокв предполагается их синонимия;
- денотационной семантикой, т.е. той сущностью, которая обозначается соответствующим знаком;
- типом (классом) обозначаемой сущности, которая может быть:
 - материальным(физическим) элементом (точкой) абстрактного пространства, множеством, которое может быть:
 - связью;
 - классом;
 - структурой;
 - реальной и вымышленной сущностью;
 - константной (конкретной) и переменной (произвольной) сущностью;
 - постоянно существующей и временно существующей сущностью (прошлой, настоящей, будущей);
- множеством тех связей, которые связывают сущность, обозначаемую данным знаком с другими сущностями, а также, если данный знак обозначает некоторую связь, множеством сущностей, которые связаны этой связью, т.е. сущностей, являющихся компонентом этой связи;
- текущим статусом самого знака в памяти кибернетической системы, который может быть:
 - логически удаленным знаком;
 - настоящим знаком;
 - предлагаемым (возможно, будущим) знаком.

]

денотат*

:= [денотат заданного знака*]

:= [объект, обозначаемый заданным знаком*]

:= [денотационная семантика заданного знака*]

:= [смысл заданного знака*]

:= [Бинарное ориентированное отношение, каждая пара которого связывает:

- некоторый знак, представленный в той или иной форме в тексте исследуемого языка;
- со знаком той сущности, которая обозначается указанным выше знаком в рамках используемого метаязыка.

]

⇒ *примечание**:

[Данное отношение используется, когда с помощью одного языка необходимо описать денотационную семантику другого языка. Фактически речь идет о переводе заданного знака, входящего в состав некоторого рассматриваемого текста, принадлежащего некоторому исследуемому языку (языку-объекту), на некоторый метаязык (в нашем случае на SC-код), денотационная семантика которого нам считается априори известной. Указанный перевод есть связь заданного знака с синонимичным ему знаком, входящим в состав текста, принадлежащего указанному метаязыку.]

⇔ *обратное отношение**:

*внешний sc-идентификатор**

:= [быть знаком, обозначающим заданную сущность*]

∈ *ключевой знак'*:

- *Описание внешних идентификаторов знаков, входящих в тексты внутреннего языка ostis-систем*
∈ *раздел Базы знаний IMS.ostis*
- *Предметная область и онтология знаков, входящих в тексты внутреннего языка ostis-систем*
∈ *раздел Базы знаний IMS.ostis*

информационная конструкция

:= [информация]

⇒ *примечание**:

[В общем случае информационная конструкция представляет собой сложную иерархическую структуру, каждому уровню иерархии которой соответствует определенный класс информационных конструкций.]

⊃ *синтаксически элементарный фрагмент информационной конструкции*

:= [атомарный фрагмент информационной конструкции]

:= [элемент информационной конструкции]

⇒ *примечание**:

[Примерами таких элементарных фрагментов информационных конструкций являются буквы]

⊃ *буква*

⊃ *простой знак*

:= [семантически элементарный фрагмент информационной конструкции]

⊂ *знак*

⊃ *выражение*

:= [сложный (непростой) знак]

:= [знак, являющийся одновременно некоторым знанием обозначаемой сущности (спецификацией этой сущности)]

:= [знак, построенный как выражение вида "тот, который..."]

:= [знак, в состав которого входят другие знаки]

⊂ *знак*

⊃ *простой текст*

:= [минимальная синтаксически целостная и корректная (правильная) информационная конструкция, включающая в себя:

- знак некоторой описываемой связи;
- минимальную спецификацию указанного знака связи (указание отношения, которому это связь принадлежит);
- указание всех компонентов описываемой связи (знаков всех сущностей, связываемых этой связью, и/или всех знаков, связываемых этой связью – описываемая связь может связывать не только "внешние" описываемые сущности, но и сами знаки);

- если описываемая связь не является бинарной, то связи с её компонентами могут потребовать явного представления знаков этих связей с дополнительным указанием роли этих компонентов.

]

С *текст*

⊃ *сложный текст*

:= [информационная конструкция, являющаяся результатом соединения нескольких простых текстов]

С *текст*

⊃ *простое знание*

:= [минимальная семантически целостная информационная конструкция]

:= [знание, в состав которого не входят другие знания]

С *знание*

⊃ *сложное знание*

:= [информационная конструкция, являющаяся результатом соединения нескольких простых знаний]

:= [знание, в состав которого не входят другие знания]

С *знание*

∈ *ключевой знак*¹:

Предметная область и онтология информационных конструкций

/*****/

смысловое представление информации

:= [смысловая форма представления информации]

:= [смысловое представление информационной конструкции]

:= [знаковая конструкция (текст), представленная в смысловой форме]

:= [смысловое представление информационной конструкции]

:= *часто используемый sc-идентификатор*^{*}:

[смысл]

:= [смысловое представление]

:= [семантическое представление информации]

⇒ *основной принцип*^{*}:

[Как можно меньше лишнего, не имеющего отношения к смыслу представляемой информации.]

:= [такое представление информационной конструкции, которое существенно прощает соответствие между структурой самой этой информационной конструкции и описываемой (отображаемой) ею конфигурацией связей между рассматриваемыми (исследуемыми) сущностями]

:= [смысловое представление знаковой конструкции]

:= [абстрактная знаковая конструкция, являющаяся инвариантом соответствующего максимального класса семантически эквивалентных знаковых конструкций]

:= [смысл информационной конструкции]

:= [денотационная семантика информационной конструкции]

:= [смысловое представление информационной конструкции]

⇒ *примечание*^{*}:

[Суть (смысл, денотационная семантика) любой информационной конструкции (информационной модели) сводится к описанию системы (конфигурации) связей между списываемыми (рассматриваемыми) сущностями. Важно, чтобы эта суть не была закамуфлирована различными "синтаксическими" деталями, не имеющими никакого отношения к указанному смыслу (синтаксическая структура знаков, многократное повторение одного и того же знака, синонимия, омонимия, местоимения, предлоги, знаки препинания, разделители, ограничители, падежи и т.п.) а обусловленными формой представления информационных конструкций, например, их линейностью.]

⇒ *пояснение*^{*}:

[Смысловое представление любой информации в конечном счете сводится:

- к перечню знаков конкретных описываемых сущностей - как первичных сущностей, так и вторичных сущностей, которые сами являются информационными конструкциями (фрагментами данной конструкции);
- к явному описанию связи между знаками вторичных сущностей и самими этими сущностями (т.е. фрагментами информационной конструкции);
- к описанию других связей между описываемыми сущностями

]

⇒ *пояснение**:

[Формализация смысла представляемой информации, т.е. строгое уточнение того, что такое *смысловое представление информации*, является объективной основой для унификации представления информации в памяти компьютерных систем и ключом к решению многих проблем семантической совместимости и эволюции компьютерных систем и технологий.

Согласно *Мартынову В. В. (Мартынов.В.В.СемиологОИ-1974кн)*, «фактически всякая мыслительная деятельность человека (не только научная), как полагают многие ученые, использует внутренний семантический код, на который переводят с естественного языка и с которого переводят на естественный язык. Поразительная способность человека к идентификации огромного множества структурно различных фраз с одинаковым *смыслом* и способность запомнить смысл вне этих фраз убеждает нас в этом.»

Приведем также слова *Мельчука И. А. (Мельчук.И.А.ОпытГЛМСТ-1999кн)*:

«Идея была следующая – язык надо описывать следующим образом: надо уметь записывать смыслы фраз. Не фразы, а их *смыслы*, что отдельно. Плюс построить систему, которая по смыслу строит фразу. Это та область или тот поворот исследований, при котором интуиция способного лингвиста работает лучше всего: как выразить на данном языке данный смысл. Это – то, для чего лингвистов учат..

Лингвистический *смысл* научного текста – это совсем не то, что ты, читая его, из него извлекаешь. Это, очень грубо говоря, инвариант синонимических перифраз. Ты можешь один и тот же смысл выразить очень многими способами. Когда ты говоришь, то можешь сказать по-разному: “Сейчас я налью тебе вина”, или: “Дай, я тебе предложу вина”, или: “Не выпить ли нам по бокалу?”, – все это имеет один и тот же смысл. И вот можно придумать, как записывать этот *смысл*. Именно его. Не фразу, а *смысл*. И работать надо от этого *смысла* к реальным фразам. Синтаксис там по дороге тоже нужен, но он нужен именно по дороге, он не может быть ни конечной целью, ни начальной точкой. Это – промежуточное дело.» (*Мельчук.И.А.КакНМЛ-1998ст*).]

⇒ *примечание**:

[Грамотная унификация (стандартизация) *смыслового представления информации* не должна привести к ограничению творческой свободы авторов различного вида публикуемых научно-технических знаний (и, в том числе, разработчиков *баз знаний*), не должна гарантировать *семантическую совместимость* различных *знаний*, представленных различными авторами (разумеется, при условии соблюдения соответствующих правил построения этих *знаний*). При этом любые *стандарты* (в том числе и принятые стандарты *смыслового представления информации*) должны постоянно эволюционировать. Текущая версия любого стандарта должна быть не догмой, а точкой опоры для дальнейшего совершенствования этого стандарта.]

⊂ УСК

:= [Универсальный Семантический Код]

⇒ *автор**:

Мартынов В. В.

⇒ *примечание**:

[Разработанный Мартыновым В. В. Универсальный Семантический Код стал важнейшим этапом создания универсальных формальных средств смыслового представления знаний. Основная методологическая идея *Мартынова В. В.*, касающаяся построения *языка смыслового представления знаний*, заключается в том, чтобы выделить смысловые "кирпичики", имеющие достаточно общий характер, а многообразие конкретных смыслов конструировать комбинаторно за счёт различных комбинаций (конфигураций) из этих "кирпичей". Это можно назвать принципом минимизации типов атомарных смысловых фрагментов.]

← *ключевой знак**:

Мартынов.В.В.УниверсСКУСК-1984кн

смысловое представление информации*

:= *пояснение**:

[*Бинарное ориентированное отношение*, каждая *пара* которого связывает некоторую *информационную конструкцию* со смысловым представлением этой *информационной конструкции**.]

⊂ *формализация**

/*****/

формализация*

∈ *бинарное ориентированное отношение*

:= [формализация информации*]

:= [пара, связывающая менее формализованное и более формализованное представление некоторой информации*]

- := [формализация информационной модели некоторой описываемой (моделируемой) системы взаимосвязанных сущностей*]
- := [Бинарное ориентированное отношение, каждая пара которого, связывает два семантически эквивалентных знания, второе из которых является более точным (более точно сформированным) знанием по сравнению с первым знанием*.]
- ⇒ *пояснение**:
 [Повышение точности (строгости) формулировки знания – минимизация (а в идеале – исключение) неоднозначной семантической интерпретации этой формулировки, т.е. несоответствия того, что хотел "сказать" автор формулировки, и того, как его поняли. Формализация знаний предполагает (1) точное (строгое) описание синтаксиса и денотационной семантики того языка, на котором формулируются знания и (2) максимально возможное упрощение синтаксических и семантических принципов, лежащих в основе указанного языка. Очевидно, что естественные языки указанным требованиям не удовлетворяют и, следовательно, не могут быть основой для точной формулировки научно-технических знаний и, соответственно, для представления этих знаний в памяти интеллектуальных компьютерных систем. Очевидно также, что разработка универсального языка формального представления научно-технических знаний является основой для глубокой конвергенции различных научно-технических дисциплин, для расширения областей применения современной математики и даже для появления новых разделов математики, которые, например, изучают общие свойства универсального смыслового пространства и, в частности, свойство семантического расстояния (семантической близости) как между различными знаками, так и между различными знаковыми конструкциями (конфигурациями знаков).]
- ⇒ *примечание**:
 [Слово "математика" означает "точное знание".]
 ⇐ *цитата**:
 Арнольд.В.И.ЧтоТМ-2012кн
- ⇒ *примечание**:
 [Формализация информационной модели есть не что иное как "движение" в сторону семантического (смыслового) представления этой модель, т.е. переход к такому представлению этой модели, в котором мы избавляемся от всего, не имеющего отношения к сути моделируемой системы и касающегося только способа построения этой модели (т.е. её синтаксической структуры).]
- ⇒ *примечание**:
 [Нет проблемы записать любое знание в компьютерную память. Для этого надо придумать соответствующий формат их кодирования. Но есть проблема представить это знание так, чтобы с ним было легко работать, чтобы с использованием этого знания можно было достаточно удобно (без лишних накладных расходов, обусловленных выбранным способом представления) решать самые различные информационные задачи (задачи интеграции знаний, информационного поиска по базе знаний, верификации и оптимизации баз знаний, логического вывода, поиска способов решения задач, хранимых в базе знаний и т. д.). Какими характеристиками должно обладать удобное представление знаний, удовлетворяющее указанным требованиям. Очевидно, что такое представление есть не что иное, как формальная (математическая) модель, семантически эквивалентная этим знаниям. Т.е. удобно представить знание – это фактически построить соответствующую этому знанию математическую модель. Для интеллектуальных компьютерных систем важно не просто приобрести знания, но и представить их в такой форме, которая была бы удобна не только для человека (пользователя и разработчика), но и для различных компьютерных систем, т.е. не требовала бы переоформление (перезаписи) этих знаний для различных компьютерных систем. Очевидно, что такая форма записи (представления) знаний должна быть абсолютно не зависящий от различных компьютерных платформ. Это и есть главная цель формализации знаний, обеспечивающей эффективную автоматизацию обработки этих знаний.]

формальное представление информации

- С информация
- := [информационная конструкция]
- ⇒ *вопрос**:
 [Почему разработка и использование формальных моделей (математических моделей) представления информации является важнейшим этапом развития любой научной и научно-технической дисциплины.]
- ⇒ *ответ**:
 {• [Формализация любой предметной области даёт возможность более конструктивно накапливать, интегрировать, понимать и систематизировать новые знания об этой предметной области]
 • [Формализация предметной области обеспечивает более строгую верификацию, обоснование (аргументацию, доказательство) и согласование различных точек зрения]

	<ul style="list-style-type: none"> • [Формализация <i>предметной области</i> создает условия для разработки строгих и легко воспроизводимых (реализуемых) <i>методов</i> решения различных <i>классов задач</i>]
	} <ul style="list-style-type: none"> ← <i>достоинства*</i>: <i>формальное представление информации</i>
:=	[формальное (формализованное) представление информационной конструкции]
С	<i>смысловое представления информации</i>
⇒	<i>примечание*</i> : [Высшим уровнем качества <i>формального представления информации</i> является смысловое представление этой информации]
:=	[формальная модель системы описываемых взаимосвязанных сущностей]
:=	[математическая модель системы описываемых взаимосвязанных сущностей]
:=	[формула]
⇒	<i>примечание*</i> : [Сам термин " <i>формальное представление информации</i> " свидетельствует о том, что при таком представлении <i>информации</i> сама <i>форма</i> представляемой информационной конструкции (т.е. синтаксическая структура этой конструкции) имеет очевидную аналогию с описываемой конфигурацией связей между соответствующими соответствующими описываемыми <i>сущностями</i> . В предельном "идеальном" случае указанная аналогия между формой и смыслом информационной конструкции должна быть изоморфизмом.]
⇒	<i>примечание*</i> : [Формализация формализации рознь и, соответственно, степень приближения формы представления информации к "идеальному" смысловому представлению может быть различной. Разработка такого "идеального" языка <i>смыслового представления информации</i> должна руководствоваться следующими основными критериями: <ul style="list-style-type: none"> □ максимально возможное упрощения синтаксиса (как можно меньше синтаксических излишеств и синтаксического разнообразия). □ обеспечение <u>универсальности</u> языка. Подчеркнем, что обеспечение универсальности <i>языка смыслового представления информации</i> является весьма нетривиальной задачей, т.к. сложно одновременно достигнуть две противоречащие друг другу цели- обеспечить простоту синтаксиса языка и его неограниченную семантическую мощьность. Косвенным подтверждением этого является большое количество созданных человечеством специализированных <i>формальных языков, языков смыслового представления информации</i> и даже <i>языков семантических сетей</i> , что свидетельствует о востребованности <i>смыслового представления информации</i> .]
⇒	<i>разбиение*</i> : { <ul style="list-style-type: none"> • <i>формальное представление информации, не являющееся смысловым</i> • <i>смысловое представление информации, не являющееся семантической сетью</i> • <i>нерафинированная семантическая сеть</i> := [смысловое представления информации 2-го уровня] • <i>рафинированная семантическая сеть</i> := [смысловое представление информации 3-го уровня] }
/*****/	
	<i>смысловое представление информации, не являющееся семантической сетью</i>
⇒	<i>примечание*</i> : [Данному уровню смыслового представления информации соответствуют предлагаемые нами универсальные формальные языки SCs-код и SCn-код]
⊃	SCs-код ∈ <i>универсальный формальный язык</i> ∈ <i>ключевой знак'</i> : <i>Описание языка линейного представления знаний ostis-систем</i>
⊃	SCn-код ∈ <i>универсальный формальный язык</i> ∈ <i>ключевой знак'</i> : <i>Описание языка структурированного представления знаний ostis-систем</i>
←	<i>принципы, лежащие в основе*</i> : { <ul style="list-style-type: none"> • [В состав <i>смыслового представления информации, не являющегося семантической сетью</i> могут входить все уровни иерархии представления информационной конструкции – }

- синтаксически элементарные фрагменты информационной конструкции, из которых строятся простые знаки описываемых сущностей, а также разделители и ограничители;
- простые знаки;
- выражения;
- простые тексты;
- сложные тексты;
- простые знания;
- сложные знания.

]

- [Множество всех описываемых сущностей, не являющихся связями, разбивается на два подмножества:

- каждой сущности, принадлежащей первому подмножеству, взаимно однозначно соответствует множество синтаксически эквивалентных (синтаксически одинаковых) *простых знаков*, каждый из которых обозначает указанную сущность;
- каждой сущности, принадлежащей второму подмножеству, соответствует в общем случае семейство множеств, како из которых является максимальным множеством синтаксически эквивалентных выражений, обозначающих указанную сущность.

]

⇒ *следовательно**:

[Здесь синонимия *простых знаков*, имеющих разную синтаксическую структуру, отсутствует, а вот синонимия *выражений*, имеющих разную синтаксическую структуру, вполне возможна. Подчеркнем при этом, что в рамках *смыслового представления информации, не являющегося семантической сетью*, *знаки* (как *простые знаки*, так и *выражения*), имеющие одинаковую синтаксическую структуру, считаются также и семантически эквивалентными, т.е. обозначающими одну и ту же сущность. Это означает отсутствие омонимии синтаксически эквивалентных знаков.]

⇒ *следовательно**:

[В рамках *смыслового представления информации, не являющегося семантической сетью*, простые знаки, обозначающие разные сущности, имеют легко устанавливаемое отличие своих синтаксических структур, а простые знаки, обозначающие одну и ту же сущность имеют легко устанавливаемое сходство своих синтаксических структур. Таким образом, в рамках *смыслового представления информации, не являющегося семантической сетью*, дублирование знаков, т.е. многократное вхождение *знаков* одной и той же сущности, допускается.]

- [Связи как вид описываемых сущностей имеют очень важные особенности:

- каждой описываемой *связи* однозначно, а в подавляющем числе случаев и взаимно однозначно соответствует *простой текст*, являющийся контекстом (спецификацией) этой *связи*;
- весьма редки *кратные связи*, т.е. *связи*, принадлежащие одному и тому же *отношению* и связывающие одинаковым образом одни и те же *сущности*;
- довольно редко *связи* являются компонентами других *связей*.

]

⇒ *следовательно**:

[Для подавляющего числа описываемых *связей* нет никакой необходимости вводить обозначающие их *знаки*, если эти *связи* описываются соответствующими *простыми текстами*. Вместо таких *знаков* можно ввести условные представления этих *связей*, отражающие их вид и направленность. Такие условные представления (изображения) описываемых *связей* можно считать *знаками*, но *знаками*, семантические свойства которых принципиально отличаются от тех *знаков* описываемых *сущностей*, которые мы рассматривали выше. Любые данного вида *разные знаки* описываемых *связей* даже, если, они являются синтаксически эквивалентными, т.е. имеют одинаковую структуру, считаются *знаками разных* описываемых *связей*. Синонимия таких *знаков* принципиально возможна, но только в том случае, если *простые тексты*, описывающие соответствующие *связи*, будут полностью продублированы.]

- [Для описания связей между описываемыми сущностями в смысловом представлении информации нет необходимости использовать такие приемы естественных языков, как склонения, спряжения, семантическая значимость последовательности знаков.]

- [В случае, если с помощью *простых текстов* необходимо описать контекст (спецификацию) нескольких кратных *связей*, все эти *связи* необходимо обозначить *знаками* первого типа – *знаками, синтак-*

сическая структура каждого из которых уникальна. Кроме этого, необходимо ввести знак, который обозначает *связь инцидентности* между описываемой *связью* и компонентом этой *связи*, и который относится к числу *знаков* второго типа – *знаков*, разные экземпляры (разные вхождения) которого считаются обозначениями разных связей]

- [Для явного указания синонимии двух разных *знаков* первого типа, имеющих разную *синтаксическую структуру*, вводится фиктивная *связь равенства*, которая сама не является описываемой *связью*, а только указывает факт синонимии двух *знаков*, по крайней мере один из которых должен быть *выражением*.]
- [Каждая описываемая *сущность* должна быть специфицирована путем указания типа этой *сущности*. Описываемая *сущность* может быть:
 - материальной сущностью*;
точкой абстрактного пространства;
множеством:
 - связью*;
 - классом*;
 - структурой*;
 - реальной сущностью*;
вымышленной сущностью;
 - константой*;
переменной;
 - постоянной сущностью*;
временной сущностью:
 - прошлой сущностью*;
 - настоящей сущностью*;
 - будущей сущностью*.

Кроме того, сам *знак* описываемой *сущности* может иметь следующий статус:

- логически удаленный знак*;
- настоящий знак*;
- будущий знак*.

]

- [Возможно дублирование информации, т.е. могут присутствовать семантически эквивалентные информационные конструкции, входящие в состав одной информационной конструкции (например, в состав информации, хранимой в памяти одной компьютерной системы). Но при этом есть принципиальная возможность обнаружить такое дублирование информации.]

)

/*****/

графовая структура

:= *определение**:

[абстрактная (математическая) структура, которая задается:

- множеством ее элементов:
 - множеством ее вершин (узлов);
 - множеством ее связей:
 - множеством ее ребер (неориентированных пар элементов графовой структуры);
 - множеством ее дуг (ориентированных пар элементов графовой структуры);
 - множеством ее гиперребер, каждое из которых является конечным множеством элементов графовой структуры, имеющим мощность больше двух
- бинарным ориентированным отношением инцидентности, связывающим каждую связку графовой структуры с каждым компонентом (элементом) этой связки.

]

*следует отличать**

- ⇒ { • [графовую структуру как абстрактный математический объект, в рамках которого не уточняется то, как выглядят (представляются, изображаются) элементы графовой структуры и связи их инцидентности]
- [представление (изображение) графовой структуры – ее рисунок, ее представление в компьютерной памяти в виде матрицы инцидентности, матрицы смежности, списковой структуры]
- }

графовая структура

:= часто используемый *sc-идентификатор**:

[дискретная информационная конструкция]

⇒ примечание*:

[Поскольку любая графовая структура является дискретной математической моделью, которая может описывать любое множество *сущностей*, связанных между собой заданным множеством *связей*, все графовые структуры с полным основанием можно считать дискретными *информационными конструкциями*. Более того, любая дискретная *информационная конструкция* (в том числе, и обычная цепочка символов) с формальной точки зрения является *графовой структурой*. Тот факт, что теория графов рассматривает "синтаксические" свойства *графовых структур* с точностью до их изоморфизма, не лишает *графовые структуры* соответствующих "семантических" свойств.]

⇒ пояснение*:

[С семантической точки зрения графовая структура – это нелинейная (в общем случае) знаковая конструкция, в состав которой могут входить знаки любых сущностей. При этом указанные знаки синтаксически разбиваются на два класса –

- на знаки сущностей, которые не являются связями между сущностями – в теории графов такие знаки называются узлами (вершинами);
- на знаки связей между *сущностями* – к таким *знакам* относятся ребра неориентированных графов, гиперребра гиперграфов, дуги ориентированных графов.

Кроме того, на множестве знаков *сущностей*, входящих в состав *графовой структуры*, задаются *отношения инцидентности*, которые связывают *знаки связей*, входящих в состав *графовой структуры*, со знаками тех *сущностей* которые являются компонентами указанных *связей*.

Теория графов рассматривает только "синтаксические" аспекты *графовых структур*. Семантика *графовой структуры* задается *онтологией*, специфицирующей систему понятий, экземплярами которых являются элементы этой графовой структуры, т.е. *знаки*, входящие в состав этой *графовой структуры*.]

семантическая сеть

:= [графовая структура, являющаяся формальным уточнением одного из видов *смыслового представления информации*]

С графовая структура

С *смысловое представление информации*

С *знаковая структура*

:= [графовая структура, вершины (узлы) которой трактуются как знаки некоторых описываемых сущностей, а связки (ребра, дуги, гиперребра, гипердуги) которой трактуются как знаки связей между описываемыми сущностями и/или знаками этих сущностей]

:= [абстрактная графовая и в общем случае нелинейная знаковая конструкция (знаковая структура), являющаяся вариантом смыслового представления соответствующей информации]

:= пояснение*:

[информационная конструкция, в которой явно выделены знаки всех описываемых сущностей, а также знаки связей, которые также считаются описываемыми сущностями и которые связывают либо сами описываемые сущности, либо описываемые сущности со знаками других описываемых сущностей, либо знаки описываемых сущностей]

⇒ примечание*:

[Теоретико-графовая трактовка (уточнение) *смыслового представления информации* является вполне естественной, т.к. любая описываемая сущность может иметь неограниченное количество связей с другими описываемыми сущностями, и очень часто анализ свойств какой-либо описываемой сущности предполагает анализ всех представленных (описанных) связей этой сущности с различными другими сущностями. Более того, для любых описываемых сущностей существует связывающая их связь (все в Мире взаимосвязано). Вопрос в том, какая это связь и нужно ли ее описывать. Далеко не все то, что можно описывать, целесообразно описывать.]

- ⇒ *общие предпосылки**:
- [Информация в знаковой конструкции содержится не в самих знаках, а в конфигурации связей между знаками, обозначающими описываемые сущности]
 - [Конфигурация связей между описываемыми сущностями в общем случае не являются линейной]
 - [Идеальным *смысловым представлением информации* следует считать такую знаковую конструкцию, синтаксическая конфигурация связей между знаками которой изоморфна конфигурации связей между описываемыми сущностями]

⇒ *примечание**:
 [Понятие семантической сети является основным понятием для *Технологии OSTIS*. Ранее семантические сети рассматривались не как основа технологии разработки интеллектуальных компьютерных систем, а как наглядная иллюстрация представления знаний, не имеющая практической перспективы из-за сложности реализации, не обладающая универсализмом.

Для нас семантические сети – это

- формальный подход к построению знаковых конструкций:
- формальный подход, позволяющий создавать целое семейство языков и в том числе языков универсальных:
- основа организации памяти нового типа – структурно перестраиваемой (реконфигурируемой) памяти, обработка информации в которой сводится к реконфигурации связей между ее элементами.

-] *достоинства**:
- [Семантическая сеть наряду с системами правил является весьма распространенным способом представления знаний в интеллектуальных системах. Особое значение этот способ представления знаний приобретает в связи с развитием сети интернет. Кроме ряда особенностей, позволяющих применять семантические сети в тех случаях, когда системы правил не применимы, *семантические сети* обладают следующим важным свойством: они дают возможность соединения в одном представлении синтаксиса и семантики или синтаксического и семантического аспектов описаний знаний предметной области. Происходит это благодаря тому, что в семантических сетях наряду с переменными для обозначения тех или иных объектов (элементов множеств, некоторых конструкций из них) присутствуют и сами эти элементы и конструкции; присутствуют и связи, сопоставляющие тем или иным переменным множества допустимых интерпретаций. Эти обстоятельства позволяют во многих случаях резко уменьшить реальную вычислительную сложность решаемых задач. Помимо изобразительных возможностей, семантические сети обладают более серьезными достоинствами. То обстоятельство, что вся информация об индивиде представлена в единственном месте – в одной вершине – означает, что вся эта информация непосредственно доступна в этой вершине, что, в свою очередь, сокращает время поиска, в частности, при выполнении унификации и подстановки в задачах логического вывода. Существует еще одна, более тонкая особенность расширенных семантических сетей – они позволяют интегрировать в одном представлении *синтаксис* и *семантику* (т.е. интерпретацию) клаузальных форм. Это позволяет в процессе вывода обеспечивать взаимодействие синтаксических и семантических, теоретико-модельных подходов, что, в свою очередь, также является фактором, зачастую делающим вывод практически более эффективным]

← *цитата**:

Осипов.Г.С.МетодыИИ-2015кн,с.43-54

← *часть**:

Осипов.Г.С.МетодыИИ-2015кн

- [Все связи между *знаками*, входящими в состав *семантической сети* представляются с помощью специальных связующих элементов *семантической сети* (дуг, ребер) и, следовательно, для описания указанных связей в *семантической сети* нет необходимости использовать такие средства, как предлоги, союзы, падежи, склонения, спряжения, различные разделители и ограничители, что существенно упрощает обработку *знаний*.]
- [Соединение синтаксических и семантических аспектов в *семантической сети* проявляется в том, что дуга или ребро, "синтаксически" соединяющая элементы *семантической сети* описывает наличие соответствующей *связи* между *сущностями*, обозначаемыми указанными элементами *семантической сети*.]

/*****

нерафинированная семантическая сеть

⇒ *примечание**:

[Переход от смыслового представления информации, не являющегося семантической сетью, к нерафинированным семантическим сетям представляет собой переход к информационным конструкциям, имеющим более простую синтаксическую структуру и денотационную семантику.

К нерафинированным семантическим сетям можно отнести тексты предлагаемого нами универсального формального SCg-кода, а также используемые в Semantic Web rdf-графы]

⊃

SCg-код

∃ универсальный формальный язык

∃ ключевой знак':

Описание языка графического представления знаний *ostis-систем*

⊃

rdf-граф

⇐

принципы, лежащие в основе*:

- [Поскольку в *информационной конструкции* информация содержится не в самих *знаках* (если не считать *знаки*, являющиеся *выражениями*), а в конфигурации связей между *знаками*, очень важно явно формально представить саму эту конфигурацию *знаков*. И как нельзя лучше для этого подходит понятие *графовой структуры* и, соответственно, понятие *семантической сети*.
Что касается *выражений*, то каждое из них легко трансформируется в *семантически эквивалентную* информационную конструкцию, не являющуюся *выражением*. Заметим, что *выражения* используются исключительно для минимизации числа вводимых *знаков* (имен) с уникальной синтаксической структурой.]
- [Все элементы, входящие в состав нерафинированной семантической сети и представленные узлами, ребрами или дугами, являются *знаками*, обозначающими соответствующие описываемые *сущности*, причём *знаками* второго типа, которые, обозначая соответствующую *сущность*, входят в *информационную конструкцию* однократно (отсутствует многократное вхождение *знаков*, обозначающих одну и ту же *сущность*). Также *знаки* могут иметь синтаксическую структуру, которая не является уникальной для обозначаемой *сущности*, а отражает только принадлежность этой сущности к соответствующим классам. Таким образом, в *нерафинированной семантической сети* в отличие от *смыслового представления информации не являющегося семантической сетью*, доминируют не *знаки* первого типа, а *знаки* второго типа, которыми в *нерафинированной семантической сети* представлены (обозначены) все описываемые *сущности*, а в *смысловом представлении информации, не являющемся семантической сетью*, представлены только *бинарные связи* и то не все.]
- [Все ребра *нерафинированной семантической сети* являются *знаками бинарных неориентированных связей* и формально трактуются как *знаки двухмощных множеств*, каждым элементом которых являются либо *знак сущности*, соединяемой указанной *бинарной связью*, либо *знак*, который сам является *сущностью*, соединяемой этой *бинарной связью*. Более того, все *двухмощные множества*, не являющиеся *кортежами* (ориентированными парами) в *нерафинированной семантической сети* обозначаются *ребрами* этой сети.]
- [Все дуги *нерафинированной семантической сети* являются *знаками бинарных ориентированных связей* и формально трактуются как *знаки двухмощных кортежей* (ориентированных пар), каждым элементом которых является либо *знак сущности*, соединяемой указанной *бинарной связью*, либо *знак*, который сам является *сущностью*, соединяемой этой *бинарной связью*. Более того, все *ориентированные пары* в *нерафинированной семантической сети* обозначаются *дугами* этой сети.]
- [Каждая *небинарная связь*, описываемая в *нерафинированной семантической сети*, трактуется как множество, мощность которого не равна двум и обозначается соответствующим узлом этой сети, который соединяется дугами, принадлежащими отношению принадлежности со всеми *знаками*, которые либо обозначаются *сущности*, связываемые рассматриваемой *небинарной связью*, либо сами являются такими *сущностями*. Для описания ориентированных *небинарных связей* (в частности, *небинарных кортежей*) выделяется несколько подмножеств отношения принадлежности, соответствующих различным ролям элементов (компонентов) ориентированных *небинарных связей*.]
- [В рамках *нерафинированной семантической сети* все рассматриваемые связи между описываемыми *сущностями* представляются явно в виде *знаков*, обозначающих эти связи.]
- [В рамках *нерафинированной семантической сети* не используются такие средства, как разделители, ограничители и др.]
- [Узлами *нерафинированной семантической сети*, которые обозначают различного вида ключевые описываемые *сущности* (прежде всего, различные *понятия*) приписываются уникальные *знаки* (имена) этих *ключевых сущностей*. Очевидно, что каждый такой *узел* и приписываемое ему *имя* – это *синонимичные знаки*, обозначающие одну и ту же *сущность*, но являющиеся *знаками* двух разных типов – (1) *знаком*, который однократно представлен в рамках *информационной конструкции*; (2) *знаком*, синтаксическая структура которого взаимно однозначно соответствует обозначаемой им *сущности*.]

- [Большинству узлов, обозначающих небинарные связи, большинству ребер и дуг, а также некоторым другим узлам нерафинированной семантической сети могут быть приписаны уникальные знаки (в частности, имена) понятий (чаще всего, отношений), которым принадлежат указанные узлы, ребра и дуги.]

)

/*****

рафинированная семантическая сеть

⇒ *основной принцип**:

[Абсолютно ничего лишнего, не имеющего отношения к смыслу представляемой информации]

:= [предельно компактная (сжатая) смысловая информационная модель соответствующей системы рассматриваемых (описываемых, исследуемых, моделируемых) сущностей]

⇒ *примечание**:

[Указанная система рассматриваемых сущностей представляет собой конфигурацию связей между этими сущностями. Подчеркнем при этом, что указанные связи между рассматриваемыми сущностями также входят в число рассматриваемых сущностей.]

:= [информационная конструкция, являющаяся результатом максимально возможного упрощения ее синтаксической структуры при обеспечении представления любой информации, что приводит к фактическому слиянию синтаксических и семантических аспектов представления информации]

:= [семантическая сеть "внутреннего" потребления, используемая для смыслового представления информации в памяти компьютерных систем]

:= [уточнение принципов смыслового представления информации, которое основано, во-первых, на четком противопоставлении внутреннего языка компьютерной системы, используемого для хранения информации в памяти компьютера, и внешних языков компьютерной системы, используемых для общения (обмена сообщениями) компьютерной системы с пользователями и другими компьютерными системами (рафинированная семантическая сеть используется исключительно для внутреннего представления информации в памяти компьютерной системы), и, во-вторых на максимально возможном упрощении синтаксиса внутреннего языка компьютерной системы при обеспечении универсальности путем исключения из такого внутреннего универсального языка средств, обеспечивающих коммуникационную функцию языка (т.е. обмен сообщениями).

Так, например, для внутреннего языка компьютерной системы излишними являются такие коммуникационные средства языка, как союзы, предлоги, разделители, ограничители, склонения, спряжения и другие. Внешние языки компьютерной системы могут быть как близки ее внутреннему языку, так и весьма далеки от него (как, например, естественные языки).]

:= [абстрактная знаковая конструкция, принадлежащая универсальному внутреннему языку компьютерных систем и являющаяся инвариантом соответствующего максимального множества семантически эквивалентных знаковых конструкций (текстов), принадлежащих самым различным языкам]

⇒ *принципы лежащие в основе**:

• [Каждый фрагмент рафинированной семантической сети является либо знаком (элементарным фрагментом, представленным либо узлом, либо ребром, либо дугой), либо множеством знаков, связанных между собой отношением инцидентности элементов рафинированной семантической сети. Указанное отношение инцидентности является бинарным ориентированным отношением, связывающим знаки описываемых связей (которые представляются ребрами, дугами и узлами, если описываемая связь является небинарной) со знаками, которые либо обозначают связываемые сущности, либо сами являются такими сущностями]

⇒ *следовательно**:

[В состав рафинированной семантической сети не входят такие средства синтаксической структуризации знаковых конструкций, как разделители и ограничители. Любая структуризация рафинированных семантических сетей описывается явно с помощью метаязыковых средств путем:

- введения узлов рафинированной семантической сети, обозначающих различные неэлементарные фрагменты этой семантической сети, являющиеся множествами узлов, ребер и дуг, входящих в состав обозначаемого фрагмента;
- введения дуг принадлежности, связывающих введенные узлы, обозначающие неэлементарные фрагменты рафинированной семантической сети, с элементами обозначаемых ими множеств;
- введения целого ряда отношений, связывающих неэлементарные фрагменты рафинированной семантической сети с другими фрагментами, а также с сущностями других видов;

- введения различных классов неэлементарных фрагментов *рафинированной семантической сети*.

]

- [Абсолютно все *знаки*, входящие в состав *рафинированной семантической сети*, являются синтаксически элементарными (атомарными) фрагментами *рафинированной семантической сети*, т.е. фрагментами, "внутренняя" структура которых не имеет никакого значения для семантического анализа и понимания *рафинированной семантической сети*. Множеству *знаков*, входящих в *рафинированную семантическую сеть*, как и множеству *букв*, входящих в обычный *текст*, ставится в соответствие *алфавит*, определяющий синтаксическую типологию таких элементарных фрагментов *рафинированной семантической сети*. При этом, если *алфавит* букв обычного *текста* не имеет никакой семантической интерпретации, то *алфавит* элементарных фрагментов *рафинированной семантической сети* имеет четкую семантическую интерпретацию – каждый элемент этого *алфавита* обозначает класс *знаков сущности*, синтаксический тип которых соответствует указанному элементу *алфавита* (задается этим элементом *алфавита* *знаков*, входящих в состав *рафинированной семантической сети*).]
 - ⇒ *следовательно**:

[Таким образом, *знаки*, входящие в *рафинированную семантическую сеть*, не являются *именами* (терминами) и, следовательно, не привязаны ни к какому *естественному языку* и не зависят от субъективных терминотворческих пристрастий различных авторов. Это значит, что при коллективной разработке *рафинированных семантических сетей*, соответствующих каким-либо информационным ресурсам, терминологические споры практически исключены.]
 - ⇒ *следовательно**:

[В *рафинированной семантической сети* нет необходимости использовать синтаксически элементарные фрагменты, не являющиеся *знаками* описываемых *сущностей*, т.е. фрагменты *информационной конструкции*, из которых строятся *простые знаки*, *выражения*, а также различные разделители и ограничители. Более того, в *рафинированной семантической сети* нет необходимости противопоставлять *простые знаки* и *выражения*. Как *простым знакам*, так и *выражениям* в *рафинированной семантической сети* соответствуют элементы этой сети, имеющие аналогичные *денотаты*. Но при этом *выражениям* дополнительно соответствуют семантически эквивалентные неэлементарные фрагменты *рафинированной семантической сети*, которые специфицируют *сущности*, обозначаемые этими *выражениями*.]
- [Абсолютно все описываемые *связи* между описываемыми сущностями в *рафинированной семантической сети* представляются явно в виде соответствующих *знаков*, обозначающих эти *связи* и инцидентных *знакам* связываемых *сущностей*. Для бинарных связей, связывающих две описываемые сущности, *знаком* связей являются *ребра* или *дуги* *рафинированной семантической сети*.]
 - ⇒ *следовательно**:

[В *рафинированных семантических сетях* нет необходимости использовать такие средства, как склонения, спряжения, род (мужской, женский, средний), семантически интерпретируемая последовательность слов.]
- [Все *знаки*, входящие в состав *рафинированной семантической сети*, входят в нее однократно. Т.е. в рамках *рафинированной семантической сети* отсутствуют пары *синонимичных знаков*, т.е. *знаков*, имеющих один и тот же *денотат*. Таким образом, разные элементы *рафинированной семантической сети* априори считаются *знаками разных* сущностей. При этом эти *знаки* могут принадлежать одному и тому же синтаксическому типу, т.е. одному и тому же элементу *алфавита* соответствующего языка *рафинированных семантических сетей*. Таким образом, в *рафинированных семантических сетях* отсутствует синонимия не только *знаков*, имеющих одинаковую синтаксическую структуру, не только *знаков*, имеющих одинаковый синтаксический тип, но также и просто разных *знаков*.]
 - ⇒ *следовательно**:

[Появление в *рафинированной семантической сети* синонимичных знаков превращает эту семантическую сеть в некорректную и требует отождествления (склеивания) обнаруженных синонимичных знаков.]
- [В рамках *рафинированной семантической сети* отсутствуют *синонимичные знаки*, т.е. *знаки*, которые имеют не один, а несколько *денотатов*, каждому из которых соответствует свой контекст (ракурс) семантической трактовки этого *знака*.]
 - ⇒ *примечание**:

[Когда речь идет об омонимии знаков в привычных нам языках, имеется в виду омонимия разных знаков, имеющих одинаковую синтаксическую структуру, т.е. омонимия разных вхождений, разных экземпляров синтаксически эквивалентных, но семантически различных знаков. Очевидным примером такого рода омонимии являются различного вида местоимения.]

[В рамках каждой *рафинированной семантической сети* отсутствует дублирование информации не только в виде многократного вхождения *синонимичных знаков*, т.е. *знаков* с одинаковыми денотатами, но также и в виде многократного вхождения *семантически эквивалентных рафинированных семантических сетей*. Две *рафинированные семантические сети* являются *семантически эквивалентными* в том и только в том случае, если:

- они *изоморфны*;
- пары соответствия указанного *изоморфизма* связывают *синонимичные знаки*.

Таким образом, полное исключение *омонимии знаков* является необходимым и достаточным условием исключения *семантически эквивалентных рафинированных семантических сетей*. Подчеркнем при этом, что запрет *семантической эквивалентности* в рамках *рафинированной семантической сети* не означает запрета *логической эквивалентности* фрагментов *рафинированной семантической сети*. Логическая эквивалентность необходима для обеспечения компактности представления некоторых знаний. Тем не менее, логической эквивалентностью хранимых в памяти знаковых конструкций увлекаться не следует, т.к. *логически эквивалентные* знаковые конструкции – это представление одного и того же *знания*, но с помощью *разных наборов понятий*. В отличие от этого *семантически эквивалентные знаковые конструкции* – это представление одного и того же *знания* с помощью одних и тех же *понятий*. Очевидно, что многообразие возможных вариантов представления одних и тех же *знаний* в памяти компьютерной системы существенно усложняет решение *задач*. Поэтому, полностью исключив *семантическую эквивалентность* в смысловой памяти, необходимо стремиться к минимизации *логической эквивалентности*. Для этого необходимо грамотное построение системы используемых *понятий* в виде иерархической системы формальных *онтологий*.]

⇒ *следовательно**:

[Интеграция (соединение, объединение) двух *рафинированных семантических сетей*, в результате чего могут появиться семантически эквивалентные фрагменты, сводится к тому, чтобы результат такого соединения был приведен в соответствие с требованием отсутствия синонимии элементов и семантической эквивалентности фрагментов *рафинированной семантической сети*.]

- [Рафинированные семантические сети должны быть *универсальными*, т.е. должны обеспечивать представление *любой* информации, в том числе, и *метаинформации*, обеспечивающей описание различных связей, свойств и закономерностей самих *рафинированных семантических сетей*, на множестве которых, в частности, заданно *отношение* "быть подструктурой*" которое связывает *рафинированные семантические сети* с их фрагментами (частями), т.е. с теми *рафинированными семантическими сетями*, которые входят в их состав.

Каждая *рафинированная семантическая сеть* трактуется как множество *знаков* *взаимно однозначно* соответствующих обозначаемым ими *сущностям* (денотатам этих *знаков*) и множество *связей* между этими *знаками*.

Каждая *связь* между *знаками* трактуется, с одной стороны, как множество *знаков*, связываемых этой *связью*, а, с другой стороны, как описание (отражение, модель) соответствующей *связи*, которая связывает денотаты указанных *знаков* или денотаты одних *знаков* непосредственно с другими *знаками*, или сами эти *знаки*. Примером первого вида *связи* между *знаками* является *связь* между *знаками материальных сущностей*, одна из которых является частью другой. Примером второго вида *связи* между *знаками* является *связь* между *знаком*, входящим в состав внутреннего смыслового представления информации, и *знаком файла*, являющегося электронным отражением структуры представления указанного *знака* во внешних *знаковых конструкциях*. Примерами третьего вида *связи* между *знаками* является *связь* между синонимичными *знаками*.

Денотатами *знаков* могут быть *любые* описываемые сущности, причем: (1) не только конкретные (константные, фиксированные), но и произвольные (переменные, нефиксированные) сущности, "пробегающие" различные множества *знаков* (возможных значений), (2) не только реальные (материальные), но и абстрактные сущности (например, числа, точки различных абстрактных пространств), (3) не только "внешние" но и "внутренние" сущности, являющиеся множествами *знаков*, входящих в состав той же самой *знаковой конструкции*, хранимой в памяти компьютерной системы.]

- [Поскольку *рафинированные семантические сети* ориентированы на *смысловое представление информации* в памяти *компьютеров нового поколения*, необходимо, с одной стороны, использовать накопленный полезный опыт представления информации в *современных компьютерах*, а, с другой стороны, обеспечить взаимодействие *компьютерных систем*, построенных на *современных компьютерах*, с *компьютерными системами*, построенными на *компьютерах нового поколения*. Для этой цели в памяти *компьютеров нового поколения* можно и нужно обеспечить обработку и хранение различного вида *информационных конструкций*, представленных в различных широко используемых форматах. И ничто не препятствует такие *информационные конструкции*, хранимые в

памяти компьютера нового поколения и не являющиеся рафинированными семантическими сетями, рассматривать как *сущности*, описываемые *рафинированной семантической сетью*, хранимой в памяти этого компьютера нового поколения. Такой вид *сущностей*, описываемых *рафинированной семантической сетью* и хранимых в той же памяти, будем называть *файлами*, описываемыми соответствующей *рафинированной семантической сетью*, т.е. "электронными" образами (копиями) соответствующих *информационных конструкций*. Таким образом, среди *узлов рафинированной семантической сети* появляются *узлы*, являющиеся знаками *файлов*, т.е. *узлы*, денотаты (обозначающие *сущности*) которых находятся (хранятся) в той же памяти, что и обозначающие их *узлы*.]

⇒ *следовательно**:

[Ничто не мешает в виде *файла*, описываемого *рафинированной семантической сетью*, хранить *имя* (термин) какой-либо *сущности*, описываемой этой же семантической сетью, а также связать это *имя* (точнее, *узел*, обозначающий это *имя*) с тем элементом *рафинированной семантической сети*, который обозначает ту же описываемую *сущность*.]

- [Следствием указанных принципов *рафинированных семантических сетей* является также то, что эти принципы приводят к нелинейным *знаковым конструкциям* (к *графовым структурам*), что усложняет реализацию *памяти компьютерных систем*, но существенно упрощает ее логическую организацию (в частности, ассоциативный доступ).

Нелинейность *рафинированных семантических сетей* обусловлена тем, что:

- каждая описываемая *сущность*, т.е. *сущность*, имеющая соответствующий ей *знак*, может иметь неограниченное число *связей* с другими описываемыми *сущностями*;
- каждая описываемая *сущность* в смысловом представлении имеет единственный *знак*, т.к. синонимия *знаков* здесь запрещена;
- все *связи* между описываемыми *сущностями* описываются (отражаются, моделируются) *связями* между *знаками* этих описываемых *сущностей*.

]

⇒ *примечание**:

[Напомним, что нелинейность *информационных конструкций* характерна не только для *рафинированных*, но и для *нерафинированных семантических сетей*.]

)

▷ SC-код

:= [Semantic Computer Code]

∈ *универсальный формальный язык*

∈ *ключевой знак'*:

Описание внутреннего языка ostis-систем

⇒ *пояснение**:

[В качестве *стандарта универсального смыслового представления информации в памяти компьютерных систем* нами предложен SC-код (Semantic Computer Code). В отличие от УСК *Мартынова В.В.*, он, во-первых, носит нелинейный характер и, во-вторых, специально ориентирован на кодирование информации в памяти компьютеров *нового поколения*, ориентированных на разработку семантически совместимых *интеллектуальных компьютерных систем* и названных нами *семантическими ассоциативными компьютерами*. Более подробно это понятие (SC-код) рассмотрено в разделе *Предметная область и онтология внутреннего языка ostis-систем*. Таким образом, основным лейтмотивом предлагаемого нами *смыслового представления информации* является ориентация на формальную модель памяти *компьютерных нефон-неймановского компьютера*, предназначенного для реализации *интеллектуальных систем*, использующих *смысловое представление информации*. Особенности такого представления являются следующие:

- ассоциативность памяти;
- поскольку при смысловом представлении информация содержится в конфигурации связей между знаками, переработка информации сводится к реконфигурации этих связей (к графодинамическим процессам);
- прозрачная семантическая интерпретируемость и, как следствие, *семантическая совместимость*.

Подчеркнем что, неявная привязка к *фон-неймановским компьютерам* присутствует во всех известных *моделях представления знаний*. Одним из примеров такой зависимости, является, например, обязательность именования описываемых объектов.]

⇒ *достоинства**:

{•

[рафинированная семантическая сеть есть объективный, не зависящий от субъективизма и многообразия синтаксических решений, способ представления информации]

- [в рамках *рафинированной семантической сети* существенно упрощается процедура *интеграции знаний* и погружения новых знаний в *базу знаний*]
- [существенно упрощается процедура приведения различного вида *знаний* к общему виду (к согласованной системе используемых *понятий*)]
- [существенно упрощается процедура интеграции различных *решателей задач* и *целых компьютерных систем*]
- [существенно упрощается автоматизация перманентного процесса *поддержки семантической совместимости* (согласованности *понятий* и *онтологий*) для *компьютерных систем* в условиях их постоянного совершенствования]
- [в рамках *рафинированных семантических сетей* достаточно легко осуществляется переход от информационных конструкций к информационным метаконструкциям путем введения узлов *семантической сети*, обозначающих *информационные конструкции*, а также дуг, связывающих эти узлы со всеми элементами обозначаемой ими *информационной конструкции*]
- [на основе *рафинированных семантических сетей* существенно упрощается интеграция различных дисциплин в области *Искусственного интеллекта*, т.е. построение *Общей формальной теории интеллектуальных компьютерных систем*, так как для построения общей формальной модели *интеллектуальных компьютерных систем* необходим базовый язык, в рамках которого можно было бы легко переходить от информации (от *знаний*) к *метаинформации* (к метазнаниям, к спецификациям исходных *знаний*). Это подтверждается тем, что:
 - подавляющее число *понятий Искусственного интеллекта* носит метаязыковой характер;
 - формальное смысловое уточнение почти каждого *понятия Искусственного интеллекта* требует предшествующего формального уточнения соответствующего языка-объекта. Так, например, как можно строго говорить о *языке онтологий* (т.е. *языке спецификации предметных областей*), не уточнив *язык* представления самих этих *предметных областей*. как можно строго говорить о *языке* описания способов обработки *информации*, не уточнив *язык* представления самой этой обрабатываемой *информации*.

]

}

/*****

язык смыслового представления информации

:= [смысловой язык]

:= [семантический язык]

⇒ разбиение*:

- язык смыслового представления информации, не являющийся языком семантических сетей
- язык семантических сетей

язык семантических сетей

⇒ пояснение*:

[Несмотря на то, что синтаксическая структура семантической сети во многом носит объективный характер, поскольку определяется конфигурацией описываемых связей между описываемыми сущностями. Тем не менее, можно говорить о разных *языках семантических сетей*, каждому из которых соответствует свой *алфавит** элементов (синтаксически атомарных фрагментов) *семантических сетей*. При этом языки семантических сетей могут быть как специализированными, так и универсальными. Задача каждого из этих *языков* – обеспечить в рамках *языка* полное отсутствие многообразия синтаксических форм представления одной и той же информации.]

С язык

:= [множество информационных конструкций, для которого существуют, причем не обязательно в формализованном виде, (1) правила построения синтаксически корректных информационных конструкций, а также (2) правила, позволяющие установить семантическую корректность правильно построенных (синтаксически корректных) информационных конструкций]

:= [язык, информационными конструкциями которого являются семантические сети и в рамках которого обеспечивается полное отсутствие многообразия форм представления одной и той же информации]

:= [графовый (нелинейный) язык смыслового представления информации]

⇒ разбиение*:

- { • *специализированный язык семантических сетей*
:= [язык семантических сетей, семантическая мощность которого ограничена соответствующей предметной областью]
- *универсальный язык семантических сетей*
⇒ *примечание**:
[Человечество давно и широко использует различные специализированные языки семантических сетей – язык принципиальных электрических схем, язык блок-схем программ, язык генеалогических деревьев и др. Но в настоящее время актуальным является создание такого *универсального языка семантических сетей*
 - синтаксис и семантика которого были бы максимально просты;
 - по отношению к которому все используемые специализированные языки были бы его подязыками*;
 - который был бы приспособлен к использованию в качестве внутреннего языка интеллектуальных компьютерных систем и компьютеров следующего поколения;
 - который был бы удобной основой как для обмена информацией между интеллектуальными компьютерными системами, так и для общения интеллектуальных компьютерных систем с их пользователями

- ⇒ *разбиение**:
- { • *язык нерафинированных семантических сетей*
 - *язык рафинированных семантических сетей*

следует отличать*

- ⇒ { • *язык семантических сетей*
:= [язык семантических сетей, рассматриваемых как абстрактные графовые структуры, в которых не уточняется способ их кодирования]
⇒ *SC-код*
- *графодинамический язык семантических сетей*
:= [язык графического изображения (визуализации) семантических сетей]
:= [язык, текстами которого являются рисунки семантических сетей]
⇒ *SCg-код*

универсальный язык семантических сетей

- ⇒ *примечание**:
[Если ставить задачу разработки универсального(!) языка, текстами которого являются графовые структуры, то классических графовых структур явно недостаточно. Так, например:
- по аналогии с переходом от ребер к ребрам и гиперребрам необходим переход от дуг к ориентированным связкам, связывающим более чем два компонента и в рамках которых эти компоненты могут иметь разные роли, которые необходимо явно указывать (классическим видом таких связок являются кортежи);
 - в семантических сетях, представляющих некоторые виды знаний, некоторые связки (ребра, дуги, гиперребра, ориентированные связки, связывающие более двух компонентов) могут быть компонентами других связок;
 - в семантических сетях, представляющих различного вида метазнания необходимо вводить узлы, обозначающие целые фрагменты (подграфы) этих же семантических сетей, и, соответственно, вводить дуги, связывающие каждый из этих узлов со всеми элементами подграфа, обозначаемого этим узлом.
-]

/*****

семантическая модель базы знаний

- := *пояснение**:
[смысловое представление всей *базы знаний интеллектуальной компьютерной системы* в виде *семантической сети*, принадлежащей *универсальному языку семантических сетей*]
- ⇒ *пояснение**:

[Для того, чтобы семантические сети могли быть использованы в качестве средства представления знаний в памяти интеллектуальной компьютерной системы необходимо:

- рассмотреть *семантические сети* как тексты, представляющие различного вида знания;
- уточнить синтаксис и семантику универсального (!) языка представления знаний, текстами которого являются *семантические сети* [ИнформЭнциклопедСловарь-2008кн – стр.195]. Считается, что *семантические сети* являются теоретической моделью представления знаний, не используемой на практике. [ИнформЭнциклопедСловарь-2008кн – стр. 207] Однако, если реализовать *графодинамическую память* и разработать языки программирования, ориентированные на обработку информации в такой памяти, то уникальные достоинства *семантических сетей* будут практически использованы в полной мере.

]

семантическая модель базы знаний

⇒ достоинства*:

- [семантическая модель базы знаний, построенная на основе универсального языка семантических сетей, обеспечивает высокий уровень ассоциативности доступа к требуемым фрагментам базы знаний благодаря широкому многообразию реализуемых видов запросов и существенному снижению реальной вычислительной сложности алгоритмов доступа (информационного поиска)]
- [семантическая модель базы знаний позволяет реализовать эффективную семантическую навигацию по текущему состоянию базы знаний (при просмотре базы знаний) путем отображения различного вида семантических окрестностей для указываемых элементов семантической сети. При этом семантическая модель базы знаний позволяет реализовать наглядную двумерную или трехмерную визуализацию просматриваемого фрагмента базы знаний (просматриваемой семантической сети). Таким образом, семантическая сеть является средством представления знаний, удобным как для самой интеллектуальной компьютерной системы, так и для её пользователей.]
- [сущностями, вписываемыми в семантической модели базы знаний и, соответственно, обозначаемыми знаками этих сущностей, могут быть не только части внешней среды соответствующие интеллектуальной компьютерной системе, но и части (фрагменты) самой базы знаний. Это дает возможность базе знаний включать в себя описание собственной структуры с любой степенью детализации и рассматривающее самые разные аспекты такой структуризации.]
- [семантическая модель базы знаний позволяет явно выделить фрагменты базы знаний, представляющие различные предметные области и соответствующие им онтологии, а также явно описать иерархию выделенных предметных областей и иные связи между ними (например, различного рода морфизмы). Такая семантическая структуризация базы знаний позволяет осуществлять локализацию области действия каждой конкретной операции обработки базы знаний, что существенно упрощает реализацию этих операций. Каждая предметная область выделенная в рамках семантической модели базы знаний, описывающая соответствующий класс исследуемых (описываемых) сущностей (объектов исследования) и соответствующих подклассов этого класса с помощью соответствующего набора отношений (в том числе, функций и алгебраических операций) и соответствующего набора свойств (параметров), представляет собой результат интеграции (соединения) текстов соответствующего специализированного языка семантической сети.]
- [размещение каждой информации, хранимой в составе семантической модели базы знаний, и, соответственно, доступ к этой информации (поиск её в базе знаний) определяются исключительно семантическими характеристиками этой информации (т.е. её смыслом) и не зависят от особенностей реализации памяти интеллектуальной компьютерной системы. Т.е. смысл информации однозначно определяет её "местоположение" в семантической модели базы знаний, а, точнее, её связи с остальной частью этой базы знаний]
- [если объем представления информационной конструкции определить как пару, состоящую (1) из количества синтаксически элементарных (атомарных) фрагментов этой конструкции и (2) из числа элементов алфавита указанных элементарных фрагментов, и если рассмотреть множество всевозможных семантически эквивалентных* представлений каждой информационной конструкции, то наиболее компактным (сжатым) её представлением окажется представление в виде рафинированной семантической сети. Более того, при расширении базы знаний (при увеличении числа описываемых сущностей и, в частности, числа описываемых связей) компактность рафинированных семантических сетей повышается, т.к. новые описываемые связи далеко не всегда рассматривают связи между новыми сущностями, которые до этого не описывались.]
- [семантические сети позволяют говорить о принципиально ином характере соединения ("конкатенации" интеграции) двух текстов в один интегрированный текст. Интеграция двух семантических сетей предполагает склеивание (отождествление) синонимичных элементов интегрируемых се-

мантических сетей. Такая интеграция, в частности, происходит при вводе (погружении) новой информации в состав семантической модели базы знаний.]

- [семантические модели баз знаний дают возможность:
 - конструктивно осуществлять анализ *семантической связности базы знаний*, путем уточнения понятия семантической силы связи между различными элементами и фрагментами базы знаний;
 - конструктивно осуществлять кластеризацию баз знаний;
 - задавать метрику семантического расстояния между знаками, входящими в состав базы знаний;
 - осуществлять в рамках базы знаний описания различного вида соответствий (морфизмов) между различными фрагментами базы знаний (изоморфизмов, гомоморфизмов, аналогий и отличий различного вида). Так, например, большое значение имеет исследование таких соответствий между различными *предметными областями*;
 - широко использовать мощный арсенал теоретико-графовых *алгоритмов* для выполнения различного рода операций обработки *баз знаний*.

]

)

/*****

*следует отличать**

- Э {
 - *предельно омонимичный класс синтаксически эквивалентных знаков*
 := [класс синтаксически эквивалентных знаков, все экземпляры (все вхождения) которого являются знаками разных сущностей]
 ⇒ *следовательно**:
 [В рамках рассматриваемого класса знаков синонимия знаков отсутствует]
 - := [максимальный класс синтаксически эквивалентных знаков, среди которых отсутствуют синонимичные знаки]
 - *частично омонимичный класс синтаксически эквивалентных знаков*
 := [класс синтаксически эквивалентных знаков, среди экземпляров которого встречаются как синонимичные знаки, так и знаки разных сущностей]
 - *неомонимичный класс синтаксически эквивалентных знаков*
 := [класс синтаксически эквивалентных знаков, все экземпляры которого являются знаками одной и той же сущности]
 := [класс синтаксически эквивалентных знаков, синтаксическая структура которых однозначно идентифицирует (соответствует) обозначаемую ими сущность]
 - *множество особенностей (характеристик), которыми обладает сущность, обозначаемая заданным знаком**
- Э {
 - *смысловое представление информации**
 := *часто используемый sc-идентификатор**:
 [смысл*]
 - *смысловое представление информации*
 ⇐ *второй домен**:
*смысловое представление информации**
 - *синтаксическая структура информационной конструкции**
 - *синтаксическая структура информационной конструкции*
 ⇐ *второй домен**:
*синтаксическая структура информационной конструкции**
 - *денотационная семантика информационной конструкции**
 := [соответствие (морфизм) между синтаксической структурой заданной информационной конструкции и ее *смысловым представлением**]
 ⇒ *примечание**:
 [соответствие между знаками входящими в состав *рафинированной семантической сети* и их *денотатами** (обозначаемыми сущностями) являются взаимно однозначными]
- Э {
 - *смысловое пространство*
 Э *SC-пространство*
 := [семантическое пространство]

- ⇒ *пояснение**:
[объединение (соединение) всевозможных корректных абстрактных семантических сетей, принадлежащих некоторому языку абстрактных *рафинированных семантических сетей*]
- := [глобальная (максимальная) абстрактная *рафинированная семантическая сеть*, включающая в себя всевозможные абстрактные рафинированные семантические сети соответствующего языка]
- := [абстрактное смысловое пространство]
- *абстрактная смысловая память*
 - := [абстрактная семантическая память]
 - := [среда, обеспечивающая хранение абстрактных рафинированных семантических сетей, а также редактирование этих семантических сетей и при этом абстрагирующаяся от деталей этих процессов]
 - := [абстрактная графодинамическая память, обеспечивающая хранение и редактирование абстрактных рафинированных семантических сетей]
- *реальная смысловая память*
 - := [физическая реализация абстрактной смысловой памяти]
- ⇔ *следует отличать**:
программная реализация модели абстрактной смысловой памяти на современных компьютерах

}

}

/* Завершили Раздел “Предметная область и онтология смыслового представления информации” */

```

/* Раздел *****/
Предметная область и онтология многоагентных моделей
решения задач, основанных на смысловом представлении
информации
⊃=
{
Предметная область многоагентных моделей решения задач, основанных на смысловом представлении
информации
∈ предметная область
⊃ максимальный класс объектов исследования1:
  многоагентный подход к обработке информации
⊃ класс объектов исследования1:
  • интеграция решателей задач
⊃ исследуемое отношение1:
  • совместимость моделей решения задач*

агентно-ориентированный подход к обработке информации
⇒ примечание*:
  [В качестве основы унификации принципов обработки информации в компьютерных системах предлагается
  использовать агентно-ориентированный подход к обработке информации, обладающий рядом важных
  достоинств.]
⇒ достоинства*:
  {
  • [Автономность (независимость) агентов, что позволяет локализовать изменения, вносимые в систему
    при ее эволюции, и снизить соответствующие трудозатраты.]
  • [Децентрализация обработки, т.е. отсутствие единого контролирующего центра, что также позволяет
    локализовать вносимые в систему изменения.]
  • [Возможность параллельной работы разных информационных процессов, соответствующих как
    одному агенту, так и разным агентам, как следствие, – возможность распределенного решения задач.
    Однако возможность параллельного выполнения информационных процессов подразумевает наличие
    средств синхронизации такого выполнения, разработка которых является отдельной задачей.]
  • [Активность агентов и многоагентной системы в целом, дающая возможность при общении с такой
    системой не указывать явно способ решения поставленной задачи, а формулировать задачу в
    декларативном ключе.]
  }
⇒ источник*:
  Wooldridge.M.IntroMAS-2009кн
⇒ недостатки современного состояния*:
  {
  • [Знания агента представляются при помощи узкоспециализированных языков, зачастую не предна-
    значенных для представления знаний в широком смысле и онтологий в частности.]
  • [Большинство современных многоагентных систем предполагает, что взаимодействие агентов осу-
    ществляется путем обмена сообщениями непосредственно от агента к агенту.]
  • [Логический уровень взаимодействия агентов жестко привязан к физическому уровню реализации
    многоагентной системы.]
  • [Среда, с которой взаимодействуют агенты, уточняется отдельно разработчиком для каждой много-
    агентной системы, что приводит к существенным накладным расходам и несовместимости таких
    многоагентных систем.]
  }
⇒ принципы устранения*:
  {
  • [Коммуникацию агентов предлагается осуществлять путем спецификации (в общей памяти
    компьютерной системы) действий (процессов), выполняемых агентами и направленных на
    решение задач.]
    ⇒ детализация*:
      [Коммуникацию агентов предлагается осуществлять по принципу “доски объявлений”,
      однако в отличие от классического подхода в роли сообщений выступают специфи-
      кации в общей семантической памяти выполняемых агентами действий (процессов),
      направленных на решение каких-либо задач, а в роли среды коммуникации выступает
      сама эта семантическая память. Такой подход позволяет:

```

- исключить необходимость разработки специализированного языка для обмена сообщениями;
- обеспечить "обезличенность" общения, т. е. каждый из агентов в общем случае не знает, какие еще агенты есть в системе, кем сформулирован и кому адресован тот или иной запрос. Таким образом, добавление или удаление агентов в систему не приводит к изменениям в других агентах, что обеспечивает модифицируемость всей системы;
- агентам, в том числе конечному пользователю, формулировать задачи в декларативном ключе, т. е. не указывать для каждой задачи способ ее решения. Таким образом, агенту заранее не нужно знать, каким образом система решит ту или иную задачу, достаточно лишь специфицировать конечный результат;
- сделать средства коммуникации агентов и синхронизации их деятельности более понятными разработчику и пользователю системы, не требующими изучения специальных низкоуровневых типов данных и форматов сообщений. Таким образом повышается доступность предлагаемых решений широкому кругу разработчиков.

Следует отметить, что такой подход позволяет при необходимости организовать обмен сообщениями между агентами напрямую и, таким образом, может являться основой для моделирования многоагентных систем, предполагающих другие способы взаимодействия между агентами.]

- [В роли внешней среды для агентов выступает та же общая память, в которой формулируются задачи и посредством которой осуществляется взаимодействие агентов. Такой подход обеспечивает унификацию среды для различных систем агентов, что, в свою очередь, обеспечивает их совместимость.]
- [Спецификация каждого агента описывается средствами языка представления знаний в той же памяти, что позволяет:
 - минимизировать число специализированных средств, необходимых для спецификации агентов, как языковых, так и инструментальных;
 - с одной стороны – минимизировать необходимую в общем случае спецификацию агента, которая включает условие его инициирования и программу, описывающую алгоритм работы агента, с другой стороны – обеспечить возможность произвольного расширения спецификации для каждого конкретного случая, в том числе возможность реализации различных современных моделей спецификации агента.
-]
- [Синхронизацию деятельности агентов предполагается осуществлять на уровне выполняемых ими процессов, направленных на решение тех или иных задач в общей семантической памяти. Таким образом, каждый агент трактуется как некий абстрактный процессор, способный решать задачи определенного класса. При таком подходе необходимо решить задачу обеспечения взаимодействия параллельных асинхронных процессов в общей семантической памяти, для решения которой можно заимствовать и адаптировать решения, применяемые в традиционной линейной памяти. При этом вводится дополнительный класс агентов – метаагенты, задачей которых является решение возникающих проблемных ситуаций, таких как взаимоблокировки]
- [Каждый информационный процесс в любой момент времени имеет ассоциативный доступ к необходимым фрагментам базы знаний, хранящейся в семантической памяти, за исключением фрагментов, заблокированных другими процессами в соответствии с соответствующим механизмом синхронизации. Таким образом, с одной стороны, исключается необходимость хранения каждым агентом информации о внешней среде, с другой стороны, каждый агент, как и в классических многоагентных системах, обладает только частью всей информации, необходимой для решения задачи.
Важно отметить, что в общем случае невозможно априори предсказать, какие именно знания, модели и способы решения задач понадобятся системе для решения конкретной задачи. В связи с этим необходимо обеспечить, с одной стороны, возможность доступа ко всем необходимым фрагментам базы знаний (в пределе – ко всей базе знаний), с другой стороны – иметь возможность локализовать область поиска пути решения задачи, например, рамками одной *предметной области*.
Каждый из агентов обладает набором ключевых элементов (как правило, понятий), которые он использует в качестве отправных точек при ассоциативном поиске в рамках базы знаний. Набор таких элементов для каждого агента уточняется на этапах проектирования многоагентной системы в соответствии с рассматриваемой ниже методикой. Уменьшение числа ключевых

элементов агента делает его более универсальным, однако снижает эффективность его работы за счет необходимости выполнения дополнительных операций.]

}

⇒

*примечание**:

[Предлагаемый подход позволяет рассматривать решатель задач как иерархическую систему. Некий целостный коллектив агентов, реализующий какую-либо подсистему решателя (например, машину дедуктивного вывода, подсистему верификации базы знаний и т. д.), может рассматриваться как единый неатомарный агент, поскольку коллективы агентов и отдельные агенты работают в соответствии с одними и теми же принципами.]

совместимость моделей решения задач*

⇒ *примечание**:

[**совместимость моделей решения задач*** – это возможность одновременного использования разными моделями решения задач одних и тех же информационных ресурсов.]

⇒ *принципы реализации**:

- { • [Вся информация, хранимая в памяти каждой ostis-системы и используемая *решателем задач* (как собственно обрабатываемая информация, так и хранимые в памяти интерпретируемые методы, например, различного вида программы), записывается в форме смыслового представления этой информации]
- [Собственно решение каждой задачи осуществляется коллективом агентов, работающих над общей для них смысловой (семантической) памятью и выполняющих интерпретацию хранимых в этой же памяти *методов*.]

}

интеграция решателей задач

С *процесс*

⇒ *алгоритм реализации**:

- { • [Объединение множества методов первого решателя и множества методов второго решателя]
- [Интеграция множества методов первого решателя и множества методов второго решателя путем взаимного погружения соответствующих информационных конструкций друг в друга, т.е. путем склеивания синонимов, а также путем выравнивания используемых ими понятий.]
- [Объединение множества агентов, входящих в состав первого решателя, со множеством агентов, входящих во второй решатель задач.]

}

⇒ *пояснение**:

[Таким образом, унификация моделей решения задач путем приведения этих моделей к виду семантических моделей (т. е. моделей обработки информации, представленной в смысловой форме) повышает уровень совместимости этих моделей благодаря наличию прозрачной процедуры интеграции информации, представленной в смысловой форме, и тривиальной процедуры объединения множеств *агентов*. Простота процедуры объединения множеств *агентов*, соответствующих разным решателям задач, обусловлена тем, что непосредственного взаимодействия между этими агентами нет, а инициирование каждого из них определяется им самим, а также текущим состоянием хранимой в памяти информации.]

}

/* Завершили Раздел “Предметная область и онтология многоагентных моделей решения задач, основанных на смысловом представлении информации” */

```

/* Раздел *****/
Предметная область и онтология логико-семантических
моделей интерфейсов компьютерных систем, основанных на
смысловом представлении информации
⊃=
{
⇒ соавтор*:
   Садовский М.Е.

Предметная область логико-семантических моделей интерфейсов компьютерных систем, основанных
на смысловом представлении информации
∈ предметная область
∋ максимальный класс объектов исследования1:
  подход к построению пользовательского интерфейса
∋ класс объектов исследования1:
  • подход к построению пользовательского интерфейса на основе специализированных языков
    описания
  • контекстно-зависимый подход к построению пользовательского интерфейса
  • подход к построению пользовательского интерфейса на основе данных
  • онтологический подход к построению пользовательского интерфейса
  • онтологический подход к построению пользовательского интерфейса на основе
    логико-семантической модели

подход к построению пользовательского интерфейса
⊃ подход к построению пользовательского интерфейса на основе специализированных языков описания
⊃ контекстно-зависимый подход к построению пользовательского интерфейса
⊃ подход к построению пользовательского интерфейса на основе данных
⊃ онтологический подход к построению пользовательского интерфейса
  ⊃ онтологический подход к построению пользовательского интерфейса на основе
    логико-семантической модели

подход к построению пользовательского интерфейса на основе специализированных языков описания
⇒ пояснение*:
  [подход на основе специализированных языков описания предполагает представление конкретного поль-
  зовательского интерфейса в платформенно независимом виде. В качестве примеров языков описания
  интерфейса можно привести UIML (Abrams.M..UIML-AIXUIL-1995cm), UsiXML (Limbourg.Q..UsiXML-
  AUIDLSMLOI-2004cm), XForms (XForms-эл) и FXML (FXML-эл). Ключевой идеей представленных языков
  является создание модели диалогов и форм интерфейса в независимом от используемой технологии виде,
  описание визуальных элементов, а также взаимосвязей между ними и их свойств для создания конкретного
  пользовательского интерфейса.]
⇒ недостатки современного состояния*:
  {
  • [как правило, спецификация модели интерфейса является неполной, что приводит к сложности
    автоматизации процесса генерации пользовательского интерфейса]
  • [как правило, созданные модели специфичны для конкретной платформы либо конкретной реализации
    пользовательского интерфейса, что препятствует их повторному использованию для других целей.]
  • [решения, которые предлагают платформенно независимое описание, позволяют генерировать лишь
    простые ограниченные по функционалу пользовательские интерфейсы (приложения-опросники,
    диаграммы и т.д.).]
  }

контекстно-зависимый подход к построению пользовательского интерфейса
⇒ пояснение*:
  [Контекстно-зависимый подход интегрирует использование структурного описания интерфейса на основе
  языков описания с поведенческой спецификацией, то есть генерация интерфейса основывается на действиях
  пользователя. В рамках подхода специфицируются переходы между различными видами конкретного
  пользовательского интерфейса. В качестве примеров языков, реализующих идеи такого подхода можно
  привести CAP3 (Van den Bergh.J..CAP3-CSAUIS-2011cm) и MARIA (Paternò.F..MARIA-AUDMALLFSOAIUE-
  2009cm).]

```

подход к построению пользовательского интерфейса на основе данных⇒ *пояснение**:

[подход на основе данных или модели ориентированный подход использует модель предметной области в качестве основы для создания пользовательских интерфейсов. Указанный подход реализуется в таких проектах, как JANUS (*Balzert.H..The-JANUS-ADEGMTUI-1996cm*) и Mecano (*Puerta.A..Beyond-DMFAUIG-1994cm*).]

онтологический подход к построению пользовательского интерфейса⇒ *пояснение**:

[существующие онтологические подходы как правило основаны на представленных ранее подходах и используют онтологии в качестве способа представления информации о конкретном пользовательском интерфейсе. Например, по аналогии с подходом на основе специализированных языков описания, был предложен фреймворк (*Liu.B..Deriving-UIFOAMBA-2005cm*), использующий онтологию для описания пользовательского интерфейса на основе понятий, хранящихся в базе знаний. По аналогии с контекстно-зависимым подходом в рамках работы *Gaulke.W..Using-POTLMDUIG-2015cm* используется модель предметной области совместно с моделью пользовательского интерфейса, ассоциированная с онтологией действий. Проект ActiveRaUL (*Butt.A..ActiveRaUL-AWFBUITCAMRD-2013cm*) совмещает UIML с модели ориентированным подходом. В рамках данного проекта онтологическая модель предметной области сопоставляется с онтологическим представлением пользовательского интерфейса. Подход, предложенный в работе *Hitz.M..UsingAOFTAGOUIFDBA-2016cm*, совмещает данные приложения с онтологией пользовательского интерфейса для создания единого описания в базе знаний с целью последующей автоматической генерации различных вариантов интерфейса для приложений-опросников с готовыми сценариями взаимодействия с пользователем. Следует также отметить работы *Грибова.В.В..АвтоматизацияРПИСДД-2011cm* и *Грибова.В.В..ГенераторКПИУО-2005cm* в рамках которых предложена концепция, позволяющая объединить однородную по содержанию информацию в компоненты модели интерфейса, освободить разработчика интерфейса от кодирования и формировать информацию для каждого компонента модели интерфейса с помощью редакторов, управляемых соответствующими моделями онтологий.]

⇒ *недостатки современного состояния**:

- { • [актуальна проблема совместимости различных онтологий в рамках единой системы]
- [отсутствие способности адаптироваться к запросам пользователя и анализировать его действия для самостоятельного совершенствования.]

⇒ *достоинства**:

- { • [позволяет интегрировать ранее предложенные подходы за счет единого способа представления знаний.]
- [позволяет создать наиболее полное описание различных аспектов пользовательского интерфейса.]
- [упрощает повторное использование интерфейса.]

онтологический подход к построению пользовательского интерфейса на основе логико-семантической модели⇒ *примечание**:

[для проектирования пользовательских интерфейсов предлагается использовать **онтологический подход к построению пользовательского интерфейса на основе логико-семантической модели**, обладающий рядом важных достоинств.]

⇒ *достоинства**:

- { • [возможность переноса пользовательских интерфейсов с одной платформы реализации на другую.]
- [наличие общих принципов построения пользовательских интерфейсов, что позволяет повторно использовать уже разработанные компоненты и снижает сроки обучения пользователя новым для него пользовательским интерфейсам.]
- [возможность модификации пользовательского интерфейса в процессе работы.]
- [возможность гибкой адаптации пользовательского интерфейса под нужды конкретного пользователя.]
-

⇒ *пояснение**:

[подход предполагает создание полной семантической модели интерфейса, содержащей "лексическое" описание интерфейса (описание компонентов, из которых формируется интерфейс), "синтаксическое" описание интерфейса (правила формирования корректного и полного интерфейса из его компонентов), но также и его семантическое описание (знание о том, знаком какой сущности является отображаемый

компонент). При этом семантическое описание также включает всебя назначение, область применения компонентов интерфейса, описание интерфейсной деятельности пользователя.]

⇒

*основные принципы**:

- { • [пользовательский интерфейс представляет собой специализированную ostis-систему, ориентированную на решение интерфейсных задач, и имеющую в составе базу знаний и машину обработки знаний пользовательского интерфейса, что даёт возможность пользователю адресовать пользовательскому интерфейсу различного рода вопросы]
- [используется онтологический подход к проектированию пользовательского интерфейса, что способствует чёткому разделению деятельности различных разработчиков пользовательских интерфейсов, а также унификации принципов проектирования]
- [используется SC-код в качестве формального языка внутреннего представления знаний (онтологий, предметных областей и др.), благодаря чему обеспечивается легкость интерпретации этих знаний и системой, и человеком - пользователем или разработчиком, а также однозначность восприятия этой информации ими]
- [средствами SC-кода с помощью соответствующих онтологий описываются синтаксис и семантика всевозможных используемых внешних языков]
- [трансляции с внутреннего языка на внешний и обратно организовываются так, чтобы механизмы трансляции не зависели от внешнего языка, для реализации нового специализированного языка в таком случае необходимо будет только описать его синтаксис и семантику, универсальная же модель трансляции не будет зависеть от данного описания]
- [каждый элемент управления пользовательского интерфейса является внешним отображением некоторого sc-элемента, хранящегося в семантической памяти (sc-памяти), что позволяет использовать их в качестве аргументов пользовательских команд и правильно трактовать прагматику и семантику объектов интерфейсной деятельности]
- [предполагается выбор стилей визуализации, осуществляемый в зависимости от вида отображаемых знаний (например, использование различных элементов визуализации для одних видов знаний и других - для других), что позволит пользователю быстрее обучаться новым специализированным языкам, а также сделать простым и понятным отображение знаний]
- [модель пользовательского интерфейса строится независимо от реализации платформы интерпретации такой модели, что позволяет легко переносить разработанную модель на разные платформы.]

}

/ Завершили Раздел “Предметная область и онтология логико-семантических моделей интерфейсов компьютерных систем, основанных на смысловом представлении информации” */*

/* Раздел *****/

Предметная область и онтология знаний и баз знаний ostis-систем

⊃=

{

⇒

дочерний раздел*:

- **Предметная область и онтология множеств**
:= [Предметная область и онтология знаний о множествах]
⇒ **примечание*:**
[знания о множествах являются частным видом знаний и, следовательно, общие свойства сущностей, описываемых знаниями, могут наследоваться Предметной областью и онтологией множеств]
- **Предметная область и онтология связок и отношений**
- **Предметная область и онтология параметров, величин и шкал**
- **Предметная область и онтология чисел и числовых структур**
- **Предметная область и онтология структур**
- **Предметная область и онтология темпоральных сущностей**
- **Предметная область и онтология темпоральных сущностей баз знаний ostis-систем**
- **Предметная область и онтология семантических окрестностей**
- **Предметная область и онтология предметных областей**
- **Предметная область и онтология онтологий**
- **Предметная область и онтология логических формул, высказываний и формальных теорий**
- **Предметная область и онтология внешних информационных конструкций и файлов ostis-систем**
- **Глобальная предметная область действий и задач и соответствующая ей онтология методов и технологий**

Предметная область знаний и баз знаний ostis-систем

∈ предметная область

∃ максимальный класс объектов исследования':

знание

∃ исследуемый класс классов':

- вид знаний
- отношение, заданное на множестве знаний

знание

:= [синтаксически корректная (для соответствующего языка) и семантически целостная информационная конструкция]

С информационная конструкция

∈ класс объектов исследования':

Предметная область и онтология информационных конструкций и языков

⇒ покрытие*:

вид знаний

:= [Множество всевозможных видов знаний]

⇒ примечание*:

[Тот факт, что семейство *видов знаний* является *покрытием* Множества всевозможных знаний, означает то, что каждое знание принадлежит по крайней мере одному выделенному нами *виду знаний*]

вид знаний

∃ спецификация

:= [описание заданной сущности]

⊃ спецификация материальной сущности

⊃ спецификация обратной сущности, не являющейся множеством

⊃ спецификация геометрической точки

⊃ спецификация числа

⊃ спецификация множества

⊃ спецификация связи

⊃ спецификация структуры

⊃ спецификация класса

	<ul style="list-style-type: none"> ⊃ спецификация класса сущностей, не являющихся множествами ⊃ спецификация отношения := [спецификация класса связей (связок)] ⊃ спецификация класса классов <ul style="list-style-type: none"> ⊃ спецификация параметра ⊃ спецификация класса структур ⊃ спецификация понятий <ul style="list-style-type: none"> ⊃ пояснение ⊃ определение ⊃ утверждение := [утверждение, описывающее свойства экземпляров (элементов) специфицируемого понятия] := [закономерность]
	<ul style="list-style-type: none"> ⊃ семантическая окрестность ⊃ однозначная спецификация ⊃ сравнительный анализ ⊃ достоинства ⊃ недостатки ⊃ структура специфицируемой сущности ⊃ принципы, лежащие в основе ⊃ обоснование предлагаемого решения := [аргументация предлагаемого решения]
Э	сравнение
Э	высказывание <ul style="list-style-type: none"> ⊃ фактографическое высказывание ⊃ закономерность
Э	формальная теория
Э	предметная область
Э	предметная область и онтология <ul style="list-style-type: none"> := [предметная область и её онтология] := [предметная область и соответствующая ей объединенная онтология]
Э	метазнание <ul style="list-style-type: none"> := [спецификация знания] ⊃ аннотация ⊃ введение ⊃ предисловие ⊃ заключение ⊃ онтология <ul style="list-style-type: none"> ⊃ онтология предметной области <ul style="list-style-type: none"> ⊃ структурная онтология предметной области ⊃ теоретико-множественная онтология предметной области ⊃ логическая онтология предметной области ⊃ терминологическая онтология предметной области ⊃ объединенная онтология предметной области
Э	задача <ul style="list-style-type: none"> := [спецификация действия]
Э	план
Э	протокол
Э	результативная часть протокола
Э	метод
Э	технология
Э	история использования предметной области и её онтологии по решению информационных задач
Э	история использования предметной области и её онтологии по решению задач во внешней среде
Э	история эволюции предметной области и её онтологии
Э	база знаний <ul style="list-style-type: none"> := [совокупность знаний, хранимых в памяти интеллектуальной компьютерной системы и достаточных для того, чтобы указанная система удовлетворяла соответствующим предъявляемым к ней требованиям (в частности, чтобы она имела соответствующий уровень интеллекта)] := [систематизированная совокупность знаний, хранимая в памяти интеллектуальной компьютерной системы и достаточная для обеспечения целенаправленного (целесообразного, адекватного) функ-

ционирования (поведения) этой системы как в своей внешней среде, так и в своей внутренней среде (в собственной базе знаний)]

⇒ *обобщенная декомпозиция**:

- { • *согласованная часть базы знаний*
:= [часть базы знаний, признанная коллективом авторов на текущий момент]
- *история эксплуатации базы знаний*
- *история эволюции базы знаний*
- *план эволюции базы знаний*
:= [система специфицированных и согласованных действий авторов базы знаний, направленных на повышение её качества]

⇒ *примечание**:

[Основным факторами, определяющими качество интеллектуальной компьютерной системы, являются:

- качественная структуризация (систематизация) и стратификация базы знаний интеллектуальной компьютерной системы, а также
- систематизация и стратификация деятельности, которая осуществляется интеллектуальной компьютерной системой и спецификация которой является важнейшей частью базы знаний этой системы (Смотрите Раздел *Глобальная предметная область действий и задач и соответствующая ей онтология методов и технологий*).

]

⇒ *примечание**:
[Даже небольшой перечень *видов знаний* свидетельствует об огромном многообразии *видов знаний*]

знание

⇒ *разбиение**:

- { • *декларативное знание*
:= [знание, имеющее только денотационную семантику, которая представляется в виде семантической спецификации системы понятий, используемых в этом знании]
- *процедурное знание*
:= [знание, имеющее не только денотационную семантику, но и операционную семантику, которая представляется в виде семейства спецификаций агентов, осуществляющих интерпретацию процедурного знания, направленную на решение некоторой инициированной задачи]
:= [функционально интерпретируемое знание, обеспечивающее решение либо конкретной задачи, либо некоторого множества иницируемых задач]
- ⊃ *задача*
:= [формулировка конкретной задачи]
⊃ *декларативная формулировка задачи*
⊃ *процедурная формулировка задачи*
- ⊃ *план*
:= [план решения конкретной задачи]
:= [контекст конкретной задачи, предоставляющий всю информацию для решения всех подзадач для указанной конкретной задачи]
:= [описание системы подзадач некоторой задачи]
- ⊃ *метод*
:= [обобщенное описание плана решения любой задачи из некоторого заданного класса задач]
- ⊃ *навык*
:= [метод, детализированный до уровня элементарных подзадач]

отношение, заданное на множестве знаний

⇒ *дочернее знание**
:= [знание, которое от "материнского" знания наследует все описанные там свойства объектов исследования]

⇒ *примечание**:

[Факт наследования свойств описываемых объектов от "материнского" знания подчеркивается использованием прилагательного "дочернее" в sc-идентификаторе данного отношения, заданного на множестве знаний]

⊃ *дочерний раздел**

:= [частный раздел*]

⊃ *дочерняя предметная область и онтология**

Э *спецификация**

:= [быть знанием, которое является спецификацией (описанием) заданной сущности]

⇒ *примечание**:

[специфицируемой сущностью может быть сущность любого вида, в том числе, и другое знание]

Э *онтология**

:= [быть семантической спецификацией заданного знания*]

Э *семантическая эквивалентность**

Э *следовательно**

:= [логическое следствие*]

Э *логическая эквивалентность**

} /* Завершили Раздел "Предметная область и онтология знаний и баз знаний *ostis-систем*" */

/* Раздел *****/	
Предметная область и онтология множеств	
$\supset=$	
{	
Предметная область множеств	
$:=$	[Теоретико-множественная предметная область]
$:=$	[Предметная область теории множеств]
$:=$	[Предметная область, объектами исследования которой являются множества]
\in	предметная область
\ni	максимальный класс объектов исследования ¹ : множество
\ni	класс объектов исследования ¹ :
	<ul style="list-style-type: none"> • конечное множество • бесконечное множество • счетное множество • несчетное множество • множество без кратных элементов • мультимножество • кратность принадлежности • класс • класс первичных <i>sc</i>-элементов • класс множеств • класс структур • класс классов • нечеткое множество • четкое множество • множество первичных сущностей • семейство множеств • нереклексивное множество • рефлексивное множество • множество первичных сущностей и множеств • сформированное множество • несформированное множество • пустое множество • синглетон • пара • пара разных элементов • пара-мультимножество • тройка • кортеж • декартово произведение • булеан • мощность множества
\ni	исследуемое отношение ¹ :
	<ul style="list-style-type: none"> • принадлежность* • пример¹ • включение* • строгое включение* • объединение* • разбиение* • пересечение* • пара пересекающихся множеств* • попарно пересекающиеся множества* • пересекающиеся множества* • пара непересекающихся множеств* • попарно непересекающиеся множества* • непересекающиеся множества* • разность множеств* • симметрическая разность множеств*

- *декартово произведение**
- *семейство подмножеств**
- *булеан**
- *равенство множеств**

множество

:= [множество sc-элементов]

:= [sc-множество]

:= [множество знаков]

:= [множество знаков описываемых сущностей]

:= [семантически нормализованное множество]

:= [sc-знак множества sc-элементов]

:= [sc-знак множества sc-знаков]

:= [sc-текст]

:= [текст SC-кода]

:= [SC-код]

⇒ разбиение*:

- *конечное множество*
- *бесконечное множество*

⇒ разбиение*:

- *множество без кратных элементов*
- *мультимножество*

⇒ разбиение*:

- *связка*
- *класс*

:= [sc-элемент, обозначающий класс sc-элементов]

:= [sc-знак множества sc-элементов, эквивалентных в том или ином смысле]

- *структура*

:= [sc-знак множества sc-элементов, в состав которого входят sc-связки или структуры, связывающие эти sc-элементы]

⇒ разбиение*:

- *четкое множество*
- *нечеткое множество*

⇒ разбиение*:

- *множество первичных сущностей*
- *множество множеств*
- *множество первичных сущностей и множеств*

⇒ разбиение*:

- *рефлексивное множество*
- *нерефлексивное множество*

⇒ разбиение*:

- *сформированное множество*
- *несформированное множество*

⇒ разбиение*:

- *кортеж*
- *неориентированное множество*

∪ *пустое множество*

∪ *синглетон*

∪ *пара*

∪ *тройка*

⇒ *пояснение**:

[Под **множеством** понимается соединение в некое целое M определённых хорошо различимых предметов m нашего созерцания или нашего мышления (которые будут называться «элементами» множества M).

множество – мысленная сущность, которая связывает одну или несколько сущностей в целое.

Более формально **множество** – это абстрактный математический объект, состоящий из элементов. Связь множеств с их элементами задается бинарным ориентированным отношением **принадлежность***.

множество может быть полностью задано следующими тремя способами:

- путем перечисления (явного указания) всех его элементов (очевидно, что таким способом можно задать только конечное множество)
- с помощью определяющего высказывания, содержащего описание общего характеристического свойства, которым обладают все те и только те объекты, которые являются элементами (т.е. принадлежат) задаваемого множества.
- с помощью теоретико-множественных операций, позволяющих однозначно задавать новые множества на основе уже заданных (это операции объединения, пересечения, разности множеств и др.)

Для любого семантически ненормализованного **множества** существует единственное семантически нормализованное **множество**, в котором все элементы, не являющиеся знаками множеств, заменены на знаки множеств.]

принадлежность*

$:=$ [принадлежность элемента множеству*]

$:=$ [отношение принадлежности элемента множеству*]

\in *бинарное отношение*

\in *ориентированное отношение*

\Rightarrow *пояснение**:

[**принадлежность*** – это бинарное ориентированное отношение, каждая связка которого связывает множество с одним из его элементов. Элементами отношения **принадлежность*** по умолчанию являются *позитивные sc -дуги принадлежности.*]

конечное множество

$:=$ [множество с конечным числом элементов]

\Rightarrow *пояснение**:

[**конечное множество** – это **множество**, количество элементов которого конечно, т.е. существует неотрицательное целое число k , равное количеству элементов этого множества.]

бесконечное множество

$:=$ [множество с бесконечным числом элементов]

\Rightarrow *разбиение**:

{ • *счетное множество*
• *несчетное множество*
}

\Rightarrow *пояснение**:

[**бесконечное множество** – это **множество**, в котором для любого натурального числа n найдётся конечное подмножество из n элементов.]

счетное множество

\Rightarrow *пояснение**:

[**счетное множество** – это **бесконечное множество**, для которого существует *взаимно-однозначное соответствие* с натуральным рядом чисел.]

несчетное множество

$:=$ [континуальное множество]

\Rightarrow *пояснение**:

[**несчетное множество** – это **бесконечное множество**, элементы которого невозможно пронумеровать натуральными числами.]

множество без кратных элементов

$:=$ [классическое множество]

$:=$ [канторовское множество]

- := [множество, состоящее из разных элементов]
- := [множество без кратного вхождения элементов]
- := [множество, все элементы которого входят в него однократно]
- := [множество, не имеющее кратного вхождения элементов]
- ⇒ *пояснение**:
[**множество без кратных элементов** - это *множество*, для каждого элемента которого существует только одна пара принадлежности, выходящая из знака этого множества в указанный элемент.]

мультимножество

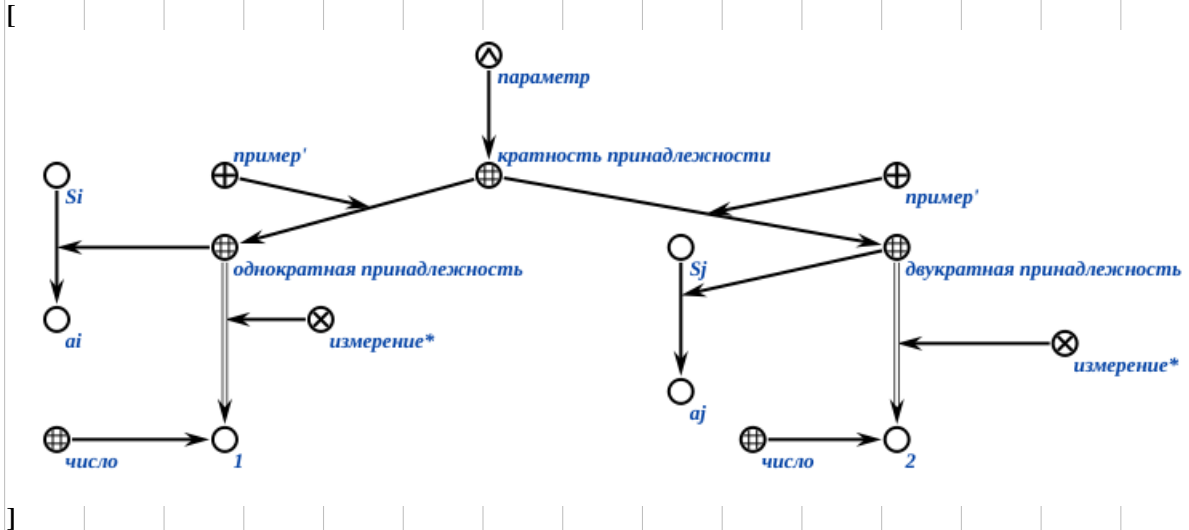
- := [множество, имеющее кратные вхождения хотя бы одного элемента]
- := [множество, по крайней мере один элемент которого входит в его состав многократно]
- ⇒ *пояснение**:
[**мультимножество** - это *множество*, для которого существует хотя бы одна кратная пара принадлежности, выходящая из знака этого множества.]

кратность принадлежности

- := [кратность принадлежности элемента]
- := [кратность вхождения элемента во множество]
- ∈ *параметр*
- ⇒ *пояснение**:
[**кратность принадлежности** - *параметр*, значением которого являются числовые величины, показывающие сколько раз входит тот или иной элемент в рассматриваемое множество. Элементами данного параметра являются классы *позитивных sc-дуг принадлежности*, связывающих данное множество с элементом, кратность вхождения которого в данное множество мы хотим задать.

Таким образом, кратное вхождение элемента в мультимножество может быть задано как явным указанием *позитивных sc-дуг принадлежности* этого элемента данному *множеству*, так и «склеиванием» этих дуг в одну и включением ее в некоторый класс **кратности принадлежности**.]

- ⇒ *описание примера**:



класс

- := [класс sc-элементов]
- ⇒ *разбиение**:
{
 - класс первичных sc-элементов
 - класс множеств
- ⇒ *пояснение**:
[**класс** – множество элементов, обладающих какими-либо явно указываемыми общими свойствами.]

класс первичных sc-элементов

- ⇒ *пояснение**:
[**класс первичных sc-элементов** – класс, элементами которого являются только *sc-элементы*, не являющиеся знаками множеств.]

класс множеств⇒ *разбиение**:

- отношение
- класс структур
- класс классов

⇒ *пояснение**:

[**класс множеств** – класс, элементами которого являются только *sc-элементы*, являющиеся знаками множеств.]

класс структур⇒ *пояснение**:

[**класс структур** – класс, элементами которого являются *структуры*.]

класс классов⇒ *пояснение**:

[**класс классов** – класс, элементами которого являются *классы*.]

нечеткое множество⇒ *пояснение**:

[**нечеткое множество** – это *множество*, которое представляет собой совокупность элементов произвольной природы, относительно которых нельзя точно утверждать – обладают ли эти элементы некоторым характеристическим свойством, которое используется для задания этого нечеткого множества. Принадлежность элементов такому множеству указывается при помощи *нечетких позитивных sc-дуг принадлежности*.]

четкое множество⇒ *пояснение**:

[**четкое множество** – это *множество*, принадлежность элементов которому достоверна и указывается при помощи *четких позитивных sc-дуг принадлежности*.]

множество первичных сущностей⊃ *класс первичных сущностей*⊂ *множество*⇒ *пояснение**:

[**множество первичных сущностей** – это *множество*, элементы которого не являются знаками множеств.]

семейство множеств

:= [множество множеств]

⊃ *класс классов*⇒ *пояснение**:

[**семейство множеств** – это *множество*, элементами которого являются знаки множеств.]

нерефлексивное множество⇒ *пояснение**:

[**нерефлексивное множество** – это *множество*, знак которого не является элементом этого множества]

рефлексивное множество⇒ *пояснение**:

[**рефлексивное множество** – это *множество*, знак которого является элементом этого множества]

множество первичных сущностей и множеств⇒ *пояснение**:

[**множество первичных сущностей и множеств** – это *множество*, элементами которого являются как знаки множеств, так и знаки сущностей, не являющихся множествами.]

сформированное множество∈ *ситуативное множество*⇒ *пояснение**:

[**сформированное множество** - это *множество*, все элементы которого известны и перечислены в данный момент.]

несформированное множество

∈ *ситуативное множество*

⇒ *пояснение**:

[**несформированное множество** - это *множество*, не все элементы которого известны и перечислены в данный момент.]

пустое множество

∈ *мощность множества*

⇒ *пояснение**:

[**пустое множество** – это *множество*, которому не принадлежит ни один элемент.]

синглетон

∈ *мощность множества*

:= [множество мощности 1]

:= [одноэлементное множество]

:= [одномощное множество]

:= [множество, мощность которого равна 1]

:= [множество, имеющее мощность равную единице]

:= [синглетон из sc-элемента]

:= [sc-синглеон]

С *конечное множество*

⇒ *пояснение**:

[**синглетон** – это *множество*, состоящее из одного элемента.

Другими словами - любое множество S_i есть **синглетон** тогда и только тогда, когда существует принадлежность этому множеству, которая совпадает с любой принадлежностью этому множеству.]

пара

∈ *мощность множества*

:= [множество мощности два]

:= [двухэлементное множество]

:= [двумощное множество]

:= [множество, мощность которого равна 2]

:= [sc-пара]

:= [пара sc-элементов]

С *конечное множество*

⇒ *разбиение**:

{ • *пара разных элементов*
• *пара-мультимножество*
}

⇒ *пояснение**:

[**пара** – это *множество*, состоящее из двух элементов.

Другими словами – любое множество есть **пара** тогда и только тогда, когда существуют две различные принадлежности этому множеству такие, что любая принадлежность этому множеству совпадает с одной из них.]

пара разных элементов

:= [канторовская пара]

:= [канторовская пара sc-элементов]

:= [канторовское двумощное множество]

пара-мультимножество

:= [пара-петля]

:= [sc-петля]

$:=$ [двумощное мультимножество]

тройка

\in *мощность множества*

$:=$ [тройка]

$:=$ [sc-тройка]

$:=$ [множество мощности три]

$:=$ [множество, мощность которого равна 3]

\subset *конечное множество*

\Rightarrow *пояснение**:

[**тройка** – это *множество*, состоящее из трех элементов.

Другими словами – любое множество есть **тройка** тогда и только тогда, когда существуют три различные принадлежности этому множеству такие, что любая принадлежность этому множеству совпадает с одной из них.]

кортеж

$:=$ [вектор]

\Rightarrow *пояснение**:

[**кортеж** – это множество, представляющее собой упорядоченный набор элементов, т.е. такое множество, порядок элементов в котором имеет значение. Пары принадлежности элементов **кортежу** могут дополнительно принадлежать каким-либо *ролевым отношениям*, при этом, в рамках каждого **кортежа** должен существовать хотя бы один элемент, роль которого дополнительно уточнена *ролевым отношением*.]

пример'

$:=$ [типичный пример']

$:=$ [типичный экземпляр заданного класса']

\in *ролевое отношение*

\Rightarrow *пояснение**:

[**пример'** – это *ролевое отношение*, связывающее некоторое *множество* с элементом, являющимся примером этого множества.]

включение*

$:=$ [включение множеств*]

$:=$ [быть подмножеством*]

\in *бинарное отношение*

\in *ориентированное отношение*

\in *транзитивное отношение*

\Rightarrow *область определения**:

множество

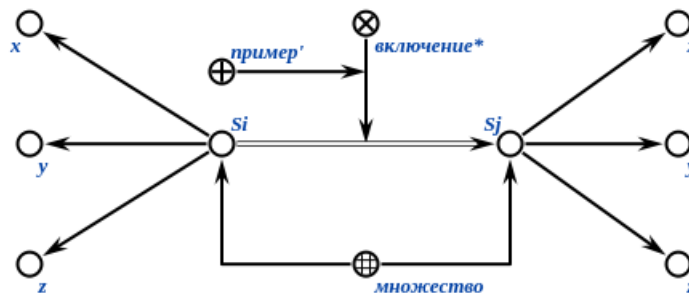
\supset *строгое включение**

\Rightarrow *определение**:

[**включение*** – это бинарное ориентированное отношение, каждая связка которого связывает два множества. Будем говорить, что *Множество S_i включает** в себя *Множество S_j* в том и только том случае, если каждый элемент *Множества S_j* является также и элементом *Множества S_i* .]

\Rightarrow *описание примера**:

[



]

\Rightarrow *пояснение**:

[Множество S_j включается во множество S_i .]

строгое включение*

:= [строгое включение множеств*]

\subset включение*

\in бинарное отношение

\in ориентированное отношение

\Rightarrow область определения*:

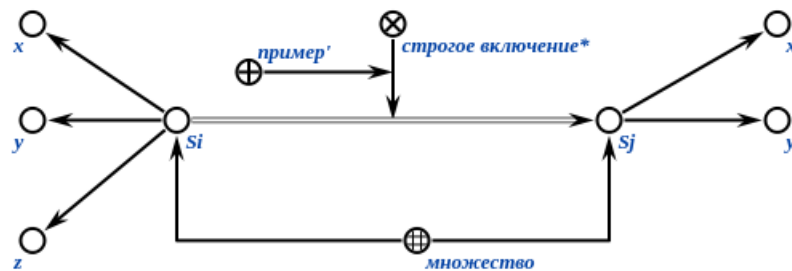
множество

\Rightarrow определение*:

[**строгое включение*** – это бинарное ориентированное отношение, областью определения которого является семейство всевозможных множеств. Будем говорить, что *Множество S_i строго включает** в себя *Множество S_j* в том и только том случае, если каждый элемент *Множество S_j* является также и элементом *Множество S_i* , при этом существует хотя бы один элемент *Множество S_i* , не являющийся элементом *Множество S_j* .]

\Rightarrow описание примера*:

[



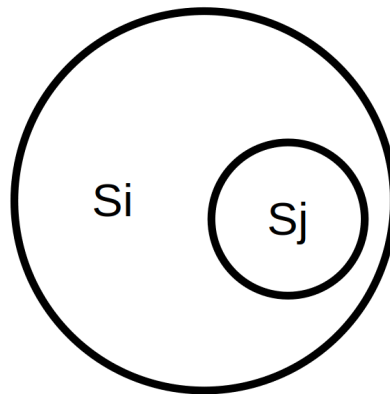
]

\Rightarrow пояснение*:

[Множество S_j строго включается во множество S_i .]

\Rightarrow изображение*:

[



]

объединение*

:= [объединение множеств*]

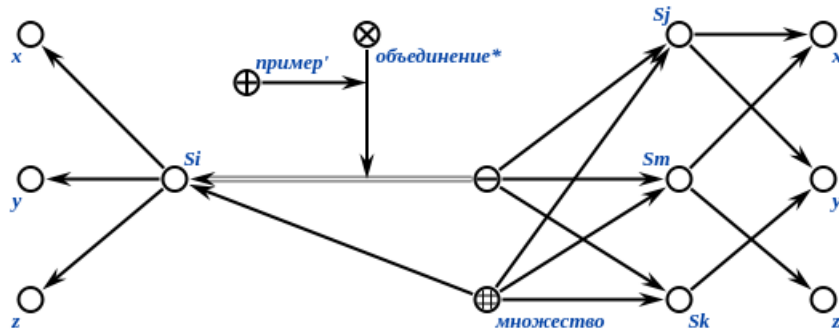
\in квазибинарное отношение

\in ориентированное отношение

\Rightarrow определение*:

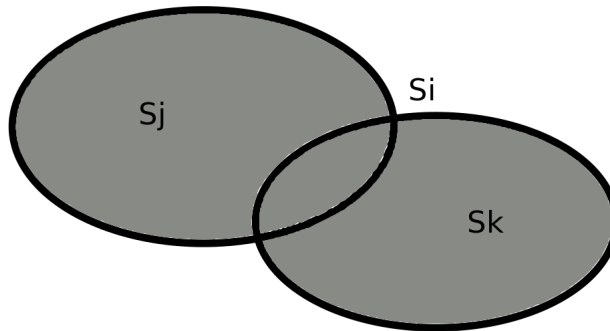
[**объединение*** – это квазибинарное ориентированное отношение, областью определения которого является семейство всевозможных множеств. Будем говорить, что *Множество S_i* является объединением *Множество S_j* и *Множество S_k* тогда и только тогда, когда любой элемент *Множество S_i* является элементом или *Множество S_j* или *Множество S_k* .]

\Rightarrow описание примера*:



⇒ *пояснение**:
 [Множество S_i является объединением множеств S_j , S_k и S_m .]

⇒ *изображение**:



разбиение*

:= [разбиение множества*]

:= [объединение попарно непересекающихся множеств*]

:= [декомпозиция множества*]

∈ *квазибинарное отношение*

∈ *ориентированное отношение*

∈ *отношение декомпозиции*

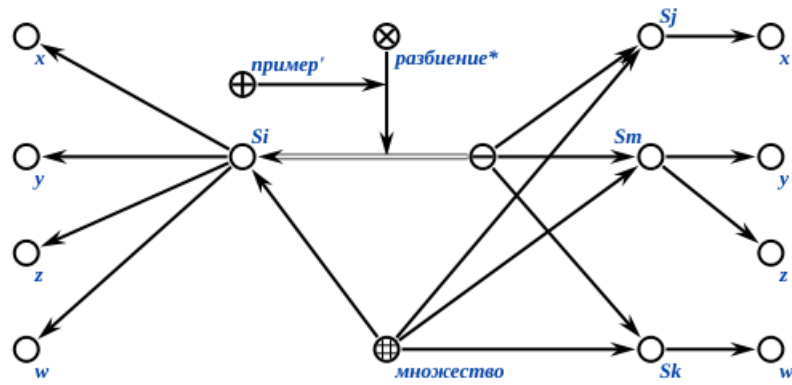
⇒ *определение**:

[**разбиение*** – это *квазибинарное ориентированное отношение*, областью определения которого является семейство всевозможных множеств. В результате разбиения множества получается множество попарно непересекающихся множеств, объединение которых есть исходное множество.

Семейство множеств $\{S_1, \dots, S_n\}$ является разбиением множества S_i в том и только том случае, если:

- семейство $\{S_1, \dots, S_n\}$ является семейством *попарно непересекающихся множеств*;
- семейство $\{S_1, \dots, S_n\}$ является *покрытием* множества S_i (или другими словами, множество S_i является *объединением* множеств, входящих в указанное выше семейство)

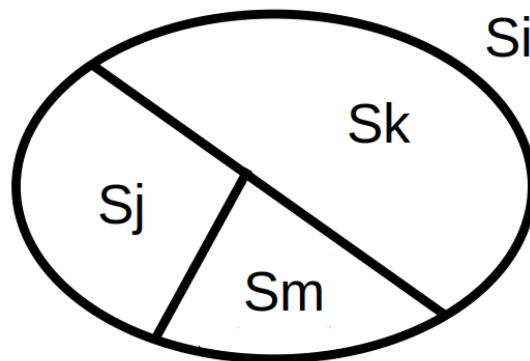
⇒ *описание примера**:



]

⇒ *пояснение**:[Множество S_i разбивается на множества S_j , S_k и S_m .]⇒ *изображение**:

[



]

пересечение*

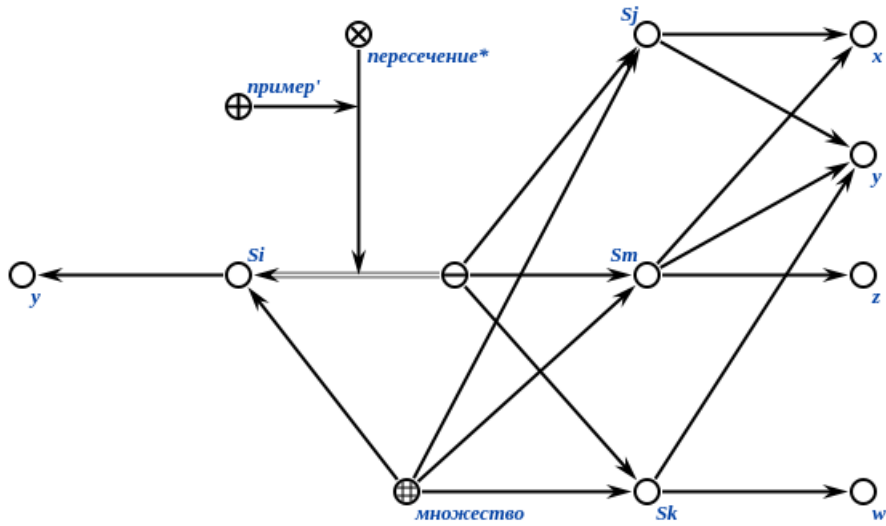
:= [пересечение множеств*]

∈ *квазибинарное отношение*∈ *ориентированное отношение*⇒ *определение**:

[***пересечение**** – это операция над множествами, аргументами которой являются два или большее число множеств, а результатом является множество, элементами которого являются все те и только те сущности, которые одновременно принадлежат каждому множеству, которое входит в семейство аргументов этой операции.]

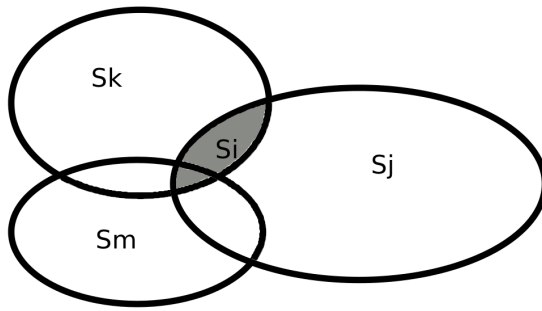
Будем говорить, что *Множество S_i* является пересечением *Множество S_j* и *Множество S_k* тогда и только тогда, когда любой элемент *Множество S_i* является элементом *Множество S_j* и элементом *Множество S_k* .]

⇒ *описание примера**:



] ⇒ *пояснение**:
 [Множество S_i является результатом пересечения множеств S_j, S_k и S_m .]

⇒ *изображение**:



пара пересекающихся множеств*

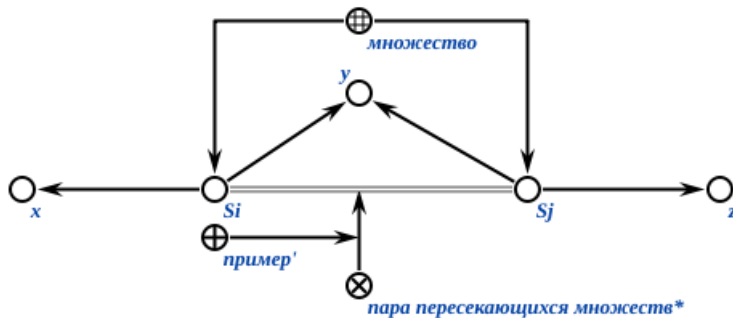
- ∈ бинарное отношение
- ∈ неориентированное отношение
- ⇒ *пояснение**:

[**пара пересекающихся множеств*** – бинарное неориентированное отношение между двумя множествами, имеющими непустое пересечение*.]

⇒ *определение**:

[**пара пересекающихся множеств*** – бинарное неориентированное отношение между двумя множествами, имеющими, по крайней мере, один общий для этих двух множеств элемент.]

⇒ *описание примера**:

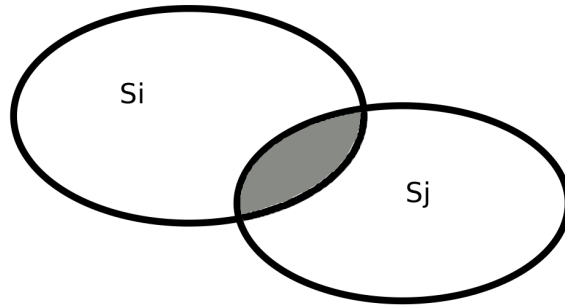


] ⇒ *пояснение**:

[Множество S_i и множество S_j являются парой пересекающихся множеств.]

⇒ изображение*:

[



]

попарно пересекающиеся множества*

:= [семейство попарно пересекающихся множеств*]

\supset пересекающиеся множества*

\in отношение

⇒ определение*:

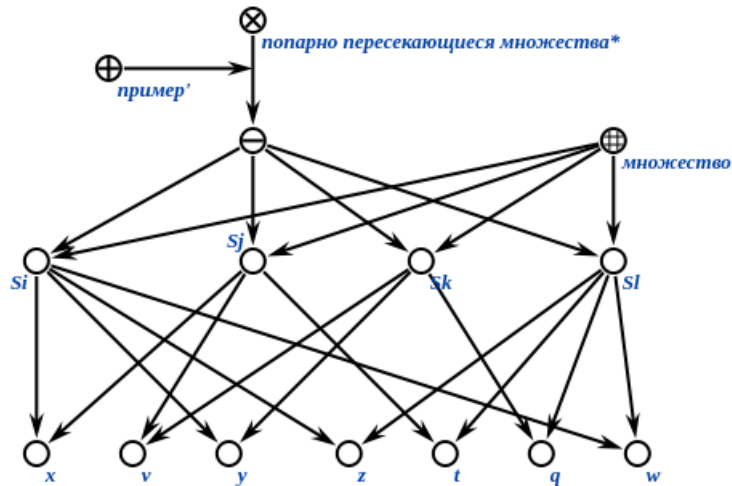
[**попарно пересекающиеся множества*** – семейство множеств, каждая пара которых является парой пересекающихся множеств, т.е. каждая пара которых имеет хотя бы один общий элемент]

⇒ примечание*:

[Не каждое семейство попарно пересекающихся множеств* является семейством пересекающихся множеств*, хотя обратное верно.]

⇒ изображение*:

[

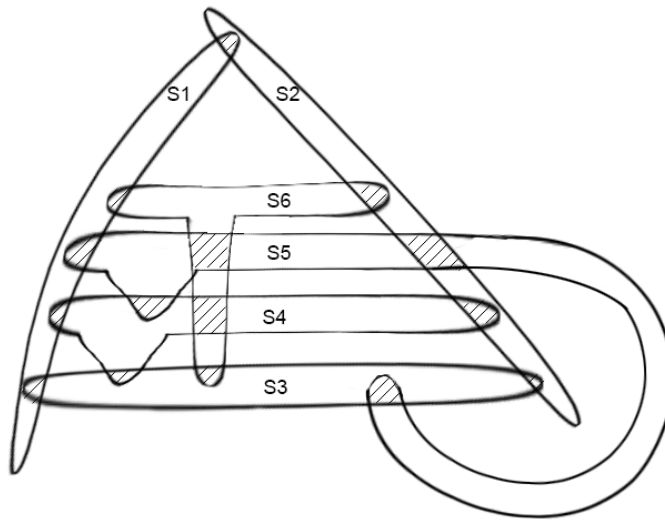


]

⇒ пояснение*:

[Множества S_i , S_j , S_k и S_l являются попарно пересекающимися множествами.]

⇒ изображение*:



пересекающиеся множества*

:= [семейство пересекающихся множеств*]

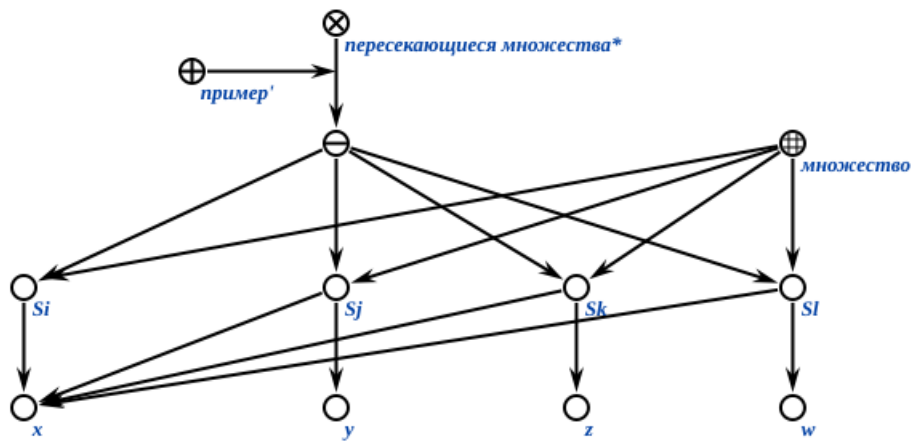
:= [быть семейством пересекающихся множеств*]

:= [семейство множеств, имеющих по крайней мере один элемент, являющийся общим для всех этих множеств*]
*попарно пересекающиеся множества**

⇒ *определение**:

[**пересекающиеся множества*** – это семейство множеств, имеющих по крайней мере один общий для всех этих множеств элемент]

⇒ *описание примера**:



⇒ *пояснение**:

[Множества S_i, S_j, S_k и S_l являются пересекающимися множествами.]

пара непересекающихся множеств*

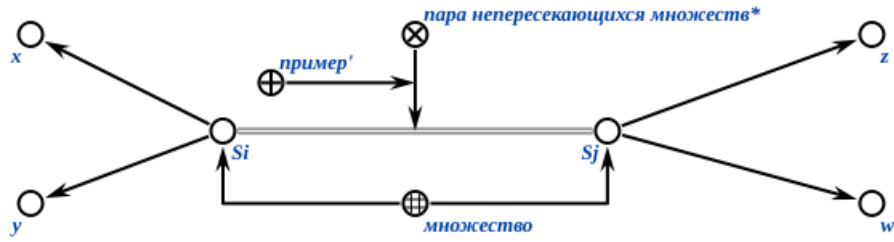
∈ *бинарное отношение*

∈ *неориентированное отношение*

⇒ *определение**:

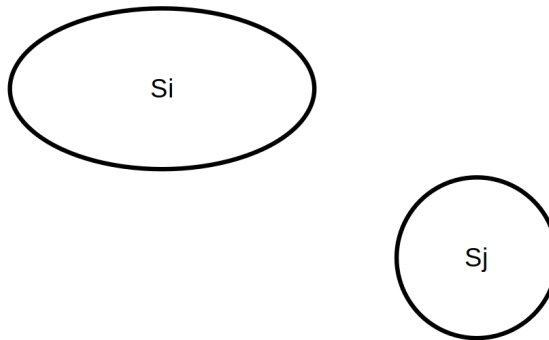
[**пара непересекающихся множеств*** – это бинарное неориентированное отношение между множествами, результатом пересечения* которых есть пустое множество.]

⇒ *описание примера**:



] ⇒ *пояснение**:
 [Множества S_i и S_j являются парой непересекающихся множеств.]

⇒ *изображение**:



попарно непересекающиеся множества*

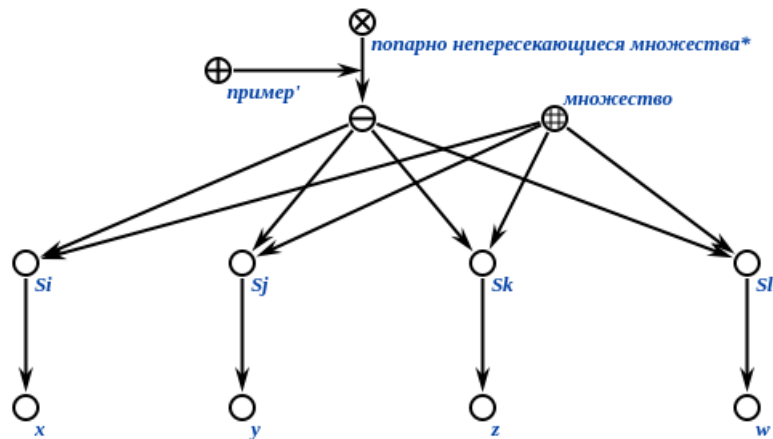
:= [семейство попарно непересекающихся множеств*]

⊂ *непересекающиеся множества**

⇒ *определение**:

[**попарно непересекающиеся множества*** – семейство множеств, каждая пара которых является парой непересекающихся множеств, т.е. каждая пара которых не имеет ни одного общего элемента]

⇒ *изображение**:



] ⇒ *пояснение**:
 [Множества S_i , S_j , S_k и S_l являются попарно непересекающимися множествами.]

непересекающиеся множества*

:= [семейство непересекающихся множеств*]

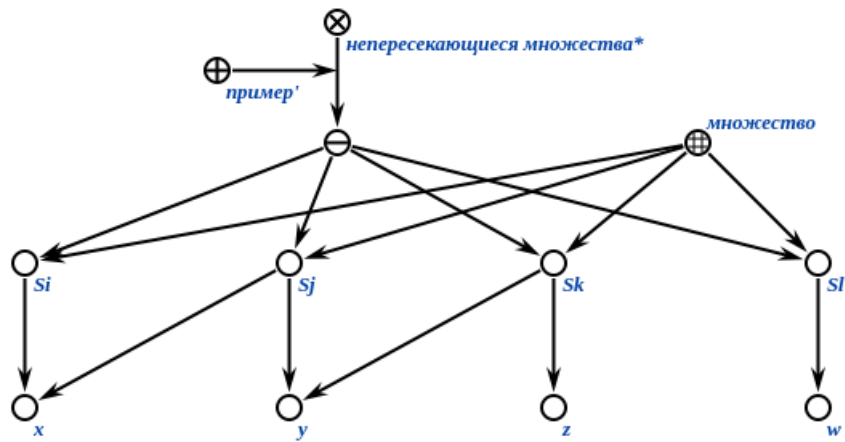
:= [быть семейством непересекающихся множеств*]

⇒ *определение**:

[**непересекающиеся множества*** – это семейство множеств, не имеющих ни одного общего элемента для всех этих множеств]

⇒ **изображение***:

[



]

⇒ **пояснение***:

[Множества S_i , S_j , S_k и S_l являются непересекающимися множествами.]

разность множеств*

∈ **бинарное отношение**

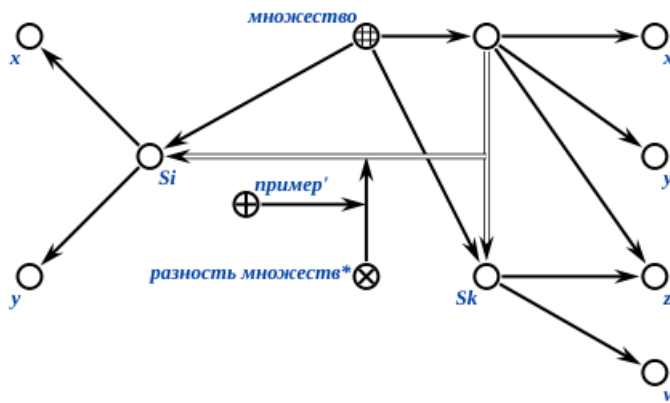
∈ **ориентированное отношение**

⇒ **определение***:

[**разность множеств*** – это бинарное ориентированное отношение, связывающее между собой ориентированную пару, первым элементом которой является уменьшаемое множество, вторым - вычитаемое множество, и множество, являющееся результатом вычитания вычитаемого из уменьшаемого, в которое входят все элементы первого множества, не входящие во второе множество.]

⇒ **описание примера***:

[

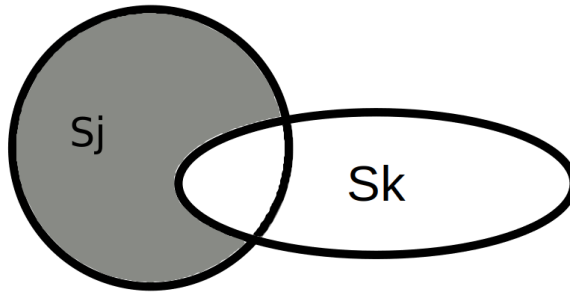


]

⇒ **пояснение***:

[Множество S_i является результатом разности множеств S_j и S_k .]

⇒ **изображение***:

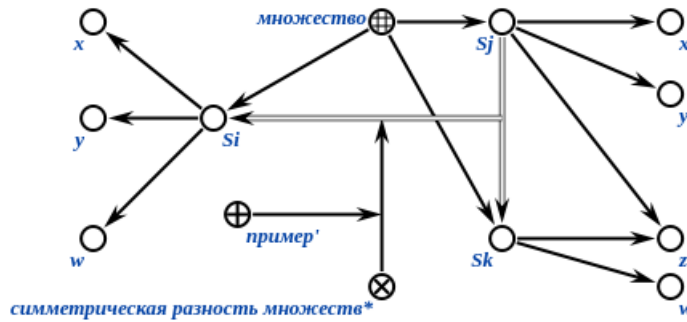


симметрическая разность множеств*

- ∈ бинарное отношение
- ∈ ориентированное отношение
- ⇒ определение*:

[**симметрическая разность множеств*** – это бинарное ориентированное отношение, связывающее между собой пару множеств и множество, являющееся результатом симметрической разности элементов указанной пары. Будем называть *Множество Si* результатом симметрической разности *Множества Sj* и *Множества Sk* тогда и только тогда, когда любой элемент *Множества Si* является или элементом *Множества Sj* или *Множества Sk*, но не является элементом обоих множеств.]

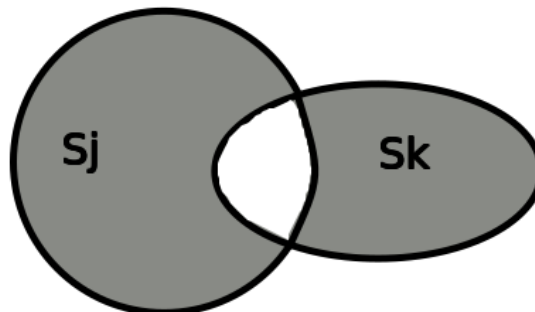
⇒ описание примера*:



⇒ пояснение*:

[Множество *Si* является результатом симметрической разности множеств *Sj* и *Sk*.]

⇒ изображение*:



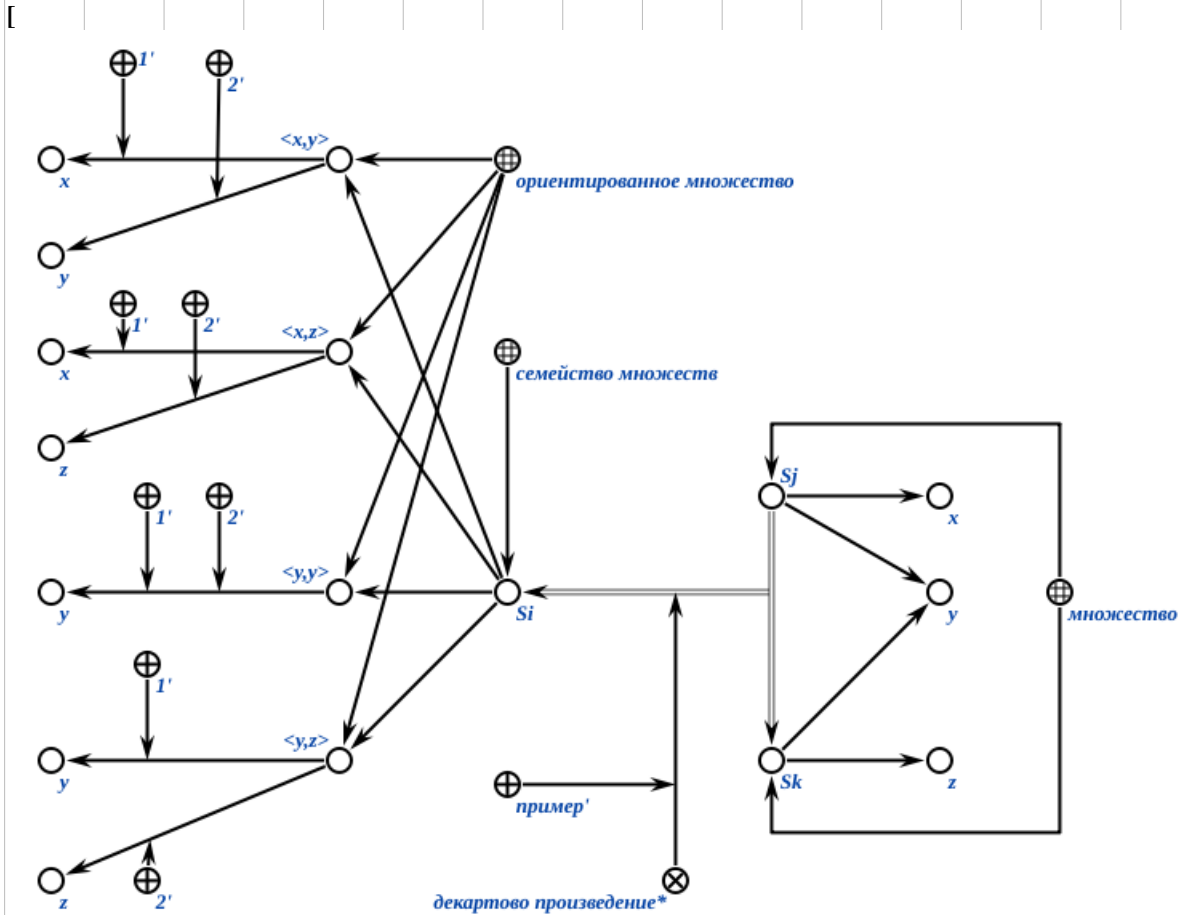
декартово произведение*

- := [декартово произведение множеств*]
- := [прямое произведение множеств*]
- ∈ бинарное отношение
- ∈ ориентированное отношение

⇒ *определение**:

[**декартово произведение*** – это бинарное ориентированное отношение между ориентированной парой множеств и множеством, элементами которого являются всевозможные упорядоченные пары, первыми элементами которых являются элементы первого из указанных множеств, вторыми – элементы второго из указанных множеств.]

⇒ *описание примера**:



] ⇒ *пояснение**:

[Множество S_i является результатом декартова произведения множеств S_j и S_k .]

декартово произведение

:= [второй домен отношения быть декартовым произведением]

⇒ *второй домен**:
декартово произведение*

семейство подмножеств*

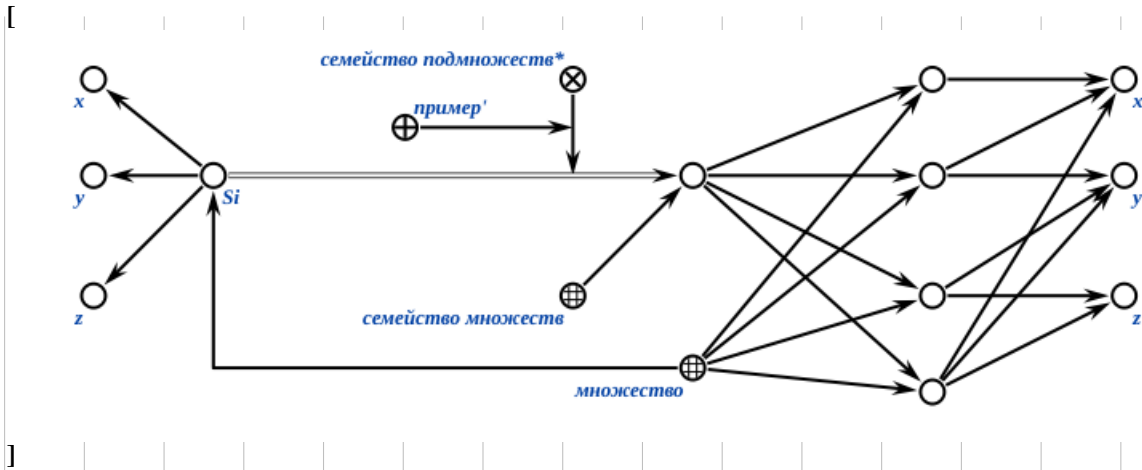
:= [семейство подмножеств заданного множества*]

∈ бинарное отношение
∈ ориентированное отношение
⊃ булеан*

⇒ *определение**:

[**семейство подмножеств*** – это бинарное ориентированное отношение между множеством и некоторым семейством множеств, каждое из которых является подмножеством первого множества.]

⇒ *описание примера**:



булеан*

:= [булеан множества*]

:= [семейство всевозможных подмножеств заданного множества*]

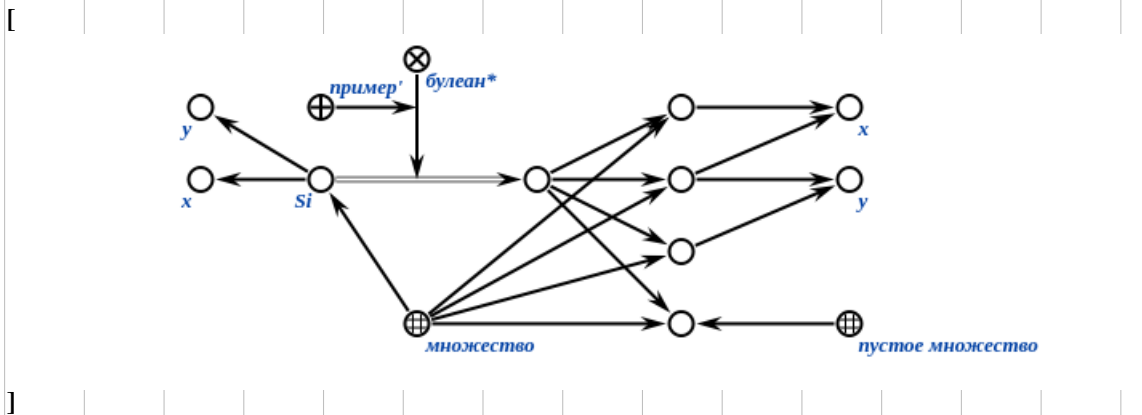
∈ бинарное отношение

∈ ориентированное отношение

⇒ определение*:

[**булеан*** – это бинарное ориентированное отношение между множеством и некоторым семейством множеств, каждое из которых является подмножеством первого множества.]

⇒ описание примера*:



булеан

:= [второй домен отношения быть булеаном]

⇒ второй домен*:

булеан*

мощность множества

:= [кардинальное число]

:= [общее число вхождений элементов в заданное множество]

:= [класс эквивалентности, элементами которого являются знаки всех тех и только тех множеств, которые имеют одинаковую мощность]

:= [класс эквивалентности, соответствующий отношению быть парой множеств, имеющих одинаковую мощность (равномощность множеств)]

:= [величина мощности множеств]

:= [трансфинитное число]

:= [мощность по Кантору]

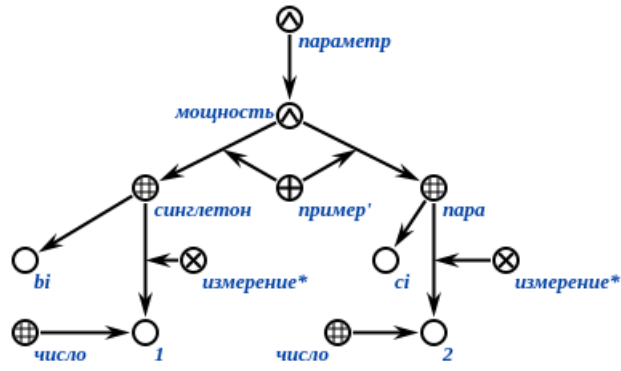
∈ параметр

⇒ пояснение*:

[**мощность множества** – это параметр, элементами которых являются множества, имеющие одинаковое количество элементов. Значением данного параметра является числовая величина, задающая количество элементов, входящих в данный класс множеств, т.е. по сути, количество *позитивных sc-дуг принадлежности*, выходящих из данного множества.]

Для двух множеств, имеющих одинаковую мощность, существует взаимно-однозначное соответствие между ними (между множествами вхождений элементов в эти множества – на случай мультимножеств).]
 ⇒ описание примера*:

[



]

равенство множеств*

- ∈ бинарное отношение
- ∈ неориентированное отношение

:= [быть равными множествами*]

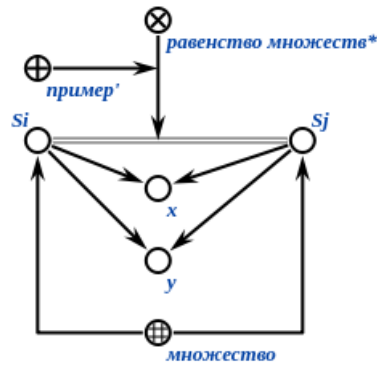
⇒ определение*:

[**равенство множеств*** - бинарное неориентированное отношение, выражающее отношение равенства множеств.

Любые два множества являются равными множествами тогда и только тогда, когда первое является включением второго и второе является включением первого.]

⇒ описание примера*:

[



]

⇒ пояснение*:

[Множество S_i равно множеству S_j .]

} /* Завершили Раздел “Предметная область и онтология множеств” */

/* Раздел *****/

Предметная область и онтология связок и отношений

$\supset=$
{

Предметная область связок и отношений

- ∈ предметная область
- ∃ максимальный класс объектов исследования':
связь
- ∃ класс объектов исследования':
 - бинарная связь
 - sc-коннектор
 - неатомарная бинарная связь
 - небинарная связь
 - неориентированная связь
 - ориентированная связь
 - отношение
 - класс равномоощных связок
 - класс связок разной мощности
 - унарное отношение
 - бинарное отношение
 - квазибинарное отношение
 - тернарное отношение
 - небинарное отношение
 - ориентированное отношение
 - неориентированное отношение
 - рефлексивное отношение
 - антирефлексивное отношение
 - частично рефлексивное отношение
 - симметричное отношение
 - антисимметричное отношение
 - частично симметричное отношение
 - транзитивное отношение
 - антитранзитивное отношение
 - частично транзитивное отношение
 - связанное отношение
 - отношение порядка
 - отношение строгого порядка
 - отношение нестрогого порядка
 - отношение толерантности
 - отношение эквивалентности
 - ролевое отношение
 - числовой атрибут
 - неролевое отношение
 - неролевое бинарное отношение
 - арность
 - метаотношение
 - отношение декомпозиции
 - отношение интеграции
- ∃ исследуемое отношение':
 - область определения*
 - атрибут отношения*
 - домен*
 - первый домен*
 - второй домен*
 - композиция отношений*
 - фактор-множество*
 - соответствие*
 - отношение соответствия*
 - область отправления'

- область прибытия'
- образ'
- прообраз'
- всюду определенное соответствие*
- частично определенное соответствие*
- сюръективное соответствие*
- несюръективное соответствие*
- однозначное соответствие*
- обратное соответствие*
- обратимое соответствие*
- неоднозначное соответствие*
- инъективное соответствие*
- взаимно однозначное соответствие*
- множество сочетаний*
- множество размещений*
- множество перестановок*

связь

:= [связка sc-элементов]

:= [sc-связка]

⇒ пояснение*:

[**связь** – множество, являющееся абстрактной моделью связи между описываемыми сущностями, которые или знаки которых являются элементами этого множества.]

⇒ примечание*:

[Напомним, что все элементы множества, представленного в SC-коде, являются знаками, но описываемыми сущностями могут быть не только сущности, обозначаемые sc-элементами, но и сами эти sc-элементы.]

⇒ разбиение*:

- { • бинарная связь
- небинарная связь

⇒ разбиение*:

- { • неориентированная связь
- ориентированная связь

бинарная связь

⇒ разбиение*:

- { • sc-коннектор
- неатомарная бинарная связь

⇒ примечание*:

[Данное разбиение осуществляется на основе синтаксического признака, а не семантического, поскольку каждый *sc-коннектор* может быть записан в памяти при помощи семантически эквивалентной конструкции, содержащей знак самой связи и пары принадлежности, ведущие к ее элементам, уточненные, при необходимости ролевыми отношениями.]

sc-коннектор

:= [атомарная бинарная связь]

⇒ пояснение*:

[Каждый **sc-коннектор** представлен в *sc-памяти* одним *sc-элементом* и семантически эквивалентен конструкции, содержащей знак некоторой *бинарной связи* и пары принадлежности, ведущие к элементам этой связи, уточненные, при необходимости ролевыми отношениями.]

Такая конструкция может быть обозначена **sc-коннектором** только в случае, когда роли компонентов соответствующей бинарной связи указываются только при помощи *числовых атрибутов 1'* и *2'* или не уточняются вообще.]

неатомарная бинарная связь

⇒ пояснение*:

[**неатомарная бинарная связь** – бинарная связь, роли компонентов которой не могут быть заданы только при помощи ролевых отношений $1'$ и $2'$, или не заданы совсем, а требуют дополнительного уточнения при помощи более частных ролевых отношений.]

небинарная связь

⇒ пояснение*:

[**небинарная связь** – связь, имеющая больше двух элементов.]

неориентированная связь

⊃ неориентированное множество

⇒ пояснение*:

[**неориентированная связь** – связь, все элементы которой имеют одинаковые роли (при этом соответствующее ролевое отношение, как правило, явно не указывается).]

ориентированная связь

⊃ кортеж

⇒ пояснение*:

[**ориентированная связь** – связь, в которой с помощью ролевых отношений, указываются роли компонентов этой связи.]

отношение

:= [класс связей]

:= [класс sc-связок]

:= [множество отношений]

:= [Множество всевозможных отношений]

⇒ определение*:

[**отношение**, заданное на множестве M – это подмножество декартового произведения этого множества самого на себя некоторое количество раз.

В более широком смысле **отношение** – это математическая структура, которая формально определяет свойства различных объектов и их взаимосвязи.]

⇒ разбиение*:

- класс равномоощных связок
- класс связок разной мощности

⇒ разбиение*:

- бинарное отношение
- небинарное отношение

⇒ разбиение*:

- ориентированное отношение
- неориентированное отношение

⇒ разбиение*:

- ролевое отношение
- неролевое отношение

класс равномоощных связок

:= [класс связок фиксированной арности]

:= [отношение, обладающее свойством арности]

⊃ унарное отношение

⊃ бинарное отношение

⊃ тернарное отношение

⇒ определение*:

[**класс равномоощных связок** – класс связок, имеющих одинаковую мощность.]

класс связок разной мощности

:= [отношение нефиксированной арности]

⊃ небинарное отношение

⇒ *определение**:
 [класс **связок разной мощности** – класс связок, имеющих разную мощность.]

унарное отношение

:= [отношение арности один]

:= [одноместное отношение]

:= [множество синглетов]

⇒ *определение**:

[**унарное отношение** – это множество таких отношений на множестве M , являющихся любым подмножеством множества M .]

бинарное отношение

:= [отношение арности два]

:= [двухместное отношение]

⊃ *квазибинарное отношение*

⊃ *отношение порядка*

⊃ *отношение толерантности*

⇒ *разбиение**:

- *рефлексивное отношение*
- *антирефлексивное отношение*
- *частично рефлексивное отношение*

⇒ *разбиение**:

- *симметричное отношение*
- *антисимметричное отношение*
- *частично симметричное отношение*

⇒ *разбиение**:

- *транзитивное отношение*
- *антитранзитивное отношение*
- *частично транзитивное отношение*

⇒ *разбиение**:

- *ролевое отношение*
- *неролевое бинарное отношение*

⇒ *определение**:

[**бинарное отношение** – это множество таких отношений на множестве M , являющихся подмножеством декартова произведения множества M .

Если **бинарное отношение** R задано на множестве M и два элемента этого множества a и b связаны данным отношением, то будем обозначать такую связь как aRb .]

квазибинарное отношение

⇒ *пояснение**:

[**квазибинарное отношение** – множество ориентированных пар, первые компоненты которых являются связками.

Таким образом, *sc-дуги*, принадлежащие **квазибинарным отношениям**, всегда выходят из связок.]

⇒ *sc-утверждение**:

[В область определения квазибинарного отношения будем включать:

- вторые компоненты ориентированных пар, принадлежащих этому отношению;
- элементы первых компонентов ориентированных пар, принадлежащих этому отношению;
- других элементов область определения квазибинарного отношения не содержит.

]

тернарное отношение

:= [отношение арности три]

:= [трехместное отношение]

⇒ *пояснение**:

[*тернарное отношение* – это множество отношений, связывающих между собой три элемента.]

небинарное отношение

⇒ *пояснение**:

[*небинарное отношение* – это множество отношений, хотя бы одна из связок каждого из которых имеет значение мощности больше двух.]

ориентированное отношение

⇒ *определение**:

[*ориентированное отношение* – это множество таких отношений, каждая связка которых является кортежем.]

неориентированное отношение

⇒ *определение**:

[*неориентированное отношение* – это множество таких отношений, каждая связка которых является неориентированным множеством.]

рефлексивное отношение

⇒ *определение**:

[*рефлексивное отношение* – это *бинарное отношение*, любая пара которого есть канторовское множество.]

антирефлексивное отношение

⇒ *определение**:

[*антирефлексивное отношение* R на множестве A – это *бинарное отношение*, в котором все элементы множества A не находятся в отношении R к самому себе.]

частично рефлексивное отношение

⇒ *определение**:

[*частично рефлексивное отношение* R на множестве A – это *бинарное отношение*, в котором хотя бы один (но не все) элемент множества A находится в отношении R к самому себе.]

симметричное отношение

⇒ *определение**:

[*симметричное отношение* R на множестве A – это *бинарное отношение*, в котором для каждой пары элементов a и b этого множества выполнение отношения aRb влечёт выполнение bRa .]

антисимметричное отношение

⇒ *определение**:

[*антисимметричное отношение* R на множестве A – это *бинарное отношение*, в котором для каждой пары элементов a и b этого множества выполнение отношений aRb и bRa влечёт равенство a и b .]

частично симметричное отношение

⇒ *определение**:

[*частично симметричное отношение* R на множестве A – это *бинарное отношение*, в котором для каждой пары элементов a и b (но не для всех таких пар) этого множества выполнение отношения aRb влечёт выполнение bRa .]

транзитивное отношение

⇒ *определение**:

[*транзитивное отношение* R на множестве A – это *бинарное отношение*, в котором для любых трёх элементов этого множества a, b, c выполнение отношений aRb и bRc влечёт выполнение отношения aRc .]

антитранзитивное отношение

⇒ *определение**:

[*антитранзитивное отношение* R на множестве A – это *бинарное отношение*, в котором для любых трёх элементов этого множества a, b, c выполнение отношений aRb и bRc не влечёт выполнение отношения aRc .]

частично транзитивное отношение

⇒ *определение**:

[**частично транзитивное отношение** R на множестве A – это бинарное отношение, в котором для каждых трёх элементов этого множества a, b, c (но не для всех таких троек) выполнение отношений aRb и bRc влечёт выполнение отношения aRc .]

связанное отношение*

∈ бинарное отношение

⇒ определение*:

[**связанное отношение*** R на множестве A – это бинарное отношение, в котором для каждой пары элементов a и b этого множества выполняется одно из двух отношений: aRb или bRa .]

отношение порядка

⇒ разбиение*:

- отношение строгого порядка
- отношение нестрогого порядка

⇒ определение*:

[**отношение порядка** – это бинарное отношение, обладающее свойством транзитивности и антисимметричности.]

отношение строгого порядка

⇒ определение*:

[**отношение строгого порядка** – это отношение порядка, обладающее свойством антирефлексивности.]

отношение нестрогого порядка

⇒ определение*:

[**отношение нестрогого порядка** – это отношение порядка, обладающее свойством рефлексивности.]

отношение толерантности

⇒ определение*:

[**отношение толерантности** – это бинарное отношение, принадлежащее классам симметричное отношение и рефлексивное отношение.]

отношение эквивалентности

:= [максимальное семейство отношений эквивалентности]

⊂ отношение толерантности

⇒ определение*:

[**отношение эквивалентности** – это отношение толерантности, принадлежащее классу транзитивных отношений]

⇒ примечание*:

[Каждое отношение эквивалентности уточняет то, что мы считаем эквивалентными сущностями, т.е. то, на какие сходства этих сущностей мы обращаем внимание и какие их отличия мы игнорируем (не учитываем).]

ролевое отношение

:= [атрибут]

:= [атрибутивное отношение]

:= [отношение, которое задает роль элементов в рамках некоторого множества]

:= [отношение, являющееся подмножеством отношения принадлежности]

⊂ семейство подмножеств*:

принадлежность*

⊂ бинарное отношение

⊃ числовой атрибут

⇒ пояснение*:

[**ролевое отношение** – это отношение, являющееся подмножеством отношения принадлежности.]

⇒ правило идентификации экземпляров*:

[В конце каждого идентификатора, соответствующего экземплярам класса **ролевое отношение**, не являющегося системным, ставится знак “/”.

Например:

ключевой экземпляр'

Из-за ограничений в разрешенном алфавите символов, в системном идентификаторе не может быть использоваться знак “*”, поэтому в начале каждого *системного идентификатора*, соответствующего экземплярам класса **ролевое отношение** ставится префикс “rrel_”.

Например:
rrel_key_sc_element]

числовой атрибут

:= [порядковый номер]

:= [номер компонента ориентированной связки]

⊃ 1'; 2'; 3'; 4'; 5'; 6'; 7'; 8'; 9'; 10'

⇒ *пояснение**:

[**числовой атрибут** – *ролевое отношение*, задающее порядковый номер элемента некоторой ориентированной связки, не уточняя при этом семантику такой принадлежности. Во многих случаях бывает достаточно использовать числовые атрибуты, чтобы различать компоненты связки, семантика каждого из которых дополнительно оговаривается, например, при определении отношения, которому данная связка принадлежит.]

неролевое отношение

⇒ *разбиение**:

- *небинарное отношение*
- *неролевое бинарное отношение*

⇒ *пояснение**:

[**неролевое отношение** – отношение, не являющееся подмножеством отношения принадлежности.]

⇒ *правило идентификации экземпляров**:

[В конце каждого *идентификатора*, соответствующего экземплярам класса **неролевое отношение**, не являющегося системным, ставится знак “*”.

Например:
включение*

Из-за ограничений в разрешенном алфавите символов, в системном идентификаторе не может быть использоваться знак “*”, поэтому в начале каждого *системного идентификатора*, соответствующего экземплярам класса **неролевое отношение** ставится префикс “nrel_”.

Например:
nrel_inclusion]

неролевое бинарное отношение

⇒ *пояснение**:

[**неролевое бинарное отношение** – *бинарное отношение*, не являющееся *ролевым отношением*.]

арность

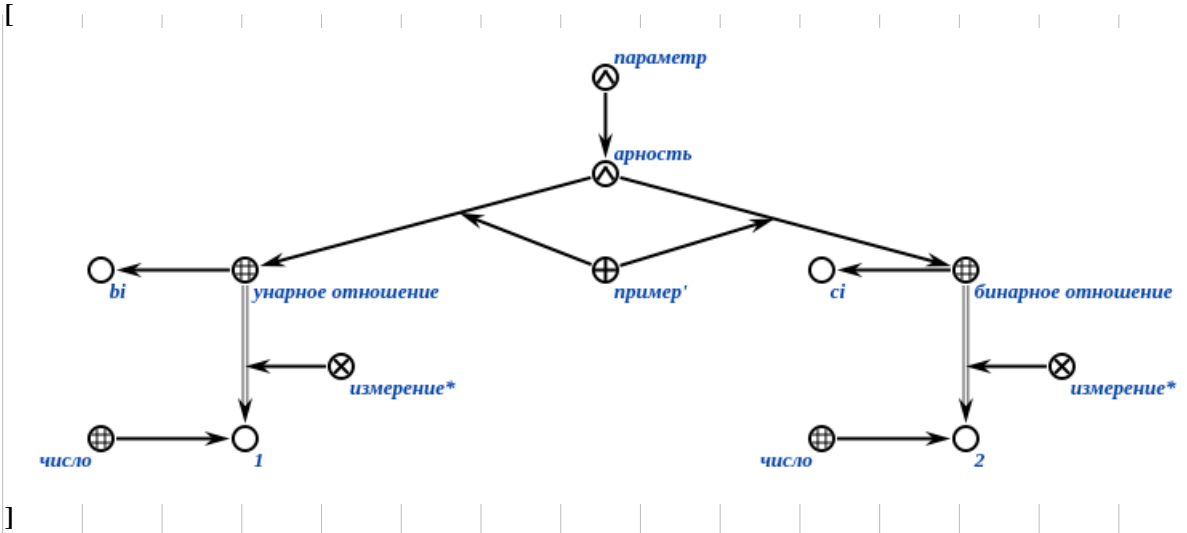
:= [арность отношения]

∈ *параметр*

⇒ *пояснение**:

[**арность** – это параметр, каждый элемент которого представляет собой класс *отношений*, каждая связка которых имеет одинаковую *мощность*. Значение данного *параметра* совпадает со значением *мощности* каждой из таких связок.]

⇒ *описание примера**:



область определения*

:= [область определения отношения*]

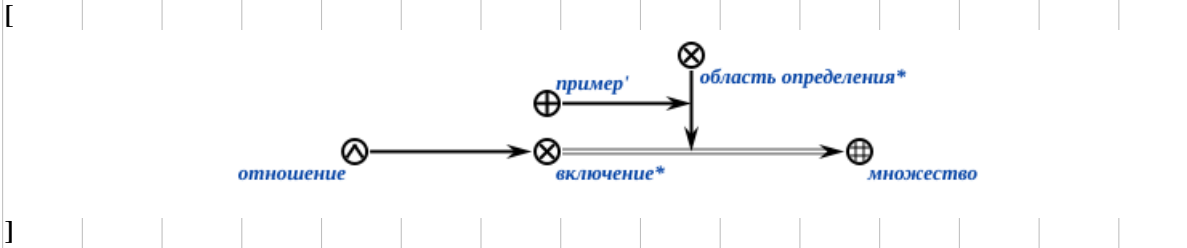
∈ бинарное отношение

⇒ пояснение*:

[**область определения*** – это бинарное отношение, связывающее отношение со множеством, являющимся его областью определения.]

Областью определения отношения будем называть результат теоретико-множественного объединения всех связок этого отношения, или, другими словами, результат теоретико-множественного объединения всех множеств, являющихся доменами данного отношения.]

⇒ описание примера*:



атрибут отношения*

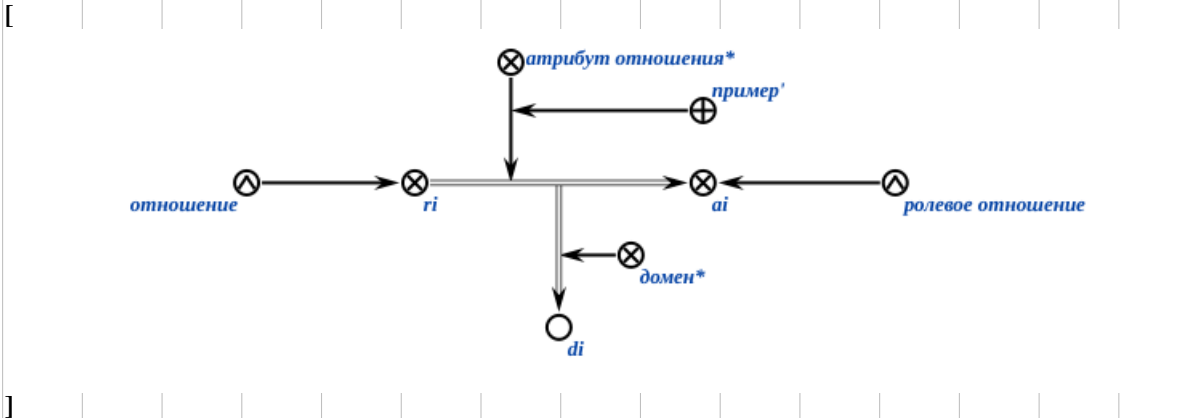
:= [ролевой атрибут, используемый в связках заданного отношения*]

∈ бинарное отношение

⇒ пояснение*:

[**атрибут отношения*** – это бинарное отношение, связывающее заданное отношение с ролевым отношением, используемым в данном отношении для уточнения роли того или иного элемента связок данного отношения.]

⇒ описание примера*:



домен*

:= [домен отношения по заданному атрибуту*]

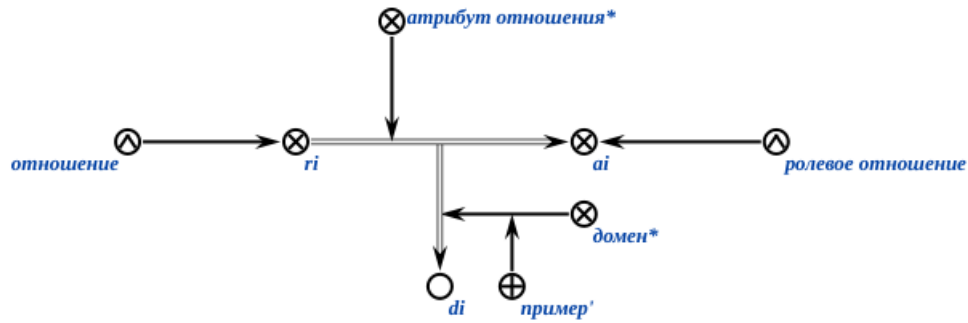
∈ бинарное отношение

⇒ пояснение*:

[**домен*** – это бинарное отношение, связывающее связку отношения *атрибут отношения** со множеством, являющимся доменом заданного отношения по заданному атрибуту. Множество *di* является доменом отношения *ri* по атрибуту *ai* в том и только том случае, если элементами этого множества являются все те и только те элементы связок отношения *ri*, которые имеют в рамках этих связок атрибут *ai*.]

⇒ описание примера*:

[



]

первый домен*

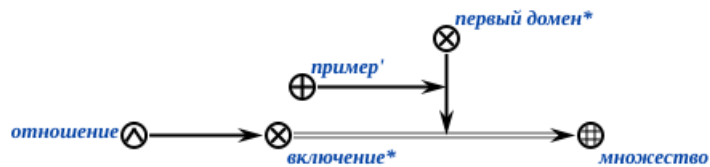
∈ бинарное отношение

⇒ определение*:

[**первый домен*** – это бинарное отношение, связывающее отношение с множеством, являющимся доменом по атрибуту *1'* данного отношения.]

⇒ описание примера*:

[



]

второй домен*

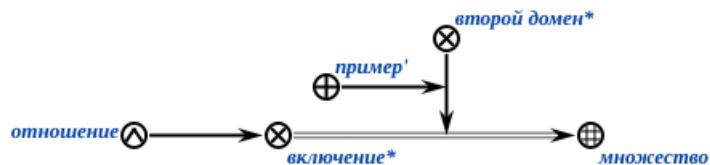
∈ бинарное отношение

⇒ определение*:

[**второй домен*** – это бинарное отношение, связывающее отношение с множеством, являющимся доменом по атрибуту *2'* данного отношения.]

⇒ описание примера*:

[



]

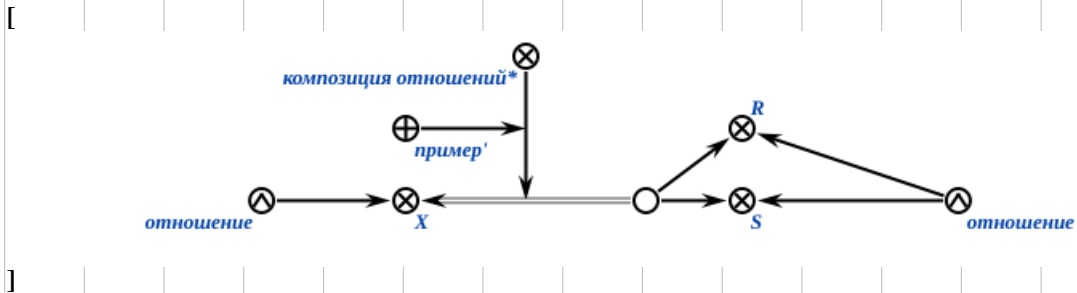
композиция отношений*

∈ квазибинарное отношение

⇒ определение*:

[**композиция отношений*** – это *квазибинарное отношение*, связывающее два бинарных отношения с отношением, являющимся их композицией. Под композицией бинарных отношений R и S будем понимать множество $\{(x, y) | \exists z(xSz \wedge zRy)\}$]

⇒ описание примера*:



фактор-множество*

:= [быть фактор-множеством*]

:= [множество всевозможных максимальных множеств из попарно эквивалентных элементов*]

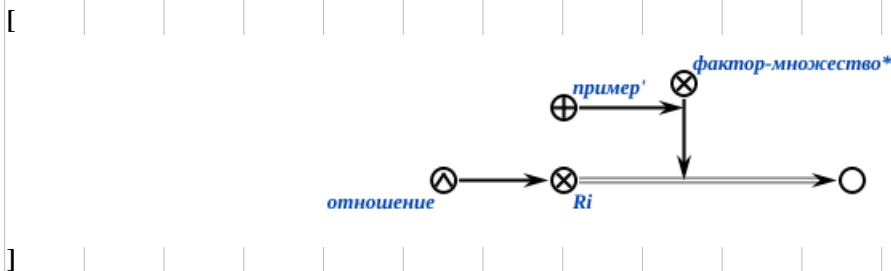
:= [множество всевозможных классов эквивалентности для заданного отношения эквивалентности*]

∈ бинарное отношение

⇒ определение*:

[**фактор множество*** – это бинарное ориентированное отношение, каждая связка которого связывает некоторое отношение эквивалентности со множеством всех соответствующих этому отношению классов эквивалентности. Каждый такой класс представляет собой максимальное множество сущностей, каждая пара которых принадлежит указанному выше отношению эквивалентности.]

⇒ описание примера*:



метаотношение

⇒ определение*:

[метаотношение – это *отношение*, в каждой связке которого есть по крайней мере один компонент, являющийся знаком некоторого *отношения*.]

отношение декомпозиции

⊃ разбиение*

⊃ декомпозиция раздела*

⊃ декомпозиция абстрактного объекта*

отношение интеграции

⊃ объединение*

соответствие*

:= [наличие соответствия*]

∈ бинарное отношение

⇒ разбиение*:

- соответствие между непересекающимися множествами*
- соответствие между строго пересекающимися множествами*
- соответствие, область отправления и область прибытия которого совпадают*

⇒ разбиение*:

- всюду определенное соответствие*
- частично определенное соответствие*

⇒ разбиение*:

- { • сюръекция*
- несюръективное соответствие*

⇒ разбиение*:

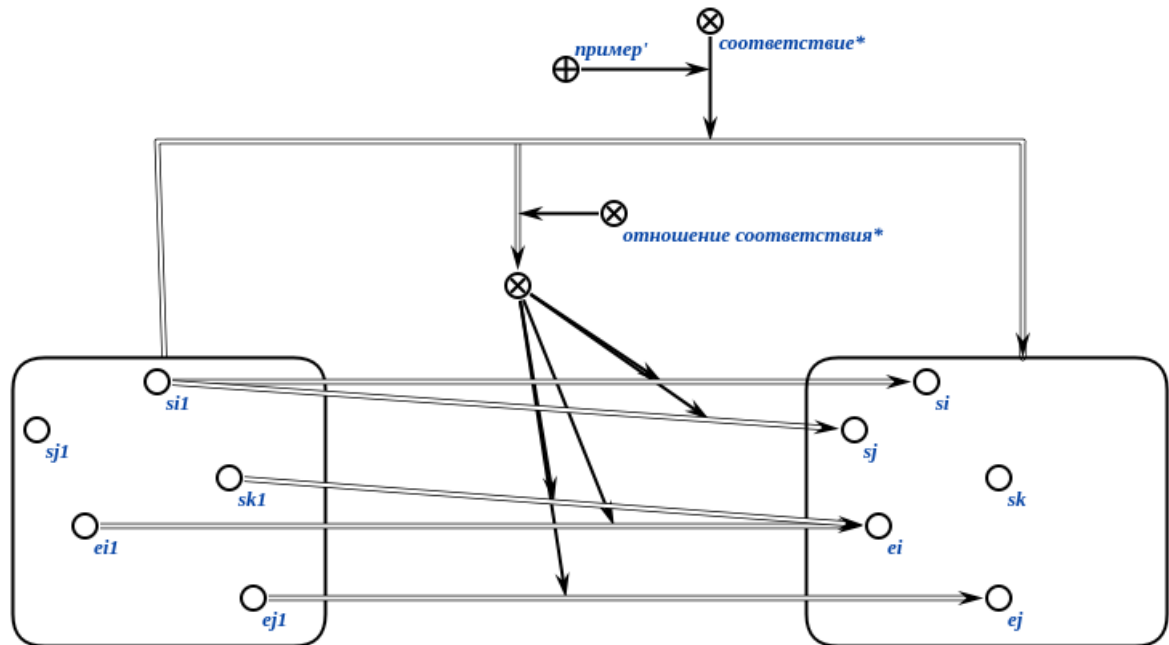
- { • однозначное соответствие*
- неоднозначное соответствие*

⇒ определение*:

[соответствие* – бинарное ориентированное отношение, каждая пара которого связывает два множества и указывает на наличие некоторого отношения, связывающего элементы этих двух множеств.]

⇒ описание примера*:

[



]

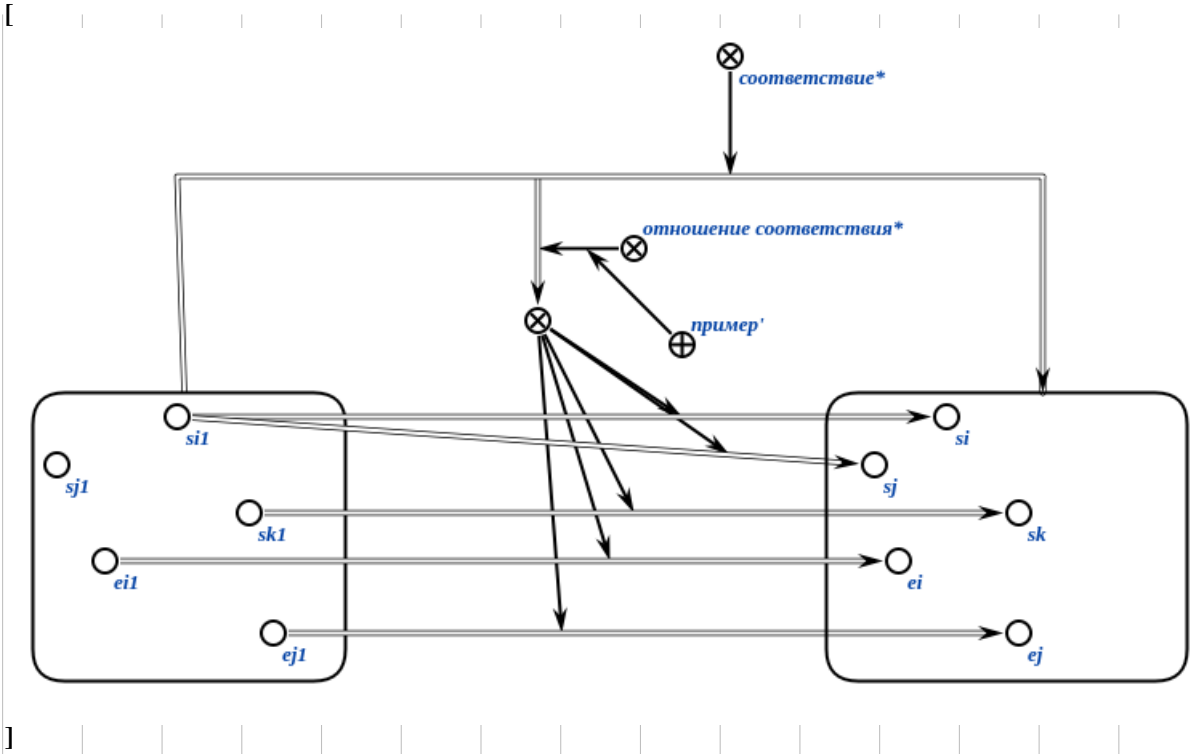
отношение соответствия*

∈ бинарное отношение

⇒ определение*:

[отношение соответствия* – бинарное отношение, связывающее ориентированную пару множеств, на которых задано соответствие* и некоторое подмножество декартова произведения* этих множеств.]

⇒ описание примера*:



область отправления'

:= [область отправления соответствия']

:= [область определения соответствия']

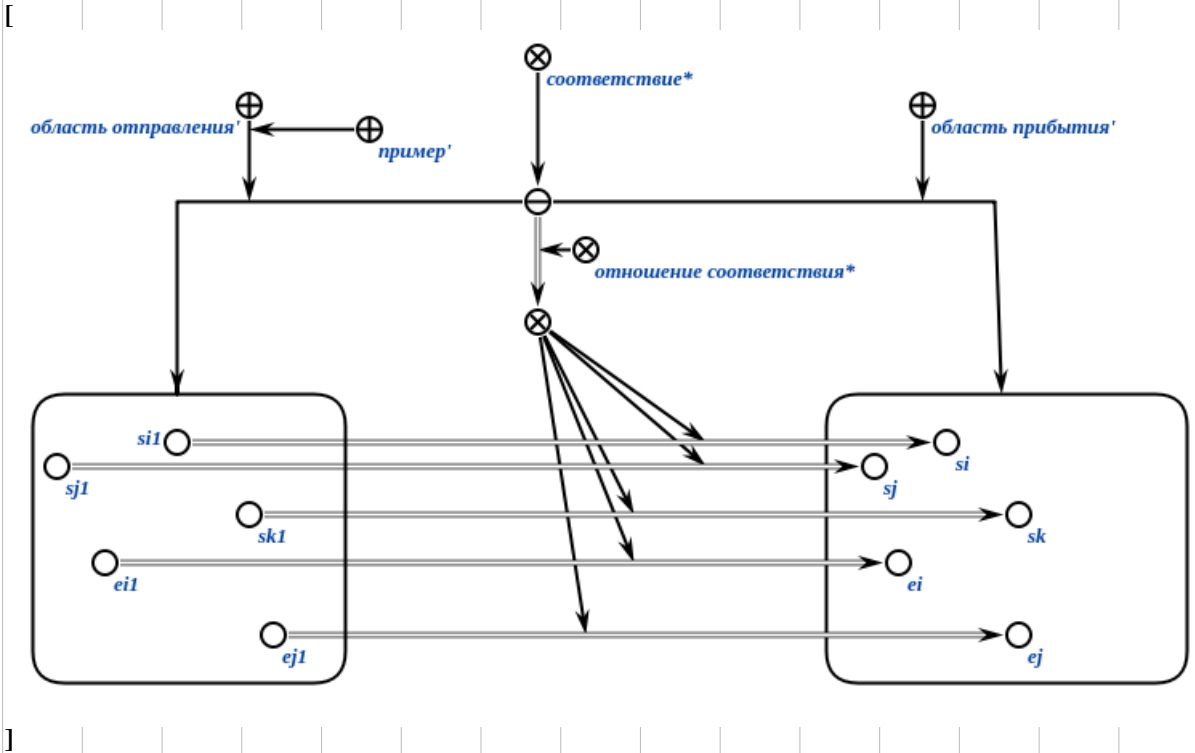
:= [первый компонент пары в отношении соответствия']

∈ ролевое отношение

⇒ определение*:

[**область отправления'** – ролевое отношение, указывающее на первый компонент пары в рамках отношения соответствие*.]

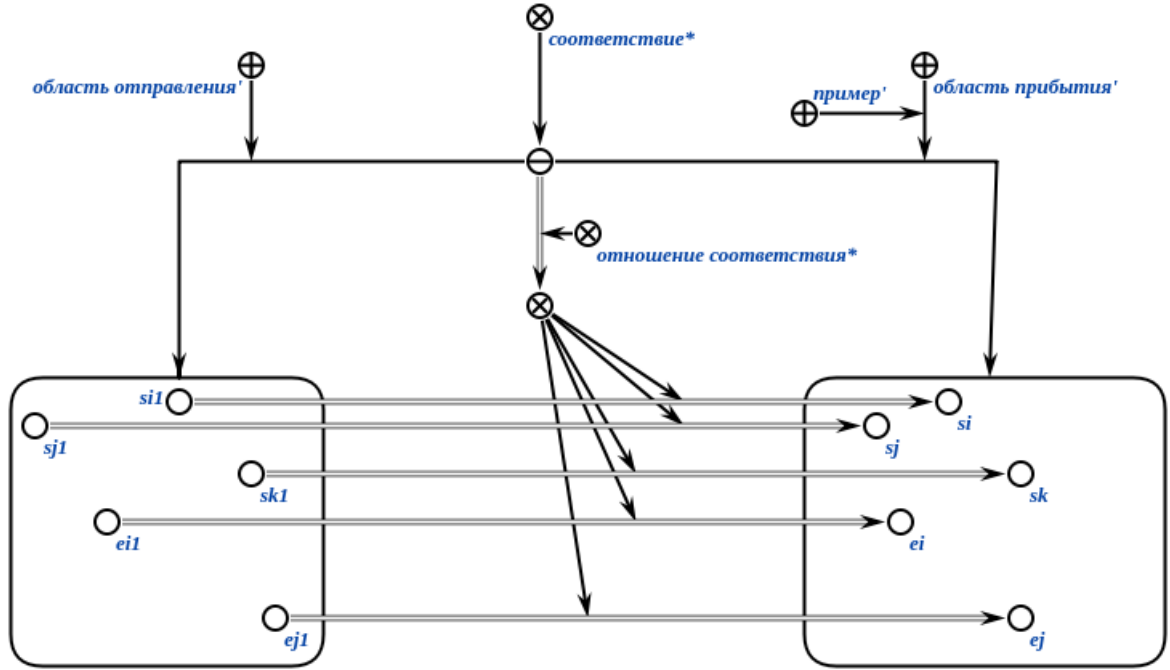
⇒ описание примера*:



область прибытия'

:= [область прибытия соответствия']

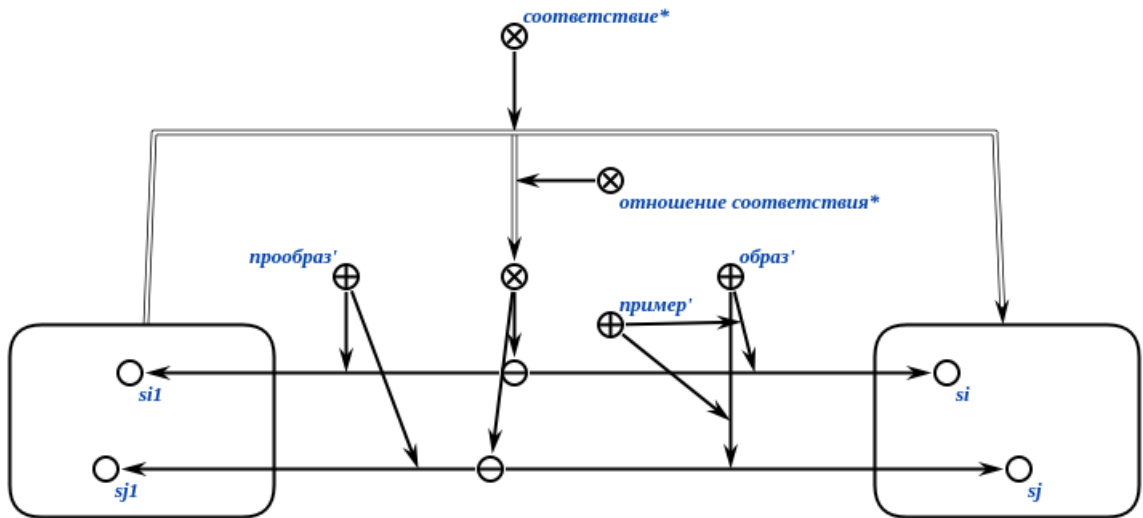
:= [область значений соответствия']
 ∈ ролевое отношение
 ⇒ определение*:
 [область прибытия' – ролевое отношение, указывающее на второй компонент пары в рамках отношения соответствия*.]
 ⇒ описание примера*:
 [



]

образ'

:= [образ соответствия']
 ∈ ролевое отношение
 ⇒ определение*:
 [образ' – ролевое отношение, указывающее на второй компонент каждой пары в рамках множества пар, которое является вторым компонентом отношения соответствия*.]
 ⇒ описание примера*:
 [



]

прообраз'

:= [прообраз соответствия']

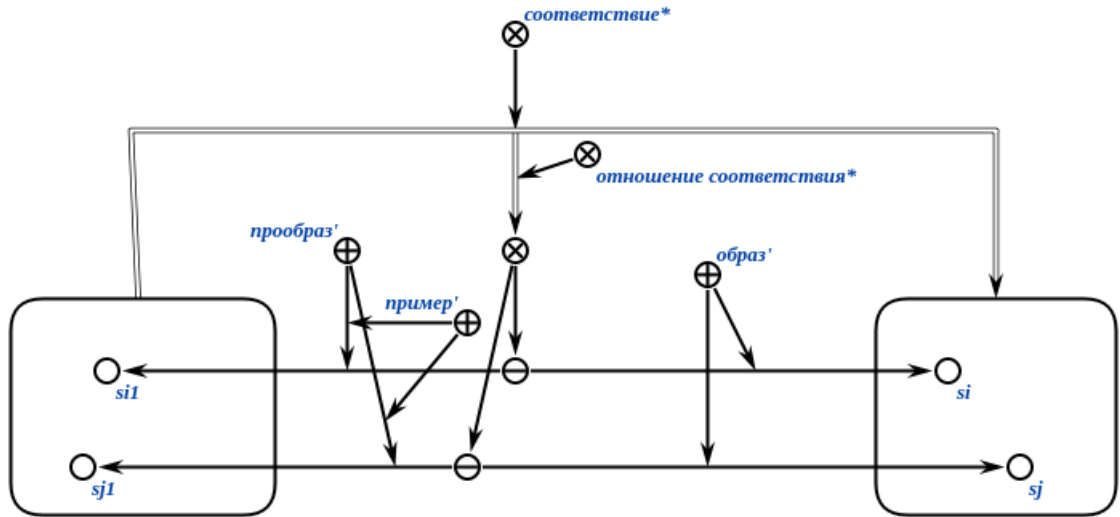
∈ ролевое отношение

⇒ определение*:

[**прообраз'** – ролевое отношение, указывающее на первый компонент каждой пары в рамках множества пар, которое является первым компонентом отношения соответствия*.]

⇒ описание примера*:

[



]

всюду определенное соответствие*

:= [полное соответствие*]

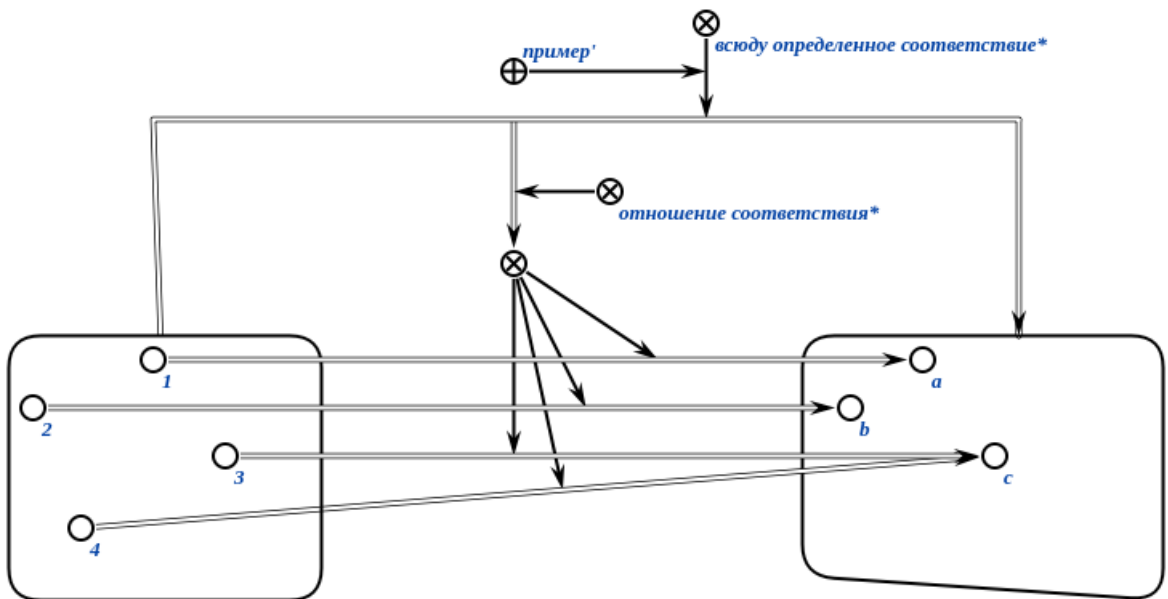
:= [наличие всюду определенного соответствия*]

⇒ определение*:

[**всюду определенное соответствие*** – это соответствие*, при котором существует образ' для каждого элемента области отправления' данного соответствия*.]

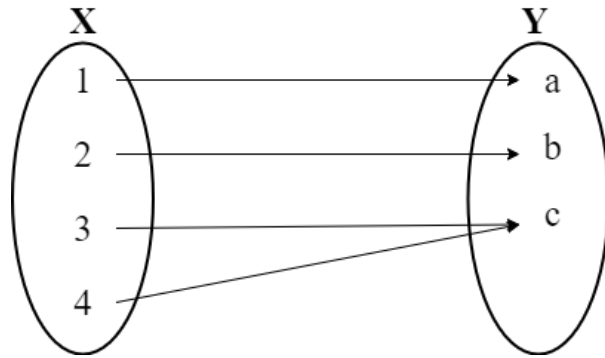
⇒ описание примера*:

[



]

⇒ изображение*:



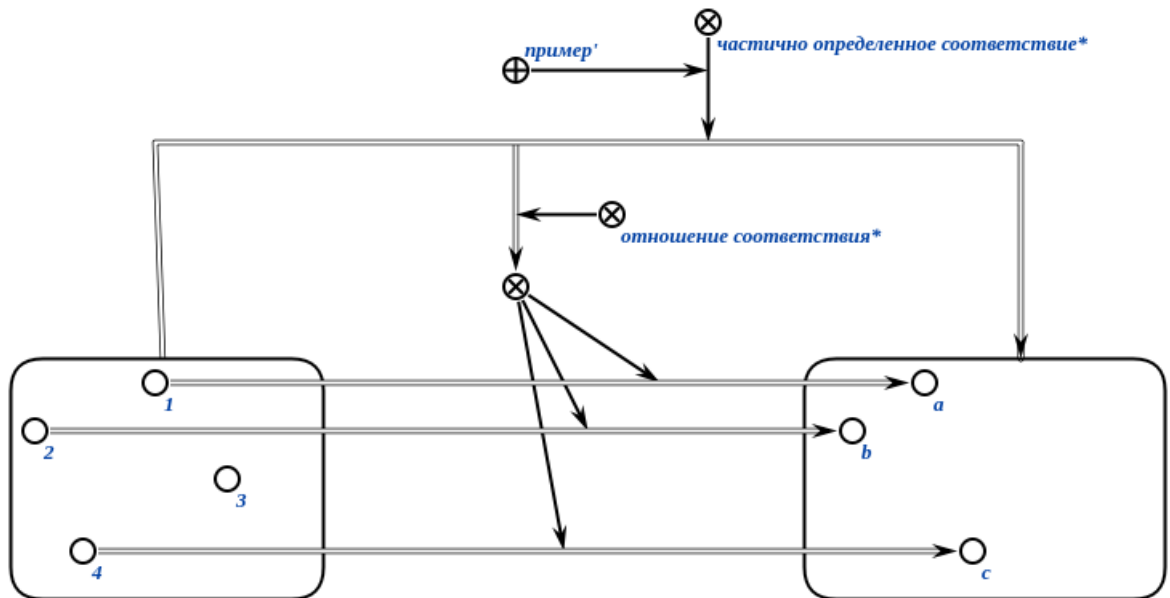
частично определенное соответствие*

:= [наличие частично определенного соответствия*]

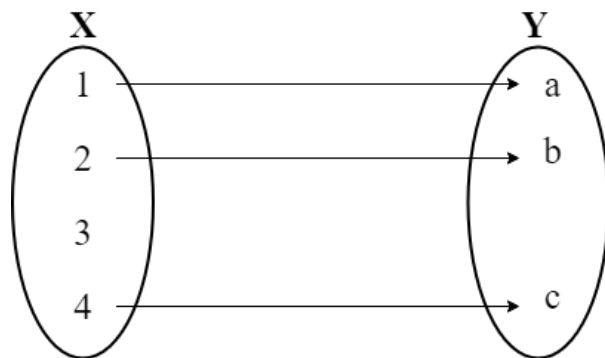
⇒ *определение**:

[*частично определенное соответствие** – это *соответствие**, при котором существует *образ'* для некоторых, но не всех элементов *области отправления'* данного *соответствия**.]

⇒ *описание примера**:



⇒ *изображение**:



сюръективное соответствие*

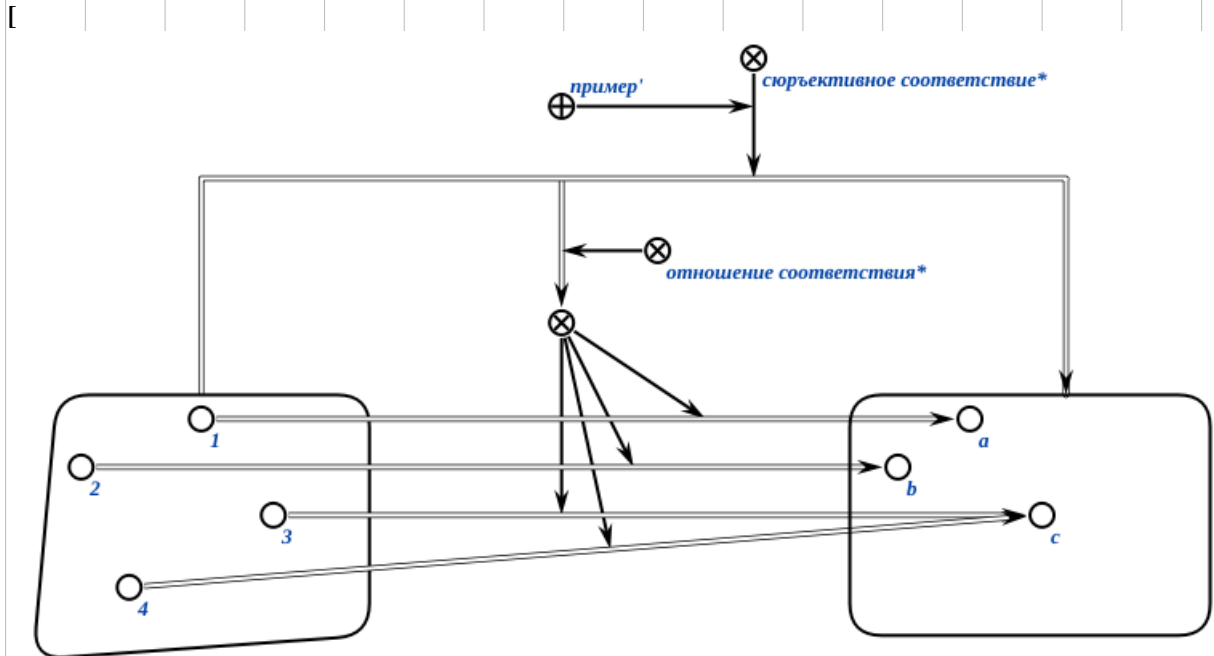
:= [наличие сюръективного соответствия*]

:= [сюръекция*]

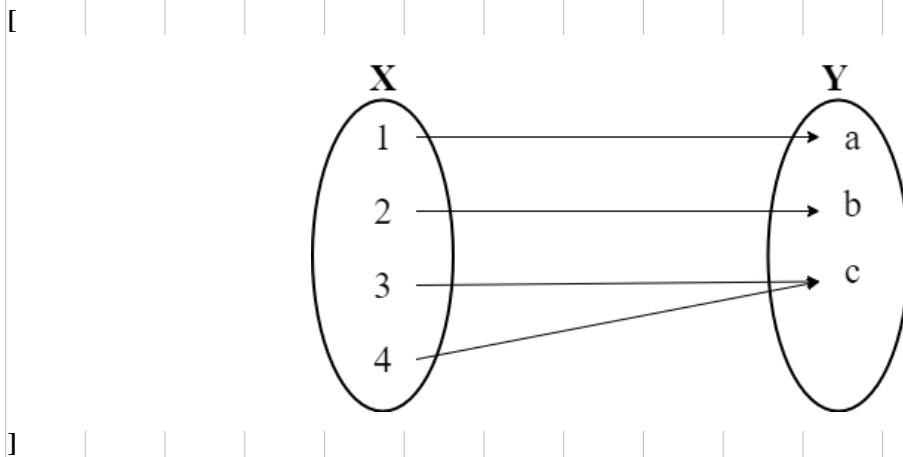
⇒ *определение**:

[**сюръективное соответствие*** – это *соответствие**, при котором существует *прообраз'* для каждого элемента *области прибытия'* данного *соответствия**.]

⇒ *описание примера**:



⇒ *изображение**:

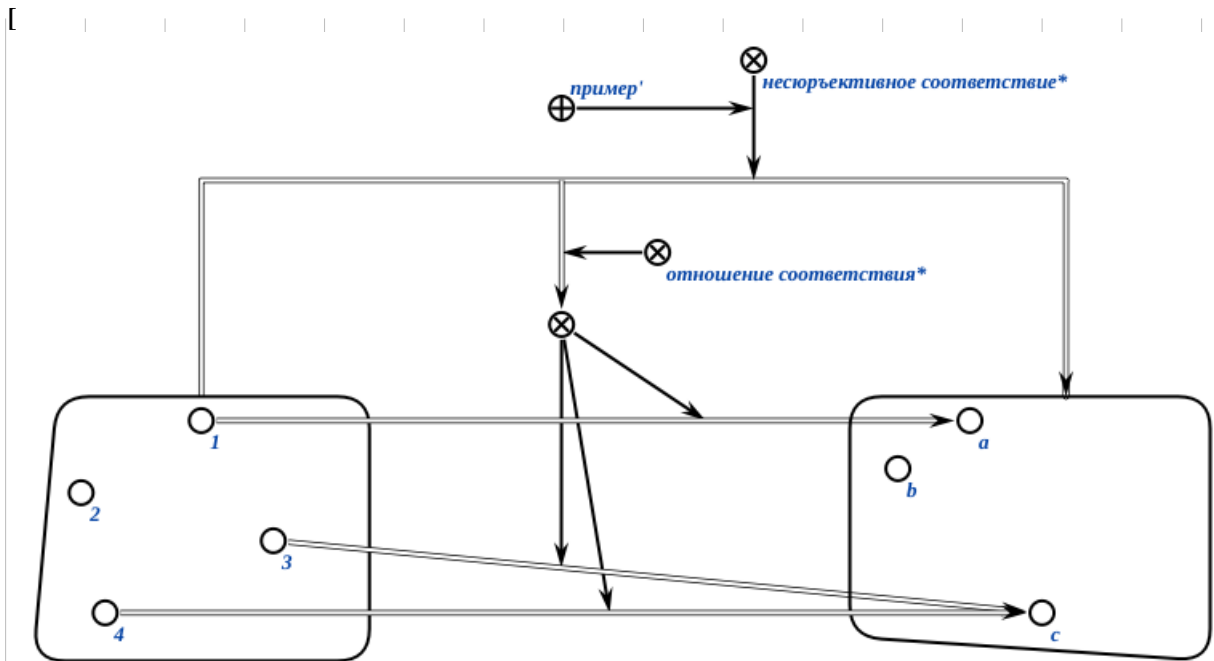
**несюръективное соответствие***

:= [наличие несюръективного соответствия*]

⇒ *определение**:

[**несюръективное соответствие*** – это *соответствие**, при котором не для каждого элемента *области прибытия'* данного *соответствия** существует *прообраз'*.]

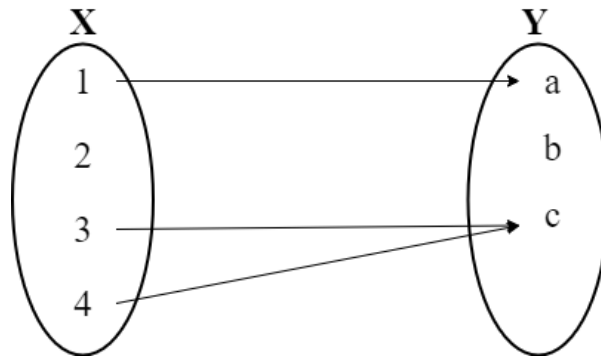
⇒ *описание примера**:



⇒

] изображение*:

[



]

однозначное соответствие*

:= [наличие однозначного соответствия*]

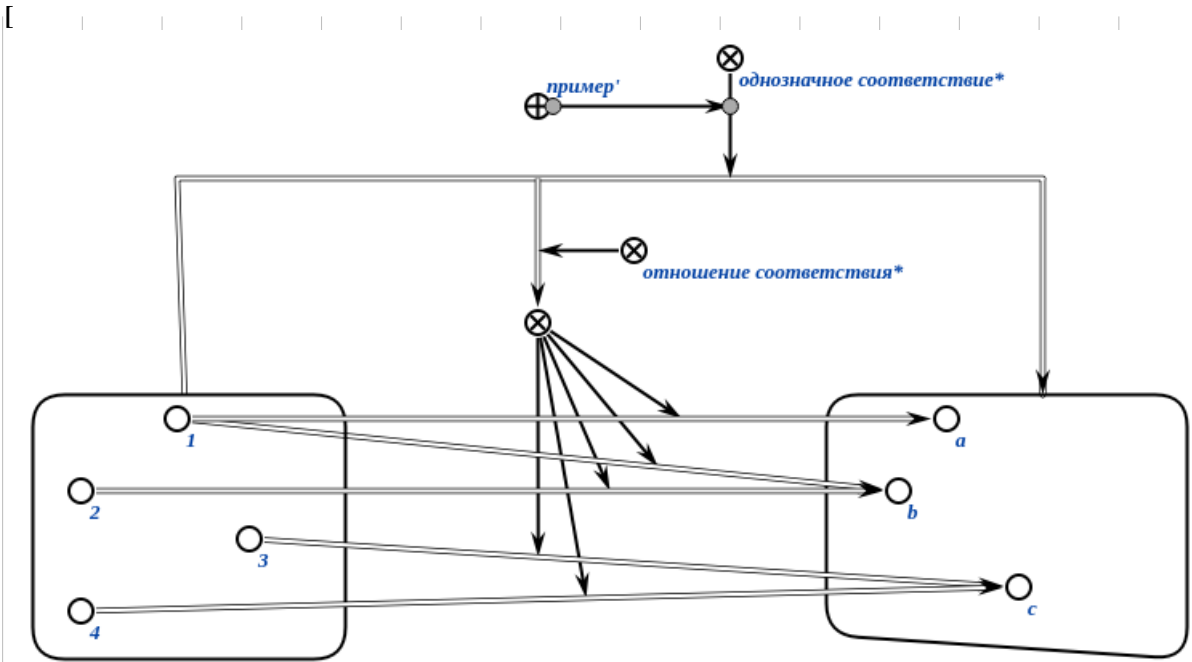
:= [функциональное соответствие*]

:= [функция*]

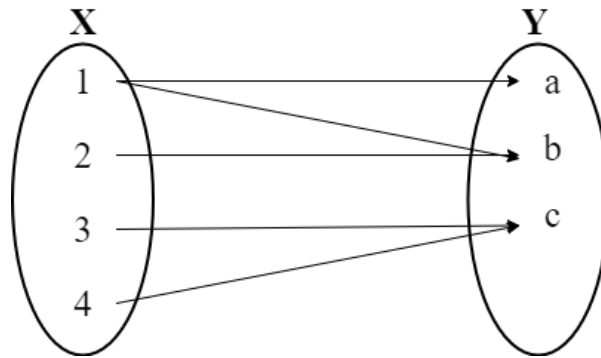
⇒ определение*:

[однозначное соответствие* – это соответствие*, при котором каждому элементу из области отправления' соответствия ставится не более, чем один элемент из области прибытия' соответствия.]

⇒ описание примера*:



⇒ изображение*:



обратное соответствие*

∈ бинарное отношение
 ⇒ область определения*:
 соответствие*
 ⇒ определение*:

[**обратное соответствие*** – бинарное отношение, связывающее два соответствия*, при этом выполняются следующие условия:

- область отправления' первого соответствия является областью прибытия' второго;
- область прибытия' первого соответствия является областью отправления' второго;
- для каждой пары, входящей в состав отношения первого соответствия, существует пара, входящая в состав отношения второго соответствия, при этом образ' и прообраз' в рамках первой указанной пары являются соответственно прообразом' и образом' в рамках второй.

обратимое соответствие*

⊆ однозначное соответствие*
 ⇒ определение*:

[**обратимое соответствие*** – такое однозначное соответствие*, для которого обратное соответствие* также является однозначным соответствием*.]

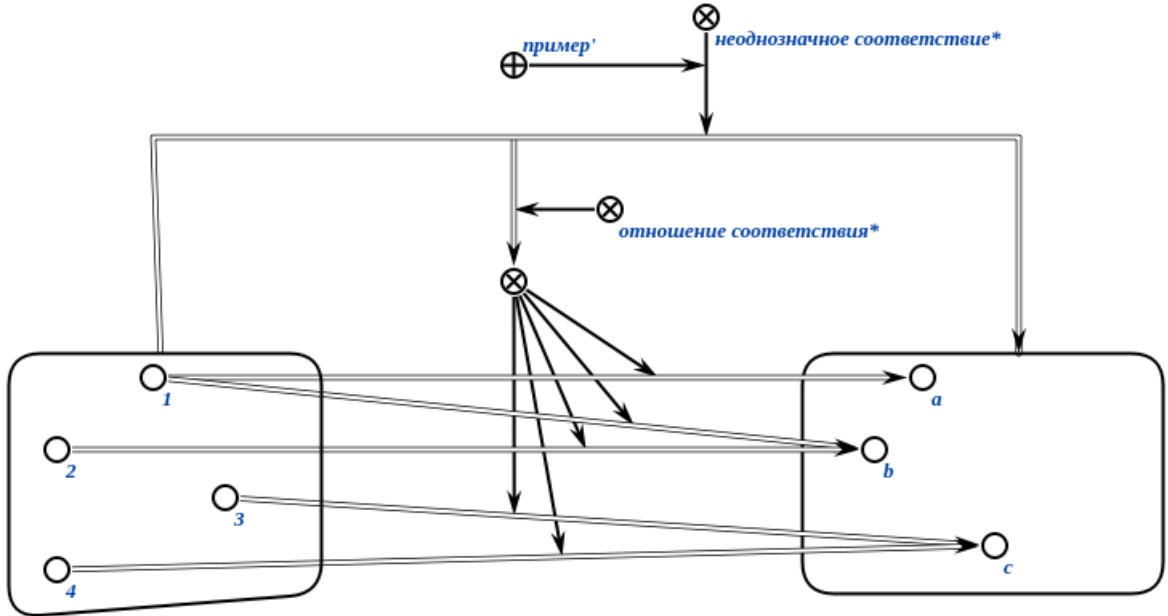
неоднозначное соответствие*

⇒ *определение**:

[**неоднозначное соответствие*** – это *соответствие**, при котором хотя бы одному элементу из *области отправления*' соответствия ставится более, чем один элемент из *области прибытия*' соответствия.]

⇒ *описание примера**:

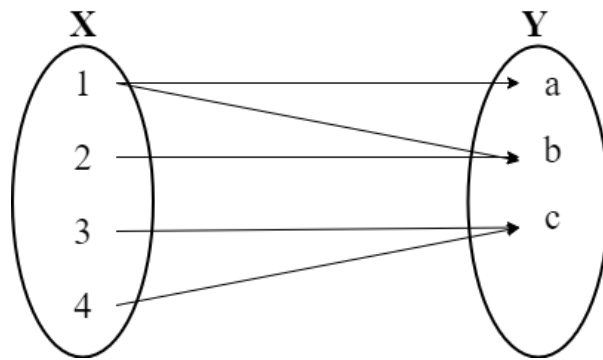
[



]

⇒ *изображение**:

[



]

инъективное соответствие*

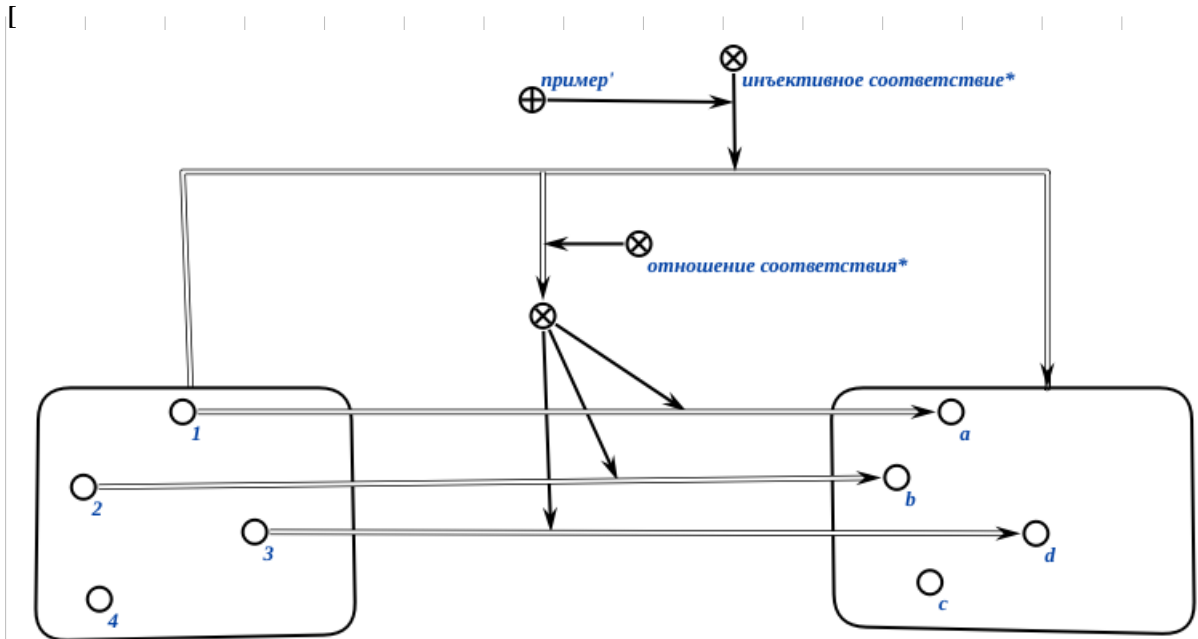
:= [инъекция*]

⊂ *однозначное соответствие**

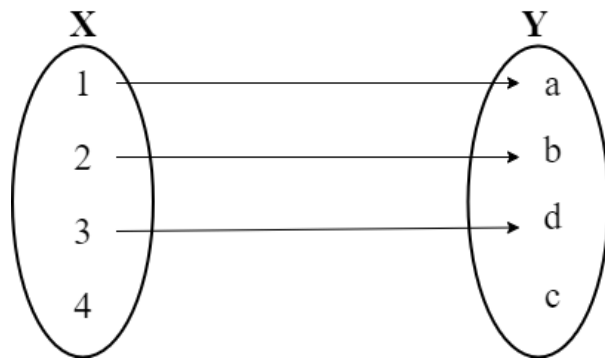
⇒ *определение**:

[**инъективное соответствие*** – это *соответствие**, при котором разным элементам из *области отправления*' соответствия всегда соответствуют разные элементы из *области прибытия*' соответствия и наоборот.]

⇒ *описание примера**:



⇒ изображение*:



взаимно однозначное соответствие*

:= [биекция*]

⊂ *всюду определенное соответствие**

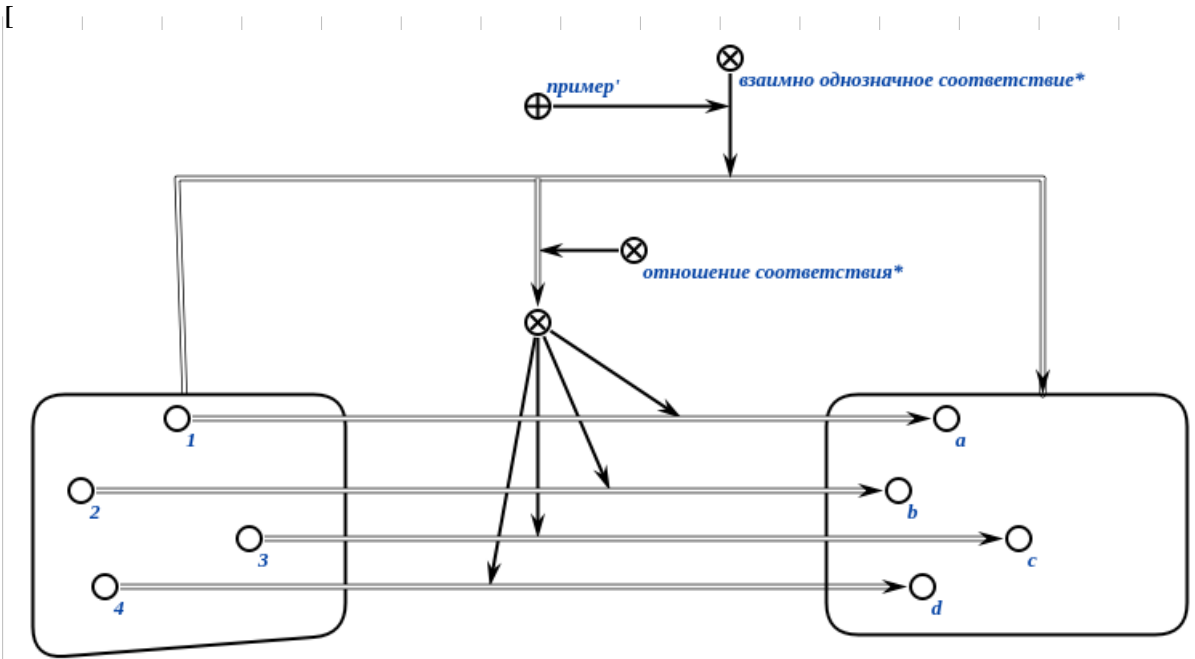
⊂ *сюръективное соответствие**

⊂ *инъективное соответствие**

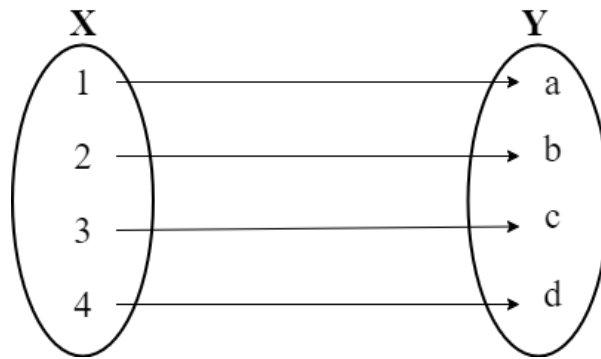
⇒ *определение**:

[**взаимно однозначное соответствие*** – это *инъективное соответствие**, являющееся *всюду определенным* и *сюръективным*.]

⇒ *описание примера**:



⇒ изображение*:



множество сочетаний*

:= [множество всевозможных сочетаний*]

:= [множество всевозможных сочетаний заданной арности из элементов заданного множества*]

:= [множество всех неориентированных связок заданной арности*]

:= [множество всех подмножеств заданной мощности*]

:= [семейство всевозможных сочетаний*]

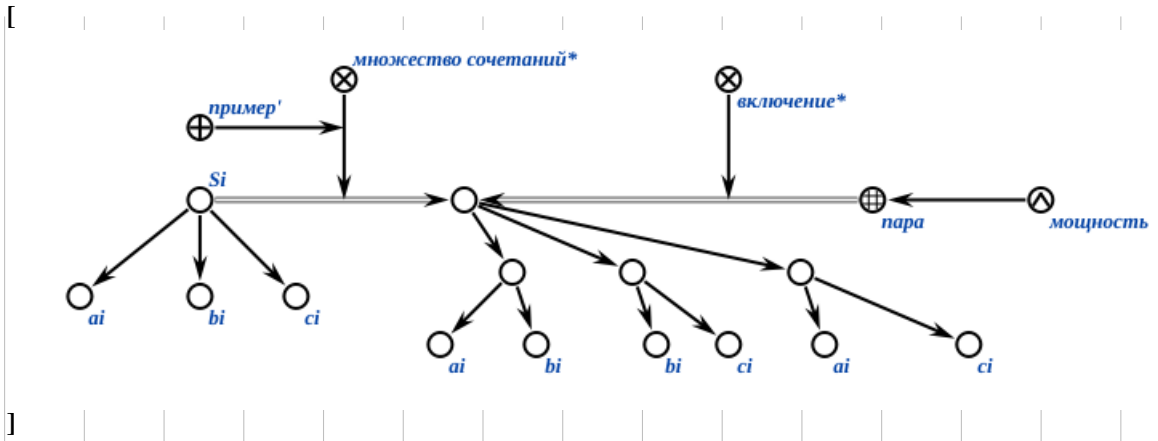
⇒ определение*:

[**множество сочетаний*** – отношение, связывающее некоторое множество и семейство всевозможных множеств, имеющих значение мощности, меньше либо равное мощности исходного множества и состоящих из тех же элементов, что и это множество.]

⇒ утверждение*:

[Мощность **множества сочетаний*** может быть вычислена как $n!/(k!(n-k)!)$, где n – мощность исходного множества, k – мощность элементов множества сочетаний.]

⇒ описание примера*:



⇒ *пояснение**:
 [Для Множества S_i представлено множество сочетаний по 2 элемента.]

множество размещений*

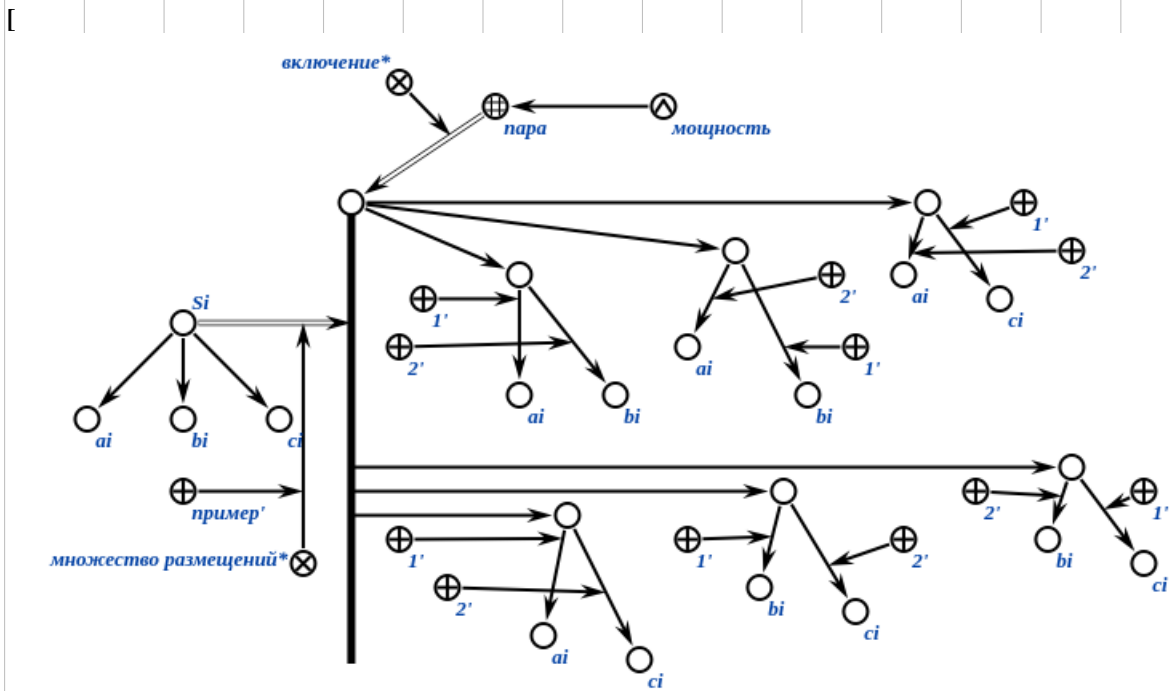
⇒ *определение**:

[**множество размещений*** – отношение, связывающее некоторое множество и семейство всевозможных кортежей, имеющих значение мощности, меньше либо равно мощности исходного множества и состоящих из тех же элементов, что и это множество.]

⇒ *утверждение**:

[Мощность **множества размещений*** может быть вычислена как $n!/(n-k)!$, где n – мощность исходного множества, k – мощность элементов множества сочетаний.]

⇒ *описание примера**:



⇒ *пояснение**:
 [Для Множества S_i представлено множество размещений по 2 элемента.]

множество перестановок*

⊂ *множество размещений**

⇒ *определение**:

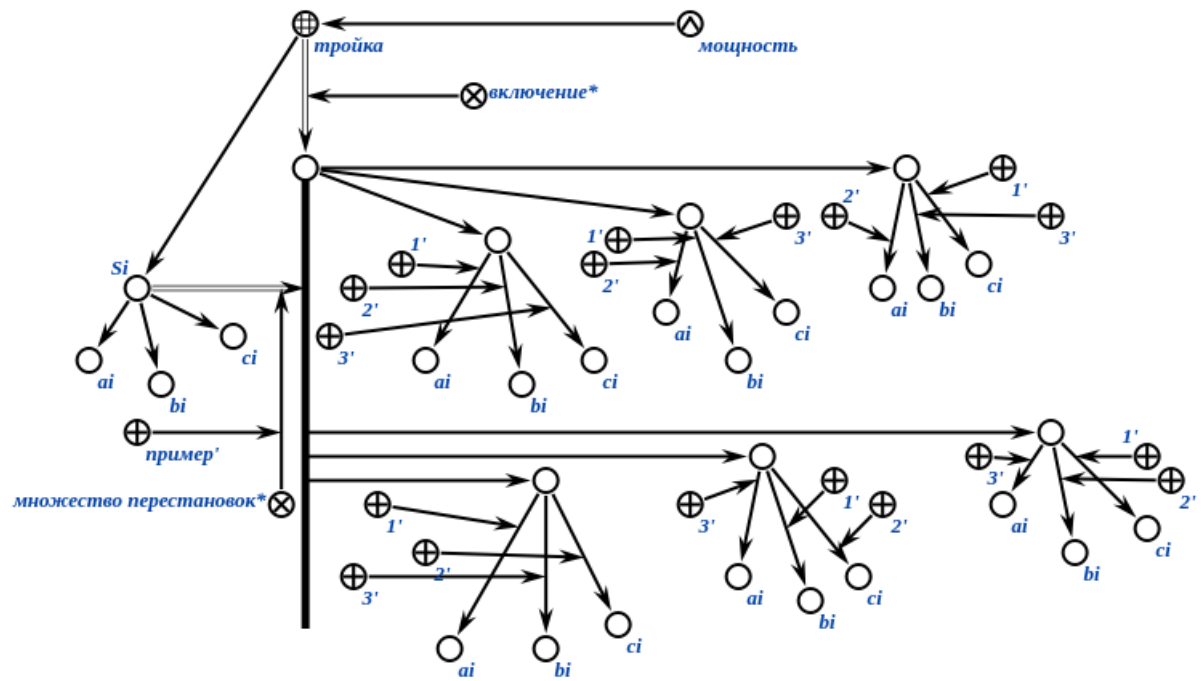
[**множество перестановок*** – отношение, связывающее некоторое множество и семейство всевозможных кортежей, равномощных исходному множеству и состоящих из тех же элементов, что и это множество.]

⇒ *утверждение**:

[Мощность **множества перестановок*** может быть вычислена как $n!$, где n – мощность исходного множества.]

⇒ описание примера*:

[



]

⇒ пояснение*:

[Для Множества S_i представлено его множество перестановок.]

} /* Завершили Раздел “Предметная область и онтология связей и отношений” */

```

/* Раздел *****/
Предметная область и онтология параметров, величин и шкал
⊃=
{
Предметная область параметров, величин и шкал
:= [[Предметная область параметров и классов эквивалентности, являющихся их элементами (значениями,
величинами)]
∈ предметная область
∋ максимальный класс объектов исследования':
параметр
∋ класс объектов исследования':
• измеряемый параметр
• неизменяемый параметр
• уровень класса эквивалентности
• величина
• точная величина
• неточная величина
• интервальная величина
• параметрическая модель
• измерение с фиксированной единицей измерения
• измерение по шкале
• арифметическое выражение на величинах
• арифметическая операция на величинах
• действие. измерение
• задача. измерение
∋ исследуемое отношение':
• область определения параметра*
• эталон'
• измерение*
• точность*
• единица измерения*
• нулевая отметка*
• единичная отметка*
• сумма величин*
• произведение величин*
• возведение величин в степень*
• большая величина*
• равенство величин*
• большая или равная величина*

параметр
:= [характеристика]
:= [свойство]
:= [признак]
:= [класс классов]
:= [измеряемое свойство]
:= [признак классификации или покрытия некоторого класса сущностей]
:= [признак разбиения или покрытия некоторого класса сущностей]
:= [семейство множеств, разбивающих или покрывающих некоторый класс сущностей]
:= [семейство классов сущностей, обладающих одинаковым соответствующим свойством]
:= [фактор-множество, соответствующее некоторому отношению эквивалентности, или аналог фактор-
множества, соответствующий некоторому отношению толерантности]
⇒ разбиение*:
{ • измеряемый параметр
• неизменяемый параметр
}
⇒ пояснение*:

```


[Каждый *параметр* представляет собой класс, являющийся семейством всевозможных классов эквивалентности или толерантности, задаваемых либо *отношением эквивалентности*, либо *отношением толерантности* (симметричным, рефлексивным, но частично транзитивным). Так, например, элементами (значениями, величинами) *параметра длина* являются либо классы эквивалентности, задаваемые отношением эквивалентности “иметь точно одинаковую длину*”, либо классы толерантности, задаваемые отношением вида “иметь приблизительно одинаковую длину с указываемой точностью*”, либо интервальные классы, задаваемые бинарными отношениями вида “иметь длину, находящуюся в одном и том же указываемом интервале*” (например, от 1 метра до 2 метров).

Примерами параметров как отношений эквивалентности являются:

- равновеликость геометрических фигур (по длине, площади, объему – в зависимости от размерности этих фигур);
- иметь одинаковый цвет (быть эквивалентными по цвету);
- эквивалентность, по вкусу, запаху, твердости и т.д.

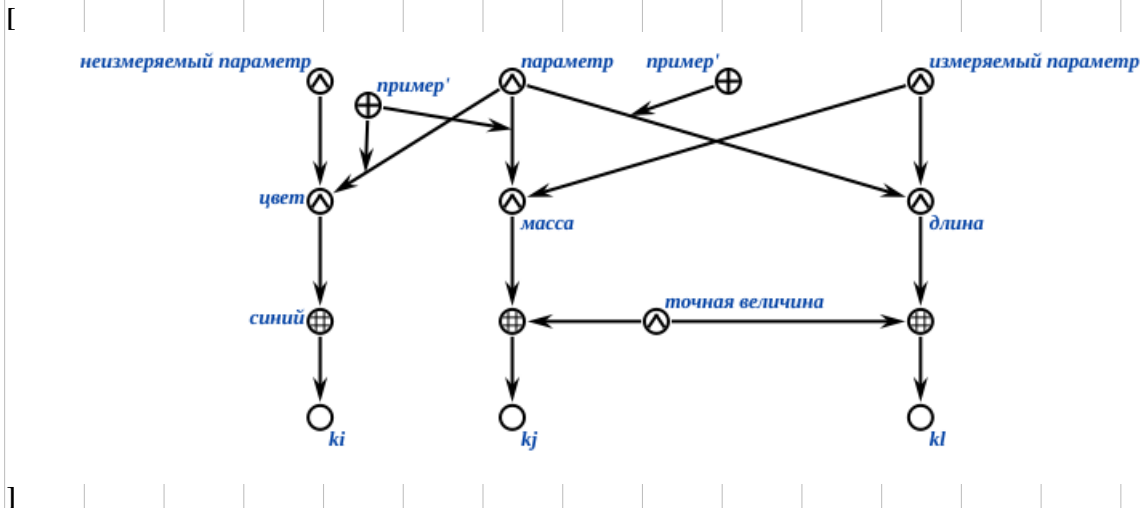
Заметим, что среди элементов (значений, величин) параметра могут встречаться пересекающиеся множества (классы), но объединение всех элементов каждого параметра есть не что иное, как класс всевозможных сущностей, обладающих этим параметром (свойством, характеристикой). Например, класс всех сущностей, имеющих длину, класс всех сущностей, обладающих цветом.

Каждый конкретный параметр (характеристика), т.е. каждый элемент класса всевозможных параметров (характеристик) есть, по сути, признак классификации сущностей, обладающих это характеристикой, по принципу эквивалентности (одинаковости значения) этой характеристики. Например, параметр *цвет* разбивает множество сущностей имеющих цвет на классы, каждый из которых включает в себя сущности, имеющие одинаковый цвет. Параметр может разбиваться на классы для уточнения некоторого свойства, например элементами параметра цвет будут классы, соответствующие конкретным цветам (синий, красный и т.д.), в свою очередь каждый конкретный цвет может включать более частные классы, уточняющие данное свойство, например, темно-синий, светло-красный и т.д.

Другими словами, каждому множеству сущностей может ставиться в соответствие набор (семейство) параметров (параметрическое пространство), которыми обладают сущности этого множества – аналог семейства отношений, определенных (заданных) на этом множестве. Часто бывает важно построить такое параметрическое пространство, “точки” которого взаимнооднозначно соответствуют параметризуемым сущностям (например, набор параметров, позволяющих однозначно идентифицировать, установить личность каждого человека).

Таким образом, для каждого используемого элемента (значения) какого-либо параметра, необходимо явно указывать спецификацию этого значения (точное значение, неточное значение, интервальное значение, точность, интервал).]

⇒ описание примера*:



*область определения параметра**

:= [множество всех тех и только тех сущностей, которые являются компонентами значений заданного параметра*]

:= [множество всех тех и только тех сущностей, которые обладают заданным параметром*]

← *включение**:
*объединение**

измеряемый параметр

:= [количественный параметр]

:= [семейство измеряемых величин]

:= [семейство классов эквивалентности, каждому из которых может быть поставлено в соответствие числовое значение]

⇒ *пояснение**:

[Каждый **измеряемый параметр** представляет собой *параметр*, значение (элемент, экземпляр) которого трактуется как *величина*, которой можно поставить в соответствие ее числовое значение на основании выбранной единицы измерения и точки отсчета (нулевой отметки выбранной шкалы).]

неизмеряемый параметр

:= [качественный параметр]

уровень класса эквивалентности

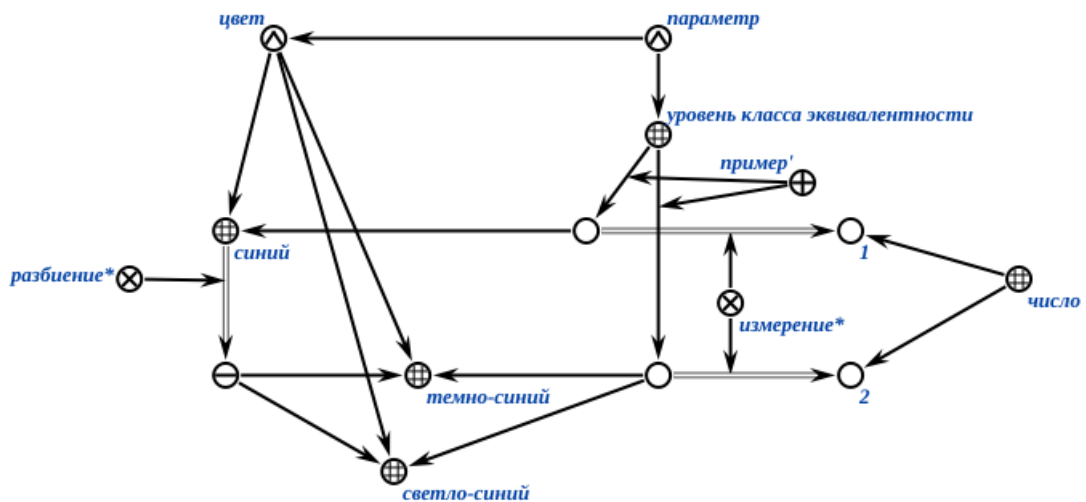
:= [уровень параметра]

∈ *параметр*⇒ *пояснение**:

[Параметр **уровень класса эквивалентности** задает уровень некоторого значения некоторого *параметра* в иерархии значений этого параметра. Уровень класса эквивалентности равен 1, если значение параметра не является частным по отношению к другому значению этого параметра, равен 2, если значение параметра является частным по отношению к значению этого параметра с уровнем 1 и т.д.]

⇒ *описание примера**:

[



]

величина

:= [значение количественного параметра]

:= [значение измеряемого параметра]

:= [класс сущностей, имеющих одинаковое значение соответствующего параметра]

⇒ *включение**:

- *точная величина*
- *неточная величина*
- *интервальная величина*

⇒ *пояснение**:

[Каждая **величина** представляет собой однозначный и независящий от шкалы измерения результат измерения некоторой характеристики у некоторой сущности.

Каждой **величине** можно поставить в соответствие ее числовое значение на основании выбранной единицы измерения и точки отсчета (нулевой отметки выбранной шкалы, в случае, если измерение осуществляется по шкале).

Нельзя путать значение параметра (*величину*) и значение величины по некоторой шкале, которое может быть скалярным и векторным.]

точная величина

:= [точное значение параметра]

:= [множество всех точных значений параметра]

:= [значение параметра, являющееся семейством классов эквивалентности, соответствующим некоторому отношению эквивалентности]

:= [класс эквивалентности]

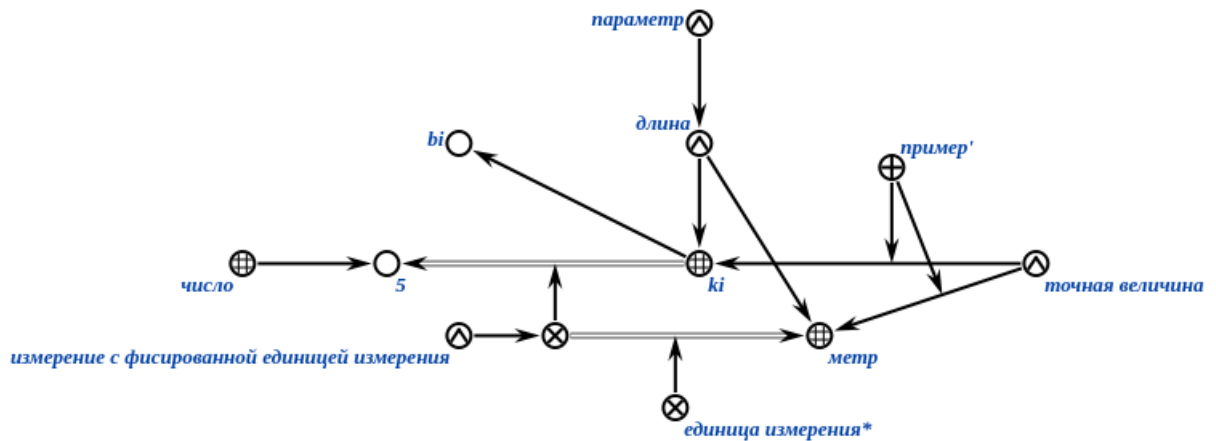
⇒ *пояснение**:

[Каждая **точная величина** имеет одно фиксированное значение в некоторой единице измерения или по какой-либо шкале. При этом считается, что все элементы такого класса имеют одинаковое значение данного параметра и отклонениями можно пренебречь.

Каждой **точной величине** можно поставить в соответствие группу *неточных величин*, являющихся ее разбиениями, а покрытиями того же множества, но с разной степенью точности.]

⇒ *описание примера**:

[



]

⇒ *пояснение**:

[В данном примере *ки* обозначает класс сущностей, имеющих длину ровно 5 метров. Конкретный пример такой сущности - *bi*.]

неточная величина

:= [множество неточных значений параметра]

:= [приблизительная величина]

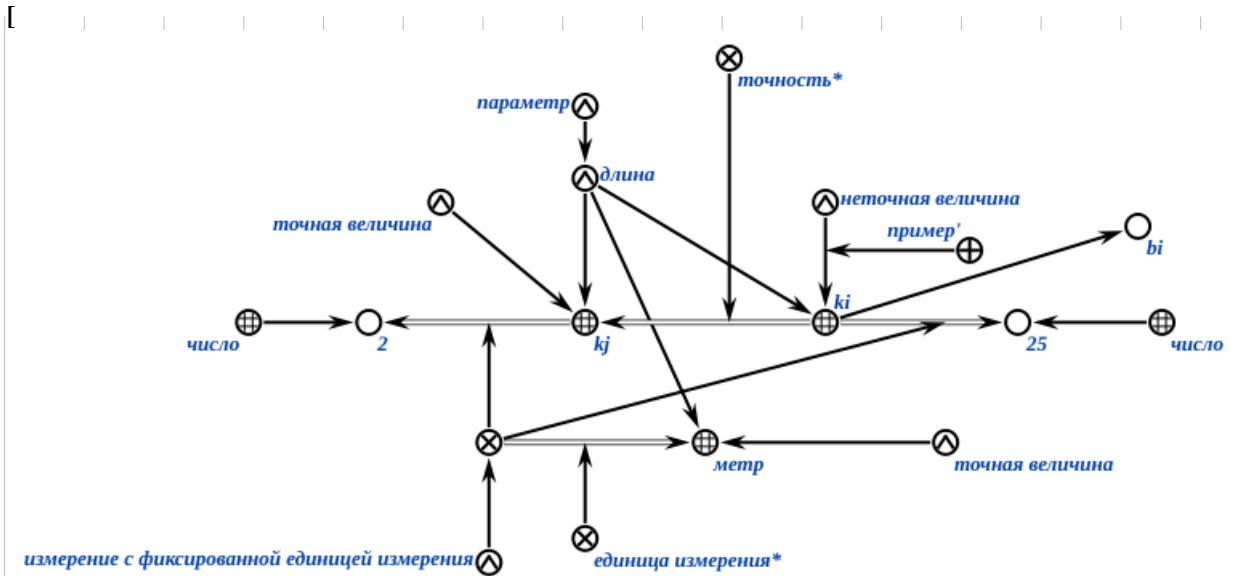
:= [приблизительное значение параметра]

:= [значение параметра в интервале с нефиксированными границами]

⇒ *пояснение**:

[Каждой **неточной величине** ставится в соответствие ее значение в некоторой единице измерения или по какой-либо шкале, а также дополнительно указывается *точность**, т.е. возможное отклонение от данного значения.]

⇒ *описание примера**:



⇒ *пояснение**:
 [В данном примере ki обозначает класс сущностей, имеющих длину примерно 25 метров, при этом максимально возможное отклонение от этого значения составляет kj , то есть 2 метра. Конкретный пример такой сущности - bi .]

интервальная величина

:= [интервальное значение параметра]

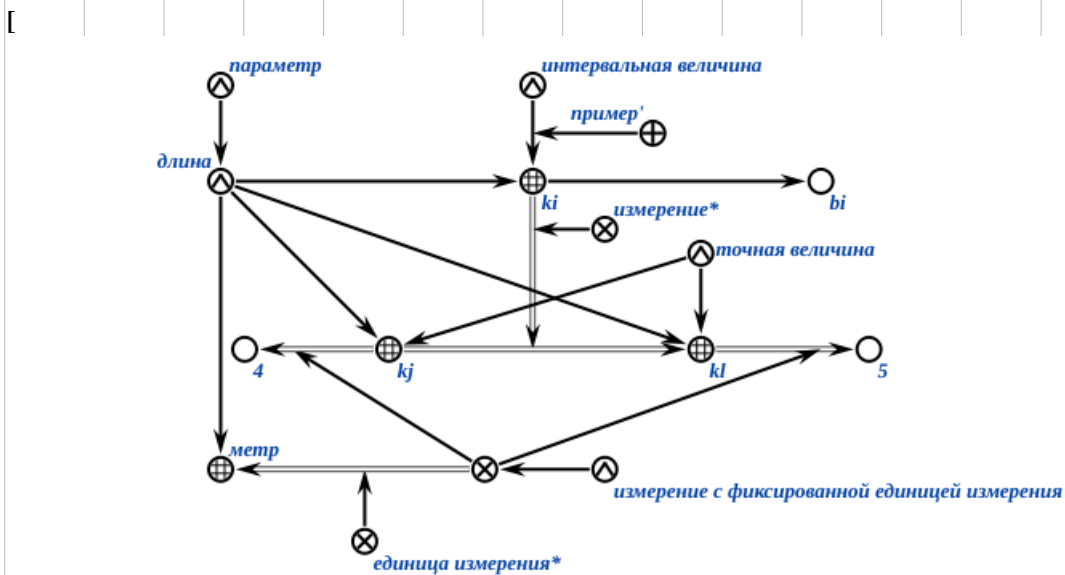
:= [значение параметра в интервале с фиксированными границами]

:= [интервал значения параметра из множества пересекающихся интервалов разной длины, имеющих нефиксированные границы]

⇒ *пояснение**:

[Каждая **интервальная величина** представляет собой класс сущностей, находящихся в рамках точно заданного интервала, минимальная и максимальная точка которого являются *точными величинами*. Результатом *измерения** такой величины является ориентированная пара, первым компонентом которой является левая (меньшая) граница интервала, вторым компонентом – правая (большая) граница интервала.]

⇒ *описание примера**:



⇒ *пояснение**:
 [В данном примере ki обозначает класс сущностей, имеющих длину, которая лежит в интервале от kj до kl , то есть в интервале от 4 до 5 метров, а bi – конкретный пример такой сущности.]

эталон'

:= [образец']

∈ *ролевое отношение*⇒ *пояснение**:

[Ролевое отношение *эталон'* указывает на тот элемент значения некоторого параметра, который в рамках данного класса эквивалентности считается эталонным, то есть он используется как образец при определении данного параметра.

эталон' может задаваться как для измеряемых, так и для неизменяемых параметров, например, эталон метра или эталон красоты.]

измерение*

:= [значение параметра*]

:= [значение заданной величины заданного параметра*]

:= [измерение как соответствие*]

:= [результат измерения заданной величины в заданной единице измерения и по заданной шкале*]

:= [бинарное ориентированное отношение, связывающее различные величины с результатами их измерения в различных единицах измерения и по различным шкалам*]

⇒ *пояснение**:

[Связки отношения *измерение** связывают величину и ее значение в некоторой единице измерения (в том числе, в интервале) или по некоторой шкале. Конкретная единица измерения или шкала указывается дополнительно при помощи соответствующего отношения. Одной величине может соответствовать только одно значение в каждой возможной единице измерения или одна точка на некоторой шкале.]

точность*

:= [отклонение*]

:= [степень точности неточного значения параметра*]

∈ *бинарное отношение*⇒ *пояснение**:

[Связки отношения *точность** связывают *неточную величину* и *точную величину* того же класса, задающую максимальное возможное отклонение указанной *неточной величины* от своего значения.]

параметрическая модель

:= [параметрическая спецификация]

:= [параметрическое sc-описание заданной сущности]

:= [описание сущности как точки в некотором параметрическом (признаковом) пространстве]

← *включение**:*семантическая окрестность*⇒ *пояснение**:

[Каждая *параметрическая модель* представляет собой описание заданной сущности в некотором параметрическом пространстве количественных и качественных *параметров*, т.е. указание того, какие значения заданных параметров (характеристик) соответствуют описываемой (заданной) сущности.]

единица измерения*∈ *бинарное отношение*

:= [единица по шкале*]

:= [единичная отметка по шкале*]

⇒ *пояснение**:

[Связки отношения *единица измерения** связывают знак конкретного *измерения с фиксированной единицей измерения* и некоторую *точную величину*, входящую в тот же конкретный *параметр*, что и первый компонент связок этого конкретного измерения, и которая используется в данном случае в качестве единицы измерения.]

измерение с фиксированной единицей измерения← *семейство подмножеств**:*измерение**⇒ *пояснение**:

[Каждая *измерение с фиксированной единицей измерения* представляет собой подмножество отношения *измерение** и характеризуется некоторой *единицей измерения**, которая является элементом того же параметра (семейством сущностей, имеющих значение данного параметра, совпадающее с этой единицей измерения).]

измерение по шкале

:= [шкала]

← *семейство подмножеств**:

*измерение**

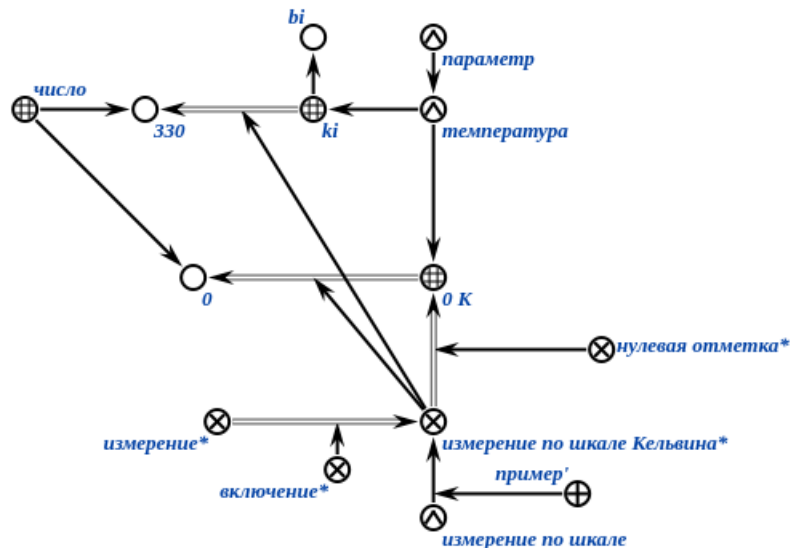
⇒ *пояснение**:

[Каждая *измерение по шкале* представляет собой подмножество отношения *измерение** и характеризуется не единицей измерения, а некоторой точкой отсчета для данной *шкалы*. Результатом *измерения по шкале* будет некоторая точка шкалы, отстоящая от точки отсчета на определенное расстояние в нужную сторону (меньшую или большую). Понятно, что это расстояние может быть измерено любыми единицами измерения, но его величина при этом останется неизменной.

Не стоит путать *измерение по шкале*, которое зависит от *нулевой отметки**, с измерением изменения того же *параметра*, которое характеризуется единицей измерения и не зависит от точки отсчета. Например, не стоит путать дату по некоторому календарю, соответствующую *началу* какого-либо процесса, и *длительность* этого процесса, которая не зависит от выбранного календаря.]

⇒ *описание примера**:

[



]

⇒ *пояснение**:

[В данном примере *ki* обозначает класс сущностей, имеющих точную температуру в 330 К, а *bi* – конкретный пример такой сущности.]

*нулевая отметка**

:= [нуль по шкале*]

:= [начало отсчета*]

:= [точка отсчета*]

∈ *бинарное отношение*

⇒ *пояснение**:

[Связки отношения *нулевая отметка** связывают знак некоторого *измерения по шкале* со знаком *точной величины* того же *параметра*, которая в рамках данной шкалы принимается за точку отсчета.]

арифметическое выражение на величинах

⇒ *пояснение**:

[Каждое *арифметическое выражение на величинах* представляет собой *связку*, компонентами которой являются элементы или подмножества некоторого *количественного параметра*.]

арифметическая операция на величинах

← семейство подмножеств*:
 арифметическое выражение на величинах
 ⇒ пояснение*:

[Каждая **арифметическая операция на величинах** представляет собой *отношение*, элементами которого являются *арифметические выражения на величинах*, то есть множество *арифметических выражений на величинах* какого-либо одного вида.]

сумма величин*

:= [сложение величин*]
 ∈ арифметическая операция на величинах
 ∈ квазибинарное отношение
 ⇒ пояснение*:

[**сумма величин*** – это арифметическая операция на величинах, аналогичная арифметической операции *сумма** для чисел.

Первым компонентом связки отношения **сумма величин*** является подмножество некоторого *количественного параметра* (слагаемые *величины*), содержащее два или более элемента, вторым компонентом – элемент этого же *количественного параметра*, значение которого в любой *единице измерения** является результатом сложения значений всех слагаемых *величин* в той же *единице измерения**. При несовпадении *единиц измерения* слагаемых величин необходимо воспользоваться соотношениями между *единицами измерения*, которые задаются при помощи операций *произведение величин** и *возведение величин в степень**.]

произведение величин*

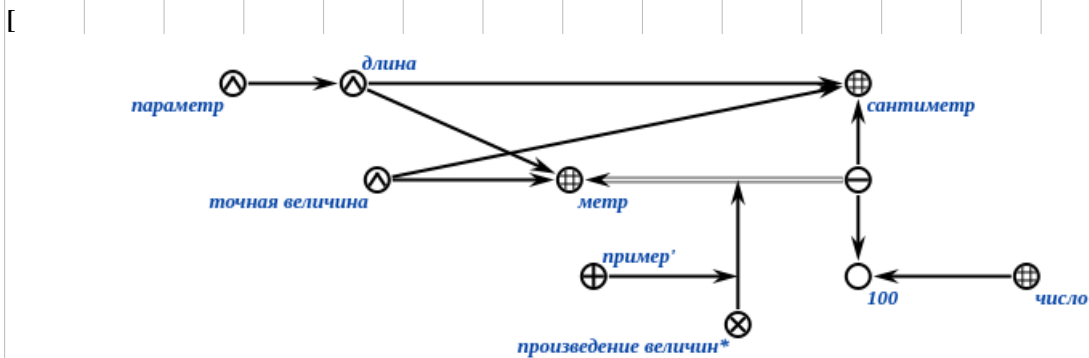
:= [умножение величин*]
 ∈ арифметическая операция на величинах
 ∈ квазибинарное отношение
 ⇒ пояснение*:

[**произведение величин*** – это арифметическая операция на величинах, аналогичная арифметической операции *произведения** для чисел.

Первым компонентом связки отношения **произведение величин*** является *связка*, элементами которой являются либо *величины количественных параметров*, либо *числа*, но при этом хотя бы один элемент должен быть *величиной*. Вторым компонентом является *величина количественного параметра*.

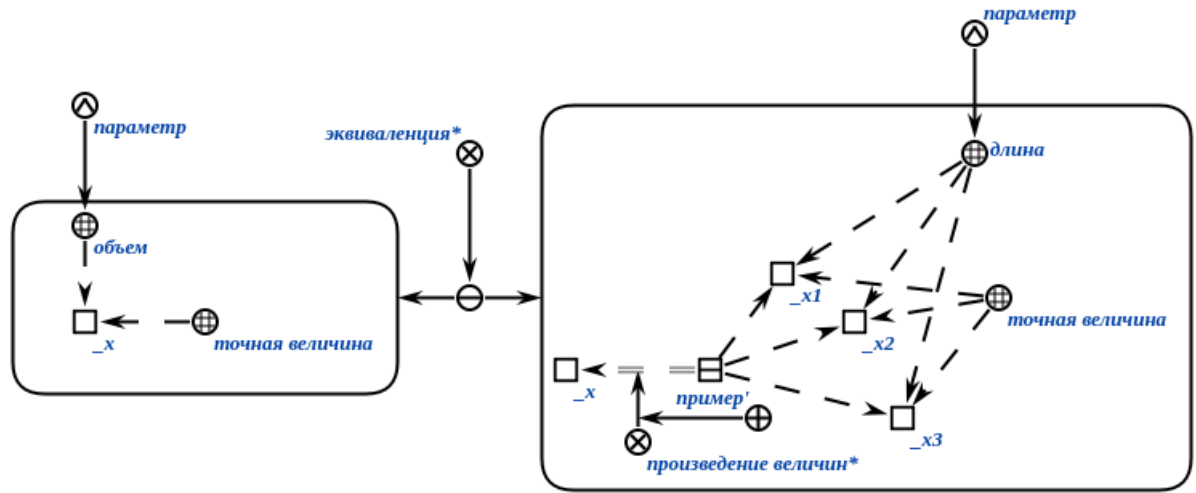
Операция **произведение величин*** может быть использована для задания соотношения между *единицами измерения** в рамках одного *количественного параметра*.]

⇒ описание примера*:



] описание примера*:

⇒



возведение величин в степень*

∈ арифметическая операция на величинах

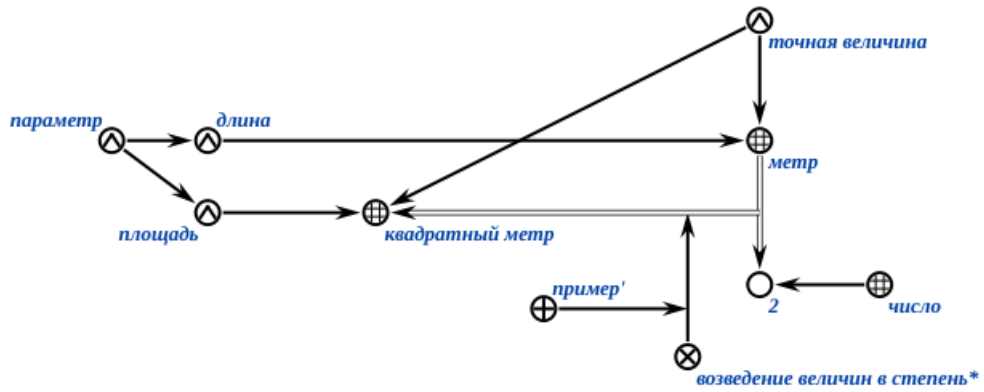
∈ бинарное отношение

⇒ пояснение*:

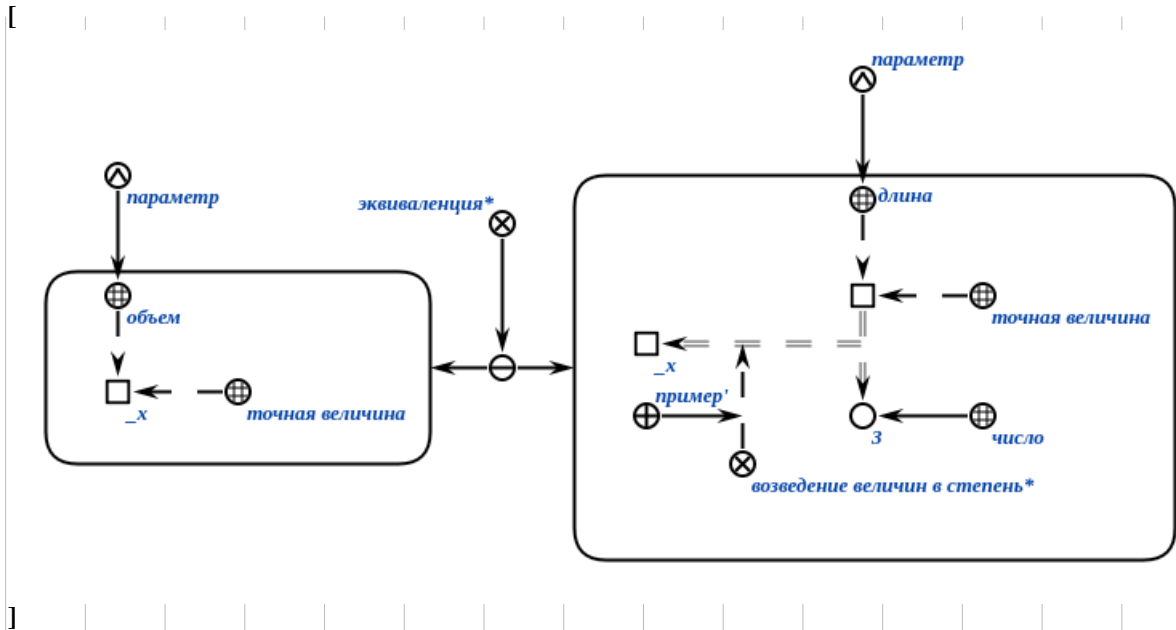
[**возведение величин в степень*** – это арифметическая операция на величинах, аналогичная арифметической операции возведение в степень* для чисел.

Первым компонентом связки отношения **возведение величин в степень*** является ориентированная пара, первым компонентом которой является величина количественного параметра (основание степени), вторым – число (показатель степени); Вторым компонентом связки отношения **возведение величин в степень*** является величина количественного параметра (результат возведения в степень).]

⇒ описание примера*:



⇒ описание примера*:



порядок величин*

- := [большая величина*]
- := [сравнение величин*]
- ∈ бинарное отношение
- ∈ отношение строгого порядка
- ⇒ пояснение*:

[**порядок величин*** – это отношение на величинах, аналогичное отношению *больше** для чисел.

Из двух величин большей является та, значение которой в любой единице измерения* больше* значения другой величины в той же единице измерения*. В ряде случаев нет возможности говорить о большей или меньшей величине, но можно говорить об их упорядоченности.]

действие. измерение

- := [измерение как действие]
- := [действие, направленное на установление связи, принадлежащей отношению измерение* и связывающей величину, которая принадлежит заданному параметру, и которой принадлежит заданная сущность, и соответствующее значение этой величины на некоторой шкале]
- := [действие, направленное на решение задачи измерения заданного параметра у заданной сущности]
- ← включение*:
действие

задача. измерение

- := [спецификация действия измерения]
- := [спецификация действия, целью которого является измерение заданного параметра у заданной сущности]
- ← включение*:
задача

} /* Завершили Раздел “Предметная область и онтология параметров, величин и шкал” */

/* Раздел *****/

Предметная область и онтология чисел и числовых структур

⊃=
{

Предметная область чисел и числовых структур

- ∈ предметная область
- ∋ максимальный класс объектов исследования¹:
число
- ∋ класс объектов исследования¹:
 - натуральное число
 - целое число
 - рациональное число
 - иррациональное число
 - действительное число
 - комплексное число
 - отрицательное число
 - положительное число
 - арифметическое выражение
 - арифметическая операция
 - Число Пи
 - Ноль
 - Единица
 - Мнимая единица
 - числовая структура
 - система счисления
 - десятичная система счисления
 - двоичная система счисления
 - шестнадцатеричная система счисления
 - дробь
 - обыкновенная дробь
 - десятичная дробь
 - цифра
 - арабская цифра
 - римская цифра
- ∋ исследуемое отношение¹:
 - противоположные числа*
 - модуль*
 - сумма*
 - произведение*
 - возведение в степень*
 - больше*
 - равенство*
 - больше или равно*

число

- := [множество чисел]
- ⊂ абстрактная терминальная сущность
- ⇒ пояснение*:

[**число** – это основное понятие математики, используемое для количественной характеристики, сравнения, нумерации объектов и их частей. Письменными знаками для обозначения чисел служат *цифры*.]

цифра

- := [множество цифр]
- ⊂ внутренний файл *ostis-системы*
- ⇒ включение*:
 - арабская цифра
 - римская цифра
- ⇒ пояснение*:

[**цифра** — это множество файлов, обозначающих вхождение этой цифры во всевозможные записи чисел с помощью этой цифры.]

натуральное число

:= [множество натуральных чисел]

⇒ *пояснение**:

[**натуральное число** — это подмножество множества *целых чисел*, которые используются при счете предметов.]

С *целое число*

целое число

:= [множество целых чисел]

⇒ *пояснение**:

[**целое число** — это подмножество множества *рациональных чисел*, получаемых объединением *натуральных чисел* с множеством чисел, *противоположных** натуральным и нулём.]

С *рациональное число*

рациональное число

:= [множество рациональных чисел]

⇒ *пояснение**:

[**рациональное число** — это число, представляемое *обыкновенной дробью*, где числитель — *целое число*, а знаменатель — *натуральное число*.]

С *действительное число*

дробь

:= [множество дробей]

⇒ *включение**:

- *обыкновенная дробь*
- *десятичная дробь*

⇒ *пояснение**:

[**дробь** — это число, состоящее из одной или нескольких равных частей (долей) единицы]

обыкновенная дробь

:= [множество обыкновенных дробей]

:= [множество простых дробей]

⇒ *пояснение**:

[**обыкновенная дробь** - запись *рационального числа* в виде $\pm \frac{m}{n}$ или $\pm m/n$, где $n \neq 0$. Горизонтальная или косая черта обозначает знак деления, в результате которого получается частное. Делимое называется числителем дроби, а делитель — знаменателем.]

десятичная дробь

:= [множество десятичных дробей]

⇒ *пояснение**:

[**десятичная дробь** - Десятичная дробь — разновидность дроби, которая представляет собой способ представления действительных чисел в виде $\pm d_m \dots d_1 d_0, d_{-1} d_{-2} \dots$, где , — десятичная запятая, служащая разделителем между целой и дробной частью числа, $d_k m$ — десятичные цифры.]

иррациональное число

:= [множество иррациональных чисел]

⇒ *пояснение**:

[**иррациональное число** — это *вещественное число*, которое не является рациональным, то есть не может быть представлено в виде дроби, где числитель — *целое число*, знаменатель — *натуральное число*. Любое **иррациональное число** может быть представлено в виде бесконечной непериодической десятичной дроби.]

С *действительное число*

действительное число

:= [вещественное число]

:= [множество вещественных чисел]

← *объединение**:

	{	• рациональное число
		• иррациональное число
	}	
⇒	разбиение*:	
	{	• положительное число
		• отрицательное число
		• {Нуль}
	}	
⇒	пояснение*:	
		[действительное число – это множество чисел, получаемое в результате объединения иррациональных и рациональных чисел.]
С	комплексное число	
	комплексное число	
:=		[множество комплексных чисел]
⇒	пояснение*:	
		[комплексное число – число вида $z=a+bi$, где a и b – вещественные числа, i – Мнимая единица.]
	отрицательное число	
:=		[множество отрицательных чисел]
⇒	пояснение*:	
		[отрицательное число – число меньше* нуля.]
	положительное число	
:=		[множество положительных чисел]
⇒	пояснение*:	
		[положительное число – число больше* нуля.]
	противоположные числа*	
∈		бинарное неориентированное отношение
⇒	пояснение*:	
		[противоположные числа* – отношение, связывающее два числа, одно из которых является отрицательным числом, второе – положительным, при этом модули* этих чисел равны*.]
	модуль*	
:=		[модуль числа*]
∈		бинарное отношение
⇒	пояснение*:	
		[Связки отношения модуль* связывают некоторое число (которое может быть как отрицательным, так и положительным) и другое число (всегда положительное), которое выражает расстояние от указанного числа до Нуля в единицах.]
	арифметическое выражение	
:=		[множество арифметических выражений]
⇒	пояснение*:	
		[Каждое арифметическое выражение представляет собой связку, компонентами которой являются числа или множества чисел.]
	арифметическая операция	
:=		[множество арифметических операций]
⇐		семейство подмножеств*:
		арифметическое выражение
⇒	пояснение*:	
		[Каждая арифметическая операция представляет собой отношение, элементами которого являются арифметические выражения, то есть множество арифметических выражений какого-либо одного вида.]
	сумма*	
:=		[сложение*]
∈		арифметическая операция

∈ квазибинарное отношение
 ⇒ пояснение*:

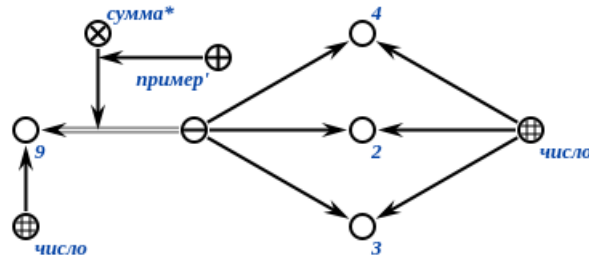
[**сумма*** – это арифметическая операция, в результате которой по данным числам (слагаемым) находится новое число (сумма), обозначающее столько единиц, сколько их содержится во всех слагаемых.

Первым компонентом связки отношения **сумма*** является множество чисел (слагаемых), содержащее два или более элемента, вторым компонентом – число, являющееся результатом сложения.

Отдельно отметим, что каждая связка отношения **сумма*** вида $a = b+c$ может также трактоваться и как запись о вычитании чисел, например $b = a-c$, в связи с чем арифметическая операция разности чисел отдельно не вводится.]

⇒ описание примера*:

[



]

произведение*

:= [умножение*]
 ∈ арифметическая операция
 ∈ квазибинарное отношение
 ⇒ пояснение*:

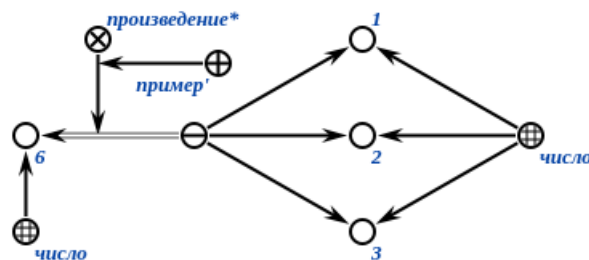
[**произведение*** – это арифметическая операция, в результате которой один аргумент складывается столько раз, сколько показывает другой, затем результат складывается столько раз, сколько показывает третий и т.д.

Первым компонентом связки отношения **произведение*** является множество чисел (множителей), содержащее два или более элемента, вторым компонентом – число, являющееся результатом произведения.

Отдельно отметим, что каждая связка отношения **произведение*** вида $a = b*c$ может также трактоваться и как запись о делении чисел, например $b = a/c$, в связи с чем арифметическая операция деления чисел отдельно не вводится.]

⇒ описание примера*:

[



]

возведение в степень*

∈ арифметическая операция
 ∈ бинарное отношение
 ⇒ пояснение*:

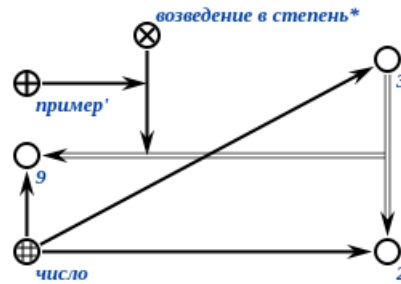
[**возведение в степень*** – это арифметическая операция, в результате которой число, называемое основанием степени, умножается само на себя столько раз, каков показатель степени.

Первым компонентом связки отношения **возведение в степень*** является ориентированная пара, первым компонентом которой является число, которое является основанием степени, вторым – число, которое является показателем степени; Вторым компонентом связки отношения **возведение в степень*** является число, которое является результатом возведения в степень.

Отдельно отметим, что каждая связка отношения **возведение в степень*** вида $a = b^c$ может также трактоваться и как запись об извлечении корня или взятии логарифма, в связи с чем *арифметические операции* извлечения корня и взятия логарифма отдельно не вводятся.]

⇒ описание примера*:

[



]

больше*

∈ бинарное отношение

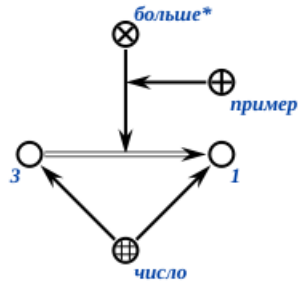
∈ отношение строгого порядка

⇒ пояснение*:

[**больше*** – это бинарное отношение сравнения чисел. Из двух чисел на координатной прямой больше то, которое расположено правее. Соответственно, первым компонентом связки отношения **больше*** является большее из двух чисел.]

⇒ описание примера*:

[



]

больше или равно*

∈ бинарное отношение

∈ отношение нестрогого порядка

⇒ пояснение*:

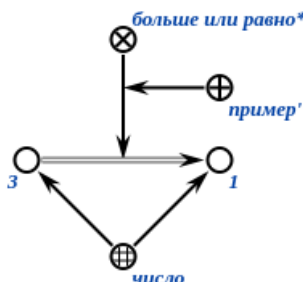
[**больше или равно*** – это бинарное отношение сравнения чисел, при которой первое число (первый компонент связки) может быть **больше*** второго или **равняться*** ему.]

⇒ примечание*:

[Отношение **равенство*** явно не вводится, поскольку в рамках SC-кода равные числа трактуются как совпадающие числа, то есть обозначаемые одним и тем же *sc-элементом*.]

⇒ описание примера*:

[



]

Число Π

∈ иррациональное число

⇒ пояснение*:

[**Число Π** – это математическая константа, равная отношению длины окружности к длине её диаметра.]

Ноль

:= [0]

∈ целое число

⇒ пояснение*:

[**Ноль** – это целое число, разделяющее на числовой прямой *положительные числа* и *отрицательные числа*.]

Единица

:= [1]

∈ целое число

∈ натуральное число

⇒ пояснение*:

[**Единица** – это наименьшее *натуральное число*.]

Мнимая единица

:= [i]

∈ комплексное число

⇒ пояснение*:

[**Мнимая единица** – это число, при возведении которого в степень 2 результатом будет число, противоположное *Единице*.]

числовая структура

С структура

⇒ пояснение*:

[**числовая структура** – структура, в состав которой входят знаки *арифметических выражений*, а также знаки их элементов и связи между выражениями и их элементами.]

система счисления

∈ параметр

⇒ пояснение*:

[Каждая **система счисления** представляет собой класс синтаксически эквивалентных файлов, хранимых в *sc-памяти*, каждый из которых может являться идентификатором какого-либо *числа*.

Каждая **система счисления** характеризуется алфавитом, т.е. конечным множеством символов (цифр), которые допускается использовать при построении файлов принадлежащих данной **системе счисления**.]

десятичная система счисления

∈ система счисления

двоичная система счисления

∈ система счисления

шестнадцатеричная система счисления

∈ система счисления

} /* Завершили Раздел “Предметная область и онтология чисел и числовых структур” */

/* Раздел *****/	
Предметная область и онтология структур	
⊃=	
{	
Предметная область структур	
∃	максимальный класс объектов исследования': структура
∃	класс объектов исследования': <ul style="list-style-type: none"> • связная структура • несвязная структура • тривиальная структура • нетривиальная структура • структура второго уровня • семантический уровень структурного элемента • количество семантических уровней элементов структуры
∃	исследуемое отношение': <ul style="list-style-type: none"> • элемент структуры' • непредставленное множество' • полностью представленное множество' • частично представленное множество' • элемент структуры, не являющийся множеством' • максимальное множество' • немаксимальное множество' • первичный элемент' • вторичный элемент' • элемент второго уровня' • метасвязь' • полиморфность* • полиморфизм* • гомоморфность* • гомоморфизм* • изоморфность* • изоморфизм* • автоморфность* • автоморфизм* • аналогичность структур* • сходство* • различие* • первичная синтаксическая структура sc-текста*
структура	
:=	[sc-структура]
:=	[структура, представленная в виде текста SC-кода]
⇒	разбиение*: <ul style="list-style-type: none"> { • связная структура • несвязная структура }
⇒	разбиение*: <ul style="list-style-type: none"> { • тривиальная структура • нетривиальная структура }
⇒	пояснение*: <p>[структура – множество sc-элементов, удаление одного из которых может привести к нарушению целостности этого множества.]</p>
⇒	разбиение*: <ul style="list-style-type: none"> { • процесс <ul style="list-style-type: none"> := [динамическая структура] := [нестационарная структура]

- *статическая структура*
:= [стационарная структура]
:= [структура, не изменяющаяся во времени]
- ⇒ разбиение*:
- *временная структура*
:= [(структура \cap временная сущность)]
 - *постоянно существующая структура*
:= [(структура \cap постоянная сущность)]

связная структура⇒ *пояснение**:

[*Структуре*, представленной в *SC-коде*, поставим в соответствие орграф, вершинами которого являются *sc-элементы*, а дугами – пары инцидентности, связывающие *sc-коннекторы* с инцидентными им *sc-элементами*, которые являются компонентами указанных *sc-коннекторов*.

Если полученный таким способом орграф является связным орграфом, то исходную структуру будем считать *связной структурой*.]

несвязная структура⇒ *пояснение**:

[*Структуре*, представленной в *SC-коде*, поставим в соответствие орграф, вершинами которого являются *sc-элементы*, а дугами – пары инцидентности, связывающие *sc-коннекторы* с инцидентными им *sc-элементами*, которые являются компонентами указанных *sc-коннекторов*.

Если полученный таким способом орграф не является связным орграфом, то исходную структуру будем считать *несвязной структурой*.]

тривиальная структура

:= [структура первого уровня]

⇒ *пояснение**:

[*тривиальная структура* – структура, не содержащая в качестве элементов связей.]

нетривиальная структура \supset структура второго уровня⇒ *пояснение**:

[*нетривиальная структура* – структура, среди элементов которой есть хотя бы одна связька.]

элемент структуры' \in неосновное понятие

⇒ разбиение*:

- *непредставленное множество'*
- *полностью представленное множество'*
- *частично представленное множество'*
- *элемент структуры, не являющийся множеством'*

⇒ разбиение*:

- *максимальное множество'*
- *немаксимальное множество'*

⇒ *пояснение**:

[*элемент структуры'* – неосновное понятие, ролевое отношение, указывающее на все элементы каждой структуры.

В рамках заданной структуры ее элементы можно классифицировать по заданным признакам:

- насколько полно в рамках заданной структуры представлено множество, обозначаемое заданным sc-элементом вместе с соответствующими дугами принадлежности;
- существуют ли в рамках заданной структуры *sc-элементы*, обозначающие множества, являющиеся надмножествами того множества, которое обозначается заданным sc-элементом;

- уровень («этаж») иерархии перехода от знаков к метазнамкам для заданного sc-элемента в рамках заданной *структуры*.

]

непредставленное множество'

:= [множество, не представленное в рамках данной структуры']

:= [быть знаком множества, элементы которого не являются элементами данной структуры']

∈ *ролевое отношение*

⇒ *пояснение**:

[**непредставленное множество'** – *ролевое отношение*, связывающее структуру со знаком множества, все элементы которого не являются элементами данной структуры.]

полностью представленное множество'

:= [множество, полностью представленное в рамках данной структуры']

:= [множество, все элементы которого являются элементами данной структуры']

:= [полностью представленный класс']

∈ *ролевое отношение*

⇒ *пояснение**:

[**полностью представленное множество'** – *ролевое отношение*, связывающее структуру со знаком множества (любого семантического типа – класса, связки или структуры), все элементы которого являются элементами данной структуры.]

частично представленное множество'

:= [множество, частично представленное в рамках данной структуры']

:= [множество, некоторые элементы которого являются элементами данной структуры']

:= [быть знаком множества, некоторые элементы которого являются элементами данной структуры']

∈ *ролевое отношение*

⇒ *пояснение**:

[**частично представленное множество'** – *ролевое отношение*, связывающее структуру со знаком множества, не все элементы которого являются элементами данной структуры.]

элемент структуры, не являющийся множеством'

∈ *ролевое отношение*

максимальное множество'

⇒ *пояснение**:

[**максимальное множество'** – *ролевое отношение*, связывающее структуру со знаком множества, для которого не существует множества, которое было бы надмножеством указанного множества и знак которого был бы элементом этой же структуры.]

немаксимальное множество'

⇒ *пояснение**:

[**немаксимальное множество'** – *ролевое отношение*, связывающее структуру со знаком множества, для которого в рамках данной структуры существует множество, являющееся надмножеством указанного множества.]

первичный элемент'

:= [первичный элемент данной структуры']

:= [sc-элемент первого уровня в рамках данной структуры']

∈ *ролевое отношение*

∈ *семантический уровень структурного элемента*

С *элемент структуры'*

⇒ *пояснение**:

[**первичный элемент'** – *ролевое отношение*, указывающее на элемент структуры, являющийся либо терминальным элементом, либо знаком множества, такого что не существует другого элемента этой же структуры, который был бы элементом множества, обозначаемого первым из указанных элементов структуры. При этом соответствующая пара принадлежности может существовать, но в состав данной структуры не входить.]

вторичный элемент'

:= [вторичный элемент данной структуры']

:= [элемент данной структуры имеющий семантический уровень более 2']

:= [непервичный элемент']

∈ *ролевое отношение*

⊂ *элемент структуры'*

⇒ *пояснение**:

[**вторичный элемент'** – ролевое отношение, указывающее на элемент структуры, обозначающий множество, все или некоторые элементы которого являются элементами указанной структуры.]

⊃ *элемент второго уровня'*

элемент второго уровня'

∈ *ролевое отношение*

∈ *семантический уровень структурного элемента*

⇒ *пояснение**:

[**элементом второго уровня'** в рамках заданной *структуры* может быть связка первичных элементов, тривиальная структура из первичных элементов или класс первичных элементов.]

структура второго уровня'

⇒ *пояснение**:

[**структура второго уровня'** - *структура*, среди элементов которой есть хотя бы один *элемент второго уровня'*.]

семантический уровень структурного элемента

∈ *параметр*

⇒ *пояснение**:

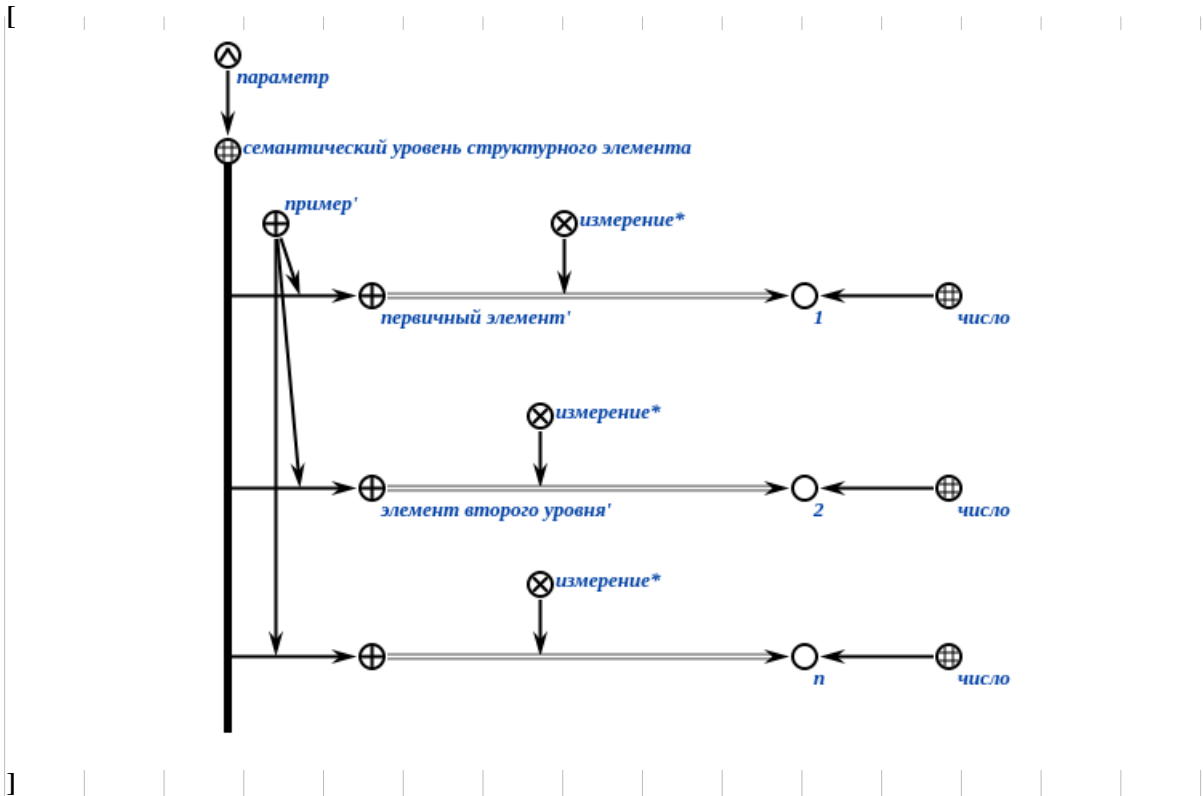
[**семантический уровень структурного элемента** представляет собой *параметр*, каждый элемент которого является классом *sc-дуг принадлежности*, связывающих некоторую *структуру* с теми ее элементами, который имеют одинаковый семантический уровень в рамках данной структуры. Значением данного параметра является число, обозначающее указанный семантический уровень.

семантический уровень структурного элемента вычисляется следующим образом:

- элементы структуры, входящие в нее с атрибутом *первичный элемент'* имеют семантический уровень 1;
- уровень элемента, не являющегося *первичным элементом'* структуры вычисляется путем прибавления 1 к максимальному из уровней элементов этого элемента (множества), входящих в эту же структуру. Например, *sc-дуга*, соединяющая два *первичных элемента'* структуры будет иметь семантический уровень 2, а *sc-элемент*, обозначающий отношение, которому принадлежит указанная *sc-дуга* – семантический уровень 3.

]

⇒ *описание примера**:



количество семантических уровней элементов структуры

∈ параметр
⇒ пояснение*:

[**количество семантических уровней элементов структуры** – параметр, каждый элемент которого представляет собой класс структур, у которых совпадает максимальный среди семантических уровней элементов этих структур.

Значением данного параметра является число, совпадающее с указанным максимальным семантическим уровнем элементов.]

метасвязь'

∈ ролевое отношение
С вторичный элемент'
⇒ пояснение*:

1. Каждая входящая в структуру связь, хотя бы одним компонентом которой является связь, входящая в эту же структуру, элементами которой являются *первичные элементы'* этой структуры, является **метасвязью'** указанной структуры;
2. Каждая входящая в структуру связь, хотя бы одним компонентом которой является **метасвязь'** этой структуры также является **метасвязью'** указанной структуры;

]

полиморфность*

С соответствие*
∈ бинарное отношение
⇒ пояснение*:

[**полиморфность*** - это соответствие, заданное на структурах, при котором каждому элементу из области определения соответствия (первой структуры) ставится в соответствие один или более элемент из области значения соответствия (второй структуры), при этом существует хотя бы один элемент области определения соответствия, которому соответствуют два или более элемента из области значения соответствия.]

полиморфизм*

∈ бинарное отношение

гомоморфность*

:= [гомоморфность структур*]

С соответствие*

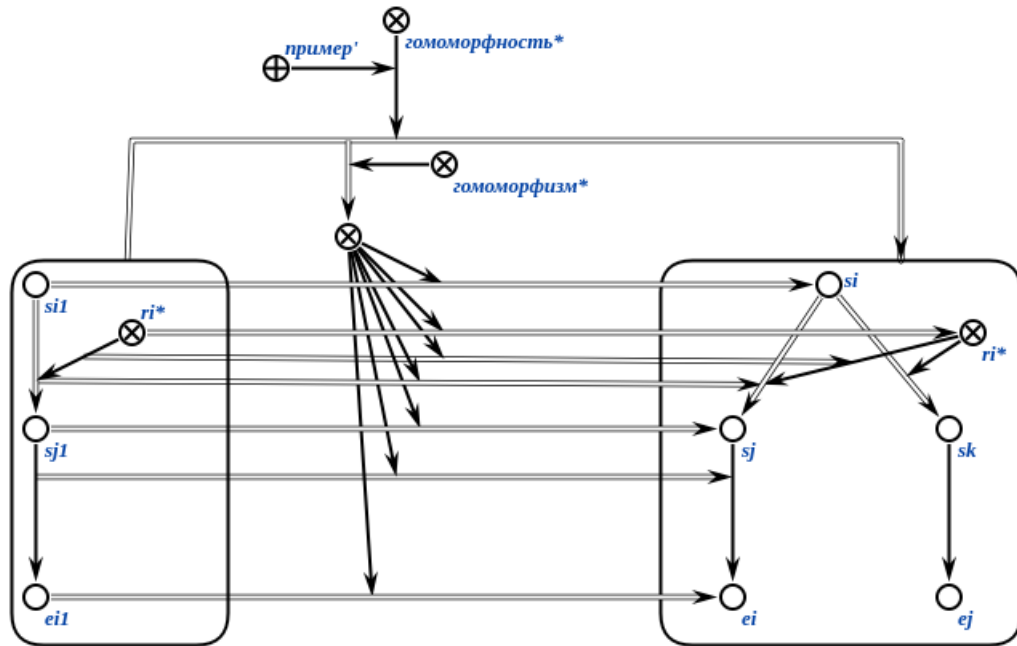
∈ бинарное отношение

⇒ пояснение*:

[**гомоморфность*** - это соответствие, заданное на структурах, при котором каждому элементу из области определения соответствия (первой структуры) ставится в соответствие только один элемент из области значения соответствия (второй структуры).]

⇒ описание примера*:

[



]

гомоморфизм*

∈ бинарное отношение

изоморфность*

:= [изоморфное соответствие*]

:= [изоморфность структур*]

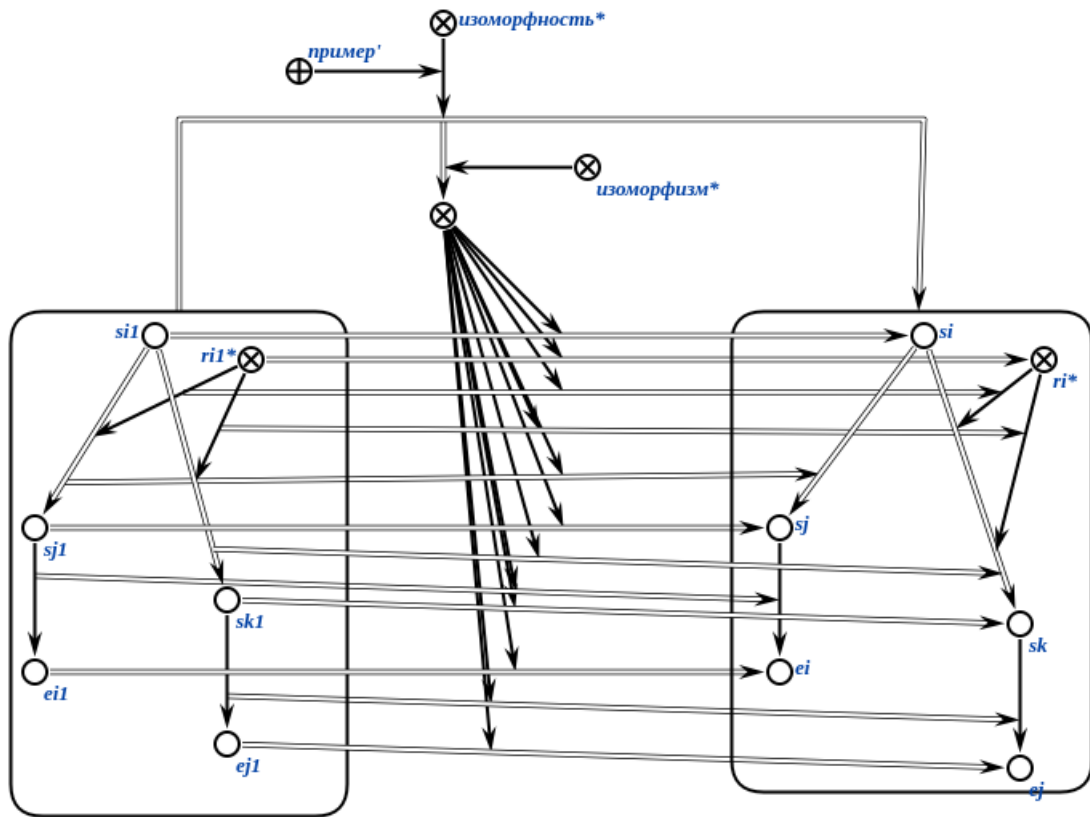
С гомоморфность*

∈ бинарное отношение

⇒ пояснение*:

[**изоморфность*** - это гомоморфность*, при которой для каждого элемента из области значения существует ровно один соответствующий элемент из области определения.]

⇒ описание примера*:



изоморфизм*

∈ *бинарное отношение*

автоморфность*

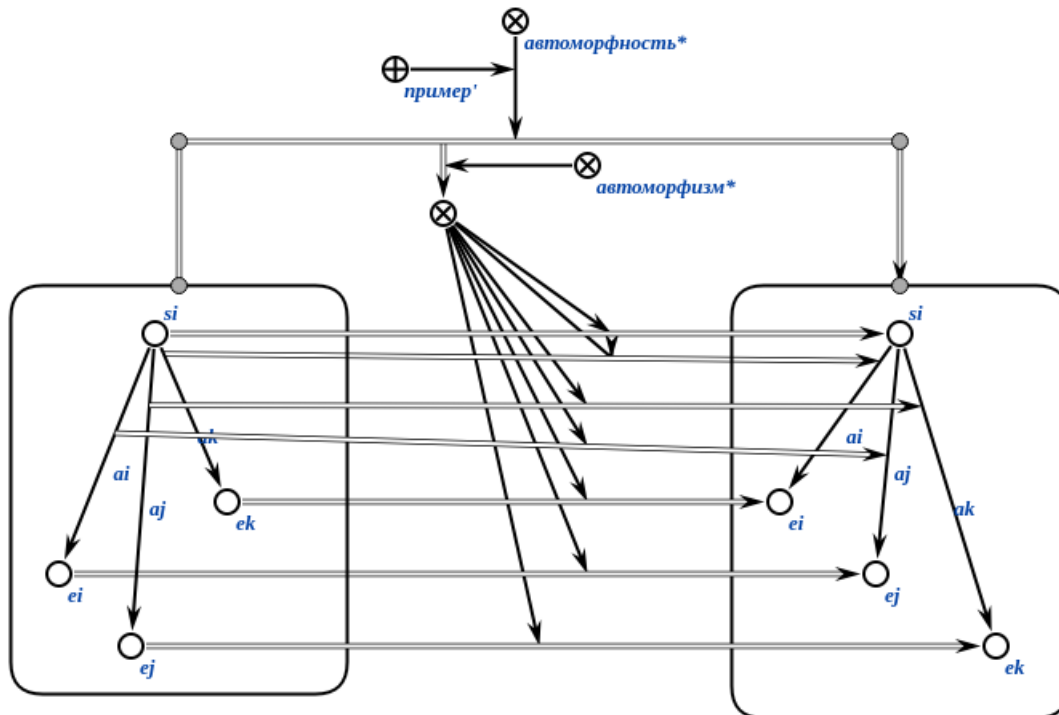
⊂ *гомоморфность**

∈ *бинарное отношение*

⇒ *пояснение**:

[***автоморфность**** - это *изоморфность**, у которой область определения соответствия и область значения соответствия совпадают.]

⇒ *описание примера**:



автоморфизм*

∈ бинарное отношение

аналогичность структур*

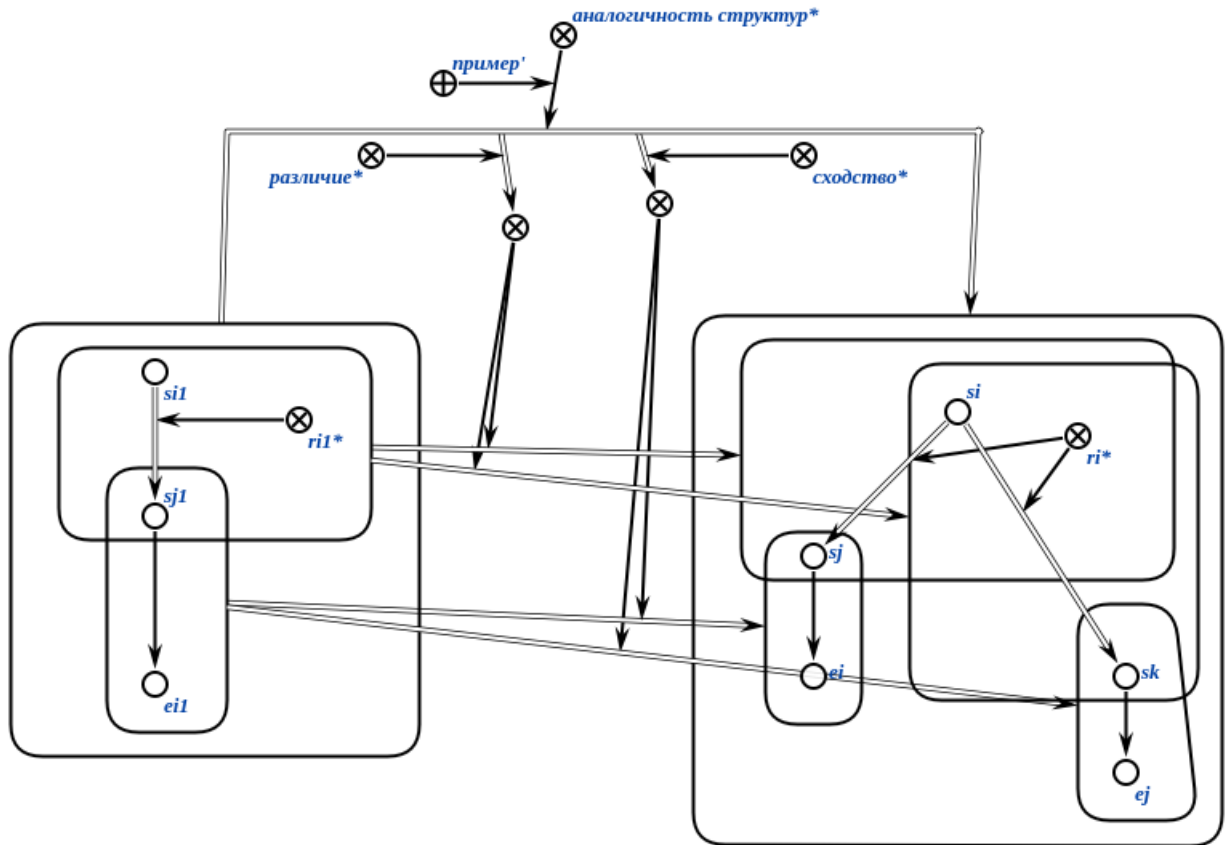
С соответствие*

∈ бинарное отношение

⇒ пояснение*:

[**аналогичность структур*** - соответствие*, задаваемое на структурах, и фиксирующее факт наличия некоторой аналогии на подструктурах (подмножествах) указанных структур. Каждой ориентированной паре, принадлежащей **аналогичности структур*** может быть поставлено в соответствие множество пар, задающих **сходства*** некоторых подструктур и **различия*** некоторых подструктур исходных структур.]

⇒ описание примера*:



сходство*

∈ бинарное отношение

различие*

∈ бинарное отношение

первичная синтаксическая структура sc-текста*

∈ бинарное отношение

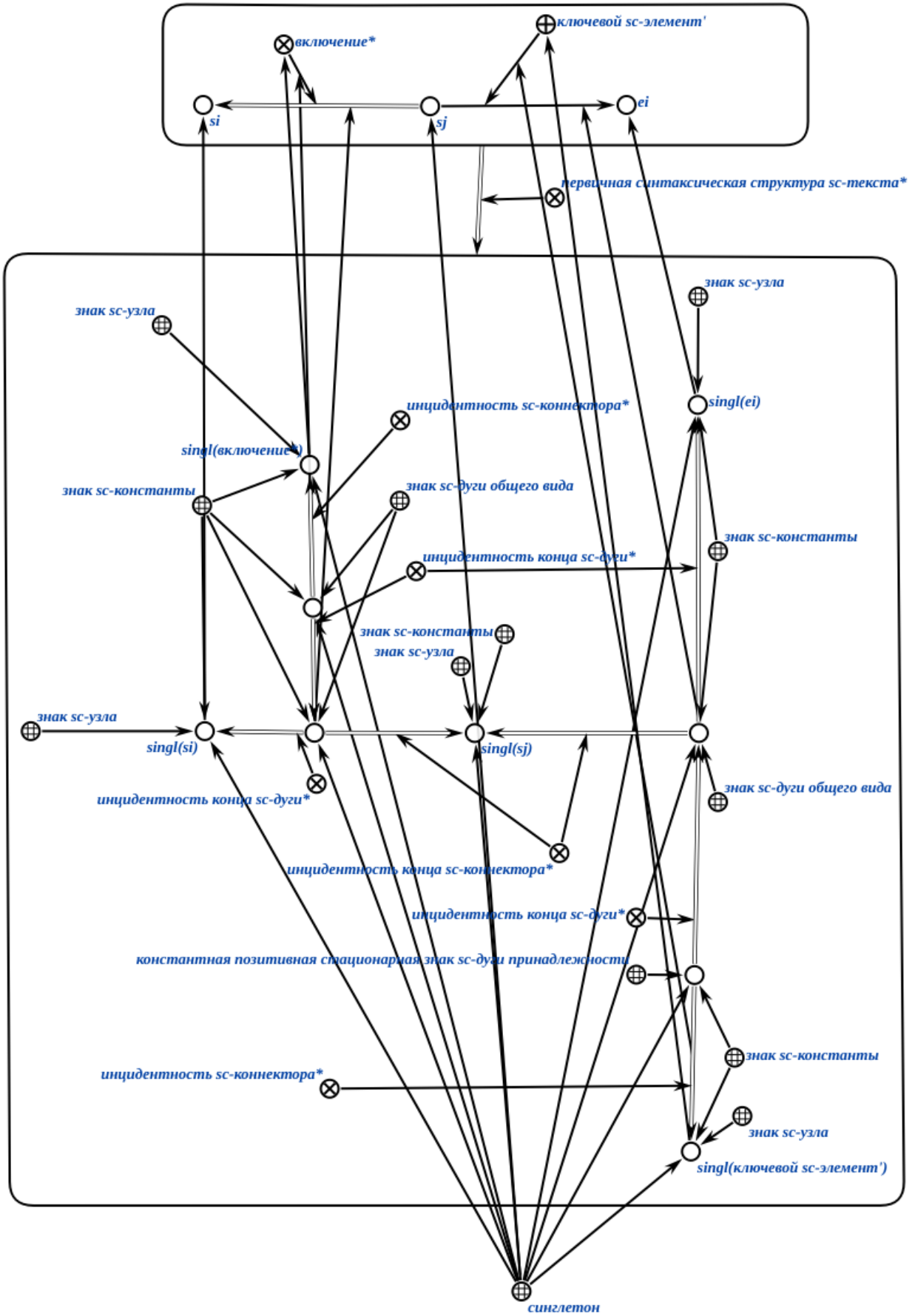
⇒ пояснение*:

[**первичная синтаксическая структура sc-текста*** - это бинарное отношение, связывающее некоторый *sc-текст* с другим *sc-текстом*, формируемым по следующим правилам:

- каждому *sc-узлу* первого *sc-текста* соответствует *синглтон* (знак *sc-узла*) в рамках второго *sc-текста*;
- каждому *sc-коннектору* из первого *sc-текста* в рамках второго *sc-текста* соответствует *синглтон*, обозначающий данный *sc-коннектор* и соединенный с другими *синглтонами* второго *sc-текста* парами инцидентности двух типов, в зависимости от того, началом или концом данного *sc-коннектора* являются обозначаемые этими *синглтонами* *sc-элементы*. В случае, когда *sc-коннектор* является *sc-ребром*, то достаточно пар инцидентности первого типа.
- для каждого *синглтона* в рамках второго *sc-текста* явно указывается синтаксический тип, определяемый типом соответствующего ему элемента из первого *sc-текста* (знак *sc-константы*, знак *sc-узла* и т.п.).

Стоит отметить, что подобным образом может быть задана синтаксическая структура любого текста, а не только *sc-текста*. В этом случае понадобятся другие отношения инцидентности другие классы синтаксических типов.]

⇒ описание примера*:



]

/* Раздел *****/	
Предметная область и онтология темпоральных сущностей	
⊃=	
{	
Предметная область темпоральных сущностей	
:=	[Предметная область темпоральных связей и отношений]
:=	[Предметная область временных сущностей]
∈	<i>предметная область</i>
⊃	<i>максимальный класс объектов исследования'</i> :
	<i>временная сущность</i>
⊃	<i>класс объектов исследования'</i> :
	• <i>прошлая сущность</i>
	• <i>настоящая сущность</i>
	• <i>будущая сущность</i>
	• <i>временная связь</i>
	• <i>ситуация</i>
	• <i>процесс</i>
	• <i>процесс в sc-памяти</i>
	• <i>процесс во внешней среде ostis-системы</i>
	• <i>материальная сущность</i>
	• <i>воздействие</i>
	• <i>отношение</i>
	• <i>класс временных связей</i>
	• <i>класс временных и постоянных связей</i>
	• <i>множество</i>
	• <i>ситуативное множество</i>
	• <i>неситуативное множество</i>
	• <i>частично ситуативное множество</i>
	• <i>темпоральная связь</i>
	• <i>темпоральное отношение</i>
	• <i>начало</i>
	• <i>завершение</i>
	• <i>длительность</i>
	• <i>тысячелетие</i>
	• <i>век</i>
	• <i>год</i>
	• <i>месяц</i>
	• <i>сутки</i>
	• <i>час</i>
	• <i>минута</i>
	• <i>секунда</i>
⊃	<i>исследуемое отношение'</i> :
	• <i>воздействующая сущность*</i>
	• <i>объект воздействия*</i>
	• <i>начальная ситуация*</i>
	• <i>причинная ситуация*</i>
	• <i>конечная ситуация*</i>
	• <i>событие*</i>
	• <i>последний добавленный sc-элемент'</i>
	• <i>темпоральное включение*</i>
	• <i>темпоральная часть*</i>
	• <i>начальный этап*</i>
	• <i>конечный этап*</i>
	• <i>промежуточный этап*</i>
	• <i>темпоральное включение без совпадения начальных и конечных моментов*</i>
	• <i>темпоральное включение с совпадением начальных моментов*</i>
	• <i>темпоральное включение с совпадением конечных моментов*</i>
	• <i>темпоральное совпадение*</i>
	• <i>темпоральное объединение*</i>

- *темпоральная декомпозиция**
- *темпоральная смежность**
- *темпоральная последовательность с промежутком**
- *темпоральная последовательность с пересечением**
- *номер тысячелетия'*
- *номер века'*
- *номер года'*
- *номер месяца в году'*
- *номер суток в месяце'*
- *номер часа в дне'*
- *номер минуты в часе'*
- *номер секунды в минуте'*

временная сущность

:= [временно существующая сущность]

:= [нестационарная сущность]

:= [сущность, имеющая и/или начало, и/или конец своего существования]

:= [sc-элемент, являющийся знаком некоторой временно существующей сущности]

:= [сущность, обладающая темпоральными характеристиками (длительностью, начальным моментом, конечным моментом и т.д.)]

⇒ *разбиение**:

- *прошлая сущность*
- *настоящая сущность*
- *будущая сущность*

⇒ *разбиение**:

- *временная связь*
- *ситуация*
- *процесс*
- *материальная сущность*

⇒ *пояснение**:

[Следует отличать:

- временный характер сущности, обозначаемой *sc-элементом*;
- временный характер существования самого *sc-элемента* в рамках *sc-памяти*;

В ходе обработки информации каждый *sc-элемент* может быть удален из *sc-памяти*.]

прошлая сущность

:= [сущность, существовавшая в прошлом времени]

:= [сущность прошлого времени]

:= [сущность, завершившая свое существование]

настоящая сущность

:= [сущность, существующая в текущий момент времени]

:= [сущность, существующая сейчас]

:= [сущность настоящего времени]

будущая сущность

:= [возможно будущая сущность]

:= [прогнозируемая временная сущность]

:= [временная сущность, которая может существовать в будущем]

:= [сущность, которая может или должна начать свое существование в будущем времени]

⇒ *включение**:

иницированное действие

⇒ *пояснение**:

[Каждой *будущей сущности* можно поставить в соответствие вероятность ее возникновения.]

временная связь

:= [нестационарная связь]

:= [временно существующая связь]

⇒ *пояснение**:

[Каждая **временная связь** представляет собой *связку*, принадлежащую множеству *временных сущностей*.

Понятие **временной связи** не следует путать с понятием *темпоральной связи*, которая сама является *постоянной сущностью*, описывающей то, как связаны во времени некоторые *временные сущности*.]

ситуация

:= [состояние]

:= [временная структура]

:= [временно существующая структура]

:= [квазистационарная структура]

:= [состояние некоторой динамической системы, описываемое с некоторой степенью детализации (подробности)]

:= [квазистационарная структура, существующая временно (в течение некоторого отрезка времени)]

⇐ *включение**:*структура*⇒ *пояснение**:

[Под ситуацией понимается *структура*, содержащая, по крайней мере, один элемент, который является *временной сущностью*. Наличие в рамках ситуации нескольких *временных сущностей* говорит о том, что существует момент времени (в прошлом, настоящем или будущем), в который все они существуют одновременно.]

процесс

:= [процесс преобразования некоторой временной сущности из одного состояния в другое]

:= [процесс перехода от одной ситуации к другой]

:= [переходный процесс]

:= [абстрактный процесс]

:= [информационная модель некоторых преобразований]

:= [динамическая sc-модель]

:= [динамическая структура]

⇒ *включение**:*воздействие*⇐ *включение**:*структура*⇒ *пояснение**:

[Каждый **процесс** определяется (задается) либо возникновением или исчезновением связей, связывающих заданную *временную сущность* с другими сущностями, либо возникновением или исчезновением связей, связывающих части указанной *временной сущности* с другими сущностями.

Многим **процессам** можно поставить в соответствие *ситуацию*, которая является его *начальной ситуацией** и *ситуацию*, которая является его *конечной ситуацией**, то есть указать *ситуации*, переход между которыми осуществляется в ходе **процесса**.

При этом знаки тех *временных сущностей*, с которыми связаны изменения, описываемые некоторым **процессом**, входят в данный **процесс** как элементы и при необходимости уточняются дополнительными *ролевыми отношениями*.]

⇒ *разбиение**:

- { ● процесс в sc-памяти
- процесс во внешней среде *ostis-системы*
- }

материальная сущность⇒ *пояснение**:

[Каждой **материальной сущности** можно поставить в соответствие различные **процессы**, описывающие ее преобразование из одного состояния в другое.]

воздействие

:= [процесс, осуществляющийся на основе (в результате) воздействия одной сущности на другую]

⇒ *включение**:

действие

⇒ *пояснение**:

[Каждому **воздействию** может быть поставлена в соответствие (1) некоторая *воздействующая сущность**, т.е. сущность, осуществляющая **воздействие** (в частности, это может быть некоторое физическое поле), и (2) некоторый *объект воздействия**, т.е. сущность, на которую воздействие направлено. Если **воздействие** связано с *материальной сущностью*, то его объектом воздействия является либо сама эта *материальная сущность*, либо некоторая ее пространственная часть.]

исходная ситуация*

:= [начальная ситуация процесса*]

:= [начальная ситуация*]

∈ *бинарное отношение*

⇒ *пояснение**:

[Связки отношения **исходная ситуация*** связывают некоторый *процесс* и некоторую ситуацию, являющуюся начальной для этого *процесса*, и, как правило, изменяемой в течение выполнения этого *процесса*.

Первым компонентом каждой связки отношения **исходная ситуация*** является знак *процесса*, вторым – знак начальной *ситуации*.]

причинная ситуация*

∈ *бинарное отношение*

⇐ *включение**:

*начальная ситуация**

⇒ *пояснение**:

[Под причинной ситуацией понимается такая *начальная ситуация**, которая обладает достаточной полнотой для однозначного задания инициируемого *процесса*.]

конечная ситуация*

:= [конечная ситуация процесса*]

:= [результатирующая ситуация*]

∈ *бинарное отношение*

⇒ *пояснение**:

[Связки отношения **конечная ситуация*** связывают некоторый *процесс* и некоторую *ситуацию*, ставшую результатом выполнения этого *процесса*, то есть его следствием.

Первым компонентом каждой связки отношения **конечная ситуация*** является знак *процесса*, вторым – знак конечной *ситуации*.]

событие*

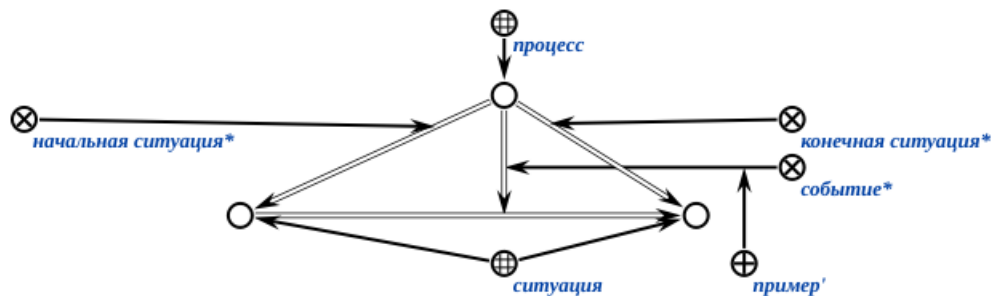
∈ *бинарное отношение*

⇒ *пояснение**:

[Связки отношения **событие*** связывают знак *процесса* и ориентированную пару, первым компонентом которой является знак *начальной ситуации** данного *процесса*, вторым компонентом – знак *конечной ситуации** данного *процесса*.]

⇒ *описание примера**:

[



]

отношение

⇒ разбиение*:
 {

- класс временных связей
- класс постоянных связей
- класс временных и постоянных связей

 }

класс временных связей

:= [отношение, все связки которого являются нестационарными]

⇒ пояснение*:

[В общем случае **класс временных связей** не является *ситуативным множеством*, поскольку факт принадлежности некоторой *временной связи* такому классу следует считать постоянным, а не временным, поскольку временность/постоянство связи и ее семантический тип, задаваемый классом (отношением), это принципиально разные параметры (характеристики, признаки) любой связи.]

класс постоянных связей

:= [отношение, все связки которого являются стационарными]

класс временных и постоянных связей

:= [отношение, некоторые (но не все) связки которого являются нестационарными]

множество

⇒ разбиение*:
 {

- ситуативное множество
- неситуативное множество
- частично ситуативное множество

 }

ситуативное множество

:= [полностью ситуативное множество]

⇒ пояснение*:

[Под **ситуативным множеством** понимается постоянное множество, у которого все выходящие из него связи принадлежности являются *временными сущностями*.

В частности, ситуативное множество может использоваться как вспомогательная динамическая структура, которая содержит элементы некоторых структур, обрабатываемые в данный момент, например, это может быть копия некоторого множества, из которой постепенно удаляются элементы по мере их просмотра и обработки. В случае, когда такая структура содержит всего один элемент, ее можно считать указателем на данный элемент, при этом в разные моменты времени это могут быть разные элементы.]

последний добавленный sc-элемент'

∈ ролевое отношение

неситуативное множество

⇒ пояснение*:

[Под **неситуативным множеством** понимается постоянное множество, у которого все выходящие из него связи принадлежности являются *постоянными сущностями*.]

частично ситуативное множество

⇒ пояснение*:

[Под **частично ситуативным множеством** понимается постоянное множество, у которого некоторые (но не все) выходящие из него связи принадлежности являются *временными сущностями*.]

темпоральная связь

:= [постоянная связь, описывающая связь во времени между временными сущностями]

темпоральное отношение

← семейство подмножеств*:
 темпоральная связь

:= [класс темпоральных связей]

- := [отношение, задающее темпоральные связи между временными сущностями]
- ⊃ *темпоральное включение**
- ⊃ *темпоральное объединение**
- ⊃ *темпоральная декомпозиция**
- ⊃ *темпоральная смежность**
- ⊃ *темпоральная последовательность с промежутком**
- ⊃ *темпоральная последовательность с пересечением**

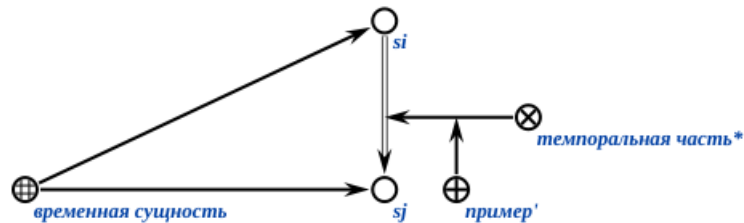
темпоральное включение*

⇒ *пояснение**:
 [Связки отношения **темпоральное включение*** связывают две *временные сущности*, период существования одной из которых полностью включается в период существования второй.
 Первым компонентом каждой связки отношения **темпоральное включение*** является знак *временной сущности*, *длительность* существования которой больше.]

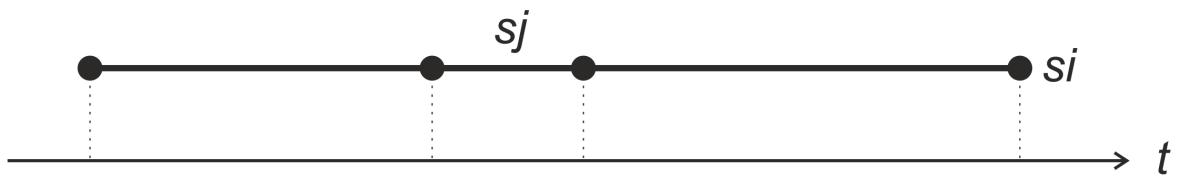
- ⇒ *включение**:
- *темпоральная часть**
 - *темпоральное включение без совпадения начальных и конечных моментов**
 - *темпоральное совпадение**
 - *темпоральное включение с совпадением начальных моментов**
 - *темпоральное включение с совпадением конечных моментов**

темпоральная часть*

- := [этап (период) заданной временной сущности*]
- := [этап процесса существования временной сущности*]
- ⇒ *включение**:
 - *начальный этап**
 - *конечный этап**
 - *промежуточный этап**
- ⇒ *описание примера**:
 [



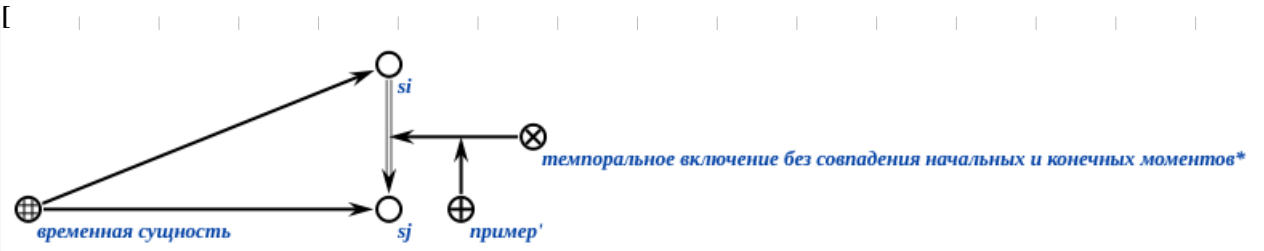
⇒ *иллюстрация**:
 [



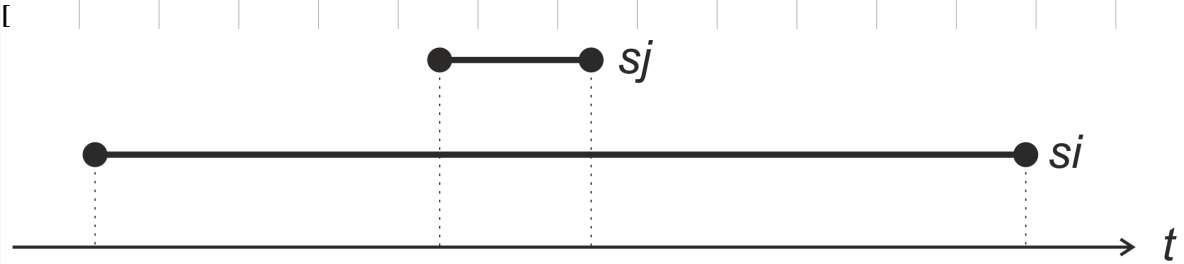
⇒ *примечание**:
 [Связки отношения **темпоральная часть*** связывают две *временные сущности*, одна из которых является частью другой, например, действие и одно из его поддействий. Соответственно, период существования одной из этих сущностей всегда будет включаться в период существования другой (большей).
 В отличие от более общего отношения **темпоральное включение***, связки которого могут связывать любые *временные сущности*, связки отношения **темпоральная часть*** связывают только *временные сущности*, одна из которых является частью другой.]

темпоральное включение без совпадения начальных и конечных моментов*

- := [строгое темпоральное включение*]
- ⇒ *описание примера**:

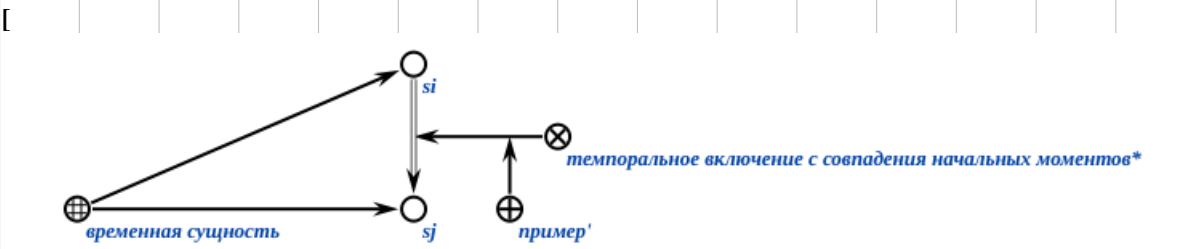


⇒ иллюстрация*:

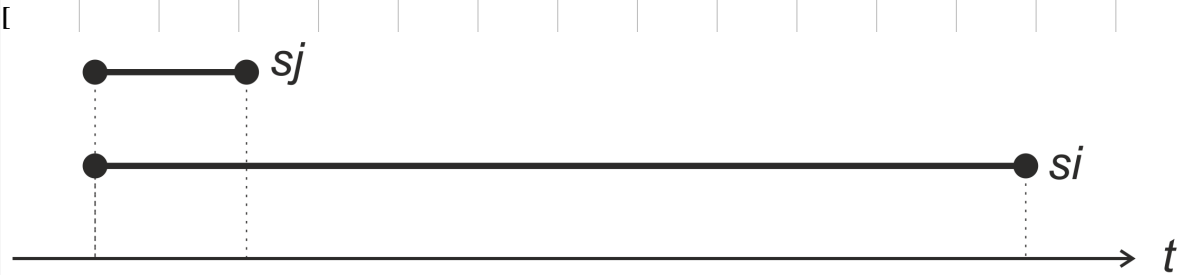


темпоральное включение с совпадением начальных моментов*

⇒ описание примера*:

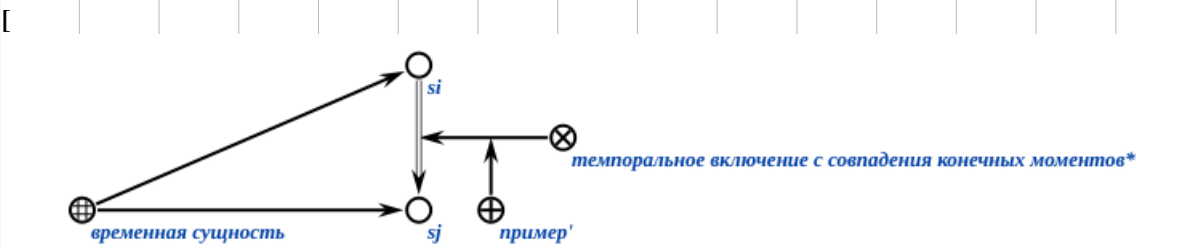


⇒ иллюстрация*:

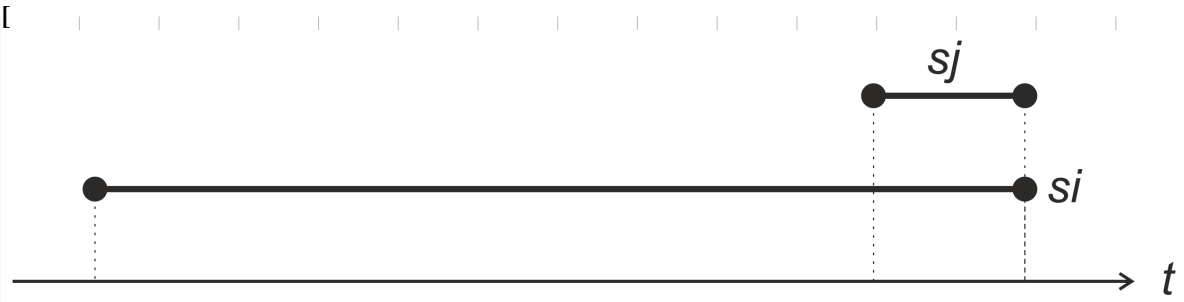


темпоральное включение с совпадением конечных моментов*

⇒ описание примера*:



⇒ иллюстрация*:

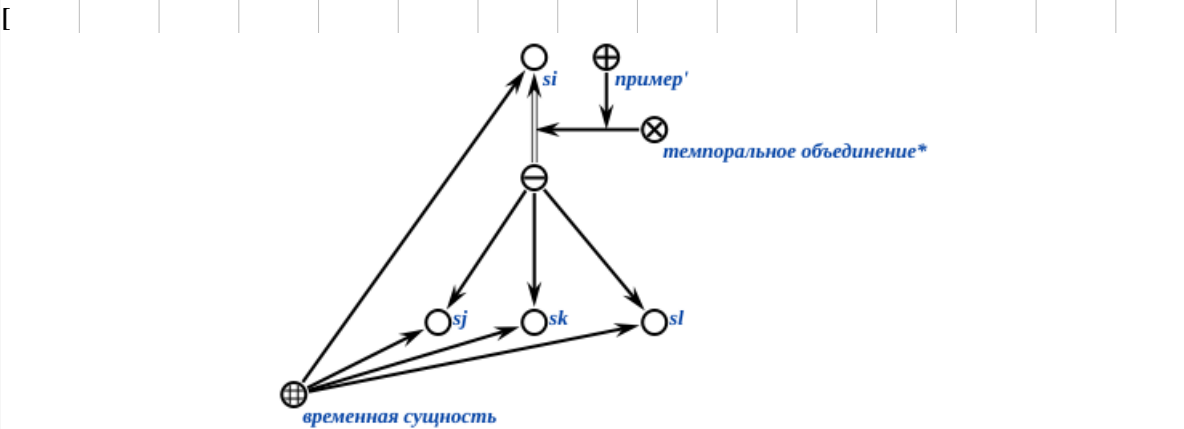


темпоральное совпадение*

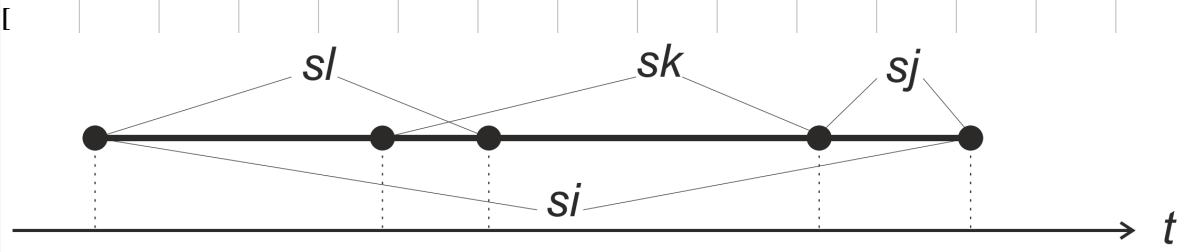
:= [совпадение начала и завершения*]

темпоральное объединение*

⇒ описание примера*:

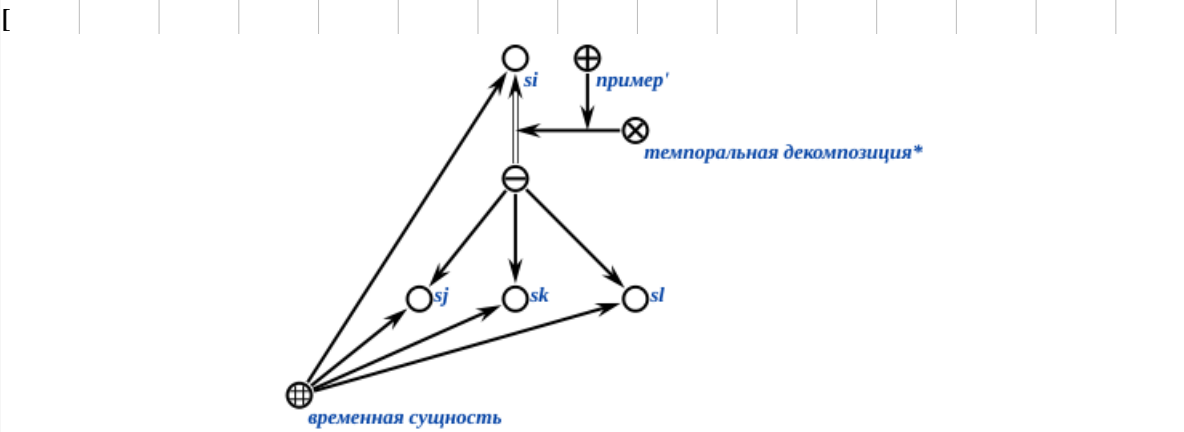


⇒ иллюстрация*:

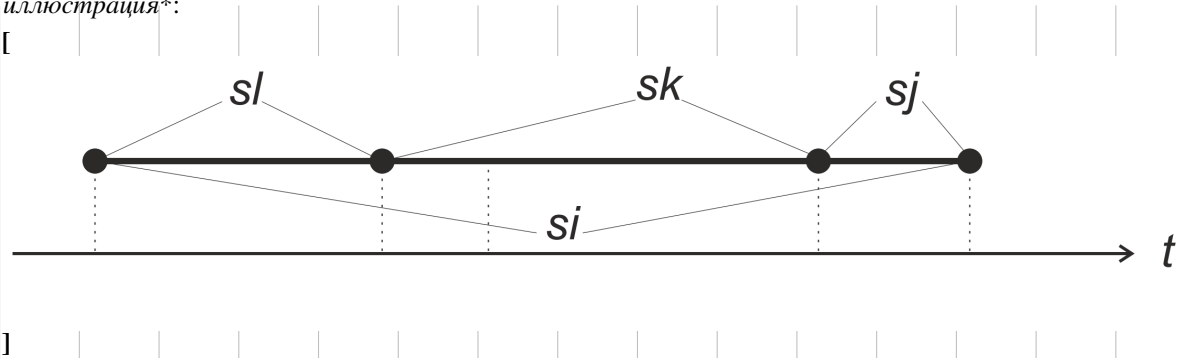


темпоральная декомпозиция*

⇒ описание примера*:



⇒ иллюстрация*:

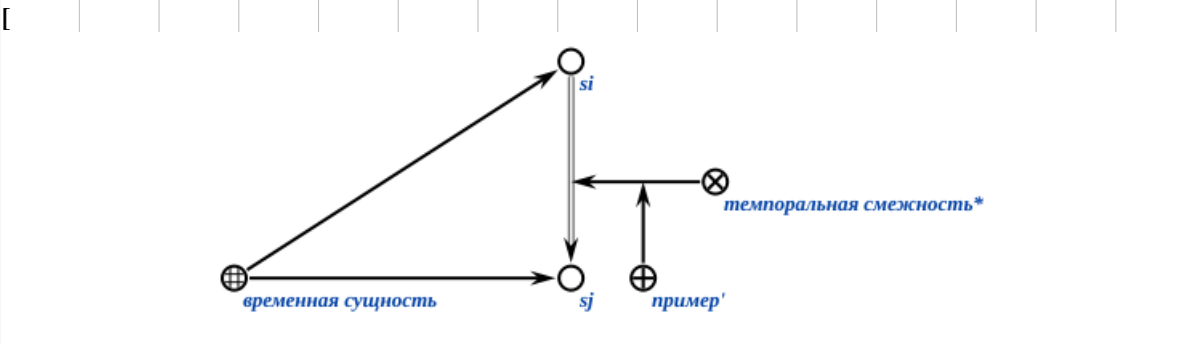


темпоральная смежность*

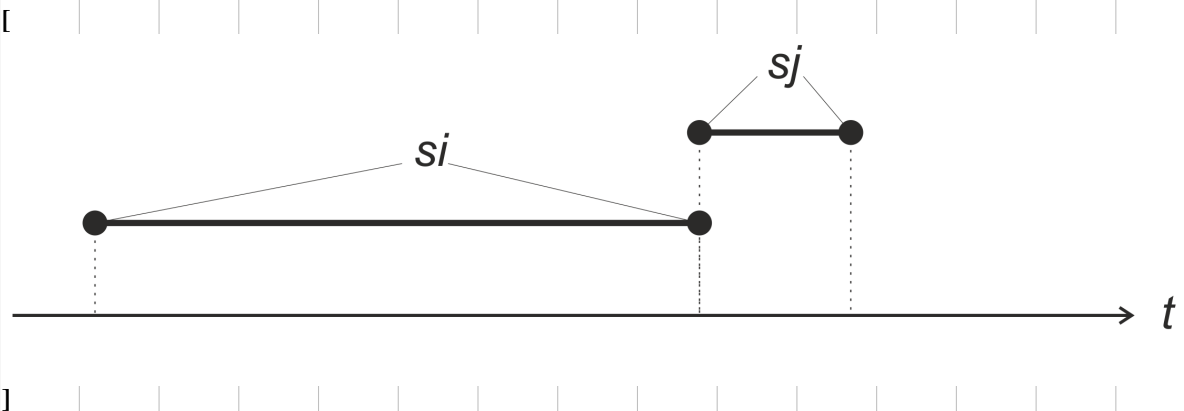
:= [строгая темпоральная последовательность (без темпорального промежутка)*]

:= [темпоральная последовательность без промежутка*]

⇒ описание примера*:

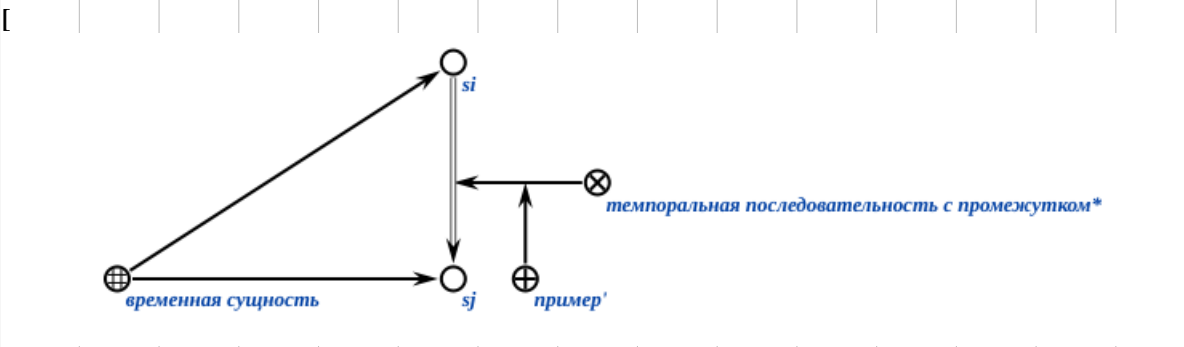


⇒ иллюстрация*:

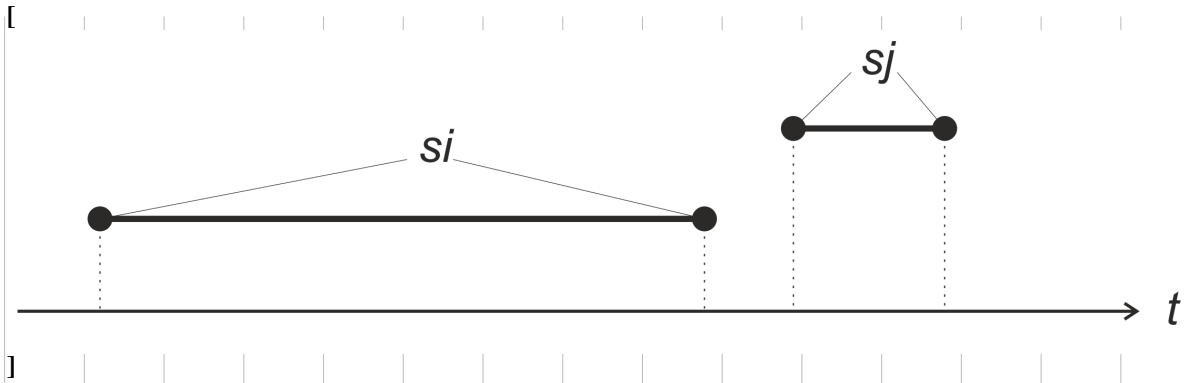


темпоральная последовательность с промежутком*

⇒ описание примера*:

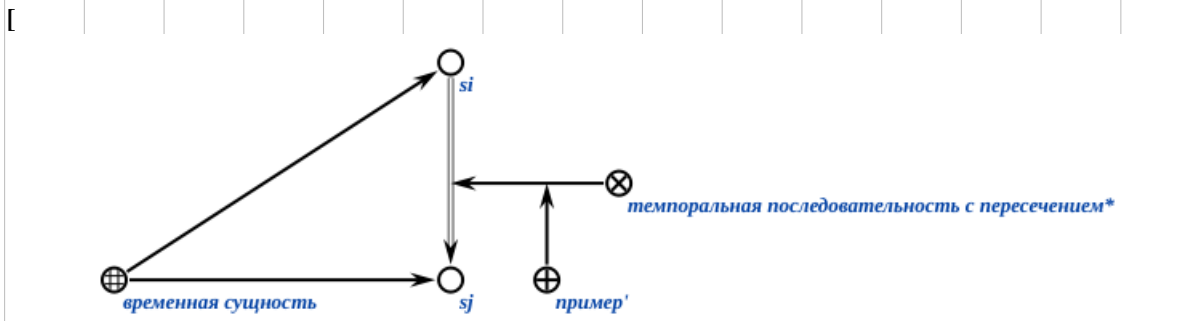


⇒ иллюстрация*:

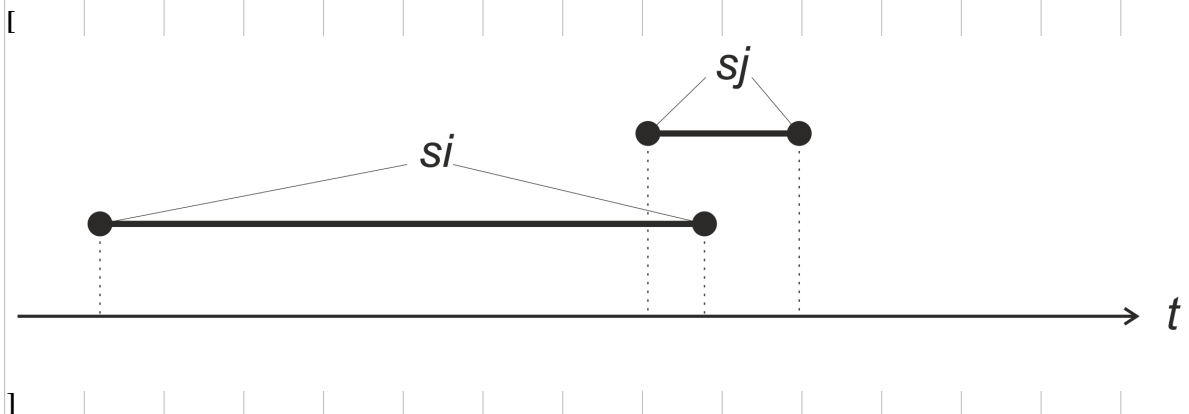


темпоральная последовательность с пересечением*

⇒ описание примера*:



⇒ иллюстрация*:



начало

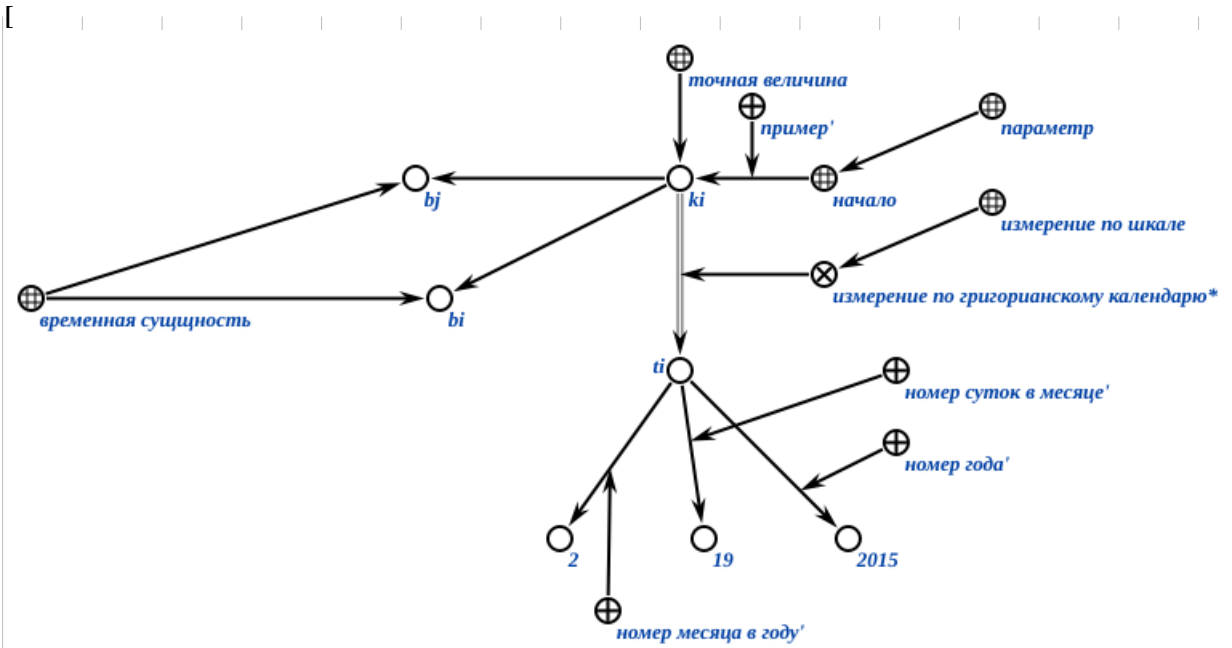
:= [класс одновременно начавшихся сущностей]

∈ параметр

⇒ пояснение*:

[Каждый элемент множества **начало** представляет собой класс *временных сущностей*, у которых совпадает момент начала их существования. Конкретное значение данного *параметра* может быть как *точной величиной*, так и *неточной величиной* или *интервальной величиной*.]

⇒ описание примера*:



⇒ *пояснение**:
 [В данном примере *ki* обозначает класс сущностей, начавших свое существование 19 февраля 2015 года по григорианскому календарю. Конкретные примеры таких сущностей – *bi* и *bj*. *ti* обозначает временную точку григорианского календаря, соответствующую 19 февраля 2015 года.]

завершение

:= [конец]

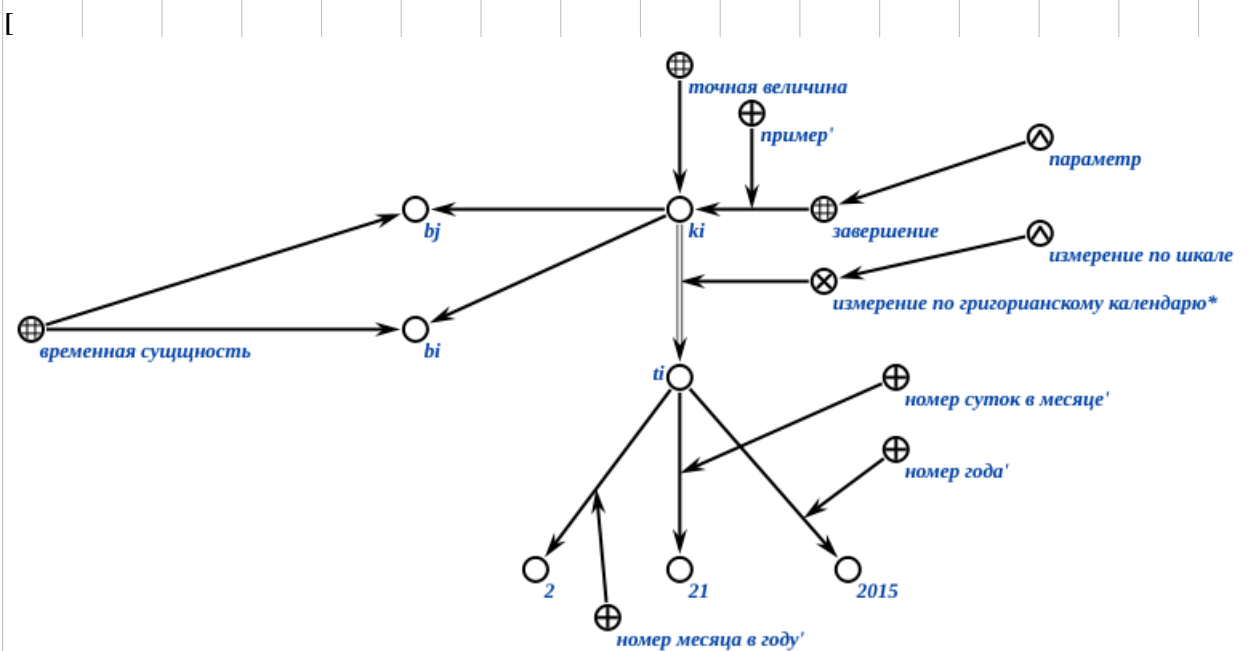
:= [класс одновременно завершившихся сущностей]

∈ *параметр*

⇒ *пояснение**:

[Каждый элемент множества **завершение** представляет собой класс *временных сущностей*, у которых совпадает конечный момент их существования (момент завершения существования). Конкретное значение данного *параметра* может быть как *точной величиной*, так и *неточной величиной* или *интервальной величиной*.]

⇒ *описание примера**:



⇒ *пояснение**:

[В данном примере *ki* обозначает класс сущностей, завершивших свое существование 21 февраля 2015 года по григорианскому календарю. Конкретные примеры таких сущностей – *bi* и *bj*. *ti* обозначает временную точку григорианского календаря, соответствующую 21 февраля 2015 года.]

длительность

:= [класс временных сущностей, имеющих одинаковую длительность]

∈ параметр

⊃ тысячелетие

⊃ век

⊃ год

⊃ месяц

⊃ день

⊃ час

⊃ минута

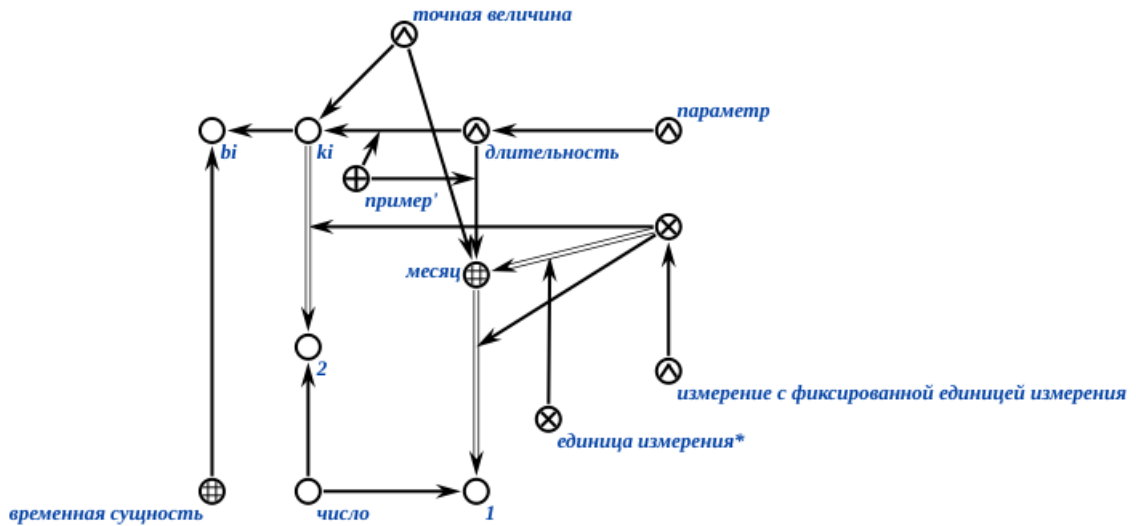
⊃ секунда

⇒ пояснение*:

[Каждый элемент множества **длительность** представляет собой класс *временных сущностей*, у которых совпадает длительность их существования. Конкретное значение данного *параметра* может быть как *точной величиной*, так и *неточной величиной* или *интервальной величиной*.]

⇒ описание примера*:

[



]

⇒ пояснение*:

[В данном примере *ki* обозначает класс сущностей, существовавших в течение 2 месяцев. Конкретный пример такой сущности – *bi*.]

} /* Завершили Раздел “Предметная область и онтология темпоральных сущностей” */

/* Раздел *****/

Предметная область и онтология темпоральных сущностей баз знаний *ostis*-систем

⊃=
{
⇒

введение*:

[Обработка информации в *sc-памяти* (т.е. динамика базы знаний, хранимой в *sc-памяти*) в конечном счете сводится:

- к появлению в *sc-памяти* новых актуальных *sc-узлов* и *sc-коннекторов*;
- к логическому удалению актуальных *sc-элементов*, т.е. к переводу их в неактуальное состояние (это необходимо для хранения протокола изменения состояния базы знаний, в рамках которого могут описываться действия по удалению *sc-элементов*);
- к возврату логически удаленных *sc-элементов* в статус актуальных (необходимость в этом может возникнуть при откате базы знаний к какой-нибудь ее прошлой версии);
- к физическому удалению *sc-элементов*;
- к изменению состояния актуальных (логически не удаленных *sc-элементов*) – *sc-узел* может превратиться в *sc-ребро*, *sc-ребро* может превратиться в *sc-дугу*, *sc-дуга* может поменять направленность, *sc-дуга* общего вида может превратиться в *константную стационарную sc-дугу принадлежности*, и т.д.;

Подчеркнем, что временный характер самого *sc-элемента* (т.к. он может появиться или исчезнуть) никак не связан с возможно временным характером сущности, обозначаемой этим *sc-элементом*. Т.е. временный характер самого *sc-элемента* и временный характер сущности, которую он обозначает – абсолютно разные вещи.

Таким образом, следует четко отличать динамику внешнего мира, описываемого базой знаний, а динамику самой базы знаний (динамику внутреннего мира). При этом очень важно, чтобы описание динамики базы знаний также входило в состав каждой базы знаний.

К числу понятий, используемых для описания динамики базы знаний относятся:

- логически удаленный *sc-элемент*;
- сформированное множество;
- вычисленное число;
- сформированное высказывание;

]

Предметная область темпоральных сущностей базы знаний *ostis*-системы

:= [Предметная область, описывающая динамику базы знаний, хранимой в *sc-памяти*]

∈ предметная область

⊃ максимальный класс объектов исследования':

ситуация

⊃ класс объектов исследования':

- *sc-элемент*
- настоящий *sc-элемент*
- логически удаленный *sc-элемент*
- число
- невычисленное число
- вычисленное число
- понятие
- основное понятие
- неосновное понятие
- понятие, переходящее из основного в неосновное
- понятие, переходящее из неосновного в основное
- специфицированная сущность
- недостаточно специфицированная сущность
- достаточно специфицированная сущность
- средне специфицированная сущность
- структура
- файл
- событие в *sc-памяти**

- элементарное событие в sc-памяти*
- событие добавления sc-дуги, выходящей из заданного sc-элемента*
- событие добавления sc-дуги, входящей в заданный sc-элемент*
- событие добавления sc-ребра, инцидентного заданному sc-элементу*
- событие удаления sc-дуги, выходящей из заданного sc-элемента*
- событие удаления sc-дуги, входящей в заданный sc-элемент*
- событие удаления sc-ребра, инцидентного заданному sc-элементу*
- событие удаления sc-элемента*
- событие изменения содержимого файла*

sc-элемент

⇐ разбиение*:
 {

- настоящий sc-элемент
- логически удаленный sc-элемент

 }

настоящий sc-элемент

∈ ситуативное множество

логически удаленный sc-элемент

∈ ситуативное множество

число

⇒ разбиение*:
 {

- невычисленное число
- вычисленное число

 }

невычисленное число

∈ ситуативное множество

вычисленное число**понятие**

⇒ разбиение*:
 {

- основное понятие
- неосновное понятие
- понятие, переходящее из основного в неосновное
- понятие, переходящее из неосновного в основное

 }

основное понятие

:= [основное понятие для данной ostis-системы]

∈ ситуативное множество

⇒ пояснение*:

[К **основным понятиям** относятся те понятия, которые активно используются в системе и могут быть ключевыми элементами sc-агентов. К **основным понятиям** относятся также все неопределяемые понятия.]

неосновное понятие

:= [дополнительное понятие]

:= [вспомогательное понятие]

:= [неосновное понятие для данной ostis-системы]

∈ ситуативное множество

⇒ пояснение*:

[Каждое **неосновное понятие** должно быть строго определено на основе *основных понятий*. Такие **неосновные понятия** используются только для понимания и правильного восприятия вводимой информации, в том числе, для выравнивания онтологий. Ключевым элементом sc-агентов **неосновные понятия** быть не могут.]

⇒ правило идентификации экземпляров*:

[В случае, когда некоторое понятие полностью перешло из *основных понятий* в неосновные, то есть стало **неосновным понятием**, и соответствующее ему *основное понятие* (через которое оно определяется) в рамках некоторого внешнего языка имеет одинаковый с ним основной идентификатор, то к идентификатору **неосновного понятия** спереди добавляется знак #. Если при этом соответствующее *основное понятие* имеет в идентификаторе знак \$, добавленный в процессе перехода, то этот знак удаляется. Если указанные понятия имеют разные основные идентификаторы в рамках этого внешнего языка, то никаких дополнительных средств идентификации не используется.

Например:

#трансляция sc-текста
#scp-программа]

понятие, переходящее из основного в неосновное

∈ *ситуативное множество*

понятие, переходящее из неосновного в основное

∈ *ситуативное множество*

⇒ *правило идентификации экземпляров**:

[В случае, когда текущее *основное понятие* и соответствующее ему **понятие, переходящее из неосновного в основное** в рамках некоторого внешнего языка имеют одинаковый основной идентификатор, то к идентификатору понятия, переходящего из неосновного в основное спереди добавляется знак \$. Если указанные понятия имеют разные основные идентификаторы в рамках этого внешнего языка, то никаких дополнительных средств идентификации не используется.

Например:

\$трансляция sc-текста
\$scp-программа]

специфицированная сущность

⇒ *разбиение**:

- { • *недостаточно специфицированная сущность*
- *достаточно специфицированная сущность*
- *средне специфицированная сущность*
- }

достаточно специфицированная сущность

⇒ *пояснение**:

[К **достаточно специфицированным сущностям** предъявляются следующие требования:

- если сущность не является понятием, то для нее должны быть указаны
 - различные варианты обозначающих ее внешних знаков;
 - классы, которым она принадлежит;
 - связи, которыми она связана с другими сущностями (с указанием соответствующего отношения);
 - значения параметров, которыми она обладает;
 - те разделы базы знаний, в которых указанная сущность является ключевой;
 - предметные области, в которые данная сущность входит.
- если специфицированная сущность является понятием, то для нее должны быть указаны:
 - различные варианты внешних обозначений этого понятия;
 - предметные области, в которых это понятие исследуется;
 - определение понятия;
 - пояснения
 - разделы базы знаний, в которых это понятие является ключевым;
 - описание примера – пример экземпляра понятия.

]

структура

⇒ *разбиение**:

- { • *сформированная структура*
- *несформированная структура*
- }

⇒	<i>разбиение*</i> : { <ul style="list-style-type: none"> • <i>недостаточно сформированная структура</i> • <i>достаточно сформированная структура</i> • <i>структура, имеющая средний уровень сформированности</i> }
файл	
⇒	<i>разбиение*</i> : { <ul style="list-style-type: none"> • <i>недостаточно сформированный внутренний файл</i> • <i>достаточно сформированный внутренний файл</i> • <i>внутренний файл, имеющий средний уровень сформированности</i> }
событие в sc-памяти*	
⇐	<i>включение*</i> : <i>событие*</i>
элементарное событие в sc-памяти*	
⇐	<i>включение*</i> : <i>событие в sc-памяти*</i>
⇒	<i>пояснение*</i> : [Под элементарным событием в sc-памяти* понимается <i>событие*</i> , соответствующее некоторому действию в <i>sc-памяти</i> , в результате выполнения которого изменяется состояние только одного <i>sc-элемента</i> .]
⇒	<i>разбиение*</i> : { <ul style="list-style-type: none"> • <i>событие добавления sc-дуги, выходящей из заданного sc-элемента*</i> • <i>событие добавления sc-дуги, входящей в заданный sc-элемент*</i> • <i>событие добавления sc-ребра, инцидентного заданному sc-элементу*</i> • <i>событие удаления sc-дуги, выходящей из заданного sc-элемента*</i> • <i>событие удаления sc-дуги, входящей в заданный sc-элемент*</i> • <i>событие удаления sc-ребра, инцидентного заданному sc-элементу*</i> • <i>событие удаления sc-элемента*</i> • <i>событие изменения содержимого файла*</i> }
}	<i>/* Завершили Раздел “Предметная область и онтология темпоральных сущностей баз знаний ostis-систем” */</i>

/* Раздел *****/

Предметная область и онтология семантических окрестностей

⊃=
{

Предметная область семантических окрестностей

- ∈ предметная область
- ⊃= максимальный класс объектов исследования': семантическая окрестность
- ⊃= класс объектов исследования':
 - семантическая окрестность по инцидентным коннекторам
 - семантическая окрестность по выходящим дугам
 - семантическая окрестность по выходящим дугам принадлежности
 - семантическая окрестность по входящим дугам
 - семантическая окрестность по входящим дугам принадлежности
 - полная семантическая окрестность
 - базовая семантическая окрестность
 - специализированная семантическая окрестность
 - пояснение
 - примечание
 - правило идентификации экземпляров
 - терминологическая семантическая окрестность
 - теоретико-множественная семантическая окрестность
 - описание декомпозиции
 - логическая семантическая окрестность
 - спецификация типичного экземпляра
 - сравнительный анализ

семантическая окрестность

- := [sc-окрестность]
- := [семантическая окрестность, представленная в виде sc-текста]
- := [sc-текст, являющийся семантической окрестностью некоторого sc-элемента]
- := [спецификация заданной сущности, знак которой указывается как ключевой элемент этой спецификации]
- := [описание заданной сущности, знак которой указывается как ключевой элемент этой спецификации]
- ⊃ знание
- ⊃ семантическая окрестность по инцидентным коннекторам
- ⊃ полная семантическая окрестность
- ⊃ базовая семантическая окрестность
- ⊃ специализированная семантическая окрестность
- := пояснение*:

[знание, являющееся спецификацией (описанием) некоторой сущности, знак которой является ключевым знаком' указанного знания. Заметим, что каждая семантическая окрестность в отличие от знаний других видов имеет только один ключевой знак' (ключевой элемент', знак описываемой сущности'). Заметим также, что многообразие видов семантических окрестностей свидетельствует о многообразии семантических видов описаний различных сущностей.]
- ⇒ примечание*:

[Понятие семантической окрестности, как и любой другой семантически выделяемый класс знаний, абсолютно не зависит от языка представления знаний. Этим языком может быть не только SC-код или другой формальный язык представления знаний или даже естественный язык, тексты которых в памяти ostis-системы представляются в виде файлов.]

семантическая окрестность по инцидентным коннекторам

- ⊃ семантическая окрестность по выходящим дугам
- ⊃ семантическая окрестность по входящим дугам
- := пояснение*:

[вид семантической окрестности, в которую входят все коннекторы, инцидентные заданному элементу, а также все элементы, инцидентные указанным коннекторам.]

семантическая окрестность по выходящим дугам

⊃ семантическая окрестность по выходящим дугам принадлежности

:= пояснение*:

[вид семантической окрестности, в которую входят все дуги, выходящие из заданного sc-элемента и вторые компоненты этих дуг. Также указывается факт принадлежности этих дуг каким-либо отношениям.]

семантическая окрестность по выходящим дугам принадлежности

:= пояснение*:

[вид семантической окрестности, в которую входят все дуги принадлежности, выходящие из заданного sc-элемента, а также их вторые компоненты. При необходимости может указываться факт принадлежности этих дуг каким-либо ролевым отношениям.]

семантическая окрестность по входящим дугам

⊃ семантическая окрестность по входящим дугам принадлежности

:= пояснение*:

[вид семантической окрестности, в которую входят все дуги, входящие в заданный sc-элемент, а также их первые компоненты. Также указывается факт принадлежности этих дуг каким-либо отношениям.]

семантическая окрестность по входящим дугам принадлежности

:= пояснение*:

[вид семантической окрестности, в которую входят все дуги принадлежности, входящие в заданный sc-элемент, а также их первые компоненты. При необходимости может указываться факт принадлежности этих дуг каким-либо ролевым отношениям.]

полная семантическая окрестность

:= [полная спецификация некоторой описываемой сущности]

:= пояснение*:

[вид семантической окрестности, включающий описание всех связей описываемой сущности.

Структура полной семантической окрестности определяется прежде всего семантической типологией описываемой сущности.

Так, например, для понятия в полную семантическую окрестность необходимо включить следующую информацию (при наличии):

- варианты идентификации на различных внешних языках (sc-идентификаторы);
- принадлежность некоторой предметной области с указанием роли, выполняемой в рамках этой предметной области;
- теоретико-множественные связи заданного понятия с другими sc-элементами;
- определение или пояснение;
- высказывания, описывающие свойства указанного понятия;
- задачи и их классы, в которых данное понятие является ключевым;
- описание типичного примера использования указанного понятия;
- экземпляры описываемого понятия.

Для понятия, являющегося отношением дополнительно указываются:

- домены;
- область определения;
- схема отношения;
- классы отношений, которым принадлежит описываемое отношение.

]

базовая семантическая окрестность

:= [минимально достаточная семантическая окрестность]

:= [минимальная спецификация описываемой сущности]

:= [сокращенная спецификация описываемой сущности]

:= [основная семантическая окрестность]

:= пояснение*:

[вид *семантической окрестности*, содержащий минимальную (краткую) информацию об описываемой сущности]

Структура базовой семантической окрестности определяется прежде всего семантической типологией описываемой сущности.

Так, например, для *понятия* в базовую семантическую окрестность необходимо включить следующую информацию (при наличии):

- варианты идентификации на различных внешних языках (sc-идентификаторы);
- принадлежность некоторой *предметной области* с указанием роли, выполняемой в рамках этой предметной области;
- определение* или пояснение.

Для *понятия*, являющегося *отношением* дополнительно указываются:

- домены*;
- область определения*;
- описание типичного примера связи указанного отношения (спецификация типичного экземпляра).

]

специализированная семантическая окрестность

- ⊃ *пояснение*
- ⊃ *примечание*
- ⊃ *правило идентификации экземпляров*
- ⊃ *терминологическая семантическая окрестность*
- ⊃ *теоретико-множественная семантическая окрестность*
- ⊃ *логическая семантическая окрестность*
- ⊃ *описание типичного экземпляра*
- ⊃ *описание декомпозиции*
- := *пояснение**:

[вид *семантической окрестности*, набор связей для которой уточняется отдельно для каждого типа такой окрестности.]

пояснение

- := [sc-пояснение]
- := *пояснение**:
- [знак sc-текста, поясняющего описываемую сущность.]

примечание

- := [sc-примечание]
- := *пояснение**:
- [знак sc-текста, являющегося примечанием к описываемой сущности. В примечании обычно описываются особые свойства и исключения из правил для описываемой сущности.]

правило идентификации экземпляров

- := [правило идентификации экземпляров заданного класса]
- := *пояснение**:
- [sc-текст являющийся описанием правил построения идентификаторов элементов заданного класса.]

терминологическая семантическая окрестность

- := *пояснение**:
- [*семантическая окрестность*, описывающая внешнюю идентификацию указанной сущности, т.е. её sc-идентификаторы]

теоретико-множественная семантическая окрестность

- := *пояснение**:
- [описание связи описываемого множества с другими множествами с помощью теоретико-множественных отношений]

описание декомпозиции

- := [*семантическая окрестность*, описывающая декомпозицию некоторой сущности]

:=	<p><i>пояснение*</i>: [<i>семантическая окрестность</i>, описывающая декомпозицию некоторой сущности на её части]</p>
логическая семантическая окрестность	
:=	<p><i>пояснение*</i>: [<i>семантическая окрестность</i>, описывающая семейство высказываний, описывающих свойства данного понятия или какого-либо конкретного экземпляра некоторого понятия]</p>
спецификация типичного экземпляра	
:=	[описание типичного экземпляра заданного класса]
:=	<p><i>пояснение*</i>: [sc-текст являющийся описанием типичного примера рассматриваемого класса.]</p>
сравнительный анализ	
:=	<p><i>пояснение*</i>: [описание сравнения некоторой сущности с другими аналогичными сущностями]</p>
сравнение	
:=	<p><i>пояснение*</i>: [описание сравнения (сходств и отличий) двух сущностей, которые заданы <i>парой</i> (двухмощными множествами), которому принадлежат знаки обеих сравниваемых сущностей]</p>
семантическая окрестность	
⇒	<p><i>примечание*</i>: [Всему классу <i>семантических окрестностей</i> и всем подклассам этого класса, а так же всем другим классам знаний, ставятся в соответствие <i>бинарные ориентированные отношения</i>, вторыми доменами которых являются указанные классы и объединение которых является <i>обратным отношением</i> для отношения "быть ключевым знаком". Эти отношения не следует причислять к основным отношениям, т.к. они вместе с выделенными классами семантических окрестностей привносят дополнительную логическую эквивалентность в базу знаний.]</p>
семантическая окрестность	
⇒	<p><i>отношение, заданное на*</i>:</p> <ul style="list-style-type: none"> • <i>семантическая окрестность*</i> • <i>семантическая окрестность по инцидентным коннекторам*</i> • <i>полная семантическая окрестность*</i> • <i>базовая семантическая окрестность*</i> • <i>специализированная синтетическая окрестность*</i> • <i>и т.д.</i>
⇒	<p><i>примечание*</i>: [Понятие семантической окрестности, дополненное уточнением таких понятий, как семантическое расстояние между знаками (семантическая близость знаков), радиус семантической окрестности, является перспективной основой для исследования свойств смыслового пространства.]</p>
}	/* Завершили Раздел "Предметная область и онтология семантических окрестностей" */

```

/* Раздел *****/
Предметная область и онтология предметных областей
⊃=
{
← конкатенация сегментов*:
{
• Что такое предметная область
• Роли знаков, входящих в состав предметных областей
• Типология предметных областей и отношения, заданных на множестве предметных областей
• Что такое sc-язык
}
}

Предметная область предметных областей
:= [Предметная область, объектами исследования которой являются предметные области]
⇒ пояснение*:
[В состав Предметной области предметных областей входят структурные спецификации всех предметных областей, входящих в состав базы знаний ostis-системы, в том числе, самой Предметной области предметных областей. Таким образом, Предметная область предметных областей является, во-первых, рефлексивным множеством, во-вторых, рефлексивной предметной областью, то есть предметной областью, одним из объектов исследования которой является она сама.]
∈ рефлексивное множество
Э максимальный класс объектов исследования':
предметная область
Э класс объектов исследования':
• статическая предметная область
• динамическая предметная область
• понятие
• sc-язык
Э исследуемое отношение':
• понятие предметной области'
• исследуемое понятие'
• максимальный класс объектов исследования'
• не максимальный класс объектов исследования'
• исследуемый класс первичных элементов'
• исследуемое отношение'
• класс исследуемых структур'
• понятие, исследуемое в дочерней предметной области'
• понятие, исследуемое в материнской предметной области'
• вспомогательное понятие'
• дочерняя предметная область*
• дочерняя предметная область по классу первичных элементов*
• дочерняя предметная область по исследуемым отношениям*
• предметная область sc-языка*

/* Сегмент *****/
Что такое предметная область
⊃=
{
предметная область
:= [sc-модель предметной области]
:= [sc-текст предметной области]
:= [sc-граф предметной области]
:= [представление предметной области в SC-коде]
С знание
С бесконечное множество
⇒ пояснение*:
[предметная область – это результат интеграции (объединения) частичных семантических окрестностей, описывающих все исследуемые сущности заданного класса и имеющих одинаковый (общий) предмет

```

исследования (то есть один и тот же набор отношений, которым должны принадлежать связки, входящие в состав интегрируемых семантических окрестностей).

предметная область – структура, в состав которой входят:

- основные исследуемые (описываемые) объекты – первичные и вторичные;
- различные классы исследуемых объектов;
- различные связки, компонентами которых являются исследуемые объекты (как первичные, так и вторичные), а также, возможно, другие такие связки – то есть связки (как и объекты исследования) могут иметь различный структурный уровень;
- различные классы указанных выше связей (то есть отношения);
- различные классы объектов, не являющихся ни объектами исследования, ни указанными выше связками, но являющихся компонентами этих связей.

При этом все классы, объявленные исследуемыми понятиями, должны быть полностью представлены в рамках данной предметной области вместе со своими элементами, элементами элементов и т.д. вплоть до терминальных элементов.

Можно говорить о типологии **предметных областей** по разным структурным признакам:

- наличие метасвязей;
- наличие исследуемых структур, входящих в состав предметной области;
- наличие исследуемых (смежных, дополнительных) объектов, которых исследуются в других предметных областях;

Понятие **предметной области** является важнейшим методологическим приемом, позволяющим выделить из всего многообразия исследуемого Мира только определенный класс исследуемых сущностей и только определенное семейство отношений, заданных на указанном классе. То есть осуществляется локализация, фокусирование внимания только на этом, абстрагируясь от всего остального исследуемого Мира.

Во всем многообразии **предметных областей** особое место занимают

- Предметная область предметных областей**, объектами исследования которой являются всевозможные **предметные области**, а предметом исследования – всевозможные **ролевые отношения**, связывающие предметные области с их элементами, отношения, связывающие предметные области между собой, отношение, связывающее предметные области с их онтологиями
- Предметная область сущностей**, являющаяся предметной областью самого высокого уровня и задающая базовую семантическую типологию *sc-элементов* (знаков, входящих в тексты *SC-кода*)
- Семейство **предметных областей**, каждая из которых задает семантику и синтаксис некоторого *sc-языка*, обеспечивающего представление онтологий соответствующего вида (например, *теоретико-множественных онтологий, логических онтологий, терминологических онтологий, онтологий задач и способов их решения* и т.д.)
- Семейство **предметных областей** верхнего уровня, в которых классами объектов исследования являются весьма "крупные" классы сущностей. К таким классам, в частности
 - класс всевозможных *материальных сущностей*,
 - класс всевозможных *множеств*,
 - класс всевозможных *связей*,
 - класс всевозможных *отношений*,
 - класс всевозможных *структур*,
 - класс всевозможных *временных (временно существующих, непостоянных сущностей) сущностей*,
 - класс всевозможных *действий* (акций),
 - класс всевозможных *параметров* (характеристик),
 - класс *знаний* всевозможного вида
 - и т.п.

Каждой **предметной области** можно поставить в соответствие:

- семейство соответствующих ей *онтологий* разного вида;
- некий язык (в нашем случае – язык, построенный на основе *SC-кода*), тексты которого представляют различные фрагменты соответствующей предметной области

Указанные языки будем называть *sc-языками*. Их синтаксис и семантика полностью задается *SC-кодом* и *онтологией* соответствующей **предметной области**. Очевидно, что в первую очередь нас должны

интересовать те *sc-языки*, которые соответствуют **предметным областям**, имеющим общий (условно говоря, предметно независимый) характер. К таким предметным областям, в частности, относятся:

- Предметная область множеств*, описывающая множества и различные связи между ними
- Предметная область отношений и соответствий*
- Предметная область структур* (в частности, графовых)
- Предметная область чисел и числовых структур*
- и т.д

Каждому типу знаний можно поставить в соответствие предметную область, которая является результатом интеграции всех знаний данного типа. Эти знания и становятся объектами исследования в рамках указанной предметной области.

Понятие **предметной области** может рассматриваться как обобщение понятия алгебраической системы. При этом семантическая структура базы знаний может рассматриваться как иерархическая система различных **предметных областей**.]

:= [система связей некоторого множества объектов исследования, ключевыми элементами которой являются:

- классы (точнее, знаки классов) объектов исследования (объектов, описываемых этой предметной областью);
- конкретные объекты исследования, обладающие особыми свойствами;
- классы связей, входящих в состав рассматриваемой системы – отношения, заданные на множестве элементов рассматриваемой системы;
- параметры, заданные на множестве элементов рассматриваемой системы;
- классы структур, являющихся фрагментами рассматриваемой системы.

]]

:= [структура, представляющая собой множество связей (точнее, знаков связей) и соответствующее множество компонентов этих связей, к числу которых относится:

- элементы (экземпляры) некоторых заданных классов объектов исследования (первичных исследуемых сущностей);
- сами связи, входящие в состав указанной структуры;
- введенные классы объектов исследования;
- введенные отношения (классы связей);
- введенные параметры (классы классов эквивалентных сущностей);
- значения параметров (и, в частности, величины для измеряемых параметров);
- введенные структуры, являющиеся фрагментами (подструктурами) рассматриваемой структуры;
- введенные классы подструктур рассматриваемой структуры.

]]

⇒ *примечание**:

[Выделяемые в рамках *базы знаний* интеллектуальной системы **предметные области** и соответствующие им *онтологии* – это, своего рода, семантические страты, кластеры, позволяющие "разложить" все хранимые в памяти *знания* по "семантическим полочкам" при наличии четких критериев, позволяющих однозначно определить то, на какой "полочке" должны находиться те или иные *знания*]

⇒ *примечание**:

[Существуют предметные области, в которых основным исследуемым понятием является множество всевозможных связей между экземплярами понятий, исследуемых в других предметных областях. Так, например, можно ввести Предметную область треугольников, Предметную область окружностей, а также Предметную область связей между треугольниками и окружностями.]

} /* Завершили Сегмент "Что такое предметная область" */

/* Сегмент *****/

Роли знаков, входящих в состав предметной области

⊃=
{

роль элемента предметной области

:= [ролевое отношения, связывающее предметные области с их ключевыми знаками]

:= [роль ключевого элемента (знака ключевой сущностей) предметной области]

:= [роль ключевого знака предметной области]

Э *класс объектов исследования'*

Э *максимальный класс объектов исследования'*

Э *ключевой объект исследования'*

Э *понятие, используемое в предметной области'*

Э *первичный исследуемый элемент предметной области'*

Э *вторичный исследуемый элемент предметной области'*

Э *неисследуемый элемент предметной области'*

класс объектов исследования'

:= [быть классом первичных (для данной предметной области) объектов исследования']

⇒ *примечание**:

[Понятие первичного объекта исследования для предметной области является понятием относительным и абсолютно не зависит от типа и уровня сложности этого объекта. Само исследование (спецификация) таких первичных исследуемых объектов осуществляется:

- путем введения различных классов объектов исследования, которым эти объекты принадлежат;
- путем введения различных связок из первичных объектов исследования и различных классов таких связок (отношений), которым принадлежат введенные связки;
- путем введения таких классов первичных объектов исследования, которые являются значениями вводимых параметров;
- путем введения различных структур, состоящих из первичных объектов исследования, из связок таких объектов, из введенных отношений и классов первичных объектов, из введенных параметров и значений этих параметров, и путем введения различных классов таких структур;
- путем введения различных связок из вторичных объектов исследования (т.е. из связок и структур) и путем введения различных классов таких связок;
- и далее можно переходить к объектам исследования более высокого уровня сложности, к параметрам, элементами значений которых являются такие объекты, а также к структурам, элементами которых являются объекты такого уровня и, соответственно, к классам таких структур.

]

⇒ *второй домен**:

класс

- ⊃ {
- *множество*
 - *отношение*
 - ⊃ *множество*
 - *параметр*
 - ⊃ *класс классов*
 - *значение параметра*
 - ⊃ *класс*
 - *структура*
 - ⊃ *множество*
 - *темпоральная сущность*
 - *темпоральная сущность базы знаний *ostis-системы**
 - *семантическая окрестность*
 - *предметная область*
 - *онтология*
 - *логическая формула*
 - *действие*
 - *задача*
 - *информационная конструкция*
 - *язык*
 - *sc-конструкция*
 - *кибернетическая система*
 - *интеллектуальная компьютерная система*
 - *знание*
 - *база знаний*
 - *решатель задач интеллектуальной компьютерной системы*
 - *интерфейс интеллектуальной компьютерной системы*
 - *компьютерная система, основанная на смысловом представлении информации*

- *смысловое представление информации*
- *многоагентная модель решения задач, основанная на смысловом представлении информации*
- *логико-семантическая модель интерфейсов компьютерных систем, основанных на смысловом представлении информации*
- *решатель задач ostis-системы*
- *действие, выполняемое ostis-системой*
- *задача, решаемая ostis-системой :план решения задачи, реализуемый ostis-системой*
- *протокол решения задачи, реализованный ostis-системой*
- *метод решения класса задач, реализуемый ostis-системой*
- *sc-агент*

:= [внутренний агент ostis-системы, осуществляющий выполнение некоторого вида действий в памяти ostis-системы]

⊃ *sc-агент обработки информации в памяти ostis-системы*

⊃ *sc-агент управления внешними действиями ostis-системы*

- *Базовый язык программирования ostis-систем*

:= [Язык SCP]

- *искусственная нейронная сеть*
- *интерфейс ostis-системы*
- *интерфейсное действие пользователя ostis-системы*
- *sc-агент интерфейса ostis-системы*
- *естественный язык*
- *базовый интерпретатор логико-семантических моделей ostis-систем*
- *базовый интерпретатор логико-семантических моделей ostis-систем, реализованный программно на современных компьютерах*
- *семантический ассоциативный компьютер*
- *обучение пользователей ostis-систем*
- *ostis-система персональной адаптивной поддержки всех видов деятельности пользователя*
- *ostis-система управления рецептурным производством*
- *ostis-система, реализующая интеллектуальный портал научно-технических знаний*

}
⇒

примечание:*

[Здесь приведено семейство тех *классов объектов исследования*, для которых в текущей версии *Стандарта OSTIS* представлены соответствующие *предметные области*. Очевидно, что это семейство должно быть существенно расширено и включить в себя, например, такие *классы сущностей*, как:

- материальная сущность*
- вещество*
- физическое поле*
- персона*
- пространственная сущность*
- юридическое лицо*
- предприятие*
- географический объект*
- и многие другие*

]

⇒ *примечание*:*

[Особого внимания требуют те *классы объектов исследования*, которые носят наиболее общий характер которым соответствуют *предметные области и онтологии высокого уровня*. Здесь важна продуманная система декомпозиции всего множества окружающих нас сущностей на иерархическую систему *классов объектов исследования*, которой соответствует иерархическая система *предметных областей и онтологий*, определяющая направления наследования свойств исследуемых объектов.]

максимальный класс объектов исследования'

:= [класс объектов исследования, для которого в заданной (!) предметной области отсутствует другой класс объектов исследования, который был бы его надмножеством']

⇒ *примечание*:*

[В некоторых предметных областях может быть несколько максимальных классов объектов исследования]

ключевой объект исследования'

- := [особый объект исследования']
- := [быть знаком особого исследуемого объекта в рамках заданной предметной области']
- := [объект исследования, обладающий особыми свойствами']
- Э *пример'*:
 ⟨Предметная область чисел; Нуль⟩
 ⇒ *примечание**:
 [Особыми свойствами Числа **Нуль** являются:
 Результатом сложения Числа **Нуль** с любым числом **x** является число **x**;
 Результатом умножения Числа **Нуль** на любое число является Число **Нуль**
]
- Э ⟨Предметная область чисел; Единица⟩
 Э ⟨Предметная область чисел; Число Пи⟩
 Э ⟨Предметная область чисел; Число E⟩

ключевой элемент предметной области'

- := [входящий в состав предметной области знак ключевой сущности']
- ⇒ *разбиение**:
 {
 • *понятие, используемое в предметной области'*
 • *ключевой объект исследования'*
 := [знак ключевого объекта исследования']
 }

понятие, используемое в предметной области'

- := [понятие, используемое в заданной предметной области не в качестве одного из объектов исследования, а в качестве ключевого понятия']
- С *используемое понятие'*
 := [понятие, используемое в sc-знании']
 С *используемое понятие**
 := [понятие, используемое в знании, которое может быть представлено не только в SC-коде*]
- ⇒ *примечание**:
 [Уточнение характера использования понятия в предметной области осуществляется по трем признакам:
 семантический тип используемого понятия;
 полнота вхождения элементов понятия в данную предметную область;
 наличие первого упоминания понятия;
 наличие определения понятия или объявления его неопределяемостис подробным пояснением и примерами;
 наличие исследования понятия.
]
- ⇒ *разбиение**:
семантический тип используемого понятия
 = {
 • *класс объектов исследования'*
 • *отношение, используемое в предметной области'*
 • *параметр, используемый в предметной области'*
 • *класс структур, используемый в предметной области'*
 }
- ⇒ *разбиение**:
полнота вхождения элементов понятия в данную предметную область
 = {
 • *используемое понятие, все элементы которого входят в данную предметную область'*
 ⇒ *примечание**:
 [Для каждого используемого отношения в предметную область здесь должны входить не только знаки связок, но и все связки целиком с их компонентами]
 • *используемое понятие, не все элементы которого входят в данную предметную область'*
 }
- ⇒ *разбиение**:
наличие первого упоминания понятия
 = {
 • *понятие, вводимое в данной предметной области'*

- *понятие, которое в данной предметной области используется, но не вводится'*

}

⇒ *примечание**:

[Будем считать, что понятие вводится в данной предметной области в том и только в том случае, если ни в одной предметной области более высокого уровня это понятие не используется. Т.е. речь идет о первом упоминании этого понятия в рамках последовательности предметных областей от родительских к дочерним]

⇒ *разбиение**:

наличие определения понятия или объявления его неопределяемости с подробным пояснением и примерами

- *понятие, которое в данной предметной области определено или объявлено как неопределяемое*
- *понятие, которое в данной предметной области не имеет ни определения, ни указания факта его неопределяемости*

}

⇒ *разбиение**:

наличие исследования понятия

- *понятие, исследуемое в данной предметной области'*
- *понятие, которое в данной предметной области используется, но не исследуется'*

}

⇒ *примечание**:

[Понятие, используемое в базе знаний, может быть введено (впервые упомянуто) в одной предметной области, определено в другой, а исследоваться – в третьей]

первичный исследуемый элемент предметной области'

:= [знак первичного объекта исследования в рамках заданной предметной области']

вторичный исследуемый элемент предметной области'

:= [знак вторичного объекта исследования в рамках предметной области']

⊃ *связка элементов предметной области'*

⊃ *связка первичных элементов предметной области'*

⊃ *метасвязка элементов предметной области'*

⊃ *метасвязка, в число компонентов которой входят связки элементов предметной области'*

⊃ *метасвязка, в число компонентов которой входят классы элементов предметной области'*

⊃ *метасвязка, в число компонентов которой входят структуры элементов предметной области'*

⊃ *класс элементов предметной области'*

⊃ *класс первичных элементов предметной области'*

⊃ *класс связок элементов предметной области'*

⊃ *класс классов элементов предметной области'*

⊃ *класс структур элементов предметной области'*

⊃ *структура элементов предметной области'*

⊃ *тривиальная структура первичных элементов предметной области'*

⊃ *структура, в число подмножеств которой входят связки элементов предметной области вместе со своими компонентами'*

⊃ *структура, в число подмножеств которой входят классы элементов предметной области вместе со своими знаками'*

⊃ *структура, в число подмножеств которой входят другие структуры вместе со своими знаками'*

неисследуемый элемент предметной области'

:= [вспомогательный элемент предметной области, исследуемый в другой (смежной) предметной области']

⇒ *примечание**:

[С помощью неисследуемых элементов предметной области описываются и исследуются различные виды связи между элементами, исследуемыми в данной предметной области с элементами, исследуемыми в других предметных областях. При этом связки, компонентами которых являются как исследуемые, так и неисследуемые элементы данной предметной области считаются исследуемыми связками этой предметной области. Примерами неисследуемых элементов, напримр, геометрической предметной области являются числа, являющиеся значениями величин таких параметров, как расстояние[^], длина[^], площадь[^], объем[^], а также различные числовые отношения (сложение*, умножение*, возведение в степень*), теоретико-множественные отношения (включение*, объединение*, пересечение*, принадлежность*)]

понятие

```

:= [концепт]
:= [класс сущностей, который входит в состав по крайней мере одной предметной области в качестве (в роли)
    ключевого исследуемого понятия]
⇒ примечание*:
    [Семейство всех введенных понятий – это, своего рода, семантическая система координат, позволяющая
    специфицировать всевозможные сущности в смысловом пространстве.]
:= [класс сущностей, который по крайней мере в одной предметной области "объявлен" как понятие (вводимое,
    исследуемое или вспомогательное)]
⇒ примечание*:
    [Каждому понятию соответствует по крайней мере одна предметная область, в которой это понятие
    является исследуемым понятием и в которой рассматриваются основные характеристики этого понятия.
    Если же в какой-либо предметной области необходимо рассмотреть дополнительные связи этого понятия
    с другими понятиями, то оно объявляется как вспомогательное понятия'.]
:= [Второй домен Отношения используемое понятие*]
⇐ второй домен*:
    используемое понятие*
:= [класс сущностей (класс связок (в т.ч. отношение), класс классов (в т.ч. параметр), класс структур), который
    по крайней мере в одной предметной области является используемым понятием']
} /* Завершили Сегмент "Роли знаков, входящих в состав предметной области" */

```

/* Сегмент *****/

Типология предметных областей и отношения, заданные на множестве предметных областей

```

⊃=
{

```

предметная область

```

⇒ разбиение*:
{
• статическая предметная область
  := [стационарная предметная область]
  := [предметная область, в которой связи между сущностями, входящими в ее состав, не зависят
    от времени (не меняются во времени), элементами статической предметной области не
    могут быть временные сущности]
• квазистатическая предметная область
  := [предметная область, решение задач в которой не требует учета темпоральных свойств
    объектов исследования]
• динамическая предметная область
  := [нестационарная предметная область]
  := [предметная область, которая описывает изменение состояния (в том числе внутренней
    структуры) объектов исследования и/или изменение конфигурации связей между объектами
    исследования]
  := [предметная область, в которой некоторые связи между сущностями, входящими в ее состав,
    меняются со временем (то есть носят ситуационный, нестационарный характер, другими
    словами, являются временными сущностями)]
}
⇒ разбиение*:
{
• первичная предметная область
  := [предметная область, объектами исследования которой являются внешние сущности (обо-
    значаемые первичными sc-элементами)]
• вторичная предметная область
  := [метапредметная область]
  := [предметная область, объектами исследования которой являются sc-множества (отношения,
    параметры, структуры, классы структур, знания, языки и др.)]
}
⇒ примечание*:
[Во всем многообразии предметных областей особое место занимают:

```

- **Предметная область предметных областей**, объектами исследования которой являются всевозможные предметные области, а предметом исследования являются – всевозможные ролевые отношения, связывающие предметные области с их элементами, отношения, связывающие предметные области между собой, отношение, связывающее предметные области с их онтологиями;
- **Предметная область сущностей**, являющаяся предметной областью самого высокого уровня и задающая базовую семантическую типологию *sc*-элементов (знаков, входящих в тексты *SC*-кода);
- Семейство *предметных областей*, каждая из которых задает семантику и синтаксис некоторого *sc*-языка, обеспечивающего представление онтологий соответствующего вида (например, теоретико множественных онтологий терминологических онтологий);
- Семейство *предметных областей* верхнего уровня, в которых классами объектов исследования являются весьма "крупные" классы сущностей. К таким классам, в частности, относятся:
 - класс всевозможных материальных сущностей,
 - класс всевозможных множеств,
 - класс всевозможных связей,
 - класс всевозможных отношений,
 - класс всевозможных структур,
 - класс всевозможных темпоральных (нестационарных) сущностей,
 - класс всевозможных действий (воздействий, акций),
 - класс всевозможных параметров (характеристик),
 - класс знаний всевозможного вида и т.п.;
- Предметные области абстрактных пространств (в том числе предметные области метрических пространств). Примерами абстрактного пространства являются Евклидово пространство геометрических точек и фигур, пространство всевозможных множеств, числовое пространство, *SC*-пространство (унифицированное смысловое пространство знаков всевозможных сущностей).

]

отношение, заданное на множестве предметных областей

Э **дочерняя предметная область***

:= [частная предметная область*]

:= [быть частной предметной областью*]

:= [близлежащий потомок предметной области*]

:= [сужение предметной области по классу объектов исследования*]

:= [предметная область, детализирующая описание одного из классов объектов исследования другой (более общей) предметной области*]

:= [предметная область, объединение классов объектов исследования которой является подмножеством объединения классов объектов исследования заданной предметной области*]

€ бинарное отношение

€ ориентированное отношение

€ неролевое отношение

⊃ частная предметная область по классу первичных элементов*

⊃ частная предметная область по исследуемым отношениям*

⇒ пояснение*:

[дочерняя предметная область* – бинарное ориентированное отношение, с помощью которого задается иерархия предметных областей путем перехода от менее детального к более детальному рассмотрению соответствующих исследуемых понятий.]

⇒ примечание*:

[Для любой предметной области все свойства ее объектов исследования наследуются всеми ее дочерними предметными областями*.]

Э **интеграция предметных областей***

:= [Отношение, связывающее заданное семейство предметных областей с предметной областью, которая является результатом их интеграции (это не только теоретико-множественное объединение заданных предметных областей, но и уточнение ролей ключевых понятий в интегрированной предметной области, поскольку одно и то же понятие в интегрируемых предметных областях может иметь разные роли).]

Э **изоморфность предметных областей***

Э **гомоморфность предметных областей***

расширение семейства исследуемых отношений*

⇒ *пояснение**:

[Переход от одной предметной области к предметной области с тем же максимальным классом объектов исследования, но с расширенным семейством отношений и, возможно, с расширенным семейством явно выделенных классов объектов исследования (подклассов максимального класса).]

переход к рассмотрению внутренней структуры объектов исследования*

⇒ *пояснение**:

[Переход от рассмотрения внешних связей объектов исследования к рассмотрению их "внутренней" структуры путем декомпозиции исследуемых объектов на части и путем включения в число исследуемых объектов тех, которые являются указанными частями.]

переход к рассмотрению структур из объектов исследования*

⇒ *пояснение**:

[Переход от описания заданного класса исследуемых объектов к описанию класса всевозможных множеств, элементами которых являются указанные объекты (например, переход от предметной области геометрических точек к предметной области геометрических фигур).]

} /* Завершили Сегмент "Типология предметных областей и отношения, заданные на множестве предметных областей" */

/* Сегмент *****/

Что такое sc-язык

⊃=

{

sc-язык

:= [максимальное множество текстов SC-кода, являющихся фрагментами соответствующей предметной области (точнее, ее sc-модели)]

⇒ *пояснение**:

[**sc-язык** — это подязык (подмножество) *SC-кода*, ориентированный на представление *sc-текстов*, являющихся фрагментами некоторой *предметной области*. Таким образом, каждому *sc-языку* взаимно однозначно соответствует некоторая *предметная область* (точнее, sc-модель некоторой *предметной области*).]

предметная область sc-языка*

:= [предметная область заданного sc-языка*]

:= [быть предметной областью соответствующей заданному sc-языку*]

:= [sc-язык и соответствующая ему предметная область*]

∈ *бинарное отношение*

⇒ *пояснение**:

[**предметная область sc-языка*** - это бинарное ориентированное отношение, каждая связка которого связывает знак некоторого *sc-языка* (первый компонент связки данного отношения) и знак соответствующей этому *sc-языку предметной области*.]

⇒ *первый домен**:

sc-язык

⇒ *второй домен**:

предметная область

sc-язык

:= [подязык *SC-кода*]

⇒ *пояснение**:

[Множество языков, синтаксис каждого из которых полностью соответствует синтаксису *SC-кода* (т.е. каждый из них является подязыком *SC-кода*), а денотационная семантика каждого из них определяется интегральной (объединенной) онтологией *предметной области*, которая взаимно однозначно соответствует этому sc-языку.]

⇒ *примечание**:

[Каждой предметной области можно поставить в соответствие множество *sc*-текстов, которые являются фрагментами *sc*-моделей этой *предметной области*, описывающими (специфицирующими) свойства объектов, исследуемых в указанной *предметной области*. Указанное множество *sc*-текстов будем называть *sc*-языком, соответствующим указанной *предметной области*. Очевидно, что *предметных областей* и соответствующих им *sc*-языков существует неограниченное количество. Синтаксис и семантика *sc*-языков полностью задается *SC*-кодом и онтологией соответствующей *предметной области*. Очевидно, что в первую очередь нас должны интересовать те *sc*-языки, которые соответствуют предметным областям, имеющим общий (условно говоря, предметно независимый) характер. К таким предметным областям, в частности, относятся:

- *Предметная область множеств*, описывающая множества и различные связи между ними
- *Предметная область структур*
- *Предметная область чисел и числовых структур*
- *Предметная область отношений*
- *Предметная область параметров, величин и шкал*
- *Предметная область логических формул, высказываний и формальных теорий*
- и другие

]

⇒ *примечание**:

[Особое место для всего семейства *sc*-языков занимают две предметные области и соответствующие им онтологии:

- *Предметная область SC-кода и онтология*, определяющая его базовую денотационную семантику, задаваемую синтаксисом *SC*-кода
- *Предметная область и онтология Базового универсального sc-языка*

]

⇒ *примечание**:

[Подчеркнем, что все *sc*-языки, кроме тех, которые задаются указанными двумя предметными областями и соответствующими им онтологиями, являются специализированными *sc*-языками.]

Предметная область SC-кода и онтология, определяющая его базовую денотационную семантику, задаваемую синтаксисом SC-кода

⇒ *пояснение**:

[Указанная онтология *SC*-кода определяет денотационную семантику *SC*-кода путем семантической интерпретации синтаксически выделяемых классов *sc*-элементов. Т.е. данная денотационная семантика уточняет смысл только тех классов *sc*-элементов, которые задаются синтаксически путем "приписывания" *sc*-элементам соответствующей "синтаксической метки".]

Предметная область и онтология Базового универсального sc-языка

⇒ *пояснение**:

[Указанная онтология Базового универсального *sc*-языка определяет денотационную семантику этого Базового универсального *sc*-языка и представляет собой онтологию верхнего уровня, которая уточняет смысл понятий, лежащий в основе указанного *sc*-языка. В эту онтологию входит уточнение таких понятий, как материальная сущность, абстрактная сущность, число, пространство, множество, связь, класс, отношение, параметр, структура, класс структур, информационная конструкция, знание, статическая сущность, динамическая сущность, процесс, ситуация, действие, постоянная сущность, временная сущность, часть*, декомпозиция*, пространственная часть*, темпоральная часть* и т.д.]

} /* Завершили Сегмент “Что такое *sc*-язык” */

} /* Завершили Раздел “Предметная область и онтология предметных областей” */


```

/* Раздел *****/
Предметная область и онтология онтологий
 $\supseteq$ 
{
 $\leftarrow$  конкатенация сегментов*:
{
• Что такое онтология
• Типология онтологий предметной области
• Понятие объединенной онтологии предметной области, понятие предметной области и онтологии
• Отношения, заданные на множестве онтологий
}
}

Предметная область онтологий
:= [Предметная область теории онтологий]
:= [Предметная область, объектами исследования которой являются онтологии]
 $\in$  предметная область
 $\ni$  максимальный класс объектов исследования':
онтология
 $\ni$  класс объектов исследования':
• объединенная онтология
• структурная спецификация предметной области
• теоретико-множественная онтология предметной области
• логическая онтология предметной области
• логическая иерархия понятий предметной области
• логическая иерархия высказываний предметной области
• терминологическая онтология предметной области
 $\ni$  исследуемое отношение':
• онтология*
• используемые константы*
• используемые утверждения*

/* Сегмент *****/
Что такое онтология
 $\supseteq$ 
{
онтология
:= [sc-онтология]
:= [онтология, представленная в SC-коде]
 $\Rightarrow$  примечание*:
[Поскольку термин “онтология” в SC-коде соответствует множеству всевозможных онтологий, представленных в SC-коде, то для формальных онтологий, представленных на других языках, необходимо использовать sc-идентификатор, содержащий явное указание этих языков, например, “owl-онтология”.]
:= [sc-текст онтологии]
:= [sc-модель онтологии]
:= [семантическая спецификация знаний]
:= [семантическая спецификация любого знания, имеющего достаточно сложную структуру, любого целостного фрагмента базы знаний – предметной области, метода решения сложных задач некоторого класса, описания истории некоторого вида деятельности, описания области выполнения некоторого множества действий (области решения задач), языка представления методов решения задач и т.д.]
:= пояснение*:
[семантическая спецификация некоторого достаточно информативного ресурса (знания)]
 $\subset$  спецификация
 $\Rightarrow$  примечание*:
[Если спецификация может специфицировать (описывать) любую сущность, то онтология специфицирует только различные знания. При этом наиболее важными объектами такой спецификации являются предметные области]
 $\subset$  метазнание
 $\in$  вид знаний

```

- ⇒ *примечание**:
 [онтологии являются важнейшим *видом знаний* (точнее, *метазнаний*), обеспечивающих семантическую систематизацию *знаний*, хранимых в памяти *интеллектуальных компьютерных систем* (в т.ч. *ostis-систем*), и, соответственно, семантическую структуризацию *баз знаний*]
- := [важнейший вид *метазнаний*, входящих в состав *базы знаний*]
- := [спецификация (уточнение) системы *понятий*, используемых в соответствующем (специфицируемом) *знании*]
- ⇒ *эпиграф**:
 [Определив точно значения слов, вы избавите человечество от половины заблуждений]
 ⇒ *автор**:
 Рене Декарт
- ⇒ *пояснение**:
 [онтология включает в себя:
 □ типологию специфицируемого *знания*;
 □ связи специфицируемого *знания* с другими *знаниями*;
 □ спецификацию ключевых *понятий*, используемых в специфицируемом *знании*, а также ключевых экземпляров некоторых таких *понятий*.
]
- ⇒ *пояснение**:
 [Основная *цель* построения *онтологии* – семантическое уточнение (пояснение, а в идеале – определение) такого семейства *знаков*, используемых в заданном *знании*, которых достаточно для понимания смысла всего специфицируемого *знания*. Как выясняется, количество *знаков*, смысл которых определяет смысл всего специфицируемого *знания*, не является большим.]
- ⇒ *разбиение**:
 { • *неформальная онтология*
 • *формальная онтология*
 := [онтология, представленная на формальном языке]
 ← *ключевой знак**:
 Лукашевич.Н.В.ТезауВЗИП-2011кн
 }
- формальная онтология**
- := [формальное описание денотационной семантики (семантической интерпретации) специфицируемого *знания*]
- ⇒ *примечание**:
 [Очевидно, что при отсутствии достаточно полных формальных онтологий невозможно обеспечить семантическую совместимость (интегрируемость) различных *знаний*, хранимых в *базе знаний*, а также приобретаемых *извне*.]
- онтология предметной области**
- ⇒ *примечание**:
 [онтология чаще всего трактуется как спецификация концептуализации (спецификация системы *понятий*) заданной *предметной области*. Здесь имеется в виду описание теоретико-множественных связей (прежде всего, классификации) используемых *понятий*, а также описание различных закономерностей для сущностей, принадлежащих этим *понятиям*. Тем не менее, важными видами спецификации *предметной области* являются также:
 □ описание связей специфицируемой *предметной области* с другими *предметными областями*;
 □ описание терминологии специфицируемой *предметной области*.
]
- ⇒ *примечание**:
 [онтологию *предметной области* можно трактовать, с одной стороны, как *семантическую окрестность* соответствующей *предметной области*, с другой стороны, как *объединение* определённого вида *семантических окрестностей* всех *понятий*, используемых в рамках указанной *предметной области*, а также, возможно, ключевых экземпляров указанных *понятий*, если таковые экземпляры имеются]
- ⇒ *пояснение**:
 [Каждая конкретная онтология заданного вида представляет собой семантическую окрестность соответствующей (специфицируемой) предметной области. Каждому *виду онтологий* однозначно соответствует *предметная область*, фрагментами которые являются конкретные *онтологии* этого вида. Следовательно, каждому *виду онтологий* соответствует свой специализированный *sc-язык*, обеспечивающий представление *онтологий* этого вида.]

```

:= [описание денотационной семантики языка, определяемого (задаваемого) соответствующей (специфицируемой) предметной областью]
:= [информационная надстройка (метаинформация) над соответствующей (специфицируемой) предметной областью, описывающая различные аспекты этой предметной области как достаточно крупного, самостоятельного и семантически целостного фрагмента база знаний]
:= [метаинформация (метазнание) о некоторой предметной области]
} /* Завершили Сегмент "Что такое онтология" */

```

```

/* Сегмент *****/
Типология онтологий предметной области

```

```

⊃=
{

```

онтология предметной области

⇒ разбиение*:

- { • **частная онтология предметной области**
:= [онтология, представляющая спецификацию соответствующей предметной области в том или ином аспекте]
- **объединённая онтология предметной области**
:= [онтология предметной области, являющаяся результатом объединения всех известных частных онтологий этой предметной области]

частная онтология предметной области

⇒ примечание*:

[Каждая частная онтология является фрагментом предметной области, включающей в себя все(!) частные онтологии, принадлежащие соответствующему виду онтологий. При этом указанная предметная область, в свою очередь, также имеет соответствующую ей онтологию, которая уже является не метазнанием (как любая онтология), а метаметазнанием (спецификацией метазнания).]

⇒ разбиение*:

вид онтологий предметных областей

- = { • **структурная спецификация предметной области**
:= [sc-окрестность (sc-спецификация) заданной предметной области в рамках Предметной области предметных областей]
:= [схема предметной области]
- **теоретико-множественная онтология предметной области**
:= [sc-спецификация заданной предметной области в рамках Предметной области множеств]
- **логическая онтология предметной области**
:= [sc-текст формальной теории заданной предметной области]
- **терминологическая онтология предметной области**

вид онтологий предметных областей

:= [вид спецификаций предметных областей]

:= [вид метазнаний, описывающих соответствующие этому виду метазнаний свойства предметных областей]

⇒ примечание*:

[Каждому виду онтологий предметных областей, представленных в SC-коде (здесь представленными в SC-коде предполагаются не только сами онтологии, но и специфицируемые ими предметные области) ставится в соответствие sc-метаязык, обеспечивающий представление метазнаний, входящих в состав указанного вида онтологий. Кроме того, обычное теоретико-множественное объединение всех sc-текстов указанного sc-метаязыка означает построение предметной области (предметной метаобласти), которая формально задаёт соответствующий вид онтологий и которая будет иметь свою онтологию. При этом каждую частную онтологию можно связать с той предметной областью, фрагментом которой эта онтология является.]

логическая онтология предметной области

⇐ семейство подмножеств*:

*Предметная область логических формул, высказываний и формальных теорий***теоретико-множественная онтология предметной области**

⇐ *семейство подмножеств**:
Предметная область множеств

структурная спецификация предметной области

⇐ *семейство подмножеств**:
Предметная область предметных областей

структурная спецификация предметной области

- := [структурная онтология предметной области]
- := [ролевая структура ключевых элементов предметной области]
- := [схема ролей понятий предметной области и её связи со смежными предметными областями]
- := [схема предметной области]
- := [спецификация предметной области с точки зрения теории графов и теории *алгебраических систем*]
- := [описание внутренней (ролевой) структуры *предметной области*, а также её внешних связей с другими *предметными областями*]
- := [описание ролей ключевых элементов предметной области (прежде всего, понятий – концептов), а также "место" специфицируемой предметной области в множестве себе подобных]
- := [семантическая окрестность знака *предметной области* в рамках самой этой *предметной области*, включающая в себя все *ключевые знаки*, входящие в состав *предметной области* (ключевые понятия и ключевые объекты исследования предметной области) с указанием их ролей (свойств) в рамках этой *предметной области* и *семантическая окрестность* знака специфицируемой *предметной области* в рамках *Предметной области предметных областей*, включающая в себя связи специфицируемой *предметной области* с другими семантически близкими ей *предметными областями* (дочерними и родительскими, аналогичными в том или ином смысле (например, изоморфными), имеющими одинаковые *классы объектов исследования* или одинаковые наборы *исследуемых отношений*)]

теоретико-множественная онтология предметной области

:= [семантическая окрестность специфицируемой *предметной области* в рамках *Предметной области множеств*, описывающая теоретико-множественные связи между *понятиями* специфицируемой *предметной области*, включая связи *отношений* с их *областями определения* и *доменами*, связи используемых *параметров* и классов структур их *областями определения*]

- := [онтология описывающая:
- классификацию объектов исследования специфицируемой предметной области;
 - соотношение областей определения и доменов используемых отношений с выделенными классами объектов исследования, а также с выделенными классами вспомогательных (смежных) объектов, не являющихся объектами исследования в специфицируемой предметной области;
 - спецификацию используемых отношений и, в том числе, указание того, все ли связки этих отношений входят в состав специфицируемой предметной области

] *пояснение**:

[теоретико-множественная онтология предметной области включает в себя:

- теоретико-множественные связи (в т.ч. таксономию) между всеми используемыми понятиями, входящими в состав специфицируемой предметной области;
- теоретико-множественную спецификацию всех *отношений*, входящих в состав специфицируемой предметной области (ориентированность, арность, область определения, домены и т.д.);
- теоретико-множественную спецификацию всех параметров, используемых в предметной области (области определения параметров, шкалы, единицы измерения, точки отсчета);
- теоретико-множественную спецификацию всех используемых классов структур

]

логическая онтология предметной области

:= [формальная теория заданной (специфицируемой) предметной области, описывающая с помощью переменных, кванторов, логических связок, формул различные свойства экземпляров понятий, используемых в специфицируемой предметной области]

:= *пояснение**:

[онтология предметной области, которая включает в себя:

- формальные определения всех понятий, которые в рамках специфицируемой предметной области являются определяемыми;
- неформальные пояснения и некоторые формальные спецификации (как минимум, примеры) для всех понятий, которые в рамках специфицируемой предметной области являются неопределяемыми;
- иерархическую систему понятий, в которой для каждого понятия, исследуемого в специфицируемой предметной области либо указывается факт неопределяемости этого понятия, либо указываются все понятия, на основе которых даётся определение данному понятию. В результате этого множество исследуемых понятий разбивается на ряд уровней:
 - неопределяемые понятия;
 - понятия 1-го уровня, определяемые только на основе неопределяемых понятий;
 - понятия 2-го уровня, определяемые на основе понятий, изменяющих 1-й уровень и ниже;
 - и т.д.
- формальную запись всех аксиом, т.е. высказываний, которые не требуют доказательств;
- формальную запись высказываний, истинность которых требует обоснования (доказательства);
- формальные тексты доказательства истинности высказываний, представляющие собой спецификацию последовательности шагов соответствующих рассуждений (шагов логического вывода, применения различных правил логического вывода);
- иерархическую систему высказываний, в которой для каждого высказывания, истинного по отношению к специфицируемой предметной области, либо указывается аксиоматичность этого высказывания, либо перечисляются *все* высказывания, на основе которых доказывается данное высказывание. В результате этого множество высказываний, истинных по отношению к специфицируемой предметной области, разбивается на ряд уровней:
 - аксиомы;
 - высказывания 1-го уровня, доказываемые только на основе аксиом;
 - высказывания 2-го уровня, доказываемые на основе высказываний, находящихся на 1-м уровне и ниже.
- формальная запись гипотетических высказываний;
- формальное описание логико-семантической типологии высказываний – высказываний о существовании, о несуществовании, об однозначности, высказывания определяющего типа (которые можно использовать в качестве определений соответствующих понятий);
- формальное описание различного вида логико-семантических связей между высказываниями (например, между высказыванием и его обобщением);
- формальное описание аналогии
 - между определениями;
 - между высказываниями любого вида;
 - между доказательствами различных высказываний.

]

терминологическая онтология предметной области

:= [онтология, описывающая правила построения терминов (sc-идентификаторов), соответствующих sc-элементам, принадлежащим специфицируемой предметной области, а также описывающая различного рода терминологические связи между используемыми терминами, характеризующие происхождение этих терминов]

:= [система терминов заданной предметной области]

:= [тезаурус соответствующей предметной области]

:= [словарь соответствующей (специфицируемой) предметной области]

:= [фрагмент глобальной *Предметной области sc-идентификаторов* (внешних идентификаторов sc-элементов), обеспечивающий терминологическую спецификацию некоторой предметной области]

} /* Завершили Сегмент “*Типология онтологии предметной области*” */

```

/* Сегмент *****/
Понятие объединённой онтологии предметной области, а также понятие предметной области и онтологии
⊃=
{
объединённая онтология предметной области
:= [объединение всех частных онтологий, соответствующих одной предметной области]
← обобщённое объединение*:
{
• структурная спецификация предметной области
• теоретико-множественная онтология предметной области
• логическая онтология предметной области
• терминологическая онтология предметной области
}

предметная область и онтология
:= [интеграция некоторой предметной области с соответствующей ей объединённой онтологией]
:= [предметная область & онтология]
← обобщённое объединение*:
{
• предметная область
• объединённая онтология предметной области
}

:= [sc-текст, являющийся объединением некоторой предметной области, представленной в SC-коде, и объединённой онтологии этой предметной области, также представленной в SC-коде]
:= [интеграция предметной области и всех онтологий, специфицирующих эту предметную область]
:= [совокупность различных фактов о структуре некоторой области деятельности некоторых субъектов, а также различного вида знаний, специфицирующих эту область деятельности]
:= [факты и знания о некоторой области деятельности]
:= [sc-модель предметной области и всевозможных онтологий, специфицирующих эту предметную область (и, в первую очередь, её ключевых понятий) в разных ракурсах]
:= [целостный с логико-семантической точки зрения фрагмент базы знаний ostis-системы, акцентирующий внимание на конкретном классе объектов исследования и на конкретном аспекте их рассмотрения]
⇒ примечание*:
[Декомпозиция базы знаний на предметные области вместе с соответствующими им обязательными онтологиями носит в известной мере условный характер. При этом надо помнить, что для исследования и формализации межпредметных (междисциплинарных) свойств и закономерностей необходимо строить иерархию предметных областей т.е. переходить от предметных областей к их объединениям.]
⇒ примечание*:
[предметные области и онтологии являются основным видом разделов баз знаний, обладающих высокой степенью их независимости друг от друга и четкими правилами их согласования, что обеспечивает их семантическую (понятную) совместимость в рамках всей базы знаний.]
:= [основной вид семантических кластеров базы знаний ostis-системы]
⇒ примечание*:
[При формализации научных знаний в самых различных областях в рамках базы знаний ostis-системы (в том числе в рамках объединённой виртуальной базы знаний Экосистемы OSTIS) каждой(!) научной дисциплине будет соответствовать своя предметная область и соответствующая ей объединённая онтология. Следует при этом подчеркнуть, что наличие в Технологии OSTIS развитых метаязыковых средств создает хорошие условия для активизации междисциплинарных исследований, для конвергенции различных научных дисциплин.]
⇒ примечание*:
[Все ролевые отношения, предметные области с ключевыми объектами исследования, а также с используемыми, вводимыми и исследуемыми понятиями, логичным образом применяются и в онтологиях, и в объединениях предметных областей с соответствующими им онтологиями, поскольку указанные ключевые объекты исследования и ключевые понятия входят и являются ключевыми не только для предметных областей, но и для соответствующих им онтологий]
}
/* Завершили Сегмент “Понятие объединённой онтологии предметной области, а также понятие предметной области и онтологии” */

```

```

/* Сегмент *****/
Отношения, заданные на множестве онтологий
⊃=
{
отношение, заданное на множестве онтологий
Э онтология*
:= [быть онтологией заданного информационного ресурса*]
⊃ онтология предметной области*
Э часть знания*
Э декомпозиция знания*
⊃ декомпозиция предметной области и онтологии*
:= [декомпозиция предметной области и онтологии на предметную область и её частные онтологий*]
Э область выполнения действий*

онтология*
С спецификация*
:= [быть спецификацией заданной сущности*]
:= [быть спецификацией знания (информационного ресурса)*]
:= [метазнание*]

онтология предметной области*
С онтология*
:= [быть онтологией заданной предметной области*]
:= [Бинарное ориентированное отношение, здорово связывает некоторую предметную область с онтологией, специфицирующей эту предметную область*]
⊃ объединённая онтология предметной области*
⊃ структурная спецификация предметной области*
⊃ теоретико множественная онтология предметной области*
⊃ логическая онтология предметной области*
⊃ терминологическая онтология предметной области*
⇒ примечание*:
[Отношение “онтология предметной области*” и его подмножества не являются обязательными, поскольку конкретный вид онтологии и вид сущности, специфицируемой данной онтологией, однозначно задаются классами, которым принадлежат эта онтология и эта сущность. Таким образом, указанные отношения вводятся исключительно для дидактических целей и не рекомендуются к использованию, поскольку увеличивают число случаев логической эквивалентности sc-текстов.]

часть знания*
С часть*
:= [знание, являющееся фрагментом (частью) другого заданного знания*]
⇒ примечание*:
[С помощью данного отношения, также различных подмножеств отношения “быть онтологией*” можно переходить от первичных знаний к метазнаниям, от метазнаний к метаметазнаниям и так далее.]

область выполнения действий*
:= [быть формальной моделью области выполнения заданной системы действий (заданного сложного действия или заданной деятельности)*]
⇒ примечание*:
[предметные области и онтологии (предметные области, интегрированные с их объединёнными онтологиями) являются часто используемыми и весьма удобными формальными моделями, обеспечивающими качественную семантическую систематизацию областей выполнения различного вида действий, осуществляемых кибернетическими системами как в собственной памяти (путём непосредственной обработки соответствующих предметных областей и их онтологий), так и в своей внешней среде.
Многим предметным областям и онтологиям будут ставиться в соответствие следующие локальные предметные области действий и задач, выполняемых на основе этих предметных областей и онтологий:
□ история эксплуатации соответствующей предметной области и онтологии при выполнении действий и задач в рамках собственной памяти;

```

- история эволюции соответствующей предметной области и онтологии;
- история выполнения действий и задач во внешней среде на основе соответствующей предметной области и онтологии.

]

} /* Завершили Сегмент “*Отношения, заданные на множестве онтологий*” */

} /* Завершили Раздел “*Предметная область и онтология онтологий*” */

/* Раздел *****/

Предметная область и онтология логических формул, высказываний и формальных теорий

$\supset=$
{

Предметная область логических формул, высказываний и формальных теорий

- ∈ предметная область
- ∃ максимальный класс объектов исследования':
- ∃ формальная теория
- ∃ класс объектов исследования':
 - высказывание
 - атомарное высказывание
 - неатомарное высказывание
 - фактографическое высказывание
 - логическая формула
 - атомарная логическая формула
 - неатомарная логическая формула
 - утверждение
 - определение
 - общезначимая логическая формула
 - противоречивая логическая формула
 - нейтральная логическая формула
 - выполнимая логическая формула
 - невыполнимая логическая формула
 - тавтология
 - квантор
 - формула существования
 - число значений переменной
 - кратность существования
 - единственное существование
 - логическая формула и единственность
 - открытая логическая формула
 - замкнутая логическая формула
- ∃ исследуемое отношение':
 - предметная область'
 - аксиома'
 - теорема'
 - подформула*
 - логическая связка*
 - импликация*
 - если'
 - то'
 - эквиваленция*
 - конъюнкция*
 - дизъюнкция*
 - строгая дизъюнкция*
 - отрицание*
 - всеобщность*
 - неатомарное существование*
 - связываемые переменные'

формальная теория

⇒ пояснение*:

[**формальная теория** — это множество высказываний, которые считаются истинными в рамках данной **формальной теории**. Высказывания могут быть как фактографическими, так и нет. Некоторые высказывания считаются аксиомами (не доказываются в рамках данной формальной теории), а другие доказываются на основе других высказываний в рамках этой же **формальной теории**.

Каждая формальная теория интерпретируется (т.е. ее высказывания являются истинными) на какой-либо **предметной области**, которая является максимальным из **фактографических высказываний** (их

объединением*), входящих в состав этой **формальной теории**. Каждой **формальной теории** соответствует одна **предметная область**, которая входит в нее под атрибутом **предметная область'**.

Каждая **формальная теория** может рассматриваться как **конъюнктивное высказывание**, априори истинное (с чьей-то точки зрения) при интерпретации на соответствующей **предметной области**.]

предметная область'

∈ *ролевое отношение*

⇒ *пояснение**:

[**предметная область'** – это *ролевое отношение*, связывающее *формальную теорию* с *предметной областью*, на которой данная *формальная теория* интерпретируется (в рамках которой истинны *высказывания*, входящие в состав этой *формальной теории*). Другими словами, эта **предметная область** является максимальным фактографическим высказыванием этой *формальной теории*.]

аксиома'

∈ *ролевое отношение*

⇒ *пояснение**:

[**аксиома'** – это *ролевое отношение*, связывающее *формальную теорию* с *высказыванием*, истинность которого не доказывается в её рамках.]

теорема'

∈ *ролевое отношение*

⇒ *пояснение**:

[**теорема'** – это *ролевое отношение*, связывающее *формальную теорию* с *высказыванием*, истинность которого доказывается в её рамках.]

высказывание

⊂ *структура*

⇒ *разбиение**:

- *атомарное высказывание*
- *неатомарное высказывание*

⇒ *разбиение**:

- *фактографическое высказывание*
- *логическая формула*

⇒ *пояснение**:

[Под **высказыванием** понимается некоторая *структура* (в которую входят *sc-константы* из некоторой предметной области и/или *sc-переменные*) или *логическая связка*, которая может трактоваться как истинная или ложная в рамках какой-либо *предметной области*. Истинность **высказывания** задается путем указания принадлежности знака этого высказывания *формальной теории*, соответствующей данной *предметной области*. Ложность высказывания задается путем указания принадлежности знака *отрицания** этого высказывания данной *формальной теории*. Явно указанная непринадлежность **высказывания** *формальной теории* может говорить как о его ложности в рамках данной теории (если это указано рассмотренным выше образом), так и о том, что данное **высказывание** вообще не рассматривается в данной *формальной теории* (например, использует понятия, не принадлежащие данной *предметной области*).

Одно и то же **высказывание** может быть истинно в рамках одной *формальной теории* и ложно в рамках другой.

Каждое высказывание может либо содержать *sc-элементы*, которые не являются знаками других **высказываний** (быть атомарным), либо содержать только знаки других **высказываний** (быть неатомарным).]

высказывание формальной теории'

∈ *неосновное понятие*

⇒ *разбиение**:

- *истинное высказывание'*
:= [высказывание, истинное в рамках данной формальной теории']
- *ложное высказывание'*
:= [высказывание, ложное в рамках данной формальной теории']
- *нечеткое высказывание'*
:= [гипотетическое высказывание']

- := [высказывание, возможно истинное или ложное в рамках данной формальной теории¹]
- := [высказывание, истинное или ложное в рамках данной формальной теории с некоторой вероятностью¹]
- *бессмысленное высказывание¹*
- := [высказывание, бессмысленное в рамках данной формальной теории¹]
- ⇒ *пояснение**:
[Высказывание является бессмысленным в рамках заданной формальной теории, если в какое-либо *атомарное высказывание* в его составе (или в само это высказывание, если оно является атомарным) входит какая-либо *sc-константа*, не являющаяся элементом предметной области, описываемой указанной *формальной теорией*.]

атомарное высказывание

- ⇒ *разбиение**:
{
 - *атомарное фактографическое высказывание*
 - *атомарная логическая формула*
- ⇒ *пояснение**:
[**атомарное высказывание** – это высказывание, которое содержит хотя бы один *sc-элемент*, не являющийся знаком другого высказывания.]

неатомарное высказывание

- ⇒ *пояснение**:
[**неатомарное высказывание** – это высказывание, в состав которого входят только знаки других высказываний.
- Следует отметить, что мы не можем говорить об истинности либо ложности **неатомарного высказывания** в рамках какой-либо *формальной теории*, в случае, когда невозможно установить истинность либо ложность любого из его элементов в рамках этой же *формальной теории*.]

фактографическое высказывание

- ⊃ *атомарное фактографическое высказывание*
- ⇒ *пояснение**:
[Под **фактографическим высказыванием** понимается:
 - атомарное высказывание*, в состав которого не входит ни одна *sc-переменная*;
 - неатомарное высказывание*, все элементы которого также являются **фактографическими высказываниями**.

логическая формула

- ⇒ *пояснение**:
[Под **логической формулой** понимается:
 - атомарное высказывание*, в состав которого входит хотя бы одна *sc-переменная*;
 - неатомарное высказывание*, хотя бы один элемент которого является **логической формулой**.
- ⇒ *разбиение**:
{
 - *атомарная логическая формула*
 - *неатомарная логическая формула*
- ⇒ *разбиение**:
{
 - *открытая логическая формула*
 - *замкнутая логическая формула*

атомарная логическая формула

- := [обобщенная структура]
- := [атомарная формула существования]
- ⇒ *пояснение**:

[Под **атомарной логической формулой** понимается *атомарное высказывание*, которое является *логической формулой*.

По умолчанию **атомарная логическая формула** трактуется как *высказывание* о существовании, то есть наличия в памяти значений, соответствующих всем *sc-переменным*, входящим в состав данной формулы и не попадающих под действие какого-либо другого *квантора* (указанного явно или по умолчанию). Таким образом, на все *sc-переменные*, входящие в состав **атомарной логической формулы** и не попадающие под действие какого-либо другого *квантора*, неявно накладывается квантор *существования**.]

неатомарная логическая формула

⇒ разбиение*:

- общезначимая логическая формула
- противоречивая логическая формула
- нейтральная логическая формула

⇒ разбиение*:

- выполнимая логическая формула
- невыполнимая логическая формула

⊃ тавтология

⇒ пояснение*:

[Под **неатомарной логической формулой** понимается *неатомарное высказывание*, которое является *логической формулой*.

Для того, чтобы рассмотреть типологию **неатомарных логических формул**, будем говорить, что исследуется истинность самой **неатомарной логической формулы** и всех ее *подформул** в рамках одной и той же *формальной теории*, при этом не важно, какой именно. Также считается, что в рассматриваемой *формальной теории* каждая *подформула** рассматриваемой **неатомарной логической формулы** в рамках этой *формальной теории* может однозначно трактоваться как либо истинная, либо ложная. В противном случае мы не можем говорить об истинности либо ложности исходной **неатомарной логической формулы** в рамках этой *формальной теории*.]

подформула*

:= [частная формула*]

∈ бинарное отношение

∈ ориентированное отношение

∈ транзитивное отношение

⇒ определение*:

[Будем называть **подформулой*** *неатомарной логической формулы* любую *логическую формулу*, являющуюся элементом исходной формулы, а также любую **подформулу*** *неатомарной логической формулы*, являющейся элементом исходной *неатомарной логической формулы*.

Будем также считать исходную *логическую формулу* **подформулой*** самой себя.]

утверждение

:= [текст логической формулы]

⇒ определение*:

[**утверждение** – это *семантическая окрестность* некоторой *логической формулы*, в которую входит полный текст этой *логической формулы*, а также факт принадлежности этой *логической формулы* некоторой *формальной теории*. Знак указанной *логической формулы* является *главным ключевым sc-элементом*¹ в рамках этого **утверждения**. Знаки понятий соответствующей *предметной области*, которые входят в состав какой-либо *подформулы** указанной *логической формулы*, будут *ключевыми sc-элементами*¹ в рамках этого **утверждения**.

Полный текст некоторой *логической формулы* включает в себя:

- знак самой этой *логической формулы*;
- знаки всех ее *подформул**;
- элементы всех *логических формул*, знаки которых попали в данную структуру;
- все пары принадлежности, связывающие *логические формулы*, знаки которых попали в данную структуру, с их компонентами.

Таким образом, факт принадлежности (истинности) логической формулы нескольким *формальным теориям* будет порождать новое утверждение для каждой такой *формальной теории*. Текст **утверждения** входит

в состав *логической онтологии*, соответствующей *предметной области*, на которой интерпретируется *главный ключевой sc-элемент'* данного утверждения.]

⇒ *правило идентификации экземпляров**:

[**утверждения** в рамках *Русского языка* именуются по следующим правилам:

- в начале идентификатора пишется сокращение **Утв.**;
- далее в круглых скобках через точку с запятой перечисляются основные идентификаторы *ключевых sc-элементов'* данного **утверждения**. Порядок определяется в каждом конкретном случае в зависимости от того, свойства каких из этих *понятий* описывает данное **утверждение** в большей или меньшей степени.

Например:

Утв. (треугольник; сторона)*

Могут быть исключения для **утверждений**, названия которых закрепились исторически, например, *Теорема Пифагора, Аксиома о прямой и точке.*]

определение

: = [текст определения]

С *утверждение*

⇒ *определение**:

[**определение** – это *утверждение*, *главным ключевым sc-элементом'* которого является связка *эквиваленции**, однозначно определяющая некоторое понятие, соответствующее понятие на основе других. Таким образом, каждое определение имеет ровно один *ключевой sc-элемент'* (не считая *главного ключевого sc-элемента'*).

Следует отметить, что для одного и того же понятия в рамках одной *формальной теории* может существовать несколько *утверждений об эквиваленции**, однозначно задающих некоторое понятие на основе других, однако только одно такое *утверждение* в рамках этой *формальной теории* может быть отмечено как **определение**.]

⇒ *правило идентификации экземпляров**:

[**определения** в рамках *Русского языка* именуются по следующим правилам:

- в начале идентификатора пишется сокращение **Опр.**;
- далее в круглых скобках через точку с запятой записывается основной идентификатор *ключевого sc-элемента'* данного **определения**.

Например:

Опр. (связный граф)]

общезначащая логическая формула

С *выполнимая логическая формула*

С *тавтология*

⇒ *определение**:

[**общезначащая логическая формула** – это *неатомарная логическая формула*, для которой не существует *формальной теории*, в рамках которой она была бы ложной с учетом истинности и ложности всех ее *подформул** в рамках этой же *формальной теории*.]

противоречивая логическая формула

С *невыполнимая логическая формула*

С *тавтология*

⇒ *определение**:

[**противоречивая логическая формула** – это *неатомарная логическая формула*, для которой не существует *формальной теории*, в рамках которой она была бы истинной с учетом истинности и ложности всех ее *подформул** в рамках этой же *формальной теории*.]

нейтральная логическая формула

С *выполнимая логическая формула*

⇒ *определение**:

[**нейтральная логическая формула** – это *неатомарная логическая формула*, для которой существует хотя бы одна *формальная теория*, в рамках которой эта формула ложна, и хотя бы одна *формальная теория*, в рамках которой эта формула истинна.]

выполнимая логическая формула

⇒ *определение**:

[**выполнимая логическая формула** – это *неатомарная логическая формула*, для которой существует хотя бы одна *формальная теория*, в рамках которой эта формула истинна.]

невыполнимая логическая формула

⇒ *определение**:

[**невыполнимая логическая формула** – это *неатомарная логическая формула*, для которой существует хотя бы одна *формальная теория*, в рамках которой эта формула ложна.]

тавтология

:= [тождественно истинная формула]

⇒ *определение**:

[**тавтология** – это *неатомарная логическая формула*, которая является либо *общезначимой логической формулой*, либо *противоречивой логической формулой*. Другими словами, это такая *неатомарная логическая формула*, которая является либо только истинной, либо только ложной в рамках всех *формальных теорий*, в которых можно установить ее истинность или ложность.]

логическая связка*

:= [неатомарная логическая формула]

:= [логический оператор*]

:= [пропозициональная связка*]

∈ *класс связок разной мощности*

← *семейство подмножеств**:

неатомарное высказывание

⇒ *пояснение**:

[**логическая связка*** – это отношение (класс связок), связками которого являются *неатомарные высказывания*. Другими словами, это *отношение*, областью определения которого является множество *высказываний*, при этом само это отношение и некоторые его подмножества могут быть *классами связок разной мощности*.]

импликация*

:= [логическое следование*]

⊂ *логическая связка**

∈ *бинарное отношение*

∈ *ориентированное отношение*

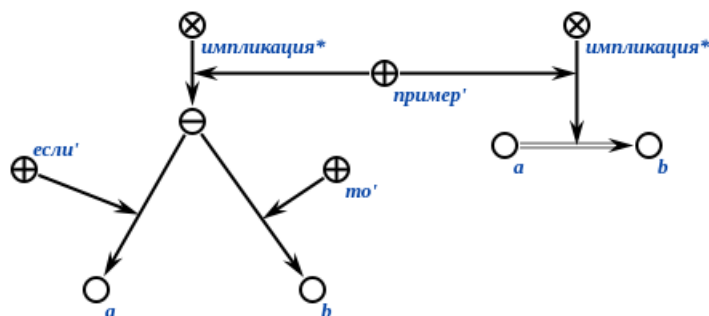
⇒ *определение**:

[**импликация*** – это множество импликативных *неатомарных высказываний*, каждое из которых состоит из посылки (первый компонент *высказывания*) и следствия (второй компонент *высказывания*). Каждое импликативное *высказывание* ложно в рамках некоторой *формальной теории* в том случае, когда его посылка истинна, а заключение ложно в рамках этой же *формальной теории*. В других случаях такое *высказывание* истинно.

По умолчанию на все переменные, входящие в обе части высказывания об *импликации** (или хотя бы одну из *подформул** каждой части) неявно накладывается квантор *всеобщности**, при условии, что эти переменные не связаны другим *квантором*, указанным явно.]

⇒ *описание примера**:

[



]

если'

C 1'

E ролевое отношение

⇒ определение*:

[**если'** – это ролевое отношение, используемое в связках импликации*, для указания посылки.]**то'**

C 2'

E ролевое отношение

⇒ определение*:

[**то'** – это ролевое отношение, используемое в связках импликации*, для указания следствия.]**эквиваленция***

:= [эквивалентность*]

C логическая связка*

E бинарное отношение

E неориентированное отношение

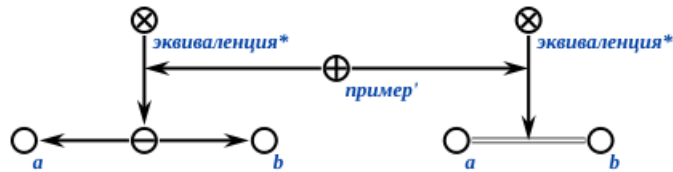
⇒ определение*:

[**эквиваленция*** – это множество неатомарных высказываний об эквивалентности, каждое из которых истинно в рамках некоторой формальной теории только в тех случаях, когда оба его компонента одновременно либо истинны в рамках этой же формальной теории, либо ложны.

По умолчанию на все переменные, входящие в обе части высказывания об эквиваленции* (или хотя бы одну из подформул* каждой части) неявно накладывается квантор всеобщности*, при условии, что эти переменные не связаны другим квантором, указанным явно.]

⇒ описание примера*:

[



]

конъюнкция*

:= [логическое и*]

:= [логическое умножение*]

C логическая связка*

E неориентированное отношение

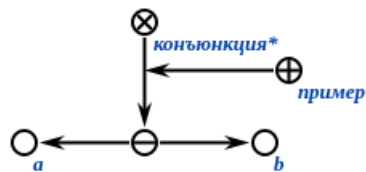
E класс связок разной мощности

⇒ определение*:

[**конъюнкция*** – это множество конъюнктивных высказываний, каждое из которых истинно в рамках некоторой формальной теории только в том случае, когда все его компоненты истинны в рамках этой же формальной теории.]

⇒ описание примера*:

[



]

дизъюнкция*

:= [логическое или*]

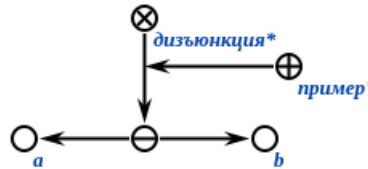
:= [логическое сложение*]

:= [включающее или*]
 C логическая связка*
 E неориентированное отношение
 E класс связок разной мощности
 => определение*:

[**дизъюнкция*** – это множество дизъюнктивных *высказываний*, каждое из которых истинно в рамках некоторой *формальной теории* только в том случае, когда хотя бы один его компонент является истинным в рамках этой же *формальной теории*.]

=> описание примера*:

[



]

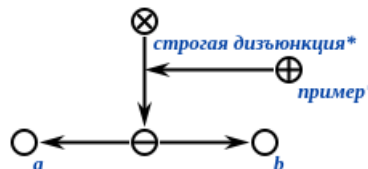
строгая дизъюнкция*

:= [сложение по модулю 2*]
 := [исключающее или*]
 := [альтернатива*]
 C логическая связка*
 E неориентированное отношение
 E класс связок разной мощности
 => определение*:

[**строгая дизъюнкция*** – это множество строго дизъюнктивных *высказываний*, каждое из которых истинно в рамках некоторой *формальной теории* только в том случае, когда ровно один его компонент является истинным в рамках этой же *формальной теории*.]

=> описание примера*:

[



]

отрицание*

C логическая связка*
 C синглетон
 => определение*:

[**отрицание*** – это множество *высказываний* об отрицании, каждое из которых истинно в рамках некоторой *формальной теории* только в том случае, когда его единственный элемент является ложным в рамках этой же *формальной теории*.]

квантор

C логическая связка*
 => определение*:

[**квантор** – это *отношение*, каждая связка которой задает истинность множества *логических формул*, входящих в ее состав, при выполнении дополнительных условий, связанных с некоторыми из переменных, входящих в состав этих *логических формул*. Будем говорить, что указанные переменные связаны **квантором** или попадают под область действия данного **квантора** (имея в виду конкретную связку конкретного **квантора**). Таким образом, в состав каждой связки каждого **квантора** входит *атомарная формула*, являющаяся *тривиальной структурой*, в которой перечислены переменные, связанные данным **квантором**.]

всеобщность*

:= [квантор всеобщности*]

:= [квантор общности*]

∈ квантор

∈ ориентированное отношение

∈ класс связок разной мощности

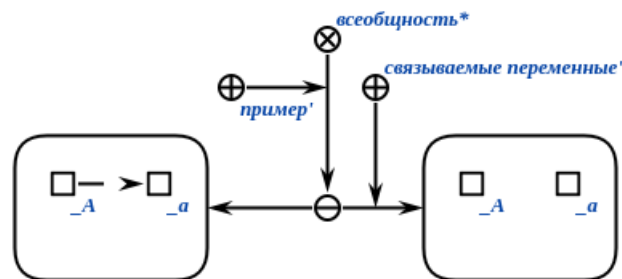
⇒ определение*:

[**всеобщность** – это квантор, для каждой связки которого, истинной в рамках некоторой *формальной теории*, выполняется следующее утверждение: все формулы, входящие в состав этой связки истинны в рамках этой же *формальной теории* при всех (любых) возможных значениях всех элементов множества *связываемых переменных'* входящего в эту связку.]

Каждая связка квантора **всеобщность*** может быть представлена как *конъюнкция** (потенциально бесконечная) исходных *логических формул*, входящих в состав этой связки, в каждой из которых все *связанные переменные'* заменены на их возможные значения.]

⇒ описание примера*:

[



]

формула существования

:= [существование*]

⇒ разбиение*:

- атомарная логическая формула
- неатомарное существование*

неатомарное существование*

:= [квантор неатомарного существования*]

∈ квантор

∈ ориентированное отношение

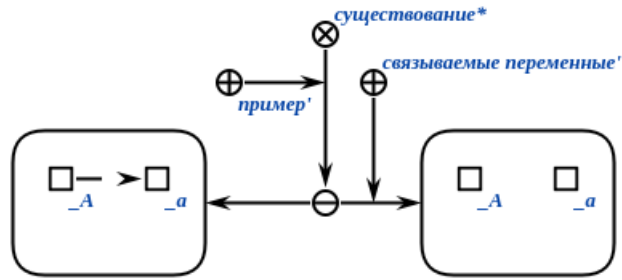
∈ класс связок разной мощности

⇒ определение*:

[**неатомарное существование*** – это квантор, для каждой связки которого, истинной в рамках некоторой *формальной теории*, выполняется следующее утверждение: существуют значения всех элементов множества *связываемых переменных'* входящего в эту связку, такие, что все формулы, входящие в состав этой связки истинны в рамках этой же *формальной теории*.]

Каждая связка квантора **неатомарное существование*** может быть представлена как *дизъюнкция** (потенциально бесконечная) исходных *логических формул*, входящих в состав этой связки, в каждой из которых все *связанные переменные'* заменены на их возможные значения.]

⇒ описание примера*:



число значений переменной

∈ параметр
⇒ пояснение*:

[Каждый элемент параметра **число значений переменной** представляет собой класс ориентированных пар, первым компонентом которых является знак *логической формулы*, вторым – *sc-переменная*, имеющая в рамках данной *логической формулы* ограниченное известное число значений, при которых данная формула является истинной в рамках соответствующей *формальной теории*.

Отметим, что в случае *атомарной логической формулы* каждая такая связка связывает знак формулы и знак принадлежащей ей *sc-переменной*, т.е. является, по сути, частным случаем пары принадлежности. В случае *неатомарной логической формулы* указанная *sc-переменная* может принадлежать любой из *подформул** исходной формулы.

Таким образом, *измерением** каждого значения параметра **число значений переменной** является некоторое число, задающее количество значений *sc-переменных* в рамках *логической формулы*.]

кратность существования

∈ параметр
⇒ область определения параметра*:
формула существования
∃ единственное существование
⇒ пояснение*:

[Каждый элемент параметра **кратность существования** представляет собой класс логических *формул существования*, для которых при интерпретации на соответствующей *предметной области* существует ограниченное общее для всех таких формул число комбинаций значений переменных, при которых указанные формулы являются истинными в рамках соответствующей *формальной теории*.

Таким образом, *измерением** каждого значения **кратности существования** является некоторое число, задающее количество таких комбинаций.]

единственное существование

:= [однократное существование]
:= [формула существования и единственности]

логическая формула и единственность

⊂ логическая формула
⊂ единственное существование
⇒ пояснение*:

[Каждый элемент множества **логическая формула и единственность** представляет собой *логическую формулу (атомарную или неатомарную)*, для которой дополнительно уточняется, что при ее интерпретации на некоторой *предметной области* существует только один набор значений переменных, входящих в эту формулу (или ее *подформулы**), при котором указанная логическая формула истинна в рамках *формальной теории*, в которую входит данная *предметная область*.]

связываемые переменные'

∈ ролевое отношение
⇒ определение*:

[**связываемые переменные'** – это *ролевое отношение*, которое связывает связку конкретного *квантора* с множеством переменных, которые связаны этим квантором.]

открытая логическая формула⇒ *определение**:

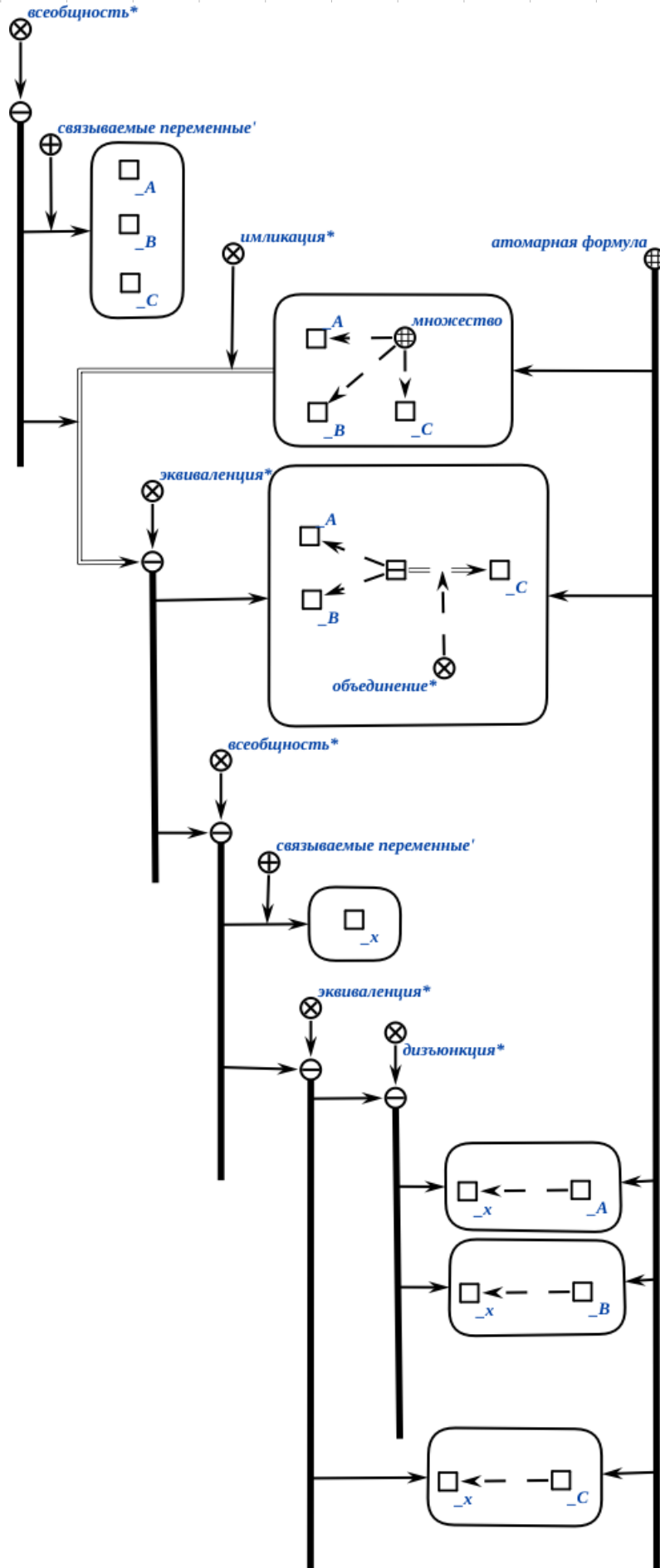
[**открытая логическая формула** – это *логическая формула*, в рамках которой (и всех ее *подформул**) существует хотя бы одна переменная, не связанная никаким *квантором*.]

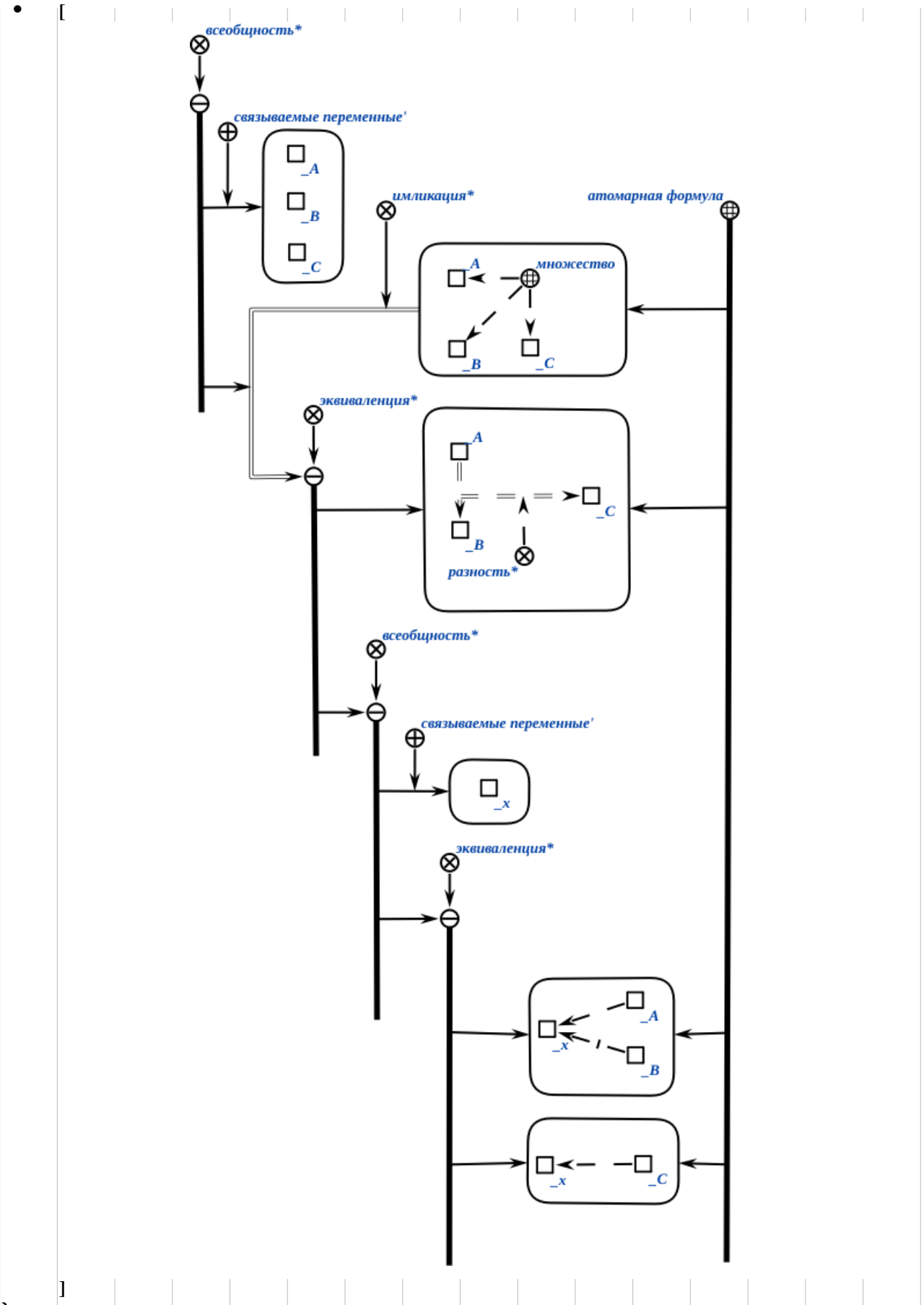
замкнутая логическая формула⇒ *определение**:

[**замкнутая логическая формула** – это *логическая формула*, в рамках которой (и всех ее *подформул**) не существует переменных, не связанных каким-либо *квантором*.]

Примеры логических утверждений

= {•





}
 /* Завершили Раздел “Предметная область и онтология логических формул, высказываний и формальных теорий” */

/* Раздел	*****/
	Глобальная предметная область действий и задачи соответствующая ей онтология методов и технологий
⊃= {	
∈	раздел
∈	предметная область и онтология
⇐	конкатенация сегментов*:
	<ul style="list-style-type: none"> • Уточнение понятия воздействия и понятия действия. Типология воздействий и действий • Уточнение понятия задачи. Типология задач • Уточнение семейства параметров и отношений, заданных на множестве воздействий, действий и задач • Предметная область и онтология субъектно-объектных спецификаций воздействий • Уточнение понятий плана сложного действия, класса задач, метода • Уточнение понятия навыка, понятия класса методов и понятия модели решения задач • Уточнение понятия деятельности, понятия вида деятельности и понятия технологии
⊃	исследуемый класс первичных объектов исследования':
	<ul style="list-style-type: none"> • <i>воздействие</i> := [процесс воздействия одних сущностей на другие] ⊂ процесс
	<ul style="list-style-type: none"> • <i>действие</i> := [процесс, "осознанно" и целенаправленно выполняемый (управляемый) некоторой кибернетической системой] ⊂ воздействие ⊂ процесс
	<ul style="list-style-type: none"> • <i>неосознанное воздействие</i>
	<ul style="list-style-type: none"> • <i>действие, выполняемое в памяти субъекта действия</i>
	<ul style="list-style-type: none"> • <i>действие, выполняемое во внешней среде субъекта действия</i>
	<ul style="list-style-type: none"> • <i>рецепторное действие</i> := [действие, выполняемое рецептором субъекта действия]
	<ul style="list-style-type: none"> • <i>эффекторное действие</i> := [действие, выполняемое эффектором субъекта действия]
	<ul style="list-style-type: none"> • <i>элементарное действие</i> := [действие, выполнение которого не требует его декомпозиции на взаимосвязанные поддействия]
	<ul style="list-style-type: none"> • <i>сложное действие</i>
	<ul style="list-style-type: none"> • <i>легко выполнимое сложное действие</i> := [сложное действие, которое известно, как выполнять]
	<ul style="list-style-type: none"> • <i>интеллектуальное действие</i> := [сложное действие, для которого априори не известно, как его выполнять]
	<ul style="list-style-type: none"> • <i>индивидуальное действие</i> := [действие, выполняемое индивидуальной кибернетической системой]
	<ul style="list-style-type: none"> • <i>коллективное действие</i> := [действие, выполняемое коллективом кибернетических систем (многоагентной системой)]
	<ul style="list-style-type: none"> • <i>планируемое действие</i>
	<ul style="list-style-type: none"> • <i>иницированное действие</i>
	<ul style="list-style-type: none"> • <i>выполняемое действие</i> := [активное действие]
	<ul style="list-style-type: none"> • <i>прерванное действие</i> := [выполняемое действие, находящееся в состоянии прерывания]
	<ul style="list-style-type: none"> • <i>выполненное действие</i>
	<ul style="list-style-type: none"> • <i>отмененное действие</i>

- действие с очень высоким приоритетом
- действие с высоким приоритетом
- действие со средним приоритетом
- действие с низким приоритетом
- действие с очень низким приоритетом

Э исследуемый класс классов первичных объектов исследования¹:

- осмысленность воздействия[^]
- длительность воздействия[^]
- место выполнения действия[^]
- функциональная сложность действия[^]
- многоагентность действия[^]
- := [коллективность субъекта действия]
- текущее состояние действия[^]
- приоритет действия[^]
- := [важность действия[^]]
- срочность действия[^]
- класс действий
- класс функционально эквивалентных действий[^]
- класс логически эквивалентных действий[^]
- класс семантических эквивалентных задач[^]
- класс логически эквивалентных задач[^]
- класс задач, для которого существует общий метод их решения[^]
- класс аналогичных семантически элементарных процессов воздействия[^]
- := [класс однотипных семантически элементарных воздействий[^]]

Э исследуемый класс классов¹:

- отношение, заданное на множестве* (действие)
- := [отношение, заданное на множестве действий]
- отношение, заданное на множестве* (задача)
- параметр, заданный на множестве* (действие)
- параметр, заданный на множестве* (задача)

/* Здесь указаны классы классов, которые не являются классами классов первичных объектов исследования */

Э исследуемое отношение, заданное на множестве первичных объектов исследования¹:

- воздействующая сущность¹
- воздействуемая сущность¹
- посредник¹
- медиатор¹
- субъект¹
- := [быть субъектом заданного действия]
- спецификация воздействия*
- ⊃ спецификация действия*
- спецификация действия*
- ⊃ задача*
- ⇒ разбиение*:
 - { • декларативная формулировка задачи*
 - процедурная формулировка задачи*
 - }
- ⊃ план сложного действия*
- ⊃ декларативная спецификация выполнения сложного действия*
- ⊃ протокол*
- ⊃ результативная часть протокола*
- декларативная формулировка задачи*
- процедурная формулировка задачи*
- план сложного действия*
- декларативная спецификация выполнения сложного действия*
- протокол*
- результативная часть протокола*

Э исследуемое отношение¹:

- спецификация класса действий*
- спецификация метода*
- спецификация класса методов*
- спецификация деятельности*
- спецификация вида деятельности*

/* Здесь указаны исследуемые отношения, которые заданы не на множестве первичных объектов исследования */

Э *исследуемый класс структур, специфицирующих первичные объекты исследования'*:

- *задача*
:= [формулировка задачи]
:= [спецификация действия]
:= [структура (sc-конструкция), содержащая в идеале достаточную информацию для выполнения соответствующего (специфицируемого) действия]
- *декларативная формулировка задачи*
:= [семантическая спецификация действия]
- *процедурная формулировка задачи*
:= [функциональная спецификация действия]
- *план сложного действия*
:= [план выполнения сложного действия]
- *процедурный план сложного действия*
- *непроцедурный план сложного действия*
:= [декларативный план сложного действия]
:= [иерархическая система подзадач заданной сложной задачи]

Э *исследуемый класс структур'*:

- *метод*
:= [спецификация класса сложных действий]
- *денотационная семантика метода*
- *операционная семантика метода*
- *навык*
- *модель решения задач*
- *технология*

/* Здесь указаны классы структур, не являющихся спецификациями первичных объектов исследования */

Э *вводимое, но не исследуемое понятие'*:

- *действие, выполняемое в памяти ostis-системы*
- *действие, выполняемое ostis-системой в своей внешней среде*
- *рецептурное действие ostis-системы*
- *эффекторное действие ostis-системы*
- *sc-агент*
:= [внутренний субъект ostis-системы]
:= [субъект, реализующий действия, выполняемые в памяти ostis-системы]

/* Здесь указаны понятия, исследуемые в предметной области (и соответствующей онтологии), которая является частной по отношению к заданной и которая наследует все свойства заданной предметной области и онтологии */

Э *используемое понятие, исследуемое в другой предметной области и онтологии'*:

- *кибернетическая система*
:= [сущность, обладающая способностью быть субъектом различного вида действий]
- *компьютерная система*
:= [искусственная кибернетическая система]
C *кибернетическая система*
- *интеллектуальная компьютерная система*
C *компьютерная система*
D *ostis-система*
- *человек*
C *кибернетическая система*
- *ostis-система*
- *спецификация**

:= [быть спецификацией (описанием, семантической окрестностью заданной сущности*)]

:= [семантическая окрестность*]

следует отличать*

- ⇒ { • { • *действие*
• *класс действий*
}
• { • *метод*
• *класс методов*
}
• { • *деятельность*
• *вид деятельности*
}

}

С *семейство подклассов**

⇒ *примечание*:*

[Все сущности, принадлежащие рассмотренным *понятиям*, требуют достаточно детальной *спецификации*. При этом не следует путать сами сущности и их *спецификации*. Так, например, не следует путать *действие* и *задачу*, которая специфицирует (уточняет) это *действие*. Особое место среди указанных понятий занимает понятие *метода*, т.к. каждый конкретный *метод*, с одной стороны, является *спецификацией* соответствующего *класса действий*, а, с другой стороны, сам нуждается в *спецификации*, которая уточняет либо *декларативную семантику* этого *метода* (т.е. обобщенную декларативную формулировку класса задач, решаемых с помощью этого *метода*), либо *операционную семантику* этого *метода*, (т.е. множество *методов*, обеспечивающих *интерпретацию* данного специфицируемого *метода*) и тем самым "преобразует" специфицируемый *метод* в *навык*.]

следует отличать*

- ⇒ { • *первый домен*(спецификация*)*
:= [специфицируемая сущность]
:= [сущность, использование которой требует вполне определенной ее спецификации]
⊃ *действие*
⊃ *класс действий*
⊃ *метод*
⊃ *класс методов*
⊃ *деятельность*
⊃ *вид деятельности*
• *второй домен*(спецификация*)*
:= [спецификация]
⊃ *задача*
⊃ *декларативная формулировка задачи*
:= [семантическая формулировка задачи]
⊃ *процедурная формулировка задачи*
:= [функциональная формулировка задачи]
⊃ *план действия*
:= [план]
:= [план выполнения действия]
⊃ *декларативная спецификация выполнения действий*
:= [иерархическая система подзадач]
⊃ *протокол*
⊃ *результативная часть протокола*
⊃ *обобщенная декларативная формулировка класса задач*
⊃ *метод*
⊃ *декларативная семантика метода*
⊃ *операционная семантика метода*
⊃ *модель решения задач*

}

⇒ *примечание*:*

[При этом следует отличать:

- спецификацию конкретного *действия* (*задачу*, *план*, *декларативную спецификацию выполнения действия*, *протокол*, *результативную часть протокола*);

- спецификацию конкретной деятельности (*контекст**, *множество используемых методов**);
- спецификацию класса действий (*обобщенную декларативную формулировку класса задач, метод*);
- спецификацию вида деятельности (*технологии*);
- спецификацию метода (*декларативную семантику метода, операционную семантику метода*);
- спецификацию класса методов (*модель решения задач*).

]

следует отличать*

\ni {

- {
 - действие
 - класс действий, метод
- {
 - деятельность
 - вид деятельности, технология

/* Сегмент ***/**

Уточнение понятия воздействия и понятия действия. Типология воздействий и действий

$\supset =$

{
 \in сегмент базы знаний

воздействие

$:=$ [процесс воздействия одной сущности (или некоторого множества сущностей) на другую сущность (или на некоторое множество других сущностей)]

$:=$ [процесс, в котором могут быть явно выделены хотя бы одна воздействующая сущность (*субъект воздействия'*) и хотя бы одна сущность, на которую осуществляется воздействие (*субъект воздействия'*)]

воздействие

\subset процесс

$:=$ [динамическая структура]

\Rightarrow примечание*:

[Поскольку воздействия являются частным видом процессов, воздействиями наследуются все свойства процессов. Смотрите Раздел *Предметная область и онтология структур*). В частности, используются все параметры, заданные на множестве процессов, например, *длительность**, *момент начала процесса**, *момент завершения процесса^*]

процесс

\Rightarrow покрытие*:

длительность^

= {

- краткосрочный процесс
- среднесрочный процесс
- долгосрочный процесс
- перманентный процесс

\Rightarrow примечание*:

[Длительность различных процессов можно уточнять до любой необходимой точности, используя различные единицы измерения длительности (с точностью до секунд, минут, часов, дней, месяцев, лет, столетий и т.д.). Кроме того, можно ссылаться на процессы, длительность (время жизни) которых соизмерима с рассматриваемым процессом.]

действие

\subset воздействие

\subset процесс

$:=$ [воздействие, в котором воздействующая сущность' осуществляет воздействие "осознанно", целенаправленно]

- := [целенаправленный процесс, выполняемый одним или несколькими субъектами (кибернетическими системами) с возможным применением некоторых инструментов]
- := [акция]
- := [акт]
- := [операция]
- := [процесс воздействия некоторой (возможно, коллективной) сущности (субъекта воздействия) на некоторую одну или несколько сущностей (объектов воздействия - исходных объектов (аргументов) или целевых (создаваемых или модифицируемых) объектов)]
- := [осознанное воздействие]
- := [активное воздействие]
- := [целенаправленный ("осознанный") процесс, выполняемый (управляемый, реализуемый) неким субъектом]
- := [акция реализации некоторого замысла]
- := [преднамеренная акция]
- := [работа]
- := [процесс выполнения некоторой задачи]
- := [дело]
- := [целостный фрагмент некоторой деятельности]
- := [целенаправленный процесс, управляемый некоторым субъектом (возможно, коллективным)]
- := [целенаправленный процесс, выполняемый некоторым субъектом (исполнителем) над некоторыми объектами]
- ⇒ *примечание**:

[Каждое *действие*, выполняемое тем или иным *субъектом*, трактуется как процесс решения некоторой задачи, т.е. процесс достижения заданной цели в заданных условиях, и, следовательно, выполняется целенаправленно. При *этом* явное указание *действия* и его связи с конкретной задачей может не всегда присутствовать в памяти. Некоторые *задачи* могут решаться определенными агентами перманентно, например, оптимизация *базы знаний*, поиск некорректностей и т.д. и для подобных задач не всегда есть необходимость явно вводить *структуру*, являющуюся формулировкой *задачи*.

Каждое *действие* может обозначать сколь угодно малое преобразование, осуществляемое во внешней среде либо в памяти некоторой *кибернетической системы*, однако в памяти явно вводятся знаки только тех *действий*, для которых есть необходимость явно хранить в памяти их спецификацию в течение некоторого времени. При выполнении *действия* можно выделить этапы:

- построение *плана действия*, декомпозиция (детализация) действия в виде системы его *поддействий*;
- выполнение построенного *плана действия*

]

⇒ *правило идентификации экземпляров**:

[Экземпляры класса *действий* в рамках *Русского языка* именуется по следующим правилам:

- в начале идентификатора пишется слово "*Действие*" и ставится точка;
- далее с прописной буквы идет либо содержащее глагол совершенного вида в инфинитиве описание сути того, что требуется получить в результате выполнения действия, либо вопросительное предложение, являющееся спецификацией запрашиваемой (ответной) информации.

Например:

Действие. Сформировать полную семантическую окрестность понятия треугольник

Действие. Верифицировать Раздел. Предметная область sc-элементов]

*следует отличать**

- Э { • *действие*
 - ⇒ *примечание**:
[Каждому *действию* становится в соответствие *кибернетическая система*, являющаяся *субъектом* этого *действия*. Указанный *субъект* может быть либо *индивидуальной*, либо *коллективной кибернетической системой*.]
- *воздействие*
 - ⊃ *действие*
 - ⊂ *процесс*
 - ⇒ *примечание**:
[Сущностью, осущетвляющей воздействие на какой-либо объект, может быть не только *кибернетическая система*, но также, например, и пассивный инструмент, управляемый некоторой *кибернетической системой*.]

- *деятельность*

}

параметр, заданный на множестве* (воздействие)

:= [признак классификации воздействий]

Э *осознанность воздействия*[^]

:= [Наличие субъекта (индивида), который запланировал и реализовал воздействие, т.е. обеспечил управление выполнением этого воздействия]

- *неосознанное действие*
- *действие*

:= [осознанное действие]

}

параметр, заданный на множестве* (действие)

:= [признак классификации действий]

Э *место выполнения действия*[^]Э *функциональная сложность действия*[^]Э *многоагентность выполнения действия*[^]Э *текущее состояние действия*[^]Э *приоритет действия*[^]**действие**⇒ *разбиение**:*место выполнения действия*[^]

- *действие, выполняемое в памяти субъекта действия*
- *действие, выполняемое во внешней среде субъекта действия*
- *рецепторное действие*
- *эффекторное действие*

}

⇒ *разбиение**:*функциональная сложность действия*[^]

- *элементарное действие*

:= [действие, выполняемое индивидуальной кибернетической системой]

⇒ *пояснение**:

[Элементарное действие выполняется одним индивидуальным субъектом и является либо элементарным действием, выполняемым в памяти этого субъекта (элементарным действием его "процессора"), либо элементарным действием одного из его эффекторов.]

- *сложное действие*

:= [неэлементарное действие]

:= [действие выполнение которого требует декомпозиции этого действия на множество его поддействий, т.е. частных действий более низкого уровня]⇒ *примечание**:

[Декомпозиция сложного действия на поддействия может иметь весьма сложный иерархический вид с большим числом уровней иерархии, т.е. поддействиями *сложного действия* могут также *сложные действия*. Уровень сложности действия можно определять (1) общим числом его поддействий и (2) числом уровней иерархии этих поддействий.]

⇒ *примечание**:

[Другим примером может служить запись одной и той же процедурной программы на языке программирования более высокого уровня и на языке программирования более низкого уровня. В данном случае элементарность действий строго определяется на уровне языка.]

⇒ *примечание**:

[Темпоральные соотношения между *поддействиями* сложного действия могут быть самые различные, но в простейшем случае *сложное действие* представляет собой строгую последовательность *действий* более низкого уровня иерархии.]

:= [система более простых действий (*поддействий*), которые могут выполняться как последовательно, так и параллельно]⇒ *примечание**:

[В состав *сложного действия* могут входить не только *собственно поддействия* этого *сложного действия*, но также и специальные *поддействия*, осуществляющие управление процессом выполнения *сложного действия*, и, в частности, *поддействия*, осуществляющие инициирование поддействий, передачу управления *поддействиям*.]

:= [неатомарное действие]

:= [неэлементарное *действие*]

:= [*действие*, выполнение которого сводится в общем случае к последовательно-параллельному выполнению некоторого множества *поддействий*]

:= [*действие*, декомпозируемое на множество более простых *действий* (*поддействий*), обеспечивающих выполнение исходного (заданного) *действия*]

⇒ *покрытие**:

{ • *легко выполнимое сложное действие*

:= [сложное действие, для выполнения которого известен соответствующий *метод* и соответствующие этому методу исходные данные, а также (для действий, выполняемых во внешней среде) имеются в наличии все необходимые исходные объекты (расходные материалы и комплектация), а также средства (инструменты)]

• *интеллектуальное действие*

:= [трудно выполнимое сложное действие]

:= [сложное действие, для выполнения которого в текущий момент либо неизвестен соответствующий *метод*, либо возможные *методы* известны, но отсутствуют условия их применения.]

⊃ *действие, у которого цель известна, на задача не совсем точна*

⊃ *действие, направленное на выявление противоречий в базе знаний*

⇒ *примечание**:

[Это действие декомпозируется на несколько самостоятельных поддействий, каждое из которых выявляет (локализует) противоречия (ошибки) конкретного формализуемого вида, для которого в базе знаний существует точное определение]

⊃ *действие, для которого априори не известен метод, обеспечивающий его выполнение*

⇒ *примечание**:

[Соответствующий метод либо не найден, либо его вообще нет в памяти.]

}

}

⇒ *разбиение**:

многоагентность выполнения действия[^]

= { • *индивидуальное действие*

:= [действие, выполняемое одним субъектом (агентом)]

:= [действие, выполняемое индивидуальной кибернетической системой]

⊃ *индивидуальное действие, выполняемое человеком*

⊃ *индивидуальное действие, выполняемое компьютерной системой*

• *коллективное действие*

:= [действие, выполняемое коллективом субъектов (многоагентной системой)]

:= [действие, выполняемое коллективом кибернетических систем (коллективом субъектов)]

⊃ *действие, выполняемое коллективом людей*

⊃ *действие, выполняемое коллективом индивидуальных компьютерных систем*

⊃ *действие, выполняемое коллективом людей и индивидуальных компьютерных систем*

⊃ *действие, выполняемое Экосистемой OSTIS*

⊃ *действие, выполняемое одним человеком во взаимодействии с одной индивидуальной компьютерной системой*

}

⇒ *разбиение**:

текущее состояние действия[^]

= { • *планируемое действие*

:= [запланированное, но не инициированное действие]

- := [будущее действие]
 := [действие, которое планируется выполнить в будущем]
 ⇒ *пояснение**:
 [Во множество *запланированных, но не иницированных действий* входят *действия*, начать выполнение которых запланировано на какой-либо момент в будущем.]
- *иницированное действие*
 - := [иницированное, но не выполняемое действие]
 - := [действие, подлежащее выполнению]
 - := [действие, включенное в план]
 - := [действие, ожидающее начала своего выполнения]
 - ⇒ *пояснение**:
 [Во множество *иницированных действий* входят *действия*, выполнение которых иницировано в результате какого-либо события. В общем случае, *действия* могут быть иницированы по следующим причинам:
 - явно путем проведения соответствующей *sc-дуги принадлежности* каким-либо *субъектом* (*заказчиком**). В случае действия в *sc-памяти*, оно может быть иницировано как внутренним *sc-агентом* системы, так и пользователем при помощи соответствующего пользовательского интерфейса. При этом, спецификация действия может быть сформирована одним *sc-агентом*, а собственно добавление во множество *иницированных действий* может быть осуществлено позже другим *sc-агентом*;
 - в результате того, что одно или несколько *действий*, предшествовавших данному в рамках некоторой декомпозиции, стали *прошлыми сущностями* (процедурный подход);
 - в результате того, что в *памяти* системы появилась конструкция, соответствующая некоторому условию иницирования *sc-агента*, который должен выполнить данное *действие* (декларативный подход).
 Следует отметить, что декларативный и процедурный подходы можно рассматривать как две крайности, использование только одной из которых не является удобным и целесообразным. При этом, например, принципы иницирования по процедурному подходу могут быть полностью сведены к набору декларативных условий иницирования, но как было сказано, это не всегда удобно и наиболее рациональным будет комбинировать оба подхода в зависимости от ситуации. По сути, попадание некоторого *действия* во множество *иницированных действий* говорит о том, что, по мнению некоторого *субъекта* (заказчика, инициатора), оно готово к выполнению и должно быть выполнено, то есть спецификация данного *действия* по мнению данного *субъекта* сформирована в степени, достаточной для решения поставленной задачи и существует некоторый другой *субъект* (исполнитель), который может приступить к выполнению *действия*. Однако стоит отметить, что с точки зрения *исполнителя* такая спецификация *действия* в общем случае может оказаться недостаточной или некорректной.]
 - *выполняемое действие*
 - := [активное действие]
 - := [непосредственно выполняемое действие]
 - := [действие, выполняемое в текущий момент]
 - := [настоящее действие]
 - ∈ *неосновное понятие*
 - С *настоящая сущность*
 - ⇒ *пояснение**:
 [Во множество *выполняемых действий* входят *действия*, к выполнению которых приступил какой-либо из соответствующих *субъектов*. Попадание *действия* в данное множество говорит о следующем:
 - рассматриваемое *действие* уже попало во множество *иницированных действий*.
 - существует как минимум один *субъект*, условие иницирования которого соответствует спецификации данного *действия*.
 После того, как собственно процесс выполнения завершился, *действие* должно быть удалено из множества *выполняемых действий* и добавлено во множество *выполненных действий* или какое-либо из его подмножеств. Понятие *выполняемое действие*

- является неосновным, и вместо того, чтобы относить конкретные действия к данному классу, их относят к классу *настоящих сущностей*.]
- *прерванное действие*
 - := [действие, ожидающее продолжения своего выполнения]
 - := [отложенное действие]
 - := [приостановленное действие]
 - ⇒ *пояснение**:
[Во множество *прерванных действий* входят *действия*, которые уже были инициированы, однако их выполнение невозможно по каким-либо причинам, например в случае, когда у исполнителя в данный момент есть более приоритетные задачи.]
 - *выполненное действие*
 - := [завершенное действие]
 - := [прошлое действие]
 - ∈ *неосновное понятие*
 - ⊂ *прошлая сущность*
 - ⇒ *пояснение**:
[Во множество *выполненных действий* попадают *действия*, выполнение которых с точки зрения субъекта, осуществлявшего их выполнение. Таким образом, понятие *выполненного действия* является относительным, поскольку с точки зрения разных субъектов одно и то же действие может считаться выполненным или все еще выполняющимся.
В зависимости от результатов конкретного процесса выполнения, рассматриваемое *действие* может стать элементом одного из подмножеств множества *выполненных действий*. Понятие *выполненное действие* является неосновным, и вместо того, чтобы относить конкретные *действия* к данному классу, их относят к классу *прошлых сущностей*.]
 - ⇒ *разбиение**:
 - { • *успешно выполненное действие*
 - ⇒ *пояснение**:
[Во множество *успешно выполненных действий* попадают *действия*, выполнение которых успешно завершено с точки зрения *субъекта*, осуществлявшего их выполнение, т.е. достигнута поставленная цель, например, получены решение и ответ какой-либо задачи, успешно преобразована какая-либо конструкция и т.д. Очень важно отметить, что в общем случае выделить критерии успешности или безуспешности выполнения действий того или иного класса невозможно, поскольку эти критерии, во-первых, зависят от контекста, во-вторых, могут быть разными с точки зрения разных субъектов. Однозначно критерии успешности выполнения действий могут быть сформулированы для некоторых частных классов действий, например, классов операторов некоторого процедурного языка программирования (например, *scp-операторов*). Таким образом, понятие *успешно выполненное действие* является относительным.
Если действие было выполнено успешно, то, в случае действия по генерации каких-либо знаний, к *действию* при помощи связки отношения *результат** приписывается *sc-конструкция*, описывающая результат выполнения указанного действия. В случае, когда действие направлено на какие-либо изменения базы знаний, *sc-конструкция*, описывающая результат действия, формируется в соответствии с правилами описания истории изменений базы знаний.
В случае, когда успешное выполнение *действия* приводит к изменению какой-либо конструкции в *sc-памяти*, которое необходимо занести в историю изменений базы знаний или использовать для демонстрации протокола решения задачи, то генерируется соответствующая связка отношения *результат**, связывающая задачу и *sc-конструкцию*, описывающую данное изменение.]
 - *безуспешно выполненное действие*
 - ⇒ *пояснение**:
[Во множество *безуспешно выполненных действий* попадают *действия*, выполнение которых не было успешно завершено с точки зрения *субъек-*

та, осуществлявшего их выполнение, по каким-либо причинам. Можно выделить две основные причины, по которым может сложиться указанная ситуация:

- соответствующая *задача* сформулирована некорректно;
- формулировка соответствующей *задачи* корректна и понятна системе, однако решение данной задачи в текущий момент не может быть получено за удовлетворительное с точки зрения заказчика или исполнителя сроки.

Для конкретизации факта некорректности формулировки задачи можно выделить ряд более частных классов *безуспешно выполненных действий*, например:

- действие, спецификация которого противоречит другим знаниям системы (например, не выполняется неравенство треугольника);
- действие, при спецификации которого использованы понятия, неизвестные системе;
- действие, выполнение которого невозможно из-за недостаточности данных (например, найти площадь треугольника по двум сторонам);
- и другие.

Для конкретизации факта безуспешности выполнения некоторого действия в системе могут также использоваться дополнительные подмножества данного множества, при необходимости снабженные естественно-языковыми или формальными комментариями.]

⊃ действие, выполненное с ошибкой

⇒ пояснение*:

[Во множество *действий, выполненных с ошибкой*, попадают действия, выполнение которых не было успешно завершено с точки зрения *субъекта*, осуществлявшего их выполнение, по причине возникновения какой-либо ошибки, например, некорректности спецификации данного *действия* или нарушения её целостности каким-либо *субъектом* (в случае *действия в sc-памяти*).]

- отмененное действие

⇒ покрытие*:

приоритет действия[^]

- = {
- действие с очень высоким приоритетом
 - действие с высоким приоритетом
 - действие со средним приоритетом
 - действие с низким приоритетом
 - действие с очень низким приоритетом
- }

действие, выполняемое в памяти субъекта действия

:= [информационное действие]

:= [действие, выполняемое в памяти]

:= [действие кибернетической системы, направленное на обработку информации, хранимой в её памяти]

⊃ действие, выполняемое кибернетической системой в собственной памяти и направленное на организацию её деятельности во внешней среде

:= [действие, выполняемое кибернетической системой в её памяти и направленное на организацию её деятельности во внешней среде и в конечном счете – на сенсо-моторную координацию деятельности её эффекторов]

⇒ пояснение*:

[Результатом выполнения *действия, выполняемого в памяти субъекта действия* является в общем случае некоторое новое состояние памяти информационной системы (не обязательно *sc-памяти*), достигнутое исключительно путем преобразования информации, хранящейся в памяти системы, то есть либо посредством генерации новых знаний на основе уже имеющихся, либо посредством удаления знаний, по каким-либо причинам ставших ненужными. Следует отметить, что если речь идет об изменении состояния *sc-памяти*,

то любое преобразование информации можно свести к ряду элементарных действий генерации, удаления или изменения инцидентности *sc-элементов* друг относительно друга.]

действие, выполняемое во внешней среде субъекта действия

:= [действие, выполняемое кибернетической системой в её внешней среде и осуществляемое (на самом низком уровне) эффекторами этой кибернетической системы]

:= [поведенческое действие]

⇒ *пояснение**:

[В случае *действия, выполняемого во внешней среде субъекта действия* результатом его выполнения будет новое состояние внешней среды. Очень важно отметить, что под внешней средой в данном случае понимаются также и компоненты системы, внешние с точки зрения памяти, то есть не являющиеся хранимыми в ней информационными конструкциями. К таким компонентам можно отнести, например, различные манипуляторы и прочие средства воздействия системы на внешний мир, то есть к поведенческим задачам можно отнести изменение состояния механической конечности робота или непосредственно вывод некоторой информации на экран для восприятия пользователем.]

поддействие*

:= [быть поддействием заданного *сложного действия**]

⇒ *разбиение**:

{ • *собственное поддействие**

• *управляющее поддействие**

⇒ *пояснение**:

[*поддействие**, которое осуществляет инициирование выполнения очередного *поддействия** (или сразу некоторого множества *поддействий**) при соблюдении некоторого априори известного условия. В общем случае таким условием является возникновение некоторой ситуации или события. В частности, это может быть:

- ситуация или событие в обрабатываемой *базе знаний* (это называют условной передачей управления);
- факт успешного завершения некоторого *поддействия* (это безусловная передача управления);
- факт успешного завершения по крайней мере одного из заданного множества *поддействий*;
- факт успешного завершения всех *поддействий* из заданного множества *поддействий*.

]

⇒ *примечание**:

[На самом деле при выполнении *сложных действий* многообразие условий инициирования (передачи управления) поддействиям может быть значительно шире]

}

} /* Завершили Сегмент “Уточнение понятия воздействия и понятия действия. Типология воздействий и действий” */

/* Сегмент *****/

Уточнение понятия задачи. Типология задач

⊃=

{

∈ *сегмент базы знаний*

задача

:= [описание желаемого состояния или события в рамках внешней среды кибернетической системы либо в рамках её базы знаний]

:= [формулировка задачи]

:= [задание на выполнение некоторого действия]

:= [постановка задачи]

:= [описание задачной ситуации]

:= [спецификация некоторого действия, обладающая достаточной полнотой для выполнения этого действия]

:= [цель плюс дополнительные условия (ограничения) накладываемые на результат или процесс получения этого результата]

:= [описание того, что требуется сделать]

⇒ *пояснение**:

[**Задача**, т.е. формальное описание условия некоторой задачи есть, по сути, формальная *спецификация* некоторого *действия*, направленного на решение данной *задачи*, достаточная для выполнения данного *действия* каким-либо *субъектом*. В зависимости от конкретного *класса задач*, описываться может как внутреннее состояние самой интеллектуальной системы, так и требуемое состояние *внешней среды*. *sc-элемент*, обозначающий *действие* входит в *задачу* под атрибутом *ключевой знак*'.

Каждая *задача* представляет собой спецификацию *действия*, которое либо уже выполнено, либо выполняется в текущий момент (в настоящее время), либо планируется (должно) быть выполненным, либо может быть выполнено (но не обязательно).

Классификация *задач* может осуществляться по дидактическому признаку в рамках каждой предметной области, например, задачи на треугольники, задачи на системы уравнений и т.п.

Каждая *задача* может включать:

- факт принадлежности *действия* какому-либо частному классу *действий* (например, *действие*. *сформировать полную семантическую окрестность указываемой сущности*), в том числе состояние *действия* с точки зрения жизненного цикла (иницированное, выполняемое и т.д.);
- описание *цели** (*результата**) *действия*, если она точно известна;
- указание *заказчика** *действия*;
- указание *исполнителя** *действия* (в том числе, коллективного);
- указание *аргумента(ов)* *действия*';
- указание инструмента или посредника *действия*;
- описание *декомпозиции действия**;
- указание *последовательности действий** в рамках *декомпозиции действия**, т.е. построение плана решения задачи. Другими словами, построение плана решения представляет собой декомпозицию соответствующего *действия* на систему взаимосвязанных между собой поддействий;
- указание области *действия*;
- указание условия инициирования *действия*;
- момент начала и завершения *действия*, в том числе планируемый и фактический, предполагаемая и/или фактическая длительность выполнения;

Некоторые *задачи* могут быть дополнительно уточнены контекстом – дополнительной информацией о сущностях, рассматриваемых в формулировке *задачи*, т.е. описанием того, что дано, что известно об указанных сущностях.

Кроме этого, *задача* может включать любую дополнительную информацию о действии, например:

- перечень ресурсов и средств, которые предполагается использовать при решении задачи, например список доступных исполнителей, временные сроки, объем имеющихся финансов и т.д.;
- ограничение области, в которой выполняется *действие*, например, необходимо заменить одну *sc-конструкцию* на другую по некоторому правилу, но только в пределах некоторого *раздела базы знаний*;
- ограничение знаний, которые можно использовать для решения той или иной задачи, например, необходимо решить задачу по алгебре используя только те утверждения, которые входят в курс школьной программы до седьмого класса включительно, и не используя утверждения, изучаемые в старших классах;
- и прочее

С одной стороны, решаемые системой *задачи*, можно классифицировать на *информационные задачи* и *поведенческие задачи*.

С точки зрения формулировки поставленной задачи можно выделить *декларативные формулировки задачи* и *процедурные формулировки задачи*. Следует отметить, что данные классы задач не противопоставляются, и могут существовать формулировки задач, использующие оба подхода.]

⇒ *правило идентификации экземпляров**:

[Экземпляры класса *задач* в рамках *Русского языка* именуется по следующим правилам:

- в начале идентификатора пишется слово “*Задача*” и ставится точка;
- далее с прописной буквы идет либо содержащее глагол совершенного вида в инфинитиве описание сути того, что требуется получить в результате выполнения действия, либо вопросительное предложение, являющееся спецификацией запрашиваемой (ответной) информации.

Например:

Задача. Сформировать полную семантическую окрестность понятия треугольник
Задача. Верифицировать Раздел. Предметная область sc-элементов]

С семантическая окрестность

⊃ процедурная формулировка задачи

⊃ декларативная формулировка задачи

⊃ вопрос

⊃ команда

:= [спецификация действия, которое выполнилось, выполняется или может быть выполнено соответствующей кибернетической системой]

⇒ *примечание**:

[Каждой задаче и, соответственно, каждому специфицируемому действию соответствует определенная кибернетическая система, являющаяся субъектом, выполняющим это действие.]

С знание

⇒ *примечание**:

[Каждая задача – это знание, описывающее то какое действие возможно потребуется выполнить.]

⊃ инициированная задача

:= [формулировка задачи, которая подлежит выполнению]

:= [спецификация (описание) соответствующего действия]

⊃ декларативная формулировка задачи

:= [задача, в формулировке которой явно указывается (описывается) целевая ситуация, т.е. то, что является результатом выполнения (решения) данной задачи]

⊃ процедурная формулировка задачи

:= [задача, в формулировке которой явно указывается характеристика действия, специфицируемого этой задачей, а именно, например, указывается:

- субъект или субъекты, выполняющие это действие,
- объекты, над которыми действие выполняется, – аргументы действия,
- инструменты, с помощью которых выполняется действие,
- момент и, возможно, дополнительные условия начала и завершения выполнения действия

]

⊃ декларативно-процедурная формулировка задачи

:= [задача, в формулировке которой присутствуют как декларативные (целевые), так и процедурные аспекты]

⇒ *примечание**:

[От качества (корректности и полноты) формулировки задачи, т.е. спецификации соответствующего действия, во многом зависит качество (эффективность) выполнения этого действия, т.е. качество процесса решения указанной задачи.]

⊃ проблема

:= [проблемная задача]

:= [сложная, трудно решаемая задача]

⊃ изобретательская задача

процедурная формулировка задачи

:= [спецификация действия, которое планируется быть выполненным]

⇒ *пояснение**:

[В случае **процедурной формулировки задачи**, в формулировке задачи явно указываются аргументы соответствующего задаче действия, и в частности, вводится семантическая типология действий. При этом явно не уточняется, что должно быть результатом выполнения данного действия. Заметим, что, при необходимости, **процедурная формулировка задачи** может быть сведена к **декларативной формулировке задачи** путем трансляции на основе некоторого правила, например определения класса действия через более общий класс.]

декларативная формулировка задачи

:= [описание ситуации (состояния), которое должно быть достигнуто в результате выполнения планируемого действия]

⇒ *пояснение**:

[В случае *декларативной формулировки задачи*, при описании условия задачи специфицируется цель *действия*, т.е. результат, который должен быть получен при успешном выполнении *действия*.]

⇐ *второй домен**:

*декларативная формулировка задачи**

:= [описание исходной (начальной) ситуации, являющейся условием выполнения соответствующего действия и целевой (конечной) ситуации, являющейся результатом выполнения этого действия]

:= [семантическая спецификация действия]

⇒ *примечание**:

[Формулировка *задачи* может не содержать указания контекста (области решения) *задачи* (в этом случае областью решения *задачи* считается либо вся *база знаний*, либо ее согласованная часть), а также может не содержать либо описания исходной ситуации, либо описания целевой ситуации. Так, например, описания целевой ситуации для явно специфицированного противоречия, обнаруженного в *базе знаний* не требуется.]

:= [формулировка (описание) *задачной ситуации* с явным или неявным описанием контекста (условий) выполнения специфицируемого действия, а также результата выполнения этого действия]

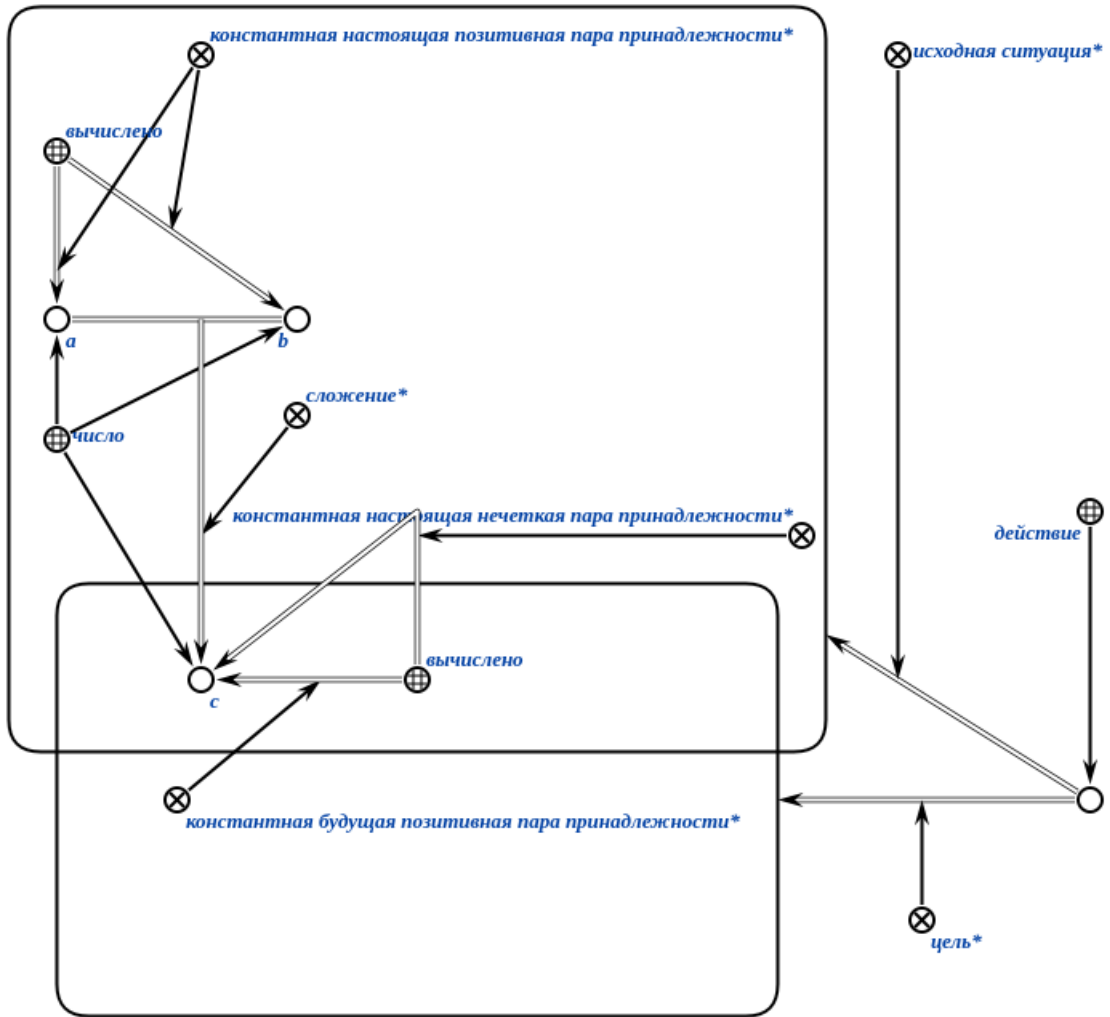
:= [явное или неявное описание

- того, что дано – исходные данные, условия выполнения специфицируемого действия,
- того, что требуется – формулировка цели, результата выполнения указанного действия

]

⇒ *пример'*:

[



]

⇒ *пояснение**:

[Выполнение данного действия сведется к следующим событиям:

- для числа *c* будет сгенерирован уникальный идентификатор, являющийся его представлением в соответствующей системе счисления

- будет сгенерирована константная настоящая позитивная пара принадлежности, соединяющая узел "вычислено" с узлом "с"
- удалится константная будущая позитивная пара принадлежности, а также константная настоящая нечеткая пара принадлежности, выходящие из узла "вычислено".

Таким образом, после выполнения действия все будущие сущности, входящие в целевую ситуацию, становятся настоящими сущностями, а некоторые настоящие сущности, входящие в исходную ситуацию, становятся прошлыми.]

задача

- ⊃ *задача, решаемая в памяти кибернетической системы*
- ⊃ *задача, решаемая в памяти индивидуальной кибернетической системы*
- ⊃ *задача, решаемая в общей памяти многоагентной системы*
- := [информационная задача]
- := [задача, направленная либо на генерацию или поиск информации, удовлетворяющей заданным требованиям, либо на некоторое преобразование заданной информации]
- ⊃ *математическая задача*
- ⊃ *элементарная информационная задача*
- ⊃ *простая информационная задача*
- ⊃ *проблемная информационная задача*
- := [интеллектуальная информационная задача]
- ⊃ *проблема Гильберта*

вопрос

- := [запрос]
- ⊃ *задача, решаемая в памяти кибернетической системы*
- := [непроцедурная формулировка задачи на поиск (в текущем состоянии базы знаний) или на генерацию знания, удовлетворяющего заданным требованиям]
- ⊃ *вопрос – что это такое*
- ⊃ *вопрос – почему*
- ⊃ *вопрос – зачем*
- ⊃ *вопрос – как*
- := [каким способом]
- := [запрос метода (способа) решения заданного (указываемого) вида задач или класса задач либо, плана решения конкретной указываемой задачи]
- := [задача, направленная на удовлетворение информационной потребности некоторого субъекта-заказчика]

команда

- := [инициированная задача]
- := [спецификация инициированного действия]
- ⇒ *пояснение**:
[Идентификатор экземпляров конкретного класса **команд** в рамках *Русского языка* пишется с прописной буквы и представляет собой либо содержащее глагол совершенного вида в инфинитиве описание сути того, что требуется получить в результате выполнения действия, соответствующего данной **команде**, либо вопросительное предложение, являющееся спецификацией запрашиваемой (ответной) информации.
Например:
Сформировать полную семантическую окрестность понятия треугольник
Верифицировать Раздел. Предметная область sc-элементов]

задача

- ⇒ *примечание**:
[Сужение бинарного ориентированного отношения *спецификация** (быть спецификацией*), связывающее действия с задачами, которые решаются в результате выполнения этих действий, не является взаимно однозначным. Каждому действию может соответствовать несколько формулировок задач, которые с разной степенью детализации или с разных аспектов специфицируют указанное действие. Кроме того, интерпретация разных формулировок семантически одной и той же задачи в общем случае приводит к разным действиям, решающим эту задачу. Подчеркнем, что разные, но семантически эквивалентные формулировки задач считаются формулировками формально разных задач.]

отношение, заданное на множестве(задача)*

⇒ *задача**

∈ *неосновное понятие*

:= [формулировка задачи*]

:= [спецификация действия, уточняющая то, что должно быть сделано*]

⇒ *разбиение**:

- *декларативная формулировка задачи**
- *процедурная формулировка задачи**

⇒ *второй домен**:

задача

∈ *основное понятие*

⊃ *задача обработки базы знаний*

⊃ *задача обработки файлов*

⊃ *задача, решаемая кибернетической системой во внешней среде*

⊃ *задача, решаемая кибернетической системой в собственной физической оболочке*

⇒ *декларативная формулировка задачи**

∈ *неосновное понятие*

:= [описание исходной ситуации и целевой ситуации специфицируемого действия*]

⇒ *пояснение**:

[декларативная формулировка задачи включает в себя:

- связку отношения *цель**, связывающую специфицируемое действие с описанием целевой ситуации;
- само описание целевой ситуации;
- связку отношения *исходная ситуация**, связывающую специфицируемое действие с описанием исходной ситуации;
- непосредственно описание исходной ситуации;
- указание контекста (области решения) задачи.

При этом указание и описание исходной ситуации может отсутствовать.]

⇒ *процедурная формулировка задачи**

∈ *неосновное понятие*

:= *пояснение**:

[указание

- класса действий*, которому принадлежит специфицируемое *действие*, а также
- субъекта* или субъектов, выполняющих это действие (с дополнительным указанием роли каждого участвующего субъекта);
- объекта* или объектов, над которыми осуществляется действие (с указанием "роли" каждого такого объекта);
- используемых материалов;
- используемых инструментов (инструментальных средств);
- дополнительных темпоральных характеристик специфицируемого действия (сроки, длительность);
- приоритета (важности) специфицируемого действия;
- если описываемое действие не является элементарным в текущем контексте, то декомпозиции описываемого действия на поддействия и этих поддействий на еще более простые поддействия, вплоть до элементарных действий.

]

задача

⇒ *примечание**:

[Для некоторых *отношений*, заданных на множестве *действий*, вводятся аналогичные *отношения*, заданные на множестве *задач*, соответствующих этим *действиям*. От *отношений*, связывающих некоторые сущности, несложно перейти к *отношениям*, связывающим *спецификации** этих сущностей.]

} /* Завершили Сегмент “Уточнение понятия задачи. Типология задач” */

```

/* Сегмент *****/
Уточнение семейства параметров и отношений, заданных на множестве
воздействий, действий и задач. Типология спецификаций воздействий,
действий и задач
⊃=
{
∈ сегмент базы знаний

отношение, заданное на множестве*(действие)
:= [отношение, заданное на множестве действий*]
∋ поддействие*
:= [частное действие*]
⊂ темпоральная часть*
⇒ пояснение*:
[Связки отношения поддействие* связывают действие и некоторое более простое частное действие,
выполнение которого необходимо для выполнения исходного более общего действия.]
∈ бинарное отношение
∈ отношение таксономии
↔ обратное отношение*:
наддействие*
:= [быть действием, являющимся частью заданного действия более высокого уровня иерархии*]
:= [быть действием, направленным на решение задачи, которая является подзадачей по отношению к
задаче, решение которой осуществляется заданным действием*]
⊃ непосредственное поддействие*
:= [быть таким поддействием заданного действия, для которого не существует наддействия,
которое было бы также поддействием заданного действия*]

спецификация*
⊃ сужение отношения по первому домену*(спецификация*; действие)*
:= часто используемый sc-идентификатор*:
[спецификация действия*]
⇒ разбиение*:
{• задача*
⇒ разбиение*:
{• декларативная формулировка задачи*
⇒ второй домен*:
декларативная формулировка задачи
• процедурная формулировка задачи*
⇒ второй домен*:
процедурная формулировка задачи
}
• исходная ситуация действия*
• цель*
• план*
• декларативная спецификация выполнения действия*
• контекст действия*
:= [информационный ресурс необходимый для выполнения заданного действия*]
• множество используемых методов*
:= [множество методов, используемых для выполнения заданного действия*]
:= [операционный (функциональный) ресурс, необходимый для выполнения заданного
действия*]
• протокол*
• результативная часть протокола*
}
⇒ примечание*:
[Таким образом, каждому действию может быть поставлен в соответствие целый ряд видов спецификации
этого действия, которые описывают различные аспекты специфицируемого действия – и то, что является
причиной (условием) инициирования этого действия, и то, что является результатом ("сухим остатком") его
выполнения, и то, и то, с помощью таких ресурсов оно может быть выполнено, и то, как управлять этими
ресурсами в процессе выполнения действия, и то, как на самом деле это действие было выполнено.]

```

*следует отличать**

- ∃ { • { • действие
• задача*
} • { • действие
• исходная ситуация*
} • { • действие
• декларативная формулировка задачи*
} • { • действие
• процедурная формулировка задачи*
} • { • действие
• план*
} • { • действие
• декларативная спецификация выполнения действия*
} • { • действие
• протокол*
} • { • действие
• результативная часть протокола*
} } }

сужение отношения по первому домену(спецификация*; действие)**

:= [спецификация действия*]

⇒ примечание*:

[спецификацию действия (базовое описание действия) условно можно разбить на следующие части:

- описание состояния действия в текущий момент времени – действие может принадлежать:
 - либо классу *прогнозируемых сущностей* (в случае действий – это планируемые действия, которые могут быть, но не обязательно выполняться в будущем);
 - либо классу *настоящих сущностей*, т.е. сущностей, существующих в настоящий (текущий) момент времени;
 - либо классу *прошлых сущностей*, завершивших свое существование (в случае действий – это действия, выполнение которых уже завершено);
- формулировки *задачи*, которая должна быть решена в результате выполнения специфицируемого действия. Такая формулировка представляет собой логико-семантическое описание *задачной продукции*, включающей в себя:
 - описание *исходной ситуации* и/или события (исходных условий того, что должно быть дано, исходных данных, исходного контекста). Для *действий во внешней среде* (действий/задач, выполняемых во внешней среде) в описании *исходной ситуации* должно быть включено описание необходимых для решения задачи материальных ресурсов (сырья, комплектации) с указанием их количества;
 - описание *целевой ситуации* и/или события (того, что требуется получить в результате решения данной задачи);
 - указание дополнительных *инструментальных средств*, используемых для выполнения специфицируемого действия (такие средства могут быть использованы только при выполнении *действий во внешней среде*).
- указание субъектов-исполнителей специфицируемого действия:
 - множество тех, кто может выполнить это действие;
 - тот, кто должен (которому поручено выполнить это действие);
- указание метода, на основании (путем интерпретации) которого специфицируемое действие может быть выполнено – таких методов в общем случае может быть несколько;
- спецификация выполненного действия, т.е. действия, отнесенного к классу *прошлых сущностей*:
 - указание отрезка времени выполнения действия (момента начала и момента завершения);

- указание числа прерываний (ожиданий) процесса выполнения действия;
- указание "чистой" длительности процесса выполнения действия;
- указание успешности выполнения процесса (в случае неуспешности – указание "штатных" причин и сбоев).

]

отношение, связывающее действия с их спецификациями

Э **декларативная спецификация выполнения действия***

С спецификация*

⇒ второй домен*:

декларативная спецификация выполнения действия

⇒ пояснение*:

[В состав такой спецификации действия входят:

- контекст действия***, содержащий информацию, достаточную для его выполнения;
- множество используемых методов*** и инструментов, достаточных для выполнения действия.

]

:= [непроцедурное описание выполнения сложного (неэлементарного) действия]

⊃ функциональная спецификация выполнения действия

⊃ логическая спецификация выполнения действия

⇒ примечание*:

[*декларативная спецификация выполнения действия* – это такой выделенный фрагмент *базы знаний* (такой *контекст** выполнения соответствующего конкретного *действия*), которого достаточно для выполнения этого *действия* с помощью заданного множества *методов*, используемых в рамках указанного *контекста**. При этом важна минимизация и самого *контекста** и *множества используемых методов**.]

Э **контекст действия***

:= [задачная ситуация*]

:= [что дано*]

:= [дополнительная информация о тех сущностях, которые входят в описание цели*]

:= [связь между некоторой задачей (формулировкой задачи) и состоянием базы знаний, возможностей и навыков некоторого субъекта, перед которым поставлена указанная задача*]

:= [связь между формулировкой задачи, т.е. описанием того, что требуется, и контекстом этой задачи, т.е. описанием имеющихся ресурсов, описанием того, что дано*]

⇒ пояснение*:

[Связки отношения *контекст действия** связывают *sc-элементы*, обозначающие *действие* и *структуры*, обозначающие контекст выполнения данного *действия*, то есть некоторую дополнительную информации о тех сущностях, которые входят в описание *цели**. Как правило, контекст используется для указания собственно условия некоторой задачи, того, что дано, т.е. тех знаний, которые можно использовать для вывода новых знаний при решении задачи. Таким образом, контекст непосредственно влияет на то, как будет решаться та или иная задача, при этом даже задачи соответствующие одному классу действий, могут решаться по-разному.

Контекст может быть представлен не только в виде атомарного фактографического высказывания, но и в виде высказывания более сложного вида. Это может быть, например:

- определение множества, используемого в описании *цели**;
- утверждение, учет которого может быть полезен в решении задач;

Первым компонентом связей отношения *контекст действия** является знак *действия*, вторым – знак *структуры*, обозначающей контекст.]

Э **множество используемых методов***

Э **протокол***

∈ неосновное понятие

:= [декомпозиция выполненного действия на систему последовательно-параллельно выполненных его поддействий*]

:= [описание того, как действительно было выполнено соответствующее действие и, в частности, описание последовательности соответствующих ситуаций и событий*]

:= [протокол выполнения сложного действия, включающий в себя протоколы выполнения всех поддействий этого действия*]

- := [протокол решения задачи*]
 := [история решения выполненной задачи*]
 ⇒ *пояснение**:
 [Каждый **протокол** представляет собой *семантическую окрестность*, *ключевым sc-элементом* является *действие*, для которого собственно описывается весь процесс его выполнения, то есть все более простые поддействия, в том числе те, выполнение которых, как выяснилось позже, не было целесообразным. Подразумевается, что *sc-элемент*, обозначающий данное действие, входит во множество прошлых сущностей.
- Таким образом, **протокол** представляет собой *sc-текст*, содержащий декомпозицию рассматриваемого *действия* на поддействия с указанием порядка их выполнения, а также необходимой спецификацией каждого такого поддействия.
- В отличие от *плана*, **протокол** всегда формируется по факту выполнения соответствующего *действия*.]
- Э **результативная часть протокола***
 ∈ *неосновное понятие*
 := [часть протокола соответствующего выполненного действия, которая включает в себя только те его поддействия, которые действительно внесли вклад в построение результата ("сухого остатка") этого выполненного действия*]
 ⇒ *примечание**:
 [протокол выполненного действия и результативная часть этого протокола могут сильно отличаться. Примером тому является, например, соотношение между протоколом доказательства некоторой конкретной теоремы и результативной частью этого протокола, которая является подтверждением корректности проведенного доказательства и, соответственно, обоснованием истинности доказанной теоремы.]
 ⇒ *пояснение**:
 [Каждая **результативная часть протокола** представляет собой *семантическую окрестность*, *ключевым sc-элементом* является *действие*, для которого собственно описывается процесс его выполнения, то есть решение соответствующей задачи. Подразумевается, что *sc-элемент*, обозначающий данное *действие*, входит во множество *успешно выполненных действий*.
- Таким образом, **результативная часть протокола** представляет собой *sc-текст*, содержащий декомпозицию рассматриваемого *действия* на поддействия с указанием порядка их выполнения, а также необходимой спецификацией каждого такого поддействия.]
- Э **декомпозиция действия***
 ∈ *отношение декомпозиции*
 ∈ *квазибинарное отношение*
 := [сведение действия ко множеству более простых взаимосвязанных действий*]
 ⇒ *пояснение**:
 [Связки отношения **декомпозиция действия*** связывают *действие*, и множество частных *действий*, на которые декомпозируется данное *действие*. При этом первым компонентом связки является знак указанного множества, вторым компонентом – знак более общего *действия*.
- Таким образом, **декомпозиция действия*** это *квазибинарное отношение*, связывающее действие со множеством действий более низкого уровня, к выполнению которых сводится выполнение исходного декомпозируемого действия.
- Стоит отметить, что каждое *действие* может иметь несколько вариантов декомпозиции в зависимости от конкретного набора элементарных действий, которые способна выполнять та или иная система *субъектов*.
- Принцип, по которому осуществляется такая декомпозиция в различных подходах к решению задач будем называть *стратегией решения задач*.]
- Э **исходная ситуация***
 := [исходная ситуация, соответствующая заданному действию*]
 := [начальная ситуация действия*]
 := [описание того, что дано (что имеется) перед началом выполнения заданного (специфицируемого) действия*]
 ⇒ *примечание**:
 [**исходная ситуация*** действия содержит описание тех исходных условий, которых оказалось достаточно для инициирования данного действия, в то время как **начальная ситуация*** является более общим понятием и в общем случае не предполагает такого описания.]

	С	<i>спецификация*</i>
	⇒	<i>второй домен*:</i> <i>ситуация</i>
Э		<i>конечная ситуация*</i>
	:=	[результатирующая ситуация, соответствующая заданному действию*]
	:=	[конечная ситуация, соответствующая заданному действию*]
	С	<i>спецификация*</i>
	⇒	<i>второй домен*:</i> <i>ситуация</i>
Э		<i>цель*</i>
	:=	[целевая ситуация*]
	С	<i>спецификация*</i>
	⇒	<i>второй домен*:</i> <i>ситуация</i>
	:=	[описание того, что требуется получить (какая ситуация должна быть достигнута) в результате выполнения заданного (специфицируемого) действия*]
	:=	[цель выполнения действия*]
	:=	[интенция, стремление, намерение, замысел, желание, устремление, направленность действия*]
		<i>следует отличать*</i>
Э	{	• <i>конечная ситуация*</i>
		• <i>цель*</i>
	}	
	⇒	<i>примечание*:</i> [Далеко не всегда конечная ситуация, сформированная в результате выполнения некоторого действия, совпадает с ситуацией, которая изначально (еще до выполнения действия или в процессе его выполнения) рассматривалась как целевая, желаемая ситуация.]
Э	{	• <i>действие</i>
		:= [процесс решения задачи]
		• <i>результат действия*</i>
		:= [результат выполнения действия*]
		:= [результат решения задачи*]
		:= [достигнутая цель*]
	}	
Э	{	• <i>процесс доказательства</i>
		:= [процесс доказательства утверждения]
		• <i>результат доказательства*</i>
		:= [текст доказательства*]
		:= [текст, обосновывающий истинность доказываемого утверждения*]
	}	
}		<i>/* Завершили Сегмент “Уточнение семейства параметров и отношений, заданных на множестве воздействий, действий и задач. Типология спецификаций воздействий, действий и задач” */</i>
		<i>/* Сегмент *****/</i>
		<i>Предметная область и онтология субъектно-объектных спецификаций воздействий</i>
	⊃=	
	{	
	⇒	<i>соавтор*:</i>
		• <i>Гордей А.Н.</i>
		• <i>Никифоров С.А.</i>
		• <i>Бобёр Е.С.</i>
		• <i>Святощик М.И.</i>
Є		<i>предметная область и онтология</i>

индивид

:= часто используемый *sc-идентификатор**:
 [субъект]
 ⇒ источник*:
 Гордей.А.Н.ОснованияКС-2005ст

участник воздействия'

:= [участник акции']
 ∈ ролевое отношение
 ⇒ первый домен*:
 индивид
 ⇒ второй домен*:
 воздействие
 ⇒ пояснение*:
 [участник акции' – это ролевое отношение, которое связывает акцию с участвующим в ней индивидом.]
 ⇒ источник*:
 Гордей.А.Н.СемантическаяРСЕОСКРЯ-2021ст
 ⇒ источник*:
 Fillmore.Ch.J.TheCCR-1977ст
 ⇒ источник*:
 Fillmore.Ch.J.FrameS-2005ст
 ⇒ разбиение*:
 {• субъект'
 ⇒ пояснение*:
 [субъект' – инициатор акции.]
 ⇒ разбиение*:
 {• инициатор'
 • вдохновитель'
 • распространитель'
 • вершитель'
 ⇒ пояснение*:
 [вершитель' завершает акцию производством из объекта продукта.]
 }
 • инструмент'
 ⇒ пояснение*:
 [инструмент' – исполнитель акции.]
 ⇒ разбиение*:
 {• активатор'
 ⇒ пояснение*:
 [активатор' непосредственно воздействует на медиатор.]
 • супрессор'
 ⇒ пояснение*:
 [супрессор' подавляет сопротивление медиатора.]
 • усилитель'
 ⇒ пояснение*:
 [усилитель' наращивает воздействие на медиатор.]
 • преобразователь'
 ⇒ пояснение*:
 [преобразователь' преобразует медиатор в инструмент.]
 }
 • медиатор'
 ⇒ пояснение*:
 [медиатор' – посредник акции.]
 ⇒ разбиение*:
 {• ориентир'
 • локус'
 ⇒ пояснение*:
 [локус' частично или полностью окружает объект и тем самым локализует его в пространстве.]
 • транспортёр'

- ⇒ *пояснение**:
[*транспортёр'* перемещает объект.]
 - *адаптер'*
 - ⇒ *пояснение**:
[*адаптер'* приспособливает инструмент к воздействию на объект.]
 - *материал'*
 - ⇒ *пояснение**:
[*материал'* используется в качестве объекта-сырья для производства продукта.]
 - *макет'*
 - ⇒ *пояснение**:
[*макет'* является исходным образцом для производства из объекта продукта.]
 - *фиксатор'*
 - ⇒ *пояснение**:
[*фиксатор'* превращает переменный локус объекта в постоянный.]
 - *ресурс'*
 - ⇒ *пояснение**:
[*ресурс'* питает инструмент.]
 - *стимул'*
 - ⇒ *пояснение**:
[*стимул'* проявляет параметр объекта.]
 - *регулятор'*
 - ⇒ *пояснение**:
[*регулятор'* служит инструкцией в производстве из объекта продукта.]
 - *хронотон'*
 - ⇒ *пояснение**:
[*хронотон'* локализует объект во времени.]
 - *источник'*
 - ⇒ *пояснение**:
[*источник'* обеспечивает инструкциями инструмент.]
 - *индикатор'*
 - ⇒ *пояснение**:
[*индикатор'* отображает параметр воздействия на объект или параметр продукта как результата воздействия на объект.]
- }
- *объект'*
 - ⇒ *пояснение**:
[*объект'* – реципиент акции.]
 - ⇒ *разбиение**:
 - { • *покрытие'*
 - ⇒ *пояснение**:
[*покрытие'* – внешняя изоляция оболочки индивида.]
 - *корпус'*
 - ⇒ *пояснение**:
[*корпус'* – оболочка индивида.]
 - *прослойка'*
 - ⇒ *пояснение**:
[*прослойка'* – внутренняя изоляция оболочки индивида.]
 - *сердцевина'*
 - ⇒ *пояснение**:
[*сердцевина'* – ядро индивида.]
- }
- *продукт'*
 - ⇒ *пояснение**:
[*продукт'* – результат воздействия субъекта на объект.]
 - ⇒ *разбиение**:
 - { • *заготовка'*
 - ⇒ *пояснение**:
[*заготовка'* – превращённый в сырьё объект.]
 - *полуфабрикат'*

⇒ *пояснение**:
[*полуфабрикат'* – наполовину изготовленный из сырья продукт.]

- *прототип'*
⇒ *пояснение**:
[*прототип'* – опытный образец продукта.]

- *изделие'*
⇒ *пояснение**:
[*изделие'* – готовый продукт.]

}

}

воздействие

:= [акция]

⇒ *источник**:

Гордей.А.Н.TheoryAGK-2017эл

⇒ *разбиение**:

Типология по характеру взаимодействия участников[^]

= {

- *воздействие активизации*
⇒ *пояснение**:
[*воздействие активизации* – воздействие, в ходе которого взаимодействие осуществляется между субъектом и инструментом.]
- *воздействие эксплуатации*
⇒ *пояснение**:
[*воздействие эксплуатации* – воздействие, в ходе которого взаимодействие осуществляется между инструментом и медиатором.]
- *воздействие трансформации*
⇒ *пояснение**:
[*воздействие трансформации* – воздействие, в ходе которого взаимодействие осуществляется между объектом и продуктом.]
- *воздействие нормализации*
⇒ *пояснение**:
[*воздействие нормализации* – воздействие, в ходе которого взаимодействие осуществляется между объектом и продуктом.]

}

⇒ *разбиение**:

Типология воздействий по виду взаимодействующих подсистем[^]

= {

- *воздействие среда-оболочка*
- *воздействие оболочка-ядро*
- *воздействие ядро-оболочка*
- *воздействие оболочка-среда*

}

воздействие

⇒ *разбиение**:

Типология воздействий по фазам наращивания воздействия[^]

= {

- *воздействие инициации*
⇒ *пояснение**:
[*воздействие инициации* – воздействие, в ходе которого оно начинается в каждой подсистеме.]
- *воздействие аккумуляции*
⇒ *пояснение**:
[*воздействие аккумуляции* – воздействие, в ходе которого происходит его накопление в каждой подсистеме.]
- *воздействие амплификации*
⇒ *пояснение**:
[*воздействие амплификации* – воздействие, в ходе которого происходит его усиление.]
- *воздействие генерации*
⇒ *пояснение**:

[*воздействие генерации* – воздействие, которое представляет собой переход в каждой подсистеме с одного уровня, например, среды-оболочки, на другой, например, оболочки-ядра.]

воздействие

⇒ *разбиение**:

Типология воздействий по виду инструмента[^]

= { • *физическое воздействие*

⇒ *пояснение**:

[*физическое воздействие* – воздействие, в котором в роли инструмента выступает оболочка субъекта.]

• *информационное воздействие*

⇒ *пояснение**:

[*информационное воздействие* – воздействие, в котором в роли инструмента выступает среда субъекта.]

Рис. Таблица воздействий

=
[

	I	II	III	IV	
A	a	1 воспринимание притягивание 57	2 запоминание скапливание 58	3 осмысливание ужимание 59	4 понимание присоединение 60
	b	5 перенимание вбирание 61	6 заучивание накапливание 62	7 обдумывание центрирование 63	8 усваивание ассимилирование 64
	c	9 прочувствование перевбирание 65	10 созерцание концентрирование 66	11 переживание центрифугирование 67	12 изведывание диссимилирование 68
	d	13 отвергание выделение 69	14 изглаживание разуплотнение 70	15 переосмысливание отжимание 71	16 изживание отъединение 72
B	a	17 сообщение подведение 73	18 рекламирование наращивание 74	19 внушение прижимание 75	20 констатирование подсоединение 76
	b	21 объяснение введение 77	22 пропагандирование нагнетание 78	23 доказывание вжимание 79	24 удостоверивание соединение 80
	c	25 ниспослание проведение 81	26 вещание распространение 82	27 просветление выжимание 83	28 явление разъединение 84
	d	29 затемнение выведение 85	30 шифрование осаживание 86	31 дискредитирование оттеснение 87	32 дезавуирование отсоединение 88
C	a	33 информирование затрагивание 89	34 заинтересовывание обволакивание 90	35 уверение обжимание 91	36 предрасположение формование 92
	b	37 наставление вскрывание 93	38 обучение наполнение 94	39 убеждение сжимание 95	40 воспитание формирование 96
	c	41 пронимание пронизывание 97	42 преисполнение переполнение 98	43 преобразование разжимание 99	44 перевоплощение выхолащивание 100
	d	45 донимание пробивание 101	46 зомбирование вздымание 102	47 умопомрачение распускание 103	48 умалишение аннигилирование 104
D	a	49 воспоминание рекристаллизование 105	50 воссоздание реинтегрирование 106	51 возобновление регенерирование 107	52 воспроизведение реформование 108
	b	53 репродуцирование рекуперирование 109	54 рекультивирование реабилитирование 110	55 возрождение реактивирование 111	56 воскрешение реанимирование 112

]]

⇒ *источник**:

Гордей, А. Н. Theory AGK-2017эл

⇒ *пояснение**:

[На изображении представлена типология *воздействий*. Любое воздействие характеризуется принадлежностью четырём классам, соответствующим признакам классификации. Заштрихованы *физические воздействия*.]

Специфицируемые классы воздействий

```

⊃ {
  • формирование
    ⇐ пересечение*:
    {
      • воздействие трансформации
      • воздействие среда-оболочка
      • воздействие генерации
      • физическое воздействие
    }

  • притягивание
    ⇐ пересечение*:
    {
      • воздействие активизации
      • воздействие среда-оболочка
      • воздействие инициации
      • физическое воздействие
    }

  • выхолощивание
    ⇐ пересечение*:
    {
      • воздействие трансформации
      • воздействие ядро-оболочка
      • воздействие генерации
      • физическое воздействие
    }

  • аннигилирование
    ⇐ пересечение*:
    {
      • воздействие трансформации
      • воздействие оболочка-среда
      • воздействие генерации
      • физическое воздействие
    }

  • введение
    ⇐ пересечение*:
    {
      • воздействие эксплуатации
      • воздействие оболочка-ядро
      • воздействие инициации
      • физическое воздействие
    }

  • распускание
    ⇐ пересечение*:
    {
      • воздействие трансформации
      • воздействие оболочка-среда
      • воздействие амплификации
      • физическое воздействие
    }

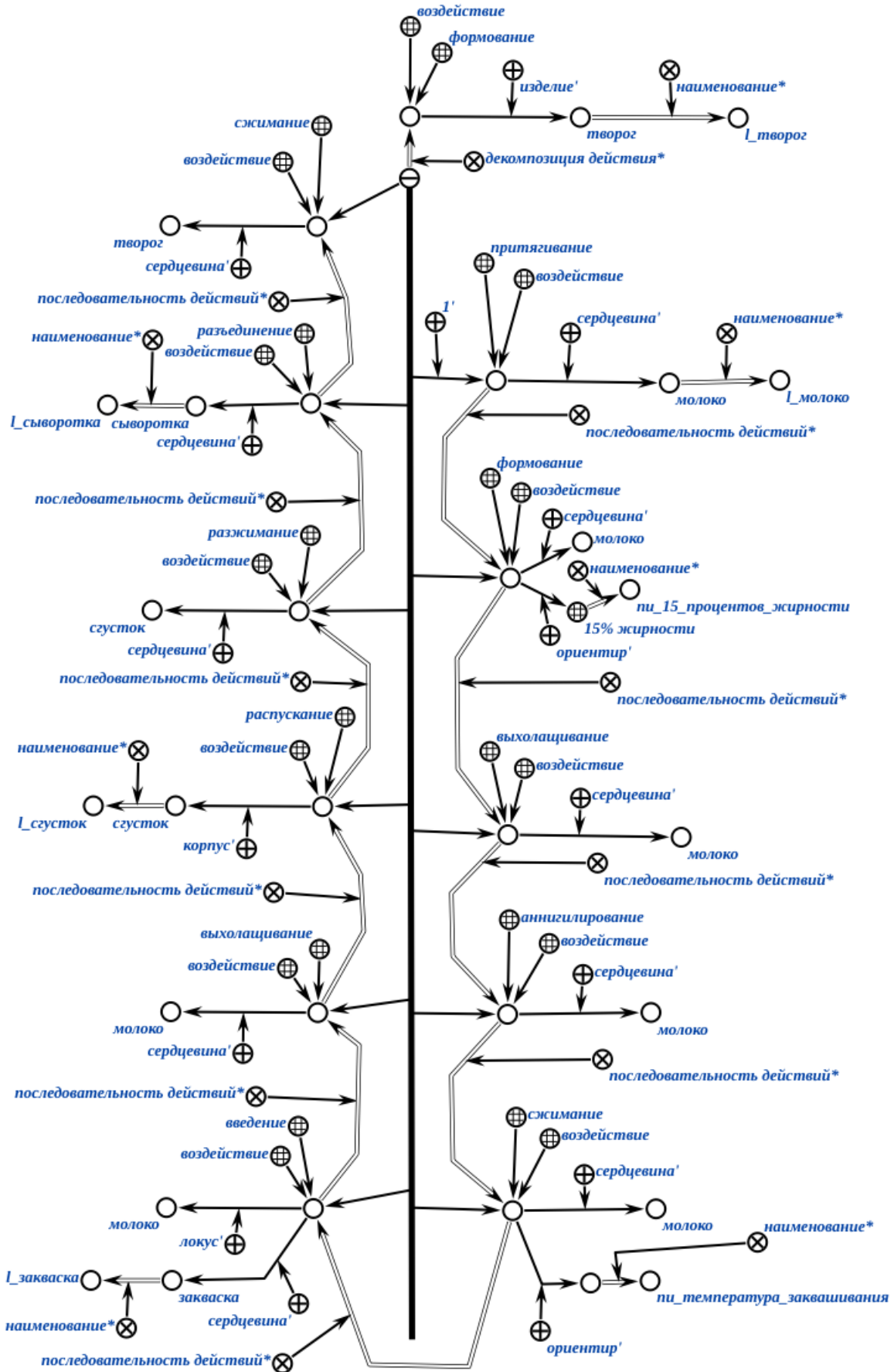
  • разжимание
    ⇐ пересечение*:
    {
      • воздействие трансформации
      • воздействие ядро-оболочка
      • воздействие амплификации
      • физическое воздействие
    }

  • разъединение
    ⇐ пересечение*:
    {
      • воздействие эксплуатации
      • воздействие ядро-оболочка
      • воздействие генерации
      • физическое воздействие
    }
}

```

Пример sc.g-текста, описывающего спецификацию воздействия

=



]

€

⇒

sc.g-текст

пояснение*:

[Представленный фрагмент базы знаний содержит декомпозицию воздействия во времени, указание принадлежности данного декомпозируемого воздействия и полученных в результате данной декомпозиции воздействий определенному их классу из приведенной выше классификации, а также указание участников данных акций.]

⇒ *пояснение**:

[Представленный фрагмент базы знаний можно протранслировать в следующий текст естественного языка: «Некто принимает молоко, затем окисляет молоко, а именно: нормализует молоко до 15-процентной жирности, затем очищает молоко, затем пастеризует молоко, затем охлаждает молоко до определённой температуры, затем вносит закваску в молоко, затем сквашивает молоко, затем режет сгусток, затем подогревает сгусток, затем обрабатывает сгусток, затем отделяет сыворотку, затем охлаждает сгусток и, в итоге, производит творог».]

субъект

:= *часто используемый sc-идентификатор**:

[индивид]

:= [активная сущность]

:= [сущность, способная самостоятельно выполнять некоторые виды действий]

:= [агент деятельности]

⊃ *Собственное Я*

⊃ *внутренний субъект ostis-системы*

⊃ *внешний субъект ostis-системы, с которым осуществляется взаимодействие*

⊃ *внешний субъект ostis-системы, с которым взаимодействие не происходит*

внутренний субъект ostis-системы

:= [субъект, входящий в состав той *ostis-системы*, в базе знаний которой он описывается]

⊃ *sc-агент*

⇒ *пояснение**:

[Под *внутренним субъектом ostis-системы* понимается такой *субъект*, который выполняет некоторые действия в той же памяти, в которой хранится его знак.

К числу *внутренних субъектов ostis-системы* относятся входящие в нее *sc-агенты*, частные *sc-машины*, целые интеллектуальные подсистемы.]

внешний субъект ostis-системы, с которым осуществляется взаимодействие

⇒ *пояснение**:

[К числу *внешних субъектов ostis-системы, с которыми осуществляется взаимодействие*, относятся конечные пользователи *ostis-системы*, ее разработчики, а также другие компьютерные системы(причем не только интеллектуальные).]

субъект действия'

С *субъект'*

:= [сущность, воздействующая на некоторую другую сущность в процессе заданного действия']

:= [сущность, создающая *причину* изменений другой сущности (объекта действия)']

:= [быть субъектом данного действия']

⊃ *субъект неосознанного воздействия'*

⊃ *субъект осознанного воздействия'*

:= [субъект целенаправленного, активного воздействия']

исполнитель*

⇒ *пояснение**:

[Связки отношения *исполнитель** связывают *sc-элементы*, обозначающие действие и *sc-элементы*, обозначающие субъекта, который предположительно будет осуществлять, осуществляет или осуществлял выполнение указанного действия. Данное отношение может быть использовано при назначении конкретного исполнителя для проектной задачи по развитию баз знаний.

В случае, когда заранее неизвестно, какой именно *субъект** будет исполнителем данного действия, отношение *исполнитель** может отсутствовать в первоначальной формулировке задачи и добавляться позже, уже непосредственно при исполнении.

Когда действие выполняется (является *настоящей сущностью*) или уже выполнено (является *прошлой сущностью*), то исполнитель этого действия в каждый момент времени уже определен. Но когда действие только инициировано, тогда важно знать:

1. кто хочет выполнить это действие и насколько важно для него стать исполнителем данного действия;
2. кто может выполнить данное действие и каков уровень его квалификации и опыта;
3. кто и кому поручает выполнить это действие и каков уровень ответственности за невыполнение (приказ, заказ, официальный договор, просьба...)

При этом следует помнить, что связь отношения *исполнитель** в данном случае также является временной прогнозируемой сущностью.

Первым компонентом связок отношений *исполнитель** является знак *действия*, вторым – знак *субъекта исполнителя*]

объект воздействия'

С *объект'*

:= [сущность, на которую осуществляется воздействие в рамках заданного действия']

:= [сущность, являющаяся в рамках заданного действия исходным условием (аргументом), необходимым для выполнения этого действия']

⇒ *примечание**:

[Для разных действий количество объектов действий может быть различным.]

⇒ *примечание**:

[Поскольку действие является процессом и, соответственно, представляет собой *динамическую структуру*, то и знак *субъекта действия'*, и знак *объекта действия'* являются элементами данной структуры. В связи с этим можно рассматривать отношения *субъект действия'* и *объект действия'* как *ролевые отношения*. Данный факт не запрещает вводить аналогичные *неролевые отношения*, однако это нецелесообразно.]

продукт'

:= [быть продуктом заданного действия']

С *продукт**

С *результат**

:= ["сухой" остаток']

:= [то, ради чего может быть выполнено, выполняется или будет выполняться заданное действие']

⇒ *примечание**:

[Продуктом действия может быть некоторая материальная сущность, некоторое множество (тираж) одинаковых материальных сущностей, некоторая информационная конструкция]

результат*

⇒ *пояснение**:

[Связки отношения *результат** связывают *sc-элемент*, обозначающий *действие*, и *sc-конструкцию*, описывающую результат выполнения рассматриваемого действия, другими словами, цель, которая должна быть достигнута при выполнении *действия*.

Результат может специфицироваться как атомарным высказыванием, так и неатомарным, т.е. конъюнктивным, дизъюнктивным, строго дизъюнктивным и т.д.

В случае, когда успешное выполнение *действия* приводит к изменению какой-либо конструкции в *sc-памяти*, которое необходимо занести в историю изменений базы знаний или использовать для демонстрации протокола решения задачи, генерируется соответствующая связка отношения *результат**, связывающая задачу и *sc-конструкцию*, описывающую данное изменение. Конкретный вид указанной *sc-конструкции* зависит от типа действия.]

⇔ *следует отличать**:

*цель**

:= [спецификация планируемого результата*]

⇒ *примечание**:

[Следует также отмечать то, что является непосредственно результатом (продуктом) некоторого действия, и то, что является предварительной (исходной, стартовой) спецификацией этого результата. Далеко не всегда результатом действия является именно то, что планировалось (было целью) изначально.]

класс выполняемых действий*

:= [класс действий, выполняемых классом субъектов*]

⇒ *пояснение**:

[Связки отношения *класс выполняемых действий** связывают классы *субъектов* и классы действий, при этом предполагается, что каждый субъект указанного класса способен выполнять действия указанного класса действий.]

заказчик*

⇒ *пояснение**:

[Связки отношения *заказчик** связывают классы *sc-элементы*, обозначающие *действие*, и *sc-элементы*, обозначающие *субъекта*, который "заинтересован" в выполнении данного действия и, как правило, инициирует его выполнение. Данное отношение может быть использовано при указании того, кто поставил проектную задачу по развитию баз знаний.

Первым компонентом связок отношения *заказчик** является знак *действия*, вторым – знак *субъекта*.]

инициатор*

⇒ *пояснение**:

[Связки отношения *инициатор** связывают *sc-элемент*, обозначающий *иницированное действие*, и знак *субъекта*, который является инициатором данного *действия*, то есть *субъектом*, который инициировал данное *действие* и, как правило, заинтересован в его успешном выполнении.]

объект'

:= [аргумент действия']

⇒ *пояснение**:

[Связки отношения *объект'* связывают *sc-элемент*, обозначающий *действие*, и знак той сущности, над которой (по отношению к которой) осуществляется данное *действие*, и, например, знак *структуры*, подлежащий верификации.]

⊃ *первый аргумент действия'*

⊃ *второй аргумент действия'*

⊃ *третий аргумент действия'*

класс аргументов*

:= [класс аргументов класса команд*]

:= [быть классом *sc-элементов*, экземпляры которого являются аргументами для заданного класса команд*]

⊃ *класс первых аргументов**

⊃ *класс вторых аргументов**

⇒ *пояснение**:

[Связки отношения *класс аргументов** связывают *классы команд* (подмножества множества *команд*) и классы *sc-элементов*, которые могут быть аргументами действий, соответствующих данному *классу команд*. В случае, когда *команды* данного класса имеют один аргумент, используется собственно отношение *класс аргументов**, в случае, когда *команды* данного класса имеют более одного аргумента, то используются подмножества данного отношения, такие как *класс первых аргументов**, *класс вторых аргументов** и т.д.

Если для некоторого *класса команд* не указан тип какого-либо из аргументов, то предполагается, что в качестве данного аргумента может выступать любой *sc-элемент*.

Первым компонентом связок отношения *класс аргументов** является знак *класса команд*, вторым – знак *класса sc-элемента*, которые могут быть *аргументами действий'*, соответствующих данному *классу команд*.]

} /* Завершили Сегмент “Предметная область и онтология субъектно-объектных спецификаций воздействий” */

/* Сегмент *****/

Уточнение понятий план сложного действия, классы действий, класса задач, метода

⊃=

{

∈ *сегмент базы знаний*

план сложного действия

- := [план]
- := [план выполнения сложного действия]
- := [план решения *сложной задачи*]
- := [план выполнения действия]
- := [спецификация выполнения действия]
- := [декомпозиция выполняемого действия на систему последовательно/параллельно выполняемых поддействий*]
- := [описание того, как может быть выполнено соответствующее сложное действие]
- := [спецификация соответствующего действия, уточняющая то, как предполагается выполнять это действие]
- := [план решения задачи (выполнения сложного действия) путем описания последовательности выполнения поддействий с описанием того, как передается управление от одних поддействий другим, как осуществляется распараллеливание, как организуется выполнение циклов]

⇒ *определение**:

[вид спецификации *сложного действия*, представляющий собой систему задач, интерпретация которой (предполагающая решение указанных задач в определенной последовательности) обеспечивает выполнение специфицируемого *сложного действия*]

С *знание*

⇒ *пояснение**:

[Каждый *план* представляет собой *семантическую окрестность*, ключевым *sc-элементом*¹ является *действие*, для которого дополнительно детализируется предполагаемый процесс его выполнения. Основная задача такой детализации – локализация области базы знаний, в которой предполагается работать, а также набора агентов, необходимого для выполнения описываемого действия. При этом детализация не обязательно должна быть доведена до уровня элементарных действий, цель составления плана – уточнение подхода к решению той или иной задачи, не всегда предполагающее составления подробного пошагового решения.

При описании *плана* может быть использован как процедурный, так и декларативный подход. В случае процедурного подхода для соответствующего *действия* указывает его декомпозиция на более частные поддействия, а также необходимая спецификация этих поддействий. В случае декларативного подхода указывается набор подцелей (например, при помощи логических утверждений), достижение которых необходимо для выполнения рассматриваемого *действия*. На практике оба рассмотренных подхода можно комбинировать.

В общем случае *план* может содержать и переменные, например в случае, когда часть плана задается в виде цикла (многократного повторения некоторого набора действий). Также план может содержать константы, значение которых в настоящий момент не установлено и станет известно, например, только после выполнения предшествующих ему *действий*.

Каждый *план* может быть задан заранее как часть спецификации *действия*, т.е. *задачи*, а может формироваться *субъектов* уже собственно в процессе выполнения *действия*, например, в случае использования стратегии разбиения задачи над подзадачи. В первом случае *план включается** в *задачу*, соответствующую тому же действию.]

⇒ *разбиение**:

- {
 - *процедурный план сложного действия*
 - := [декомпозиция *сложного действия* на множество последовательно и/или параллельно выполняемых *поддействий*]
 - *непроцедурный план сложного действия*
 - := [декомпозиция исходной *задачи*, соответствующей заданному *сложному действию*, на иерархическую систему и/или подзадач]

процедурный план сложного действия

⇒ *примечание**:

[В *процедурном плане* выполнения *сложного действия* соответствующие *поддействия** декомпозируемого *сложного действия* представляются специфицирующими их *задачами*. Но, кроме такого рода *задач*, в *процедурный план* выполнения *сложного действия* входят также *задачи*, которые специфицируют *действия*, обеспечивающие:

- синхронизацию выполнения *поддействий** заданного *сложного действия*;

- передачу управления указанным *поддействиям** (а точнее, соответствующим им *задачам*), т.е. инициирование указанных *поддействий** (и соответствующих им *задач*).

]

действие управления интерпретацией процедурного плана сложного действия

- ⇔ семантически близкий знак*:
задача управления интерпретацией процедурного плана сложного действия
- ⊃ безусловная подзадача управления от одного поддействия к другому
- ⊃ инициирование заданного поддействия при возникновении в базе знаний ситуации или события, заданного вида
- ⊃ инициирование заданного множества поддействий при успешном завершении выполнения всех поддействий другого заданного множества
- ⊃ инициирование заданного множества поддействий при успешном завершении выполнения по крайней мере одного поддействия другого заданного множества

класс действий

- ⇐ семейство подклассов*:
действие
- := пояснение*:
[максимальное множество аналогичных (похожих в определенном смысле) действий, для которого существует (но не обязательно известных в текущий момент) по крайней мере один *метод* (или средство), обеспечивающий выполнение любого действия из указанного множества действий]
- := [множество однотипных действий]
- ⊃ класс элементарных действий
- ⊃ класс легковыполнимых сложных действий
- ⇒ примечание*:
[Тот факт, что каждому выделяемому *классу действий* соответствует по крайней мере один общий для них *метод* выполнения этих *действий*, означает то, что речь идет о семантической "кластеризации" множества *действий*, т.е. выделении *классов действий* по признаку семантической близости (сходства) *действий*, входящих в состав выделяемого *класса действий*. При этом прежде всего учитывается аналогичность (сходство) *исходных ситуаций* и *целевых ситуаций* рассматриваемых *действий*, т.е. аналогичность *задач*, решаемых в результате выполнения соответствующих *действий*. Поскольку одна и та же *задача* может быть решена в результате выполнения нескольких разных действий, принадлежащих разным классам действий, следует говорить не только о *классах действий* (множествах аналогичных действий), но и о *классах задач* (о множествах аналогичных задач), решаемых этими *действиями*. Так, например, на множестве *классом действий* заданы следующие *отношения*:
- *отношение*, каждая связка которого связывает два разных (непересекающихся) *класса действий*, осуществляющих решение одного и того же *класса задач*;
- *отношение*, каждая связка которого связывает два разных *класса действий*, осуществляющих решение разных *классов задач*, один из которых является *надмножеством* другого.

]

- ⇒ *правило идентификации экземпляров**:

[Конкретные *классы действий* в рамках *Русского языка* именуется по следующим правилам:

- в начале идентификатора пишется слово "*действие*" и ставится точка;
- далее со строчной буквы идет либо содержащее глагол совершенного вида в инфинитиве описание сути того, что требуется получить в результате выполнения действий данного класса, либо вопросительное предложение, являющееся спецификацией запрашиваемой (ответной) информации.

Например:

действие, сформировать полную семантическую окрестность указываемой сущности
действие, верифицировать заданную структуру

Допускается использовать менее строгие идентификаторы, которые, однако, обязаны оперировать словом "*действие*" и достаточно четко специфицировать суть действий описываемого класса.

Например:

действие редактирования базы знаний
действие, направленное на установление темпоральных характеристик указываемой сущности]

класс элементарных действий

:= [множество элементарных действий, указание принадлежности которому является необходимым и достаточным условием для выполнения этого действия]

⇒ *примечание**:

[Множество всевозможных элементарных действий, выполняемых каждым субъектом, должно быть разбито на классы элементарных действий.]

⇒ *пояснение**:

[Принадлежность некоторого *класса действий* множеству *классу элементарных действий*, фиксирует факт того, что при указании всех необходимых аргументов принадлежности *действия* данному классу достаточно для того, чтобы некоторый субъект мог приступить к выполнению этого действия.

При этом, даже если *класс действий* принадлежит множеству *класс элементарных действий*, не запрещается вводить более частные *классы действий*, для которых, например, заранее фиксируется один из аргументов.

Если конкретный *класс элементарных действий* является более частным по отношению к *действиям в sc-памяти*, то это говорит о наличии в текущей версии системы как минимум одного *sc-агента*, ориентированного на выполнение действий данного класса.]

класс легковыполнимых сложных действий

:= [множество сложных действий, для которого известен и доступен по крайней мере один *метод*, интерпретация которого позволяет осуществить полную (окончательную, завершающуюся элементарными действиями) декомпозицию на поддействия каждого сложного действия из указанного выше множества]

:= [множество всех сложных действий, выполнимых с помощью известного *метода*, соответствующего этому множеству]

⇒ *пояснение**:

[Принадлежность некоторого *класса действий* множеству *класс легковыполнимых сложных действий* фиксирует факт того, что даже при указании всех необходимых аргументов принадлежности *действия* данному классу недостаточно для того, чтобы некоторый *субъект* приступил к выполнению этого действия, и требуются дополнительные уточнения.]

сужение отношения по первому домену*(спецификация*;класс действий*)

:= *часто используемый идентификатор**:

[спецификация класса действий*]

⇒ *разбиение**:

{ • *обобщенная формулировка задач соответствующего класса**

⇒ *разбиение**:

{ • *обобщенная декларативная формулировка задач соответствующего класса**

• *обобщенная процедурная формулировка задач соответствующего класса**

}

• *метод**

:= [метод решения задач заданного класса*]

:= [метод выполнения действий соответствующего (заданного) класса*]

⇒ *разбиение**:

{ • *процедурный метод выполнения действий соответствующего класса**

:= [обобщенный план выполнения действий заданного класса*]

• *декларативный метод выполнения действий соответствующего класса**

:= [обобщенная декларативная спецификация выполнения действий заданного класса*]

}

}

класс задач

:= [множество аналогичных задач]

:= [множество задач, для которого можно построить обобщенную формулировку задач, соответствующую всему этому множеству задач]

⇒ *примечание**:

[Каждая *обобщенная формулировка задач соответствующего класса* по сути есть не что иное, как строгое логическое определение указанного класса задач.]

← *семейство подмножеств**:

задача

⇒ *правило идентификации экземпляров**:

[Конкретные *классы задач* в рамках *Русского языка* именуется по следующим правилам:

- в начале идентификатора пишется слово “задача” и ставится точка;
- далее с прописной буквы идет либо содержащее глагол совершенного вида в инфинитиве описание сути того, что требуется получить в результате решения данного *класса задач*, либо вопросительное предложение, являющееся спецификацией запрашиваемой (ответной) информации.

Например:

задача. сформировать полную семантическую окрестность указываемой сущности
задача. верифицировать заданную структуру

Допускается использовать менее строгие идентификаторы, которые, однако, обязаны оперировать словом “задача” и достаточно четко специфицировать суть задач описываемого класса.

Например:

задача на установление значения величины
задача на доказательство]

класс команд

⇐ *семейство подмножеств**:

задача

⊃

класс интерфейсных пользовательских команд

⊃

класс интерфейсных команд пользователя ostis-системы

⊃

класс команд без аргументов

⊃

класс команд с одним аргументом

⊃

класс команд с двумя аргументами

⊃

класс команд с произвольным числом аргументов

⇒

*пояснение**:

[Идентификатор конкретного класса *класса команд* в рамках *Русского языка* пишется со строчной буквы и представляет собой либо содержащее глагол совершенного вида в инфинитиве описание сути того, что требуется получить в результате выполнения действий, соответствующих данному *классу команд*, либо вопросительное предложение, являющееся спецификацией запрашиваемой (ответной) информации.

Например:

сформировать полную семантическую окрестность указываемой сущности
верифицировать заданную структуру

Допускается использовать менее строгие идентификаторы, которые, однако, обязаны оперировать словом “команда” и достаточно четко специфицировать суть задач описываемого класса.

Например:

команда редактирования базы знаний
команда установления темпоральных характеристик указываемой сущности]

⇒

*разбиение**:

```
{
• атомарный класс команд
• неатомарный класс команд
}
```

атомарный класс команд

⇒ *пояснение**:

[Принадлежность некоторого *класса команд* множеству *атомарных классов команд* фиксирует факт того, что данная спецификация является достаточной для того, чтобы некоторый субъект приступил к выполнению соответствующего действия.

При этом, даже если *класс команд* принадлежит множеству *атомарных классов команд* не запрещается вводить более частные *классы команд*, в состав которых входит информация, дополнительно специфицирующая соответствующее *действие*.

Если соответствующий данному *классу команд* *класс действий* является более частным по отношению к *действиям в sc-памяти*, то попадание данного класса команд во множество *атомарных классов команд* говорит о наличии в текущей версии системы как минимум одного *sc-агента*, условие инициирования которого соответствует формулировке команд данного класса.]

неатомарный класс команд

⇒ *пояснение**:

[Принадлежность некоторого *класса команд* множеству *неатомарных классов команд* фиксирует факт того, что данная спецификация не является достаточной для того, чтобы некоторый субъект приступил к выполнению соответствующего действия, и требует дополнительных уточнений.]

класс действий

⇒ разбиение*:

- { • класс действий, однозначно задаваемый решаемым классом задач
:= [класс действий, обеспечивающих решение соответствующего класса задач и использующих при этом любые, самые разные методы решения задач этого класса]
- класс действий, однозначно задаваемый используемым методом решения задач

метод

← второй домен*:

метод*

- := [описание того, как может быть выполнено любое или почти любое действие, принадлежащее соответствующему классу действий]
 - := [метод решения соответствующего класса задач, обеспечивающий решение любой или большинства задач указанного класса]
 - := [обобщенная спецификация выполнения действий соответствующего класса]
 - := [обобщенная спецификация решения задач соответствующего класса]
 - := [программа решения задач соответствующего класса, которая может быть как процедурной, так и декларативной (непроцедурной)]
 - := [знание о том, как можно решать задачи соответствующего класса]
 - С *знание*
 - ∈ *вид знаний*
 - := [способ]
 - := [знание о том, как надо решать задачи соответствующего класса задач (множества эквивалентных (однотипных, похожих) задач)]
 - := [метод (способ) решения некоторого (соответствующего) класса задач]
 - := [информация (знание), достаточная для того, чтобы решить любую задачу, принадлежащую соответствующему классу задач с помощью соответствующей модели решения задач]
 - := [обобщенный план выполнения некоторого класса сложных действий (или обобщенный план решения соответствующего класса сложных задач), "привязка" которого к конкретной задаче указанного класса и последующая интерпретация обеспечивает решение любой или почти любой задачи этого класса]
- ⇒ примечание*:
- [Очевидно, что трудоемкость разработки метода определяется не столько мощностью класса задач, решаемых с помощью разрабатываемого метода, сколько их семантической близостью, аналогичностью.]
- := [метод решения соответствующего класса задач]
 - := [метод выполнения соответствующего класса действий]
 - := ["пассивный" метод, хранимый в базе знаний и используемый соответствующими коллективами агентов при соответствующем их инициировании]
 - := [метод выполнения действий некоторого класса или метод решения задач некоторого класса или метод, который может быть использован для выполнения некоторого конкретного действия или для решения некоторой конкретной задачи]
 - := [обобщенное описание того, как можно выполнить действия из соответствующего класса действий или как можно решить задачу из соответствующего класса задач]

⊃ метод сведения задач к подзадам

- := [класс логически эквивалентных методов, обеспечивающих решение задач путем сведения этих задач к подзадам и отличающихся только деталями формализации]

⇒ примечание*:

[В состав спецификации каждого *класса задач* входит описание способа "привязки" *метода* к исходным данным конкретной задачи, решаемой с помощью этого *метода*. Описание такого способа "привязки" включает в себя:

- набор переменных, которые входят как в состав *метода*, так и в состав *обобщенной формулировки задач соответствующего класса* и значениями которых являются соответствующие элементы исходных данных каждой конкретной решаемой задачи;

- часть *обобщенной формулировки задач* того класса, которому соответствует рассматриваемый *метод*, являющихся описанием условия применения этого *метода*.

Сама рассматриваемая "привязка" *метода* к конкретной *задаче*, решаемой с помощью этого *метода* осуществляется путем поиска в базе знаний такого фрагмента, который удовлетворяет условиям применения указанного *метода*. Одним из результатов такого поиска и является установление соответствия между указанными выше переменными используемого *метода* и значениями этих переменных в рамках конкретной решаемой *задачи*.

Другим вариантом установления рассматриваемого соответствия является явное обращение (вызов, call) соответствующего *метода* (программы) с явной передачей соответствующих параметров. Но такое не всегда возможно, т.к. при выполнении процесса решения конкретной *задачи* на основе декларативной спецификации выполнения этого действия нет возможности установить:

- когда необходимо инициировать вызов (использование) требуемого *метода*;
- какой конкретно *метод* необходимо использовать;
- какие параметры, соответствующие конкретной инициируемой *задаче*, необходимо передать для "привязки" используемого *метода* к этой *задаче*.

Процесс "привязки" *метода* решения *задач* к конкретной *задаче*, решаемой с помощью этого *метода*, можно также представить как процесс, состоящий из следующих этапов:

- построение копии используемого *метода*;
- склеивание основных (ключевых) переменных используемого *метода* с основными параметрами конкретной решаемой *задачи*.

В результате этого на основе рассматриваемого *метода* используемого в качестве образца (шаблона) строится спецификация процесса решения конкретной задачи – процедурная спецификация (*план*) или декларативная.]

⇒ *примечание**:

[Заметим, что *методы* могут использоваться даже при построении *планов* решения конкретных *задач*, в случае, когда возникает необходимость многократного повторения неких цепочек *действий* при априори неизвестном количестве таких повторений. Речь идет о различного вида *циклах*, которые являются простейшим видом процедурных *методов* решения задач, многократно используемых (повторяемых) при реализации *планов* решения некоторых *задач*.]

:= [программа]

:= [программа выполнения действий некоторого класса]

⇒ *примечание**:

[Одному *классу действий* может соответствовать несколько *методов* (программ).]

⊃ программа в *sc-памяти*

⊃ процедурная программа

:= [обобщенный процедурный план]

:= [обобщенный процедурный план выполнения некоторого класса действий]

:= [обобщенный процедурный план решения некоторого класса задач]

:= [обобщенная спецификация декомпозиции любого действия, принадлежащего заданному классу действий]

:= [знание о некотором классе действий (и соответствующем классе задач), позволяющее для каждого из указанных действий достаточно легко построить процедурный план его выполнения]

С алгоритм

⇒ *пояснение**:

[Каждая *процедурная программа* представляет собой обобщенный процедурный план выполнения *действий*, принадлежащих некоторому классу, то есть *семантическую окрестность, ключевым *sc-элементом** является *класс действий*, для элементов которого дополнительно детализируется процесс их выполнения.

В остальном описание *процедурной программы* аналогично описанию *плана* выполнения конкретного *действия* из рассматриваемого *класса действий*.

Входным параметрам *процедурной программы* в традиционном понимании соответствуют аргументы, соответствующие каждому *действию* из *класса действий*, описываемого *процедурной программой*. При генерации на основе *процедурной программы плана* выполнения конкретного *действия* из данного класса эти аргументы принимают конкретные значения.

Каждая *процедурная программа* представляет собой систему описанных действий с дополнительным указанием для действия:

- либо *последовательности выполнения действий** (передачи инициирования), когда условием выполнения (инициирования) действий является завершение выполнения одного из указанных или всех указанных действий;
- либо события в базе знаний или внешней среде, являющегося условием его инициирования;
- либо ситуации в базе знаний или внешней среде, являющейся условием его инициирования;

]

⇒ *примечание**:

[Отметим, что понятие *метода* фактически позволяет локализовать область решения задач соответствующего класса, то есть ограничить множество знаний, которых достаточно для решения задач данного класса определенным способом. Это, в свою очередь, позволяет повысить эффективность работы системы в целом, исключая число лишних действий.]

⇒ *примечание**:

[Каждый конкретный метод рассматривается нами как важный вид спецификации соответствующего класса задач, но также и как *объект*, который и сам нуждается в спецификации, обеспечивающей непосредственное применение этого метода. Другими словами, метод является не только спецификацией (спецификацией соответствующего класса задач), но и объектом спецификации.]

эквивалентность задач*

:= [быть эквивалентной задачей*]

∈ *отношение*

⇒ *определение**:

[Задачи являются эквивалентными в том и только в том случае, если они могут быть решены путем интерпретации одного и того же *метода* (способа), хранимого в памяти кибернетической системы.]

⇒ *примечание**:

[Некоторые *задачи* могут быть решены разными *методами*, один из которых, например, является обобщением другого.]

отношение, заданное на множестве*(метод)

∋ *подметод**

:= [подпрограмма*]

:= [быть методом, использование которого (обращение к которому) предполагается при реализации заданного метода*]

⇔ *следует отличать**:
*частный метод**

:= [быть методом, обеспечивающим решение класса задач, который является подклассом задач, решаемых с помощью заданного метода*]

стратегия решения задач

С *метод*

:= [метаметод решения задач, обеспечивающий либо поиск одного релевантного известного метода, либо синтез целенаправленной последовательности акций применения в общем случае различных известных методов]

⇒ *примечание**:

[Можно говорить об универсальном метаметод (универсальной стратегии) решения задач, объясняющем всевозможные частные стратегии.]

⇒ *пояснение**:

[Можно говорить о нескольких глобальных *стратегиях решения информационных задач* в базах знаний. Пусть в базе знаний появился знак инициированного действия с формулировкой соответствующей информационной цели, т.е. цели, направленной только на изменение состояния базы знаний. И пусть текущее состояние базы знаний не содержит контекста (исходных данных), достаточного для достижения указанной выше цели, т.е. такого контекста, для которого в доступном пакете (наборе) методов (программ) имеется метод (программа), использование которого позволяет достигнуть указанную выше цель. Для достижения такой цели, контекст (исходные данные) которой недостаточен, существует три подхода (три стратегии):

- декомпозиция (сведение изначальной цели к иерархической системе и/или подцелей (и/или подзадач) на основе анализа текущего состояния базы знаний и анализа того, чего в базе знаний не хватает для использования того или иного метода.)

При этом наибольшее внимание уделяется методам, для создания условий использования которых требуется меньше усилий. В конечном счете мы должны прийти (на самом нижнем уровне иерархии) до подцелей, контекст которых достаточен для применения одного из имеющихся методов (программ) решения задач;

- генерация новых знаний в семантической окрестности формулировки изначальной цели с помощью любых доступных методов в надежде получить такое состояние базы знаний, которое будет содержать нужный контекст (достаточные исходные данные) для достижения изначальной цели с помощью какого-либо имеющегося метода решения задач;
- комбинация первого и второго подхода.

Аналогичные стратегии существуют и для поиска пути решения задач, решаемых во внешней среде.]

сужение отношения по первому домену(спецификация*, метод)

```
:= [спецификация метода*]
⇒ разбиение*:
{ • денотационная семантика метода*
  := [обобщенная формулировка класса задач, решаемых с помощью данного метода*]
  ⇔ семантически близкий знак***:
    обобщенная формулировка задач соответствующего класса*
    ⇒ примечание*:
      [Данное отношение связывает обобщенную формулировку задач не с методом, а с
      классом задач]
  • операционная семантика метода*
    := [перечень обобщенных агентов, обеспечивающих интерпретацию метода*]
    := [семейство методов интерпретации данного метода*]
    := [формальное описание интерпретатора заданного метода*]
}
} /* Завершили Сегмент "Уточнение понятий план сложного действия, классы действий, класса задач,
метода" */
```

/* Сегмент ***/**

Уточнение понятия навыка, понятия класса методов и понятия модели решения задач

```
⊃=
{
∈ сегмент базы знаний

навык
:= [умение]
:= [объединение метода с его исчерпывающей спецификацией – полным представлением операционной
семантики метода]
:= [метод, интерпретация (выполнение, использование) которого полностью может быть осуществлено данной
кибернетической системой, в памяти которой указанный метод хранится]
:= [метод, который данная кибернетическая система умеет (может) применять]
:= [метод + метод его интерпретации]
:= [умение решать соответствующий класс эквивалентных задач]
:= [метод плюс его операционная семантика, описывающая то, как интерпретируется (выполняется, реализуется)
этот метод, и являющаяся одновременно операционной семантикой соответствующей модели решения
задач]
⇒ разбиение*:
{ • активный навык
  := [самоиницирующийся навык]
  • пассивный навык
}
⇒ пояснение*:
```

[*Навыки* могут быть *пассивными навыками*, то есть такими *навыками*, применение которых должно явно инициироваться каким-либо агентом, либо *активными навыками*, которые инициируются самостоятельно при возникновении соответствующей ситуации в базе знаний. Для этого в состав *активного навыка* помимо *метода* и его операционной семантики включается также *sc-агент*, который реагирует на появление соответствующей ситуации в базе знаний и инициирует интерпретацию *метода* данного *навыка*.

Такое разделение позволяет реализовать и комбинировать различные подходы к решению задач, в частности, *пассивные навыки* можно рассматривать в качестве способа реализации концепции интеллектуального пакета программ.]

класс методов

- ← семейство подклассов*:
метод
- := [множество методов, для которых можно унифицировать представление (спецификацию) этих методов]
- := [множество всевозможных методов решения задач, имеющих общий язык представления этих методов]
- := [множество всевозможных методов, представленных на данном языке]
- := [множество методов, для которых задан язык представления этих методов]
- Э процедурный метод решения задач
 - ⊃ алгоритмический метод решения задач
- Э логический метод решения задач
 - ⊃ продукционный метод решения задач
 - ⊃ функциональный метод решения задач
- Э искусственная нейронная сеть
 - := [класс методов решения задач на основе искусственных нейронных сетей]
- Э генетический "алгоритм"
- := [множество методов основанных на общей онтологии]
- := [множество методов, представленных на одинаковом языке]
- := [множество методов решений задач, которому соответствует специальный язык (например, sc-язык), обеспечивающий представление методов из этого множества]
- := [множество методов, которому ставится в соответствие отдельная модель решения задач]

язык представления методов

- := [язык методов]
- := [язык представления методов, соответствующих определенному классу методов]
 - ⇒ примечание*:
[Таких специализированных языков может быть выделено целое множество, каждому из которых будет соответствовать своя модель решения задач (т.е. свой интерпретатор)]
- := [язык (например sc-язык) представлений методов соответствующего класса методов]
- С язык
- := [язык программирования]
 - ⊃ язык представления методов обработки информации
 - := [язык программирования внутренних действий кибернетической системы, выполняемых в их памяти]
 - := [язык представления методов решения задач в памяти кибернетических систем]
 - ⊃ язык представления методов решения задач во внешней среде кибернетических систем
 - := [язык программирования внешних действий кибернетических систем]

модель решения задач

- := [метаметод интерпретации соответствующего класса методов]
- С метод
- := [метаметод]
- := [абстрактная машина интерпретации соответствующего класса методов]
- := [иерархическая система "микропрограмм", обеспечивающих интерпретацию соответствующего класса методов]
 - ⊃ алгоритмическая модель решения задач
 - ⊃ процедурная параллельная синхронная модель решения задач
 - ⊃ процедурная параллельная асинхронная модель решения задач
 - ⊃ продукционная модель решения задач
 - ⊃ функциональная модель решения задач
 - ⊃ логическая модель решения задач

- ⊃ четкая логическая модель решения задач
- ⊃ нечеткая логическая модель решения задач
- ⊃ "нейросетевая" модель решения задач
- ⊃ "генетическая" модель решения задач
- ⇒ примечание*:
- [Для интерпретации всех моделей решения задач может быть использован агентно-ориентированный подход]
- ⇒ пояснение*:
- [Каждая модель решения задач задается:
 - соответствующим классом методов решения задач, т.е. языком представления методов этого класса;
 - предметной областью этого класса методов;
 - онтологией этого класса методов (т.е. денотационной семантикой языка представления этих методов);
 - операционной семантикой указанного класса методов.
-]

модель решения задач*

- = сужение отношения по первому домену(спецификация*; класс методов)*
- := [спецификация класса методов*]
- := [спецификация языка представления методов*]
- ⇐ обобщенное объединение*:
- {
 - синтаксис языка представления методов соответствующего класса*
 - денотационная семантика языка представления методов соответствующего класса*
 - операционная семантика языка представления методов соответствующего класса*
- }
 - ⇒ примечание*:
 - [Каждому конкретному классу методов взаимно однозначно соответствует язык представления методов, принадлежащих этому (специфицируемому) классу методов. Таким образом, спецификация каждого класса методов сводится к спецификации соответствующего языка представления методов, т.е. к описанию его синтаксической, денотационной семантики и операционной семантики. Примерами языков представления методов являются все языки программирования, которые в основном относятся к подклассу языков представления методов – к языкам представления методов обработки информации. Но сейчас все большую актуальность приобретает необходимость создания эффективных формальных языков представления методов выполнения действий во внешней среде кибернетических систем. Без этого комплексная автоматизация, в частности, в промышленной сфере невозможна.]

денотационная семантика языка представления методов соответствующего класса

- ⇐ второй домен*:
- денотационная семантика языка представления методов соответствующего класса*
- := [онтология соответствующего класса методов]
- := [денотационная семантика соответствующего класса методов]
- := [денотационная семантика языка (sc-языка), обеспечивающего представление методов соответствующего класса]
- := [денотационная семантика соответствующей модели решения задач]
- ⇒ примечание*:
- [если речь идет о языке, обеспечивающем внутреннее представление методов соответствующего класса в ostis-системе, то синтаксис этого языка совпадает с синтаксисом SC-кода]
- С онтология

операционная семантика языка представления методов соответствующего класса

- ⇐ второй домен*:
- операционная семантика языка представления методов соответствующего класса*
- := [метаметод интерпретации соответствующего класса методов]
- := [семейство агентов, обеспечивающих интерпретацию (использования) любого метода, принадлежащего соответствующему классу методов]
- := [операционная семантика соответствующей модели решения задач]

язык представления обобщенных формулировок задач для различных классов задач

- ⇒ примечание*:

[Поскольку каждому *методу* соответствует *обобщенная формулировка задач*, решаемых с помощью этого *метода*, то каждому *классу методов* должен соответствовать не только определенный *язык представления методов*, принадлежащих указанному *классу методов*, но и определенный *язык представления обобщенных формулировок задач для различных классов задач*, решаемых с помощью *методов*, принадлежащих указанному *классу методов*.]

} /* Завершили Сегмент “Уточнение понятия навыка, понятия класса методов и понятия модели решения задач” */

/* Сегмент *****/

Уточнение понятия деятельности, понятия вида деятельности и понятия технологии

⊃=

{

Є *сегмент базы знаний*

деятельность

:= *пояснение**:

[сложный процесс, состоящий из действий, направленных на достижение нескольких разных целей (т.е. целей, не связанных отношением "цель-подцель"). При этом некоторые из указанных "максимальных" целей могут достигаться с помощью одного и того же метода или одного и того же (фиксированного) семейства методов]

⊃ *физическая деятельность*

:= [деятельность по преобразованию материальных сущностей (физических объектов)]

⊃ *информационная деятельность*

:= [деятельность, направленная на обработку информации]

:= [умственная деятельность]

⇒ *примечание**:

[Информационная деятельность является необходимым компонентом физической деятельности, обеспечивающим принятие решений, планирование и управление физическим процессом]

С *процесс*

:= [целостный, целенаправленный процесс поведения (функционирования) одного субъекта или сообщества субъектов, осуществляемый на основе хорошо или не очень хорошо продуманной и согласованной *технологии* в последнем случае качество деятельности определяется уровнем интеллекта единоличного или коллективного субъекта, осуществляющего этот целенаправленный процесс.]

:= [система действий, являющаяся некоторым кластером семантически близких действий, обладающих семантической близостью, семантической связностью и семантической целостностью]

:= [трудно выполнимая семантически целостная система действий]

:= [кластер множества действий, определяемый семантической близостью этих действий]

:= [система связанных между собой действий, имеющих общий контекст, общую область выполнения этих действий]

⇒ *примечание**:

[В состав каждой конкретной *деятельности* входят *действия*, являющиеся *поддействиями** других *действий*, входящих в состав этой же *деятельности*. При этом для каждого *действия*, входящего в состав *деятельности*, все поддействия этого *действия* также входят в состав этой *деятельности*.

В состав каждой конкретной *деятельности* входят также *действия*, не являющиеся *поддействиями** других *действий*, входящих в состав этой же *деятельности*. Такие "первичные" ("независимые", "самостоятельные", "автономные") *действия* для заданной *деятельности* могут инициироваться извне этой *деятельности* с помощью соответствующих инициирующих эти *действия ситуаций* или *событий*. Примерами таких инициирующих ситуаций, "порождающих" соответствующие действия, являются:

- появление в *базе знаний* каких-либо противоречий, информационных дыр, информационного мусора;
- появление в *базе знаний* описаний (информационных моделей) каких-либо нештатных ситуаций в сложном объекте управления, на которые необходимо реагировать;
- появление в *базе знаний* формулировок различного рода задач с явным указанием инициирования соответствующих действий, направленных на решение этих задач.

К числу указанных "первичных" ("независимых") *действий*, входящих в состав *объединенной деятельности кибернетической системы*, также относятся:

- сложное действие, целью которого является перманентное обеспечение комплексной *безопасности кибернетической системы*;
- сложное действие, целью которого является перманентное повышение качества информации (базы знаний), хранимой в памяти *кибернетической системы*;
- сложное действие, целью которого является перманентное повышение *качества решателя задач кибернетической системы*;
- сложное действие, целью которого является перманентная поддержка высокого уровня *семантической совместимости* кибернетической системы со своими партнерами.

]

отношение, заданное на множестве*(деятельность)

Э *субъект**

:= [быть субъектом заданного действия или деятельности*]

:= [кибернетическая система, которая в рамках заданного действия или деятельности выполняет ту или иную роль, воздействует на некий объект действия, используя тот или иной инструмент*]

Є *отношение, заданное на множестве*(действие)*

Э *контекст**

:= [информационный контекст, в рамках которого осуществляется выполнение заданного действия или деятельности*]

:= [область исполнения действия или деятельности*]

:= [область действия или деятельности*]

⇒ *первый домен**:
(*действие* ∪ *деятельность*)

:= [совокупность знаний, достаточных для информационного обеспечения заданного действия или заданной деятельности]

Є *отношение, заданное на множестве* (действие)*

⇒ *примечание**:

[Локализация (минимизация) *контекста* заданного действия или деятельности является важнейшим "подготовленным" этапом, обеспечивающим существенное снижение "накладных расходов" при непосредственном выполнении этого *действия* или *деятельности*.]

⇒ *примечание**:

[Чаще всего *контекстом* заданного *действия* или *деятельности* является некоторая *предметная область* вместе с соответствующей ей интегрированной (объединенной) *онтологией*. Поэтому хорошо продуманная декомпозиция *базы знаний* интеллектуальной компьютерной системы на иерархическую систему *предметных областей* и соответствующих им *онтологий* имеет важное "практическое" значение, существенно повышающее качество (в частности, быстродействие) *решателя задач* интеллектуальной компьютерной системы благодаря априорному разбиению множества выполняемых *действий* (решаемых задач) по соответствующим им *контекстам*.]

следует отличать*

⊃ { • *действие*

:= [процесс достижения конкретной цели конкретных обстоятельствах]

:= [процесс решения конкретной задачи в конкретных условиях]

:= [процесс задуманный, инициированный и осуществленный некоторым (или некоторыми) субъектами (кибернетическими системами)]

⇒ *примечание**:

[*действие* (точнее, соответствующая форма участия в его выполнении) является частью (фрагментом) *деятельности* всех участвующих в этом субъектов (кибернетических систем)]

• *деятельность*

:= [система действий выполняемых соответствующим субъектом (кибернетической системой) "скрепленное" общим контекстом и определенным набором используемых навыков и инструментов]

⇒ *примечание**:

[В отличие от *действия*, *деятельность* носит чаще всего перманентный характер в рамках времени существования соответствующего субъекта]

}

деятельность кибернетической системы

:= [полная система действий, выполняемых соответствующей кибернетической системой]

:= [деятельность субъекта]

:= [система всех действий соответствующего субъекта]

⇒ *разбиение**:

{ • *внутренняя деятельность субъекта*

:= [внутренняя деятельность соответствующего субъекта]

:= [деятельность некоторого субъекта по обработке информации]

:= [информационная деятельность]

• *поведение субъекта*

:= [внешнее поведение соответствующего субъекта]

:= [деятельность субъекта во внешней среде]

}

вид деятельности

← *семейство подклассов**:

деятельность

:= [класс семантически целостных систем действия, для которых можно унифицировать используемые методы, информационные ресурсы и инструменты]

:= [класс трудно выполнимых и семантически целостных систем сложных действий]

:= [класс кластеров систем действий]

:= [множество деятельностей, которые могут быть реализованы с помощью общей технологии]

Э *устранение противоречий в базе знаний*

Э *устранение информационных дыр в базе знаний*

Э *ликвидация информационного мусора в базе знаний*

Э *управление сложным внешним объектом*

Э *поддержка семантической совместимости с партнерами*

Э *проектирование*

:= [проектная деятельность]

:= [построение такого описания (в частности, описания структуры) некоторого материального объекта, которого достаточно для воспроизводства (реализации, материализации) этого объекта либо при одиночном (уникальном), либо при массовом (промышленном) воспроизводстве указанного объекта]

⇒ *примечание**:

[Примерами проектирования являются:

- проектирование здания;
- проектирование машиностроительной конструкции;
- проектирование микросхемы;
- проектирование ostis-системы;
- разработка системы шунтирования сердца;
- разработка такого описания сложной геометрической фигуры, которого было бы достаточно для построения изображения (рисунка) этой фигуры с помощью, например, циркуля и линейки.

]

Э *разработка плана производства материального объекта по заданному проекту этого объекта*

⊃ *разработка плана единичной реализации материального объекта по заданному проекту этого объекта*

⊃ *разработка плана массовой реализации материальных объектов по заданному их типовому проекту*

⇒ *примечание**:

[Примерами данного вида действий являются:

- разработка плана-графика строительства конкретного здания;
- разработка типового плана строительства зданий по заданному их типовому проекту;
- разработка типового плана операций шунтирования сердца;
- разработка алгоритма построения изображения заданной геометрической фигуры с помощью циркуля и линейки

]
 Э *производство*
 := [воспроизводство материального объекта по заданному его проекту и плану реализации]
 := [производственная деятельность]
 ⇒ *примечание**:
 [Примерами данного вида действий являются:
 непосредственно строительство конкретного здания;
 проведение конкретной хирургической операции;
 процесс построения изображения (рисунка) геометрической фигуры с помощью циркуля и линейки.

]
 Э *реинжиниринг*
 Э *анализ*
 Э *интеграция*
 := [синтез]
 Э *деятельность в области здравоохранения*
 Э *образовательная деятельность*
 Э *эксплуатация сложного объекта*
 Э *научно-исследовательская деятельность*
 Э *управление*
 ⊃ *целенаправленная координация деятельности нескольких субъектов*
 := [управление целенаправленной коллективной деятельностью нескольких субъектов]

проектирование

:= [действие, направленное на построение (разработку) такой информационной модели (проекта) некоторой материальной сущности, которой достаточно, чтобы соответствующий индивидуальный или коллективный субъект по соответствующей технологии (т.е. с помощью соответствующих методов и средств (инструментов)) смог воспроизвести (изготовить) указанную материальную сущность либо в одном экземпляре, либо в достаточно большом количестве таких экземпляров (копий), т.е. воспроизвести в промышленном масштабе]

производство

:= [воспроизводство]
 := [изготовление]
 := [реализация]
 := [материализация]
 := [построение, синтез материальной сущности (артефакта)]
 := [изготовление материальной сущности в одной или во множестве экземпляров (копий)]
 := [производство (как действие)]

реинжиниринг

:= [модификация]
 := [внесение изменений в некую сущность]
 := [обновление]
 := [реинжиниринг]
 := [перепроектирование]
 := [реконфигурация]
 := [трансформирование]
 ⊃ *совершенствование*
 := [модификация, направленной на повышение качества модифицируемой сущности]
 := [повышение качества]
 := [улучшение]
 ⊃ *самосовершенствование*
 := [совершенствование, выполняемое самой совершенствуемой сущностью]
 ⊃ *совершенствование, осуществляемое извне*
 ⇒ *примечание**:

[Самосовершенствоваться и обучаться могут только достаточно развитые кибернетические системы. Но совершенствоваться усилиями внешних субъектов могут любые сущности.]

анализ

:= [построение (разработка, создание) спецификации (описания) основных связей и/или структуры, свойств, закономерностей, соответствующих (описываемой) сущности]

⇒ *примечание**:

[Объектом анализа может быть не только материальная сущность, но и процесс, ситуация, статическая структура, внешняя информационная конструкция, знание, понятие и другие абстрактные сущности]

интеграция

:= [синтез]

:= [соединение]

:= [объединение]

:= [сборка]

⇒ *разбиение**:

{ • *эkleктичная интеграция*

:= [интеграция без разрушения целостности интегрируемых сущностей]

:= [интеграция без взаимопроникновения]

:= [соединение систем по их входам/выходам]

• *глубокая интеграция*

:= [интеграция, в результате которой получается гибридная сущность]

:= [интеграция с разрушением целостности (взаимопроникновением "диффузий") интегрируемых сущностей]

:= ["бесшовная" интеграция]

}

вид деятельности

⇒ *пояснение**:

[Если классу легко выполнимых сложных действий ставится в соответствие чаще всего один метод и, возможно, некоторый набор инструментальных средств, используемых в этом методе, то каждому виду деятельности ставится в соответствие своя *технология*, включающая в себя некоторый набор используемых *методов*, а также набор *инструментальных средств*, используемых в этих *методах*. Сложность здесь заключается:

- в нетривиальности организации использования всего арсенала имеющейся *технологии* для реализации (выполнения) каждой соответствующей *деятельности*;
- в трудности, а часто и в принципиальной невозможности полностью автоматизировать реализацию соответствующей *деятельности*.

]

следует отличать*

Э { • *действие*

Э *пример'*:

Процесс доказательства Теоремы Пифагора

∈ *действие направленное на построение доказательства теоремы Геометрии Евклида*

• *класс действий*

Э *пример'*:

процесс доказательства теоремы

:= *имя нарицательное**:

[действие, направленное на построение доказательства (логического обоснования) теоремы]

:= *имя собственное**:

[Класс действий, направленных на построение доказательств (логических обоснований) всевозможных теорем в различных формальных теориях]

• *деятельность*

Э *пример'*:

Процесс эволюции Геометрии Евклида

:= [Процесс эволюции формальной теории, являющейся формальным представлением Геометрии Евклида]

⇒ *пояснение**:

[В данный процесс входит и генерация гипотез в рамках Геометрии Евклида, и доказательство теорем, и выявление противоречий между высказываниями, и разрешение этих противоречий, и минимизация числа используемых определяемых понятий, и многое другое]

⇒ *примечание**:

[*деятельность* – это то, что "превращает" множество самостоятельных и в определенной степени независимых *действий*, принадлежащих разным *классам действий*, в целостную, целенаправленную, сбалансированную систему *действий*, ориентированную, прежде всего на поддержание качества и эволюцию *кибернетических систем*, а также на обеспечение их адаптации к новым, ранее не предусмотренным обстоятельствам.]

• *вид деятельности*

∃ *пример'*:

процесс эволюции формальной теории

:= *имя собственное**:

[Класс процессов, направленных на эволюцию всевозможных формальных теорий (логических онтологий), которая также включает в себя возможность коррекции этих теорий.]

}

сужение отношения по первому домену(спецификация*; вид деятельности)*

:= *часто используемый sc-идентификатор**:

[спецификация вида деятельности*]

= *технология**

:= [технология реализации (выполнения) деятельности соответствующего (заданного) вида*]

⇒ *второй домен**:

технология

:= [технология соответствующего вида деятельности]

⇔ *аналог**:

декларативный метод выполнения действий соответствующего класса

⇔ *аналог**:

декларативная спецификация выполнения действия

⇒ *пояснение**:

[*технология* (как спецификация соответствующего вида деятельности) включает в себя:

- указание *контекста** специфицируемого *вида деятельности*;
- указание *множества используемых методов**, множества используемых инструментов, а также используемых материалов.

]

технология

⇒ *пояснение**:

[Каждая *технология* представляет собой комплекс *методов* (методик) и средств, обеспечивающих выполнение некоторого множества *действий*, входящих в состав соответствующего *вида деятельности*. Каждая *технология* задается:

- множеством методов (методик), которое разбивается на классы методов, эквивалентных по своей операционной семантике (по набору агентов, осуществляющих интерпретацию соответствующего класса методов);
- множеством агентов, являющихся средством интерпретации методов из указанного выше множества.

Указанное множество агентов также разбивается на подмножества, каждое из которых соответствует своему классу методов и обеспечивает интерпретацию методов только этого класса.]

:= [множество (комплекс) навыков, обеспечивающих выполнение такого множества действий (задач), для которых отсутствует общий метод их выполнения]

:= [методика, инструментарий и дополнительные ресурсы, которые обеспечивают выполнение каждой конкретной деятельности, принадлежащей соответствующему виду деятельности]

⇒ *пояснение**:

[с формальной точки зрения каждая технология задается ориентированной связкой, компонентами которой являются

- знак множества используемых методов
- знак множества используемых инструментов
- знак множества дополнительных используемых ресурсов

]

:= [комплекс методов и средств (инструментов), с помощью которого некий субъект (который может быть как индивидуальным, так и коллективным) осуществляет некоторую деятельность (некоторое целенаправленное множество действий, входящих в состав этой деятельности)]

⊃ *технология научно-теоретической деятельности*

⊃ *технология проектирования*

:= [технология проектной деятельности]

:= [технология построения такой информационной модели соответствующей сущности (артефакта), которой достаточно для воспроизводства этой сущности]

⊃ *технология производства*

:= [технология производственной деятельности]

:= [технология воспроизводства некоторого вида сущностей по заданным проектам этих сущностей]

⊃ *технология здравоохранения*

⊃ *технология образования*

:= [технология подготовки молодых специалистов]

:= [технология образовательной деятельности]

отношение, заданное на множестве* (технология*)

Э *методы**

:= [семейство методов, используемых в специфицируемой технологии с дополнительным указанием их иерархии (т.е. с указанием того, какие методы используются при реализации других методов)]

Э *активные инструменты**

:= [средства, которые сами способны выполнять некоторые действия, но при этом ими надо как-то управлять (например, транспортные средства, компьютеры, . . .)]

:= [средства автоматизации*]

Э *пассивные инструменты**

:= [средства, которые сами ничего делать не могут (например, молоток, лопата, ножницы, . . .)]

Э *комплектация**

Э *расходные средства**

Э *сырье**

Э *продукты**

Э *общий продукт**

:= [объединенный (интегрированный) продукт*]

Э *реализация технологии**

:= [вариант (форма) реализации технологии*]

Э *частная технология**

:= [быть частной технологией по отношению к заданной технологии*]

продукты*

:= [производимые сущности*]

:= [изготавливаемые материальные сущности*]

:= [продукция*]

:= [результаты выполнения соответствующего множества действий, осуществляемых во внешней среде*]

:= [продукты технологии*]

:= [множество материальных сущностей, производимых (создаваемых, порождаемых, изготавливаемых) с помощью заданной технологии*]

:= [то, что является "сухим остатком" при использовании данной технологии*]

технология

⇒ *примечание**:

[Поскольку разработка каждой конкретной *технологии* требует больших затрат, очень важно, чтобы *технологии* создавались не под конкретные *деятельности*, а для целых классов деятельностей (*видов деятельности*). При этом важно, чтобы разрабатываемые *технологии* охватывали как можно большее количество деятельностей, входящих в состав указанных *видов деятельности*. Из этого следует целесообразность конвергенции и унификации различных сфер *деятельности* для того, чтобы повысить мощность применения (использования) каждой разрабатываемой *технологии*. Кроме того важна *совместимость технологий*, позволяющая решать *задачи*, требующие одновременного использования нескольких *технологий*, причем, в непредсказуемых сочетаниях. Очень важно также, кроме *видов деятельности*, которым соответствуют конкретные *технологии*, ввести *обобщенные виды деятельности* и построить их иерархии явно фиксировать стандарты, которым должны соответствовать все виды соответствующего обобщенного *вида деятельности*. Это необходимо для обеспечения совместимости *технологий*. Все используемые технологии должны "пронизывать" друг друга и составлять стройную иерархическую систему совместимых технологий (сумму технологий).]

класс технологий

- := [множество похожих технологий, использующих, например, одинаковые методики и/или одинаковые активные инструменты и/или одинаковые пассивные инструменты и/или похожие множества продуктов]
- Э *технология проектирования*
 - ⊃ *технология проектирования интеллектуальных компьютерных систем*
 - ⊃ *технология проектирования программных систем*
 - ⊃ *технология проектирования микросхем*
 - ⊃ *технология машиностроительного проектирования*
- Э *технология рецептурного производства*
 - ⊃ *технология производства молочных продуктов*
 - ⊃ *технология производства мясных продуктов*
 - ⊃ *технология фармацевтического производства*
- } /* Завершили Сегмент “Уточнение понятия деятельности, понятия вида деятельности и понятия технологий” */
- } /* Завершили Раздел “Глобальная предметная область действий и задач и соответствующая ей онтология методов и технологий” */

/* Раздел *****/

Предметная область и онтология решателей задач ostis-систем

⊃=

{

⇒ дочерний раздел*:

- Предметная область и онтология действий, задач, планов, протоколов и методов, реализуемых ostis-системой, а также внутренних агентов, выполняющих эти действия
- Предметная область и онтология Базового языка программирования ostis-систем
- Предметная область и онтология искусственных нейронных сетей и соответствующая им предметная область и онтология действий по обработке искусственных нейронных сетей

решатель задач ostis-системы

:= [совокупность всех навыков, которыми обладает ostis-система на текущий момент времени]

⇐ семейство подмножеств*:

навык

⇒ примечание*:

[Предлагаемый в рамках *Технологии OSTIS* подход к построению решателей задач позволяет обеспечить их модифицируемость, что, в свою очередь, позволяет *ostis-системе* при необходимости легко приобретать новые *навыки*, модифицировать (совершенствовать) уже имеющиеся, и даже избавляться от некоторых навыков с целью повышения производительности системы. Таким образом, имеет смысл говорить не о жестко фиксированном решателе задач, который разрабатывается один раз при создании первой версии системы и далее не меняется, а о совокупности навыков, фиксированной в каждый текущий момент времени, но постоянно эволюционирующей.]

⊃ объединенный решатель задач ostis-системы

:= [полный решатель задач ostis-системы]

:= [интегрированный решатель задач ostis-системы]

:= [решатель задач ostis-системы, реализующий все ее функциональные возможности, как основные, так и вспомогательные]

⇒ пояснение*:

[В общем случае *объединенный решатель задач ostis-системы*, решает задачи, связанные с:

- обеспечением основных функциональных возможностей системы (например, решение явно сформулированных задач по требованию пользователя);
- обеспечением корректности и оптимизацией работы самой ostis-системы (перманентно на протяжении всего жизненного цикла ostis-системы);
- обеспечением повышения квалификации конечных пользователей и разработчиков ostis-системы;
- обеспечением автоматизации развития и управления развитием ostis-системы.

]

⊃ гибридный решатель задач ostis-системы

:= [решатель задач ostis-системы, реализующий две и более модели решения задач]

машина обработки знаний

⊂ sc-агент

⇒ пояснение*:

[Под *машиной обработки знаний* будем понимать совокупность интерпретаторов всех *навыков*, составляющих некоторый *решатель задач*. С учетом многоагентного подхода к обработке информации, используемого в рамках *Технологии OSTIS*, *машина обработки знаний* представляет собой *sc-агент* (чаще всего – *неавтономный sc-агент*), в состав которого входят более простые *sc-агенты*, обеспечивающие интерпретацию соответствующего множества *методов*. Таким образом, *машина обработки знаний* в общем случае представляет собой иерархическую систему *sc-агентов*.]

решатель задач ostis-системы

⊃ Решатель задач *Метасистемы IMS.ostis*

⊃ решатель задач вспомогательной ostis-системы

⊃ решатель задач интерфейса компьютерной системы

⇒ разбиение*:

- решатель задач пользовательского интерфейса компьютерной системы

- решатель задач интерфейса компьютерной системы с другими компьютерными системами
- решатель задач интерфейса компьютерной системы с окружающей средой

- ⊃ решатель задач ostis-подсистемы поддержки проектирования компонентов определенного класса
 - ⊃ решатель задач ostis-подсистемы поддержки проектирования баз знаний
 - ⊃ решатель задач повышения качества базы знаний
 - ⊃ решатель задач верификации базы знаний
 - ⊃ решатель задач поиска и устранения некорректностей в базе знаний
 - ⊃ решатель задач поиска и устранения неполноты
 - ⊃ решатель задач оптимизации структуры базы знаний
 - ⊃ решатель задач выявления и устранения информационного мусора
 - ⊃ решатель задач ostis-подсистемы поддержки проектирования решателей задач ostis-систем
 - ⇒ разбиение*:
 - решатель задач ostis-подсистемы поддержки проектирования программ обработки знаний
 - решатель задач ostis-подсистемы поддержки проектирования агентов обработки знаний
- ⊃ решатель задач подсистемы управления проектирования компьютерных систем и их компонентов
- ⊃ решатель задач самостоятельной ostis-системы

решатель задач ostis-системы

- ⊃ решатель задач с использованием хранимых методов
 - := [решатель, способный решать задачи тех классов, для которых на данный момент времени известен соответствующий метод решения]
 - ⊃ решатель задач на основе нейросетевых моделей
 - ⊃ решатель задач на основе генетических алгоритмов
 - ⊃ решатель задач на основе императивных программ
 - ⊃ решатель задач на основе процедурных программ
 - ⊃ решатель задач на основе объектно-ориентированных программ
 - ⊃ решатель задач на основе декларативных программ
 - ⊃ решатель задач на основе логических программ
 - ⊃ решатель задач на основе функциональных программ
- ⊃ решатель задач в условиях, когда метод решения задач данного класса в текущий момент времени не известен
 - := [решатель, реализующий стратегии решения задач, позволяющие породить метод решения задачи, который в текущий момент времени не известен ostis-системе]
 - := [решатель, использующий для решения задач метаметоды, соответствующие более общим классам задач по отношению к заданной]
 - := [решатель задач, позволяющий породить метод, который является частным по отношению какому-либо известному ostis-системе методу и интерпретируется соответствующей машиной обработки знаний]
 - ⊃ решатель, реализующий стратегию поиска путей решения задачи в глубину
 - ⊃ решатель, реализующий стратегию поиска путей решения задачи в ширину
 - ⊃ решатель, реализующий стратегию проб и ошибок
 - ⊃ решатель, реализующий стратегию разбиения задачи на подзадачи
 - ⊃ решатель, реализующий стратегию решения задач по аналогии
 - ⊃ решатель, реализующий концепцию интеллектуального пакета программ

машина обработки знаний

- ⊃ машина логического вывода
 - ⊃ машина дедуктивного вывода
 - ⊃ машина прямого дедуктивного вывода
 - ⊃ машина обратного дедуктивного вывода
 - ⊃ машина индуктивного вывода
 - ⊃ машина абдуктивного вывода
 - ⊃ машина нечеткого вывода
 - ⊃ машина вывода на основе логики умолчаний
 - ⊃ машина логического вывода с учетом фактора времени

решатель задач ostis-системы

- ⊃ решатель задач информационного поиска
 - ⇒ разбиение*:
 - решатель задач поиска информации, удовлетворяющей заданным критериям
 - решатель задач поиска информации, не удовлетворяющей заданным критериям
 - ⊃ решатель явно сформулированных задач
 - := [решатель задач, для которых явно сформулирована цель]
 - ⊃ решатель задач поиска или вычисления значений заданного множества величин
 - ⊃ решатель задач установления истинности заданного логического высказывания в рамках заданной формальной теории
 - ⊃ решатель задач формирования доказательства заданного высказывания в рамках заданной формальной теории
 - ⊃ машина верификации ответа на указанную задачу
 - ⊃ машина верификации решения указанной задачи
 - ⊃ машина верификации доказательства заданного высказывания в рамках заданной формальной теории
 - ⊃ решатель задач классификации сущностей
 - ⊃ машина соотнесения сущности с одним из заданного множества классов
 - ⊃ машина разделения множества сущностей на классы по заданному множеству признаков
 - ⊃ решатель задач синтеза информационных конструкций
 - ⊃ решатель задач синтеза естественно-языковых текстов
 - ⊃ решатель задач синтеза изображений
 - ⊃ решатель задач синтеза сигналов
 - ⊃ решатель задач синтеза речи
 - ⊃ решатель задач анализа информационных конструкций
 - ⊃ решатель задач анализа естественно-языковых текстов
 - ⊃ решатель задач понимания естественно-языковых текстов
 - ⊃ решатель задач верификации естественно-языковых текстов
 - ⊃ решатель задач анализа изображений
 - ⊃ решатель задач сегментации изображений
 - ⊃ решатель задач понимания изображений
 - ⊃ решатель задач анализа сигналов
 - ⊃ решатель задач анализа речи
 - ⊃ решатель задач понимания речи
- /* Завершили Раздел “Предметная область и онтология решателей задач ostis-систем” */

/* Раздел *****/

Предметная область и онтология действий, задач, планов, протоколов и методов, реализуемых ostis-системой, а также внутренних агентов, выполняющих эти действия

⊃=
{

Предметная область и онтология действий, задач, планов, протоколов и методов, реализуемых ostis-системой в ее памяти, а также внутренних агентов, выполняющих эти действия

- ∈ предметная область
- ⊃ максимальный класс объектов исследования¹:
 - действие в sc-памяти
 - абстрактный sc-агент
 - sc-агент
- ⊃ класс объектов исследования¹:
 - абстрактный sc-агент, не реализуемый на Языке SCP
 - абстрактный sc-агент, реализуемый на Языке SCP
 - Абстрактный программный sc-агент
 - неатомарный абстрактный sc-агент
 - атомарный абстрактный sc-агент
 - платформенно-независимый абстрактный sc-агент
 - платформенно-зависимый абстрактный sc-агент
 - внутренний абстрактный sc-агент
 - эффекторный абстрактный sc-агент
 - рецепторный абстрактный sc-агент
 - абстрактный sc-агент, не реализуемый на Языке SCP
 - абстрактный sc-агент, реализуемый на Языке SCP
 - абстрактный sc-агент интерпретации scp-программ
 - абстрактный программный sc-агент
 - абстрактный программный sc-агент, реализуемый на Языке SCP
 - абстрактный sc-метаагент
 - sc-агент
 - активный sc-агент
 - описание поведения sc-агента
 - тип блокировки
 - полная блокировка
 - блокировка на любое изменение
 - блокировка на удаление
- ⊃ исследуемое отношение¹:
 - декомпозиция абстрактного sc-агента*
 - ключевые sc-элементы sc-агента*
 - программа sc-агента*
 - первичное условие инициирования*
 - условие инициирования и результат*
 - блокировка*

обработка информации в ostis-системах

- ⇒ принципы, лежащие в основе*:
 - [В основе решателя задач каждой ostis-системы лежит многоагентная система, агенты которой взаимодействуют между собой **только(!)** через общую для них sc-память посредством спецификации в этой памяти выполняемых ими **действий в sc-памяти**. При этом пользователи ostis-системы также считаются агентами этой системы. Кроме того, sc-агенты делятся на внутренние, рецепторные и эффекторные. Взаимодействие между агентами через общую sc-память сводится к следующим видам действий:
 1. К использованию общедоступной для соответствующей группы sc-агентов части хранимой базы знаний;
 2. К формированию (генерации) новых фрагментов базы знаний и/или к корректировке (редактированию) каких-либо фрагментов доступной части базы знаний;

3. К интеграции (погружению) новых и/или обновленных фрагментов в состав доступной части базы знаний;

Подчеркнем, что *sc*-агенты не общаются между собой напрямую путем отправки сообщений, как это делается в большинстве современных подходов к построению многоагентных систем. Кроме того, *sc*-агенты имеют доступ к общей для них базе знаний за счет чего гарантируется семантическая совместимость (взаимопонимание) между агентами, включая и пользователей *ostis*-систем]

- [Пользователь *ostis-системы* не может сам непосредственно выполнить какое-либо действие в *sc*-памяти, но он может средствами пользовательского интерфейса инициировать построение (генерацию, формирование в *sc*-памяти) *sc*-текста, являющегося спецификацией действия в *sc*-памяти, выполняемого либо одним атомарным *sc*-агентом за один акт, либо одним атомарным *sc*-агентом за несколько актов, либо коллективом *sc*-агентов (неатомарным *sc*-агентом). В спецификации каждого такого действия в *sc*-памяти, инициированного пользователем, этот пользователь указывается как заказчик этого действия. Таким образом, пользователь *ostis-системы* дает поручения (задания, команды) *sc*-агентам этой системы на выполнение различных специфицируемых им действий в *sc*-памяти.]
- [Каждый *sc*-агент, выполняя некоторое действие в *sc*-памяти, должен "помнить", что *sc*-память, над которой он работает, является общим ресурсом не только для него, но и для всех остальных *sc*-агентов, работающих над этой же *sc*-памятью. Поэтому *sc*-агент должен соблюдать определенную этику поведения в коллективе таких *sc*-агентов, которая должна минимизировать помехи, которые он создает другим *sc*-агентам.]
- [Деятельность каждого агента *ostis-системы* дискретна и представляет собой множество элементарных действий (актов). При этом при выполнении каждого акта агент может устанавливать блокировки нескольких типов на фрагменты базы знаний. Указанные блокировки позволяют запретить другим агентам изменять указанный фрагмент базы знаний или вообще сделать его "невидимым" для других агентов. Блокировки устанавливаются самим агентом при выполнении соответствующего акта и снимаются им же на последнем этапе выполнения этого акта или раньше, если это возможно.]
- [Если некий *sc*-агент выполняет некоторое действие в *sc*-памяти, то он на время выполнения этого действия может:
 1. Запретить другим *sc*-агентам изменять состояние некоторых *sc*-элементов, хранимых в *sc*-памяти – удалять их, изменять тип;
 2. Запретить другим *sc*-агентам добавлять или удалять элементы некоторых множеств, обозначаемых соответствующими *sc*-узлами;
 3. Запретить другим *sc*-агентам доступ на просмотр некоторых *sc*-элементов, то есть эти *sc*-элементы становятся полностью "невидимыми" (полностью заблокированными) для других *sc*-агентов но только на время выполнения соответствующего действия.
 Указанные блокировки должны быть полностью сняты до завершения выполнения соответствующего действия. Подчеркнем, что число *sc*-элементов, блокируемых на время выполнения некоторого действия, в основном входят атомарные и неатомарные связи, и не должны входить *sc*-узлы, обозначающие бесконечные классы каких-либо сущностей, и, тем более, *sc*-узлы, обозначающие различные понятия (ключевые классы различных предметных областей).
 Этичное (неэгоистичное) поведение *sc*-агента, касающееся блокировки *sc*-элементов (то есть ограничения к ним доступа другим *sc*-агентам) предполагает соблюдение следующих правил:
 1. Не следует блокировать больше *sc*-элементов, чем это необходимо для решения задачи;
 2. Как только для какого-либо *sc*-элемента необходимость его блокировки отпадает до завершения выполнения соответствующего действия, этот *sc*-элемент желательно сразу деблокировать (снять блокировку);
 Для того, чтобы *sc*-агент имел возможность работы с каким-либо произвольным *sc*-элементом, он должен либо убедиться в том, что этот *sc*-элемент не входит во фрагмент базы знаний, входящий в полную блокировку, либо убедиться в том, что эта блокировка не установлена самим этим агентом. Особой группой полностью заблокированных *sc*-элементов (на время выполнения действия *sc*-агентом) являются вспомогательные *sc*-элементы ("леса"), создаваемые только на время выполнения этого действия. Эти *sc*-элементы в конце выполнения действия должны не деблокироваться, а удаляться.]
- [Если действие в *sc*-памяти, выполняемое *sc*-агентом, завершилось (т.е. стало прошлой сущностью), то *sc*-агент оформляет результат этого действия, указывая (1) удаленные *sc*-элементы и (2) сгенерированные *sc*-элементы. Это необходимо, если по каким-либо причинам придется сделать откат этого действия, т.е. возвратиться к состоянию базы знаний до выполнения указанного действия.]

```

/* Сегмент *****/
Понятие действия в sc-памяти
⊃=
{
действие в sc-памяти
:= [внутреннее действие ostis-системы]
:= [действие, выполняемое в sc-памяти]
:= [действие, выполняемое в абстрактной унифицированной семантической памяти]
:= [действие, выполняемое машиной обработки знаний ostis-системы]
:= [действие, выполняемое агентом или коллективом агентов ostis-системы]
:= [информационный процесс над базой знаний, хранимой в sc-памяти]
:= [процесс решения информационной задачи в sc-памяти]
С процесс в sc-памяти
⇒ пояснение*:
[Каждое действие в sc-памяти обозначает некоторое преобразование, выполняемое некоторым sc-агентом (или коллективом sc-агентов) и ориентированное на преобразование sc-памяти. Спецификация действия после его выполнения может быть включена в протокол решения некоторой задачи.

Преобразование состояния базы знаний включает, в том числе и информационный поиск, предполагающий (1) локализацию в базе знаний ответа на запрос, явное выделение структуры ответа и (2) трансляцию ответа на некоторый внешний язык.

Во множество действий в sc-памяти входят знаки действий самого различного рода, семантика каждого из которых зависит от конкретного контекста, т.е. ориентации действия на какие-либо конкретные объекты и принадлежности действия какому-либо конкретному классу действий.

Следует четко отличать:
□ Каждое конкретное действие в sc-памяти, представляющее собой некоторый переходный процесс, переводящий sc-память из одного состояния в другое;
□ Каждый тип действий в sc-памяти, представляющий собой некоторый класс однотипных (в том или ином смысле) действий;
□ sc-узел, обозначающий некоторое конкретное действие в sc-памяти;
□ sc-узел, обозначающий структуру, которая является описанием, спецификацией, заданием, постановкой соответствующего действия.

]
⊃ действие в sc-памяти, инициируемое вопросом
⊃ действие редактирования базы знаний ostis-системы
⊃ действие установки режима ostis-системы
⊃ действие редактирования файла, хранимого в sc-памяти
⊃ действие интерпретации программы, хранимой в sc-памяти

действие в sc-памяти, инициируемое вопросом
:= [действие, направленное на формирование ответа на поставленный вопрос]
⊃ действие. сформировать заданный файл
⊃ действие. сформировать заданную структуру
⊃ действие. верифицировать заданную структуру
⊃ действие. установить истинность или ложность указываемого логического высказывания
⊃ действие. установить корректность или некорректность указываемой структуры
⊃ действие. сформировать структуру, описывающую некорректности, имеющиеся в указываемой структуре
⊃ действие. уточнить тип заданного sc-элемента
⊃ действие. установить позитивность/негативность указываемой sc-дуги принадлежности или непринадлежности
⊃ действие. сформировать семантическую окрестность
⊃ действие. сформировать полную семантическую окрестность указываемой сущности
⊃ действие. сформировать базовую семантическую окрестность указываемой сущности
⊃ действие. сформировать частную семантическую окрестность указываемой сущности
⊃ действие. сформировать структуру, описывающую связи между указываемыми сущностями
⊃ действие. сформировать структуру, описывающую сходства указываемых сущностей

```

- ⊃ действие. сформировать структуру, описывающую различия указываемых сущностей
- ⊃ действие. сформировать структуру, описывающую способ решения указываемой задачи
- ⊃ действие. сформировать план генерации ответа на указанный вопрос
- ⊃ действие. сформировать протокол выполнения указываемого действия
- ⊃ действие. сформировать обоснование корректности указываемого решения
- ⊃ действие. верифицировать обоснование корректности указываемого решения
- ⊃ действие, направленное на установление темпоральных характеристик указываемой сущности
- ⊃ действие, направленное на установление пространственных характеристик указываемой сущности

действие редактирования базы знаний

- ⊃ действие. изменить направление указанной sc-дуги
- ⊃ действие. исправить ошибки в заданной структуре
- ⊃ действие. преобразовать указанную структуру в соответствии с указанным правилом
- ⊃ действие. отождествить два указанных sc-элемента
- ⊃ действие. включить множество
- := [сделать все элементы множества S_i явно принадлежащими множеству S_j , то есть сгенерировать соответствующие sc-дуги принадлежности]
- ⊃ действие генерации sc-элементов
 - ⊃ действие генерации, одним из аргументов которого является некоторая обобщенная структура
 - ⊃ действие. сгенерировать структуру, изоморфную указываемому образцу
 - ⊃ действие. сгенерировать sc-элемент указанного типа
 - ⊃ действие. сгенерировать sc-коннектор указанного типа
 - ⊃ действие. сгенерировать sc-узел указанного типа
 - ⊃ действие. сгенерировать файл с заданным содержимым
 - ⊃ действие. установить указанный файл в качестве основного идентификатора указанного sc-элемента для указанного внешнего языка
 - ⊃ действие. обновить понятия
 - := [действие. заменить неосновные понятия на их определения через основные понятия]
 - ⊃ действие. интегрировать информационную конструкцию в текущее состояние базы знаний
 - ⊃ действие. интегрировать содержимое указанного файла в текущее состояние базы знаний
 - ⊃ действие. протранслировать содержимое указанного файла в sc-память
 - ⊃ действие. интегрировать указанную структуру в текущее состояние базы знаний
 - ⊃ действие. дополнить описание прошлого состояния *ostis*-системы
 - ⊃ действие. дополнить структуру, описывающую историю эволюции *ostis*-системы
 - ⊃ действие. дополнить структуру, описывающую историю эксплуатации *ostis*-системы
 - ⊃ действие удаления sc-элементов
 - ⊃ действие. удалить указанные sc-элементы
 - ⊃ действие. удалить sc-элементы, входящие в состав указанной структуры и не являющиеся ключевыми узлами каких-либо sc-агентов

действие. отождествить два указанных sc-элемента

- := [действие. совместить два указанных sc-элемента]
- := [действие. склеить два указанных sc-элемента]
- ⇒ разбиение*:
 - {
 - действие. физически отождествить два указанных sc-элемента
 - действие. логически отождествить два указанных sc-элемента

действие. отождествить два указанных sc-элемента

- ⇒ пояснение*:

[Каждое **действие. отождествить два указанных sc-элемента** может быть выполнено как **действие. физически отождествить два указанных sc-элемента** или **действие. логически отождествить два указанных sc-элемента**. В случае логического отождествления в протоколе деятельности агентов сохраняется само действие с его спецификацией, включающей обязательное указание того, какие элементы были сгенерированы, а какие удалены. В случае физического отождествления протокол действия не сохраняется.]

действие. обновить понятия

- := [действие. заменить некоторое множество понятий на другое множество понятий]
- ⇒ пояснение*:

[Каждое *действие. обновить понятия* обозначает переход от какой-то группы понятий, использовавшихся ранее, к другой группе понятий, которые будут использоваться вместо первых, и станут *основными понятиями*. В общем случае *действие. обновить понятия* состоит из следующих этапов:

- Определить заменяемые понятия на основе заменяющих;
- Внести соответствующие изменения в программы *sc*-агентов, ключевыми узлами которых являются обновляемые понятия;
- Заменить все конструкции в базе знаний, содержащие заменяемые понятия, в соответствии с определениями этих понятий через заменяющие их понятия;
- При необходимости, *sc-элементы*, обозначающие замененные таким образом понятия, могут быть полностью выведены из текущего состояния базы знаний.

Первым аргументом (входящим в знак *действия* под атрибутом 1') *действия. обновить понятия* является знак множества *sc-узлов*, обозначающих заменяемые понятия, вторым (входящим в знак *действия* под атрибутом 2') - знак множества *sc-узлов*, обозначающих заменяющие понятия. В общем случае любое или оба этих множества могут быть *синглтонами*.]

действие. удалить указанные sc-элементы

⇒ *разбиение**:

- { • *действие. физически удалить указанные sc-элементы*
- *действие. логически удалить указанные sc-элементы*
- }

⇒ *пояснение**:

[Каждое *действие. удалить указанные sc-элементы* может быть выполнено как *действие. физически удалить указанные sc-элементы* или *действие. логически удалить указанные sc-элементы*. В случае логического удаления в протоколе деятельности агентов сохраняется само действие с его спецификацией, включающей обязательное указание того, какие элементы были удалены, т.е. по сути, элементы просто исключаются из текущего состояния базы знаний. В случае физического удаления протокол действия не сохраняется.

В случае удаления какого-либо *sc-элемента*, инцидентные ему *связки*, в том числе *sc-коннекторы*, также удаляются.]

действие. интегрировать указанную структуру в текущее состояние базы знаний

⇒ *пояснение**:

[Для того, чтобы выполнить *действие. интегрировать указанную структуру в текущее состояние базы знаний*, необходимо склеить *sc-элементы*, входящие в интегрируемую *структуру* с синонимичными им *sc-элементами*, входящими в текущее состояние базы знаний, заменить неиспользуемые (например, устаревшие) понятия, входящие в интегрируемую *структуру*, на используемые (т.е. заменить неиспользуемые понятия на их определения через используемые), явно включить все элементы интегрируемой *структуры* в число элементов утвержденной части базы знаний и явно включить все элементы интегрируемой *структуры* в число элементов одного из атомарных разделов утвержденной части базы знаний.]

действие интерпретации программы, хранимой в sc-памяти

⊃ *действие интерпретации scr-программы*

задача, решаемая в sc-памяти

⊂ *задача*

:= [спецификация действия, выполняемого в *sc-памяти*]

:= [структура, являющаяся таким описанием (постановкой, заданием) соответствующего действия в *sc-памяти*, которое обладает достаточной полнотой для выполнения указанного действия]

:= [семантическая окрестность некоторого действия в *sc-памяти*, обеспечивающая достаточно полное задание этого действия]

класс действий

⊃ *класс действий в sc-памяти*

⇐ *семейство подмножеств**:
действие в sc-памяти

⇒ *разбиение**:

- { • *класс логически атомарных действий*
:= [класс автономных действий]
- *класс логически неатомарных действий*

:= [класс неавтономных действий]
}

класс логически атомарных действий

⇒ *пояснение**:

[Каждое *действие*, принадлежащее некоторому конкретному *классу логически атомарных действий*, обладает двумя необходимыми свойствами:

- выполнение действия не зависит от того, является ли указанное действие частью декомпозиции более общего действия. При выполнении данного действия также не должен учитываться тот факт, что данное действие предшествует каким-либо другим действиям или следует за ними (что явно указывается при помощи отношения *последовательность действий**);
- указанное действие должно представлять собой логически целостный акт преобразования, например, в семантической памяти. Такое действие по сути является транзакцией, т. е. результатом такого преобразования становится новое состояние преобразуемой системы, а выполняемое действие должно быть либо выполнено полностью, либо не выполнено совсем, частичное выполнение не допускается.

В то же время логическая атомарность не запрещает декомпозировать выполняемое действие на более частные, каждое из которых, в свою очередь, также будет являться логически атомарным.]

⊃ **класс логически атомарных действий в sc-памяти**

⇒ *пояснение**:

[На логически атомарные действия предлагается делить всю деятельность, направленную на решение каких-либо задач *ostis*-системой. Соответственно *решатель задач ostis-системы* предлагается делить на компоненты, соответствующие таким *классам логически атомарных действий в sc-памяти*, что является основой для обеспечения его *модифицируемости*.]

} /* Завершили Сегмент “Понятие действия в sc-памяти” */

/* Сегмент *****/

Понятие sc-агента и абстрактного sc-агента

⊃=
{

sc-агент

:= [единственный вид *субъектов*, выполняющих преобразования в *sc-памяти*]

:= [*субъект*, способный выполнять *действия в sc-памяти*, принадлежащие некоторому определенному *классу логически атомарных действий*]

⇒ *пояснение**:

[Логическая атомарность выполняемых *sc-агентом* действий предполагает, что каждый *sc-агент* реагирует на соответствующий ему класс ситуаций и/или событий, происходящих в *sc-памяти*, и осуществляет определенное преобразование *sc-текста*, находящегося в семантической окрестности обрабатываемой ситуации и/или события. При этом каждый *sc-агент* в общем случае не имеет информацию о том, какие еще *sc-агенты* в данный момент присутствуют в системе и осуществляет взаимодействие в другими *sc-агентами* исключительно посредством формирования некоторых конструкций (как правило – спецификаций действий) в общей *sc-памяти*. Таким сообщением может быть, например, вопрос, адресованный другим *sc-агентам* в системе (заранее не известно, каким конкретно), или ответ на поставленный другими *sc-агентами* вопрос (заранее не известно, каким конкретно). Таким образом, каждый *sc-агент* в каждый момент времени контролирует только фрагмент базы знаний в контексте решаемой данным агентом задачи, состояние всей остальной базы знаний в общем случае непредсказуемо для *sc-агента*.]

абстрактный sc-агент

⇒ *примечание**:

[Поскольку предполагается, что копии одного и того же *sc-агента* или функционально эквивалентные *sc-агенты* могут работать в разных *ostis*-системах, будучи при этом физически разными *sc-агентами*, то целесообразно рассматривать свойства и классификацию не *sc-агентов*, а классов функционально эквивалентных *sc-агентов*, которые будем называть *абстрактными sc-агентами*.]

⇒ *пояснение**:

[Под **абстрактным sc-агентом** понимается некоторый класс функционально эквивалентных *sc-агентов*, разные экземпляры (т.е. представители) которого могут быть реализованы по-разному.

Каждый **абстрактный sc-агент** имеет соответствующую ему спецификацию. В спецификацию каждого **абстрактного sc-агента** входит:

- указание ключевых *sc-элементов* этого *sc-агента*, т.е. тех *sc-элементов*, хранимых в *sc-памяти*, которые для данного *sc-агента* являются «точками опоры»;
- формальное описание условий инициирования данного *sc-агента*, т.е. тех *ситуация* в *sc-памяти*, которые инициируют деятельность данного *sc-агента*;
- формальное описание первичного условия инициирования данного *sc-агента*, т.е. такой *ситуации* в *sc-памяти*, которая побуждает *sc-агента* перейти в активное состояние и начать проверку наличия своего полного условия инициирования (для *внутренних абстрактных sc-агентов*);
- строгое, полное, однозначно понимаемое описание деятельности данного *sc-агента*, оформленное при помощи каких-либо понятных, общепринятых средств, не требующих специального изучения, например на естественном языке.
- описание результатов выполнения данного *sc-агента*.

]

⇒ разбиение*:

- неатомарный абстрактный *sc-агент*
- атомарный абстрактный *sc-агент*

⇒ разбиение*:

- внутренний абстрактный *sc-агент*
- эффекторный абстрактный *sc-агент*
- рецепторный абстрактный *sc-агент*

⇒ разбиение*:

- абстрактный *sc-агент*, не реализуемый на Языке SCP
- абстрактный *sc-агент*, реализуемый на Языке SCP

⇒ разбиение*:

- абстрактный *sc-агент* интерпретации *scp-программ*
- абстрактный программный *sc-агент*
- абстрактный *sc-метаагент*

⇒ разбиение*:

- платформенно-зависимый абстрактный *sc-агент*
 \supset абстрактный *sc-агент*, не реализуемый на Языке SCP
- платформенно-независимый абстрактный *sc-агент*

абстрактный *sc-агент*, не реализуемый на Языке SCP

:= [абстрактный *sc-агент*, который не может быть реализован на платформенно-независимом уровне]

⇒ разбиение*:

- эффекторный абстрактный *sc-агент*
- рецепторный абстрактный *sc-агент*
- абстрактный *sc-агент* интерпретации *scp-программ*

абстрактный *sc-агент*, реализуемый на Языке SCP

:= [абстрактный *sc-агент*, который может быть реализован на платформенно-независимом уровне]

⇒ разбиение*:

- абстрактный *sc-метаагент*
- абстрактный программный *sc-агент*, реализуемый на Языке SCP

абстрактный программный *sc-агент*

⇒ разбиение*:

- эффекторный абстрактный *sc-агент*
- рецепторный абстрактный *sc-агент*
- абстрактный программный *sc-агент*, реализуемый на Языке SCP

неатомарный абстрактный *sc-агент*

⇒ пояснение*:

[Под **неатомарным абстрактным sc-агентом** понимается *абстрактный sc-агент*, который декомпозируется на коллектив более простых *абстрактных sc-агентов*, каждый из которых в свою очередь может быть как *атомарным абстрактным sc-агентом*, так и **неатомарным абстрактным sc-агентом**. При этом в каком либо варианте *декомпозиции абстрактного sc-агента** дочерний **неатомарный абстрактный sc-агент** может стать *атомарным абстрактным sc-агентом*, и реализовываться соответствующим образом.]

атомарный абстрактный sc-агент

⇒ *пояснение**:

[Под **атомарным абстрактным sc-агентом** понимается *абстрактный sc-агент*, для которого уточняется платформа его реализации, т.е. существует соответствующая связка отношения *программа sc-агента**.]

⇒ *разбиение**:

- { • *платформенно-независимый абстрактный sc-агент*
- *платформенно-зависимый абстрактный sc-агент*
- }

платформенно-независимый абстрактный sc-агент

⇒ *пояснение**:

[К **платформенно-независимым абстрактным sc-агентам** относят *атомарные абстрактные sc-агенты*, реализованные на базовом языке программирования Технологии OSTIS, т.е. на *Языке SCP*.

При описании **платформенно-независимых абстрактных sc-агентов** под платформенной независимостью понимается платформенная независимость с точки зрения Технологии OSTIS, т.е. реализация на специализированном языке программирования, ориентированном на обработку семантических сетей (*Языке SCP*), поскольку *атомарные sc-агенты*, реализованные на указанном языке могут свободно переноситься с одной платформы интерпретации *sc-моделей* на другую. При этом языки программирования, традиционно считающиеся платформенно-независимыми в данном случае не могут считаться таковыми.

Существуют *sc-агенты*, которые принципиально не могут быть реализованы на платформенно-независимом уровне, например, собственно *sc-агенты* интерпретации *sc-моделей* или рецепторные и эффекторные *sc-агенты*, обеспечивающие взаимодействие с внешней средой.]

платформенно-зависимый абстрактный sc-агент

⇒ *пояснение**:

[К **платформенно-зависимым абстрактным sc-агентам** относят *атомарные абстрактные sc-агенты*, реализованные ниже уровня *sc-моделей*, т.е. не на *Языке SCP*, а на каком-либо другом языке описания программ.

Существуют *sc-агенты*, которые принципиально должны быть реализованы на платформенно-зависимом уровне, например, собственно *sc-агенты* интерпретации *sc-моделей* или рецепторные и эффекторные *sc-агенты*, обеспечивающие взаимодействие с внешней средой.]

внутренний абстрактный sc-агент

⇒ *пояснение**:

[Каждый **внутренний абстрактный sc-агент** обозначает класс *sc-агентов*, которые реагируют на события в *sc-памяти* и осуществляют преобразования исключительно в рамках этой же *sc-памяти*.]

эффекторный абстрактный sc-агент

⇒ *пояснение**:

[Каждый **эффекторный абстрактный sc-агент** обозначает класс *sc-агентов*, которые реагируют на события в *sc-памяти* и осуществляют преобразования во внешней относительно данной *ostis-системы* среде.]

рецепторный абстрактный sc-агент

⇒ *пояснение**:

[Каждый **рецепторный абстрактный sc-агент** обозначает класс *sc-агентов*, которые реагируют на события во внешней относительно данной *ostis-системы* среде и осуществляют преобразования в памяти данной системы.]

абстрактный sc-агент, не реализуемый на Языке SCP

⇒ *пояснение**:

[Каждый *абстрактный sc-агент, не реализуемый на Языке SCP* должен быть реализован на уровне платформы интерпретации sc-моделей, в том числе, аппаратной. К таким *абстрактным sc-агентам* относятся абстрактные sc-агенты интерпретации scr-программ, а также эффекторные и рецепторные абстрактные sc-агенты.]

абстрактный sc-агент, реализуемый на Языке SCP

⇒ *пояснение**:

[Каждый *абстрактный sc-агент, реализуемый на Языке SCP* может быть реализован на Языке SCP, то есть платформенно-независимом уровне, но при необходимости, может реализовываться и на уровне платформы, например, с целью повышения производительности.]

абстрактный sc-агент интерпретации scr-программ

⇒ *пояснение**:

[К *абстрактным sc-агентам интерпретации scr-программ* относятся не реализуемые на платформенно-независимом уровне *абстрактные sc-агенты*, обеспечивающие интерпретацию *scr-программ* и *scr-метапрограмм*, в том числе создание *scr-процессов*, собственно интерпретацию *scr-операторов*, а также другие вспомогательные действия. По сути, агенты данного класса обеспечивают работу sc-агентов более высоких уровней (программных sc-агентов и sc-метаагентов), реализованных на Языке SCP, в частности, обеспечивают соблюдение указанными агентами общих принципов синхронизации.]

абстрактный программный sc-агент

⇒ *пояснение**:

[К *абстрактным программным sc-агентам* относятся все *абстрактные sc-агенты*, обеспечивающие основной функционал системы, то есть ее возможность решать те или иные задачи. Агенты данного класса должны работать в соответствии с общими принципами синхронизации деятельности субъектов в sc-памяти.]

абстрактный sc-метаагент

⇒ *пояснение**:

[Задачей *абстрактных sc-метаагентов* является координация деятельности *абстрактных программных sc-агентов*, в частности, решение проблемы взаимоблокировок. Агенты данного класса могут быть реализованы на Языке SCP, однако для синхронизации их деятельности используются другие принципы, соответственно, для реализации таких агентов требуется Язык SCP другого уровня, типология операторов которого полностью аналогична типологии scr-операторов, однако эти операторы имеют другую операционную семантику, учитывающую отличия в принципах синхронизации (работы с *блокировками**). Программы такого языка будем называть *scr-метапрограммами*, соответствующие им процессы в sc-памяти – *scr-метапроцессами*, операторы – *scr-метаоператорами*.]

*декомпозиция абстрактного sc-агента**

∈ *отношение декомпозиции*

⇒ *пояснение**:

[Отношение *декомпозиции абстрактного sc-агента** трактует *неатомарные абстрактные sc-агенты* как коллективы более простых *абстрактных sc-агентов*, взаимодействующих через *sc-память*.

Другими словами, *декомпозиция абстрактного sc-агента** на *абстрактные sc-агенты* более низкого уровня уточняет один из возможных подходов к реализации этого *абстрактного sc-агента* путем построения коллектива более простых *абстрактных sc-агентов*.]

sc-агент

:= [агент над sc-памятью]

С *субъект*

⇒ *семейство подмножеств**:

абстрактный sc-агент

⇒ *пояснение**:

[Под *sc-агентом* понимается конкретный экземпляр (с теоретико-множественной точки зрения - элемент) некоторого *атомарного абстрактного sc-агента*, работающий в какой-либо конкретной интеллектуальной системе.

Таким образом, каждый *sc-агент* - это субъект, способный выполнять некоторый класс однотипных действий либо только над *sc-памятью*, либо над *sc-памятью* и внешней средой (для эффекторных *sc-агентов*). Каждое такое действие инициируется либо состоянием или ситуацией в *sc-памяти*, либо состоянием или ситуацией во внешней среде (для рецепторных *sc-агентов-датчиков*), соответствующей условию инициирования

атомарного абстрактного sc-агента, экземпляром которого является заданный *sc-агент*. В данном случае можно провести аналогию между принципами объектно-ориентированного программирования, рассматривая *атомарный абстрактный sc-агент* как класс, а конкретный *sc-агент* – как экземпляр, конкретную имплементацию этого класса.

Взаимодействие *sc-агентов* осуществляется только через *sc-память*. Как следствие, результатом работы любого *sc-агента* является некоторое изменение состояния *sc-памяти*, т.е. удаление либо генерация каких-либо *sc-элементов*.

В общем случае один *sc-агент* может явно передать управление другому *sc-агенту*, если этот *sc-агент* априори известен. Для этого каждый *sc-агент* в *sc-памяти* имеет обозначающий его *sc-узел*, с которым можно связать конкретную ситуацию в текущем состоянии базы знаний, которую иницируемый *sc-агент* должен обработать.

Однако далеко не всегда легко определить того *sc-агента*, который должен принять управление от заданного *sc-агента*, в связи с чем описанная выше ситуация возникает крайне редко. Более того, иногда условие иницирования *sc-агента* является результатом деятельности непредсказуемой группы *sc-агентов*, равно как и одна и та же конструкция может являться условием иницирования целой группы *sc-агентов*.

При этом общаются через *sc-память* не программы *sc-агентов**, а сами описываемые данными программами *sc-агенты*.

В процессе работы *sc-агент* может сам для себя порождать вспомогательные *sc-элементы*, которые сам же удаляет после завершения акта своей деятельности (это вспомогательные *структуры*, которые используются в качестве "информационных лесов" только в ходе выполнения соответствующего акта деятельности и после завершения этого акта удаляются).]

активный sc-агент

С *sc-агент*
⇒ *пояснение**:

[Под **активным sc-агентом** понимается *sc-агент* *ostis*-системы, который реагирует на события, соответствующие его условию иницирования, и, как следствие, его *первичному условию иницирования**. Не входящие во множество **активных sc-агентов** *sc-агенты* не реагируют ни на какие события в *sc-памяти*.]

ключевые sc-элементы sc-агента*

⇒ *пояснение**:

[Связки отношения **ключевые sc-элементы sc-агента*** связывают между собой *sc-узел*, обозначающий *абстрактный sc-агент* и *sc-узел*, обозначающий множество *sc-элементов*, которые являются ключевыми для данного *абстрактного sc-агента*, то данные *sc-элементы* явно упоминаются в рамках программ, реализующих данный *абстрактный sc-агент*.]

программа sc-агента*

⇒ *пояснение**:

[Связки отношения **программа sc-агента*** связывают между собой *sc-узел*, обозначающий *атомарный абстрактный sc-агент* и *sc-узел*, обозначающий множество программ, реализующих указанный *атомарный абстрактный sc-агент*. В случае *платформенно-независимого абстрактного sc-агента* каждая связка отношения **программа sc-агента*** связывает *sc-узел*, обозначающий указанный *абстрактный sc-агент* с множеством *scr-программ*, описывающих деятельность данного *абстрактного sc-агента*. Данное множество содержит одну *агентную scr-программу*, и произвольное количество (может быть, и ни одной) *scr-программ*, которые необходимы для выполнения указанной *агентной scr-программы*.

В случае *платформенно-зависимого абстрактного sc-агента* каждая связка отношения **программа sc-агента*** связывает *sc-узел*, обозначающий указанный *абстрактный sc-агент* с множеством файлов, содержащих исходные тексты программы на некотором внешнем языке программирования, реализующей деятельность данного *абстрактного sc-агента*.]

первичное условие иницирования*

⇒ *пояснение**:

[Связки отношения **первичное условие иницирования*** связывают между собой *sc-узел*, обозначающий *абстрактный sc-агент* и бинарную ориентированную пару, описывающую *первичное условие иницирования* данного *абстрактного sc-агента*, т.е. такой спецификацию *ситуации* в *sc-памяти*, возникновение которой побуждает *sc-агента* перейти в активное состояние и начать проверку наличия своего полного условия иницирования.

Первым компонентом данной ориентированной пары является знак некоторого класса *элементарных событий в sc-памяти**, например, *событие добавления sc-дуги, выходящей из заданного sc-элемента**.

Вторым компонентом данной ориентированной пары является произвольный в общем случае *sc-элемент*, с которым непосредственно связан указанный тип события в *sc-памяти*, т.е., например, *sc-элемент*, из которого выходит либо в который входит генерируемая либо удаляемая *sc-дуга*, либо *файл*, содержимое которого было изменено.

После того, как в *sc-памяти* происходит некоторое событие, активизируются все *активные sc-агенты*, *первичное условие инициирования** которых соответствует произошедшему событию.]

условие инициирования и результат*

⇒ *пояснение**:

[Связки отношения **условие инициирования и результат*** связывают между собой *sc-узел*, обозначающий *абстрактный sc-агент* и бинарную ориентированную пару, связывающую условие инициирования данного *абстрактного sc-агента* и результаты выполнения данного экземпляров данного *sc-агента* в какой-либо конкретной системе.

Указанную ориентированную пару можно рассматривать как логическую связку импликации, при этом на *sc-переменные*, присутствующие в обеих частях связки, неявно накладывается квантор всеобщности, на *sc-переменные*, присутствующие либо только в посылке, либо только в заключении неявно накладывается квантор существования.

Первым компонентом указанной ориентированной пары является логическая формула, описывающая условие инициирования описываемого *абстрактного sc-агента*, то есть конструкции, наличие которой в *sc-памяти* побуждает *sc-агент* начать работу по изменению состояния *sc-памяти*. Данная логическая формула может быть как атомарной, так и неатомарной, в которой допускается использование любых связок логического языка.

Вторым компонентом указанной ориентированной пары является логическая формула, описывающая возможные результаты выполнения описываемого *абстрактного sc-агента*, то есть описание произведенных им изменений состояния *sc-памяти*. Данная логическая формула может быть как атомарной, так и неатомарной, в которой допускается использование любых связок логического языка.]

описание поведения sc-агента

С *семантическая окрестность*

⇒ *пояснение**:

[**описание поведения sc-агента** представляет собой *семантическую окрестность*, описывающую деятельность *sc-агента* до какой-либо степени детализации, однако такое описание должно быть строгим, полным и однозначно понимаемым. Как любая другая *семантическая окрестность*, **описание поведения sc-агента** может быть протранслировано на какие-либо понятные, общепринятые средства, не требующие специального изучения, например на естественный язык.

Описываемый *абстрактный sc-агент* входит в соответствующее **описание поведения sc-агента** под атрибутом *ключевой sc-элемент'*.]

} /* Завершили Сегмент "Понятие sc-агента и абстрактного sc-агента" */

/* Сегмент *****/

Принципы синхронизации деятельности sc-агентов

⊃=
{

процесс в sc-памяти

⇒ *примечание**:

[Понятия *действие в sc-памяти*, и *процесс в sc-памяти* (информационный процесс, выполняемый агентом в семантической памяти), являются синонимичными, поскольку все процессы, протекающие в *sc-памяти*, являясь осознанными и выполняются каким-либо *sc-агентами*. Тем не менее, когда идет речь о синхронизации выполнения каких-либо преобразований в памяти компьютерной системы, в литературе принято использовать именно термины "процесс", "взаимодействие процессов" Дийкстра.Э.ВзаимодействиеПП-1972ст; Хоар.Ч.ВзаимодейстПП-1989кн, в связи с чем будем использовать этот термин при описании принципов синхронизации деятельности *sc-агентов* при выполнении ими параллельных процессов в *sc-памяти*.]

⇒ *разбиение**:

{• *процесс в sc-памяти, соответствующий платформенно-зависимому sc-агенту*

- *scr-процесс*

}
⇒ разбиение*:

- {
 - *scr-процесс, не являющийся scr-метапроцессом*
 - *scr-метапроцесс*
}

процесс в sc-памяти, соответствующий платформенно-зависимому sc-агенту

⇒ разбиение*:

- {
 - *процесс в sc-памяти, соответствующий платформенно-зависимому sc-агенту и не являющийся действием абстрактной scr-машины*
 - *действие абстрактной scr-машины*
 - ⊃ *действие интерпретации scr-программы*
}

блокировка*

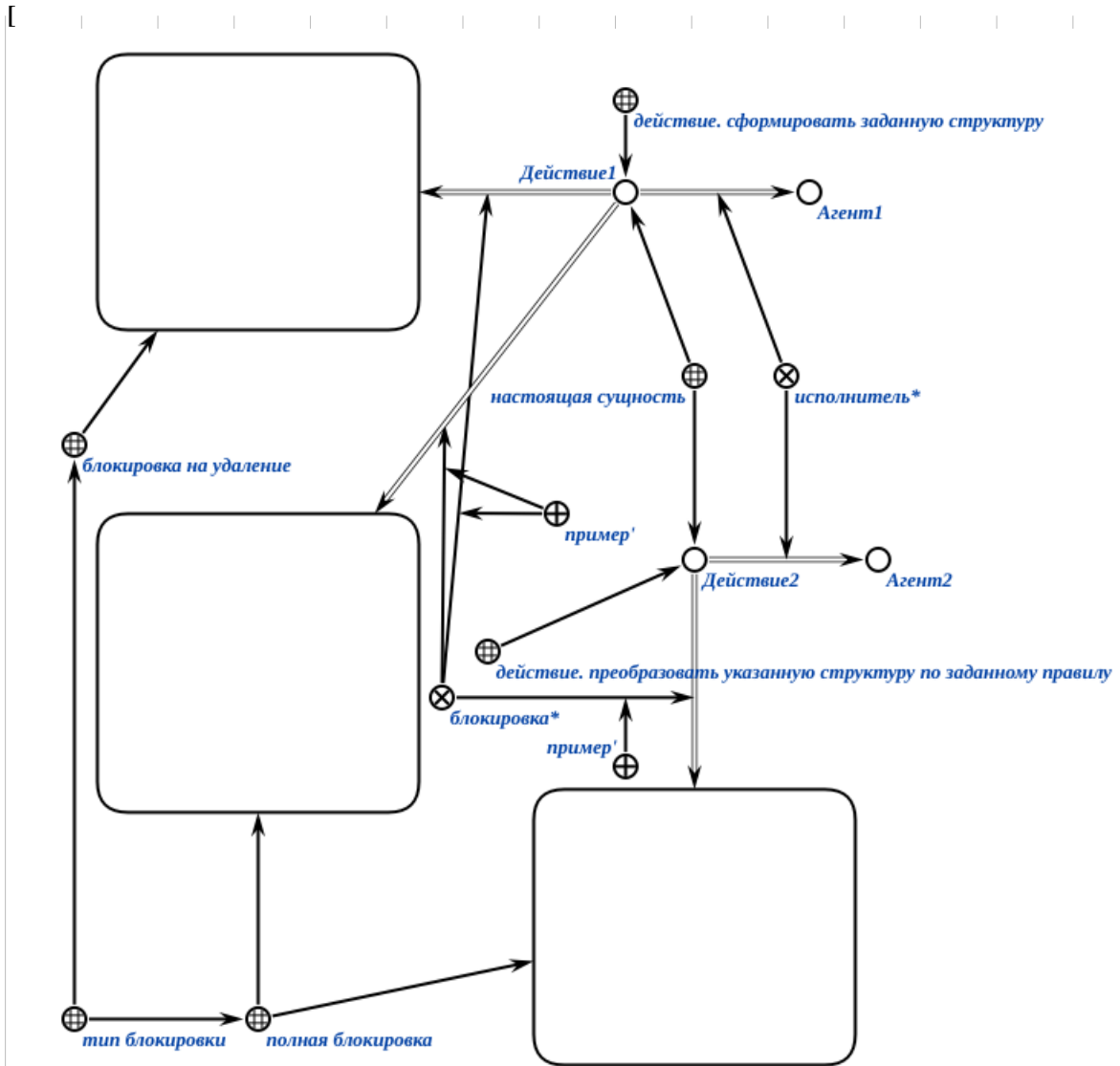
∈ бинарное отношение

⇒ пояснение*:

[Для синхронизации выполнения *процессов в sc-памяти* используется механизм блокировок. Отношение **блокировка*** связывает знаки *действий в sc-памяти* со знаками *структур* (ситуативных), которые содержат элементы, заблокированные на время выполнения данного действия или на какую-то часть этого периода. Каждая такая *структура* принадлежит какому-либо из *типов блокировки*.

Первым компонентом связок отношения **блокировка*** является знак *действия в sc-памяти*, вторым – знак заблокированной *структуры*.]

⇒ описание примера*:



тип блокировки

⇒ пояснение*:

[Множество **тип блокировки** содержит все возможные классы блокировок, т.е. *структуры*, содержащие *sc-элементы*, заблокированные каким-либо *sc-агентом* на время выполнения им некоторого *действия* в *sc-памяти*.]

⊃ *полная блокировка*

⊃ *блокировка на любое изменение*

⊃ *блокировка на удаление*

полная блокировка

⇒ пояснение*:

[Каждая *структура*, принадлежащая множеству **полная блокировка** содержит *sc-элементы*, просмотр и изменение (удаление, добавление инцидентных *sc-коннекторов*, удаление самих *sc-элементов*, изменение содержимого в случае файла) которых запрещены всем *sc-агентам*, кроме собственно *sc-агента*, выполняющего соответствующее данной структуре *действие* в *sc-памяти*, связанное с ней отношением *блокировка**.

Для того, чтобы исключить возможность реализации *sc-агентов*, которые могут внести изменения в конструкции, описывающие блокировки других *sc-агентов*, все элементы этих конструкций, в том числе, сам знак *структуры*, содержащей заблокированные *sc-элементы* (принадлежащей как множеству **полная блокировка**, так и любому другому *типу блокировки*) и связи отношения *блокировка**, связывающие эту *структуру* и конкретное *действие* в *sc-памяти*, добавляются в **полную блокировку**, соответствующую данному *действию* в *sc-памяти*. Таким образом, каждой **полной блокировке** соответствует петля принадлежности, связывающая ее знак с самой собой.]

блокировка на любое изменение⇒ *пояснение**:

[Каждая *структура*, принадлежащая множеству **блокировка на любое изменение** содержит *sc-элементы*, изменение (физическое удаление, добавление инцидентных *sc-коннекторов*, физическое удаление самих *sc-элементов*, изменение содержимого в случае файла) которых запрещено всем *sc-агентам*, кроме собственно *sc-агента*, выполняющего соответствующее данной структуре *действие в sc-памяти*, связанное с ней отношением *блокировка**. Однако не запрещен просмотр (чтение) этих *sc-элементов* любым *sc-агентом*.]

блокировка на удаление⇒ *пояснение**:

[Каждая *структура*, принадлежащая множеству **блокировка на удаление** содержит *sc-элементы*, удаление которых запрещено всем *sc-агентам*, кроме собственно *sc-агента*, выполняющего соответствующее данной структуре *действие в sc-памяти*, связанное с ней отношением *блокировка**. Однако не запрещен просмотр (чтение) этих *sc-элементов* любым *sc-агентом*, добавление инцидентных *sc-коннекторов*.]

блокировка*⇒ *принципы работы**:

- { • [в каждый момент времени одному процессу в *sc-памяти* может соответствовать только одна блокировка каждого типа]
- [в каждый момент времени одному процессу в *sc-памяти* может соответствовать только одна блокировка, установленная на некоторый конкретный *sc-элемент*]
- [при завершении выполнения любого процесса в *sc-памяти* все установленные им блокировки автоматически снимаются]
- [для повышения эффективности работы системы в целом каждый процесс должен в каждый момент времени блокировать минимально необходимое множество *sc-элементов*, снимая блокировку с каждого *sc-элемента* сразу же, как это становится возможным (безопасным)]
- [В случае когда в рамках *процесса в sc-памяти* явно выделяются более частные подпроцессы (при помощи отношений *темпоральная часть**, *поддействие**, *декомпозиция действия** и т. д.), то каждый такой подпроцесс с точки зрения синхронизации выполнения рассматривается как самостоятельный процесс, которому в соответствие могут быть поставлены все необходимые блокировки.]

⇒ *детализация**:

- [все дочерние процессы в *sc-памяти* имеют доступ к блокировкам родительского процесса так же, как если бы это были блокировки соответствующие каждому из таких дочерних процессов]
- [в свою очередь, родительский процесс не имеет какого-либо привилегированного доступа к *sc-элементам*, заблокированным дочерними процессами, и работает с ними так же, как любой другой процесс в *sc-памяти*. Исключение составляют *sc-элементы*, обозначающие сами дочерние процессы, поскольку родительский процесс должен иметь возможность управления дочерним, например, приостановки или прекращения их выполнения]
- [все дочерние процессы по отношению друг к другу работают так же, как и по отношению к любым другим процессам]
- [в случае, когда родительский процесс приостанавливает выполнение (становится *отложенным действием*), все его дочерние процессы также приостанавливают выполнение. В свою очередь, приостановка одного из дочерних процессов в общем случае не инициирует явно остановку всего родительского процесса и соответственно других дочерних.]

}

полная блокировка⇒ *принципы работы**:

- { • [если *sc-элемент*, инцидентный некоторому *sc-коннектору*, попадает в какую-либо полную блокировку, то сам этот *sc-коннектор* по умолчанию также считается заблокированным этой же блокировкой. Обратное в общем случае неверно, т. к. часть *sc-коннекторов*, инцидентных некоторому *sc-элементу*, может быть полностью заблокирована, при этом сам этот элемент заблокирован не будет. Такая ситуация типична, например, для *sc-узлов*, обозначающих классы понятий]
-

[каждый процесс в sc-памяти может свободно изменять или удалять любые sc-элементы, попадающие в полную блокировку, соответствующую этому процессу.]

}

⇒

*примечание**:

[Принципы работы с *полными блокировками*, с одной стороны, наиболее просты, поскольку все процессы, кроме установившего такую блокировку, не имеют доступа к заблокированным sc-элементам и конфликты возникнуть не могут. С другой стороны, частое использование блокировок такого типа может привести к тому, что система не сможет использовать в полной мере имеющиеся у нее знания и давать неполные или даже некорректные ответы на поставленные вопросы.]

блокировка на любое изменение

⇒

*принципы работы**:

{

- [на один и тот же sc-элемент в один момент времени может быть установлена только одна блокировка одного типа, но разные процессы могут одновременно установить на один и тот же элемент блокировки двух разных типов. Это касается случая, когда первый процесс установил на некоторый sc-элемент блокировку на удаление, а второй процесс затем устанавливает блокировку на любое изменение. В других случаях возникает конфликт блокировок]
- [установка блокировки любого типа также считается изменением, таким образом, если на некоторый sc-элемент была установлена блокировка на любое изменение, то другой процесс не сможет установить на этот же sc-элемент блокировку любого типа, пока первый процесс не снимет свою]
- [если блокировка на удаление устанавливается на некоторый sc-коннектор, то по умолчанию та же блокировка устанавливается на инцидентные этому sc-коннектору sc-элементы, поскольку удаление этих элементов приведет к удалению этого коннектора.]

}

←

*принципы работы**:

блокировка на удаление

процесс в sc-памяти

:= [действие в sc-памяти]

⇒

*разбиение**:

Классификация процессов в sc-памяти с точки зрения синхронизации их выполнения

=

- *действие поиска sc-элементов*
- *действие генерации sc-элементов*
- *действие удаления sc-элементов*
- *действие установки блокировки некоторого типа на некоторый sc-элемент*
- *действие снятия блокировки с некоторого sc-элемента*

}

транзакция в sc-памяти

⇒

*пояснение**:

[В некоторых случаях для того, чтобы обеспечить синхронизацию, необходимо объединять несколько элементарных действий над sc-памятью в одно неделимое действие (*транзакцию в sc-памяти*), для которого гарантируется, что ни один сторонний процесс не сможет прочитать или изменить участвующие в этом действии sc-элементы, пока действие не завершится. При этом, в отличие от ситуации с полной блокировкой, процесс, пытающийся получить доступ к таким элементам, не продолжает выполнение так, как если бы этих элементов просто не было в sc-памяти, а ожидает завершения транзакции, после чего может выполнять с данными элементами любые действия согласно общим принципам синхронизации процессов. Проблема обеспечения транзакций не может быть решена на уровне SC-кода и требует реализации таких неделимых действий на уровне *платформы интерпретации sc-моделей*.]

действие поиска sc-элементов

⇒

*пояснение**:

[В случае осуществления поиска все найденные и сохраненные в рамках какого-либо процесса sc-элементы попадают в соответствующую данному процессу *блокировку на любое изменение*. Таким образом, гарантируется целостность фрагмента базы знаний, с которым работает некоторый процесс в sc-памяти. При этом поиск и автоматическая установка такой блокировки должны быть реализованы как *транзакция в sc-памяти*.

Такой подход также позволяет избежать ситуации, когда один процесс заблокировал некоторый sc-элемент на любое изменение, а второй процесс пытается сгенерировать или удалить *sc-коннектор*, инцидентный

данному *sc-элементу*. В таком случае второй процесс должен будет предварительно найти и заблокировать указанный *sc-элемент* на любое изменение, что вызовет конфликт блокировок (*взаимоблокировку**.)]

действие генерации *sc-элементов*

⇒ *пояснение**:

[В случае генерации любого *sc-элемента* в рамках некоторого процесса он автоматически попадает в полную блокировку, соответствующую данному процессу. При этом генерация и автоматическая установка такой блокировки должны быть реализованы как *транзакция в sc-памяти*. При необходимости сгенерированные элементы могут быть удалены (т. е. их временное существование вообще никак не отразится на деятельности других процессов) или разблокированы в случае, когда сгенерирована информация, которая может иметь некоторую ценность в дальнейшем.]

действие установки блокировки некоторого типа на некоторый *sc-элемент*

⇒ *пояснение**:

[В случае если какой-либо процесс пытается установить блокировку любого типа на какой-либо *sc-элемент*, уже заблокированный каким-либо другим процессом, то, с одной стороны, блокировка не может быть установлена, пока другой процесс не разблокирует указанный *sc-элемент*; с другой стороны, для того чтобы обеспечить возможность поиска и устранения *взаимоблокировок*, необходимо явно указывать тот факт, что какой-либо процесс хочет получить доступ к какому-либо заблокированному другим процессом *sc-элементу*. Для того чтобы иметь возможность указать, какие процессы пытаются заблокировать уже заблокированный *sc-элемент*, предлагается наряду с отношением *блокировка** использовать отношение *планируемая блокировка**, полностью аналогичное отношению *блокировка**.

Описанный механизм регулирует также и процессы поиска, поскольку поиск и сохранение некоторого *sc-элемента* предполагает установку *блокировки на любое изменение*. Кроме того, следует учитывать, что на один *sc-элемент* *блокировка на любое изменение* может быть установлена после *блокировки на удаление*, соответствующей другому процессу. В этом случае использовать отношение *планируемые блокировки** нет необходимости.]

⇒ *примечание**:

[Действие проверки наличия на некотором *sc-элементе* блокировки и в зависимости от результата проверки, установки блокировки или планируемой блокировки (с указанием приоритета при необходимости) должно быть реализовано как транзакция.]

планируемая блокировка*

С *блокировка**

⇒ *пояснение**:

[Процесс, которому в соответствие поставлена *планируемая блокировка**, приостанавливает выполнение до тех пор, пока уже установленные блокировки не будут сняты, после чего *планируемая блокировка** становится реальной *блокировкой** и процесс продолжает выполнение в соответствии с общими правилами.]

приоритет блокировки*

⇒ *область определения**:

*планируемая блокировка**

⇒ *пояснение**:

[В случае, когда на один и тот же *sc-элемент* планируют установить блокировку сразу несколько процессов, используется отношение *приоритет блокировки**, связывающее между собой пары отношения *планируемая блокировка**. Как правило, приоритет блокировки определяется тем, какой из процессов раньше попытался установить блокировку на рассматриваемый *sc-элемент*, хотя в общем случае приоритет может устанавливаться или меняться в зависимости от дополнительных критериев.]

действие удаления *sc-элементов*

⇒ *примечание**:

[В случае попытки удаления некоторого *sc-элемента* некоторым процессом удаление может быть осуществлено только в случае, когда на данный *sc-элемент* не установлена (и не планируется) ни одна блокировка каким-либо другим процессом.

В других случаях необходимо обеспечить корректное завершение выполнения всех процессов, работающих с данным *sc-элементом*, и только потом удалить его физически.

Для реализации такой возможности каждому процессу в соответствии может быть поставлено множество удаляемых данным процессом *sc-элементов*.]

⇒ *примечание**:

[Действие проверки наличия блокировок или планируемых блокировок на удаляемый sc-элемент и собственно его удаление или добавление во множество удаляемых sc-элементов для соответствующего процесса должно быть реализовано как транзакция.]

удаляемые sc-элементы*

- ⇒ *первый домен**:
 процесс в sc-памяти
 ⇒ *пояснение**:

[Sc-элементы, попавшие во множество удаляемых sc-элементов некоторого процесса в sc-памяти, доступны процессам, уже установившим (или планирующим установить) на эти sc-элементы блокировки ранее (до попытки его удаления), а для всех остальных процессов эти sc-элементы уже считаются удаленными. Процесс, пытающийся удалить sc-элемент, приостанавливает свое выполнение до того момента, пока все заблокировавшие и планирующие заблокировать данный sc-элемент процессы не разблокируют его. В общем случае один sc-элемент может входить во множества удаляемых элементов одновременно для нескольких процессов, в этом случае все такие процессы одновременно продолжают выполнение после снятия с этого sc-элемента всех блокировок. Если удаление пытается осуществить один из процессов, уже установивший на указанный sc-элемент блокировку, то алгоритм действий остается прежним – sc-элемент добавляется во множество удаляемых данным процессом sc-элементов, и будет физически удален, как только все остальные процессы, установившие на данный sc-элемент блокировки, снимут их.]

действие снятия блокировки с некоторого sc-элемента

- ⇒ *алгоритм выполнения**:

- {
 - [если на данный sc-элемент установлена одна или несколько *планируемых блокировок**, то первая из них по приоритету (или единственная) становится *блокировкой**, соответствующий ей процесс продолжает выполнение (становится настоящей сущностью); связка отношения приоритет выполнения, соответствовавшая удаленной связке отношения *планируемая блокировка** также удаляется, т. е. приоритет смещается на одну позицию]
 - [если *планируемых блокировок**, установленных на данный sc-элемент, нет, но он попадает во множество удаляемых sc-элементов для одного или нескольких процессов, то рассматриваемый sc-элемент физически удаляется, а приостановленные до его удаления процессы продолжают свое выполнение (становятся настоящими сущностями)]
 - [если на данный sc-элемент не установлены планируемые блокировки и он не входит во множество удаляемых для какого-либо процесса, то блокировка просто снимается без каких-либо дополнительных изменений.]

транзакция в sc-памяти

- ⇒ *разбиение**:

- {
 - *поиск некоторой конструкции в sc-памяти и автоматическая установка блокировки на любое изменение на найденные sc-элементы*
 - *генерация некоторого sc-элемента и автоматическая установка на него полной блокировки*
 - *проверка наличия на некотором sc-элементе блокировки и в зависимости от результата проверки установка блокировки или планируемой блокировки*
 - *проверка наличия блокировок или планируемых блокировок на удаляемый sc-элемент и собственно его удаление или добавление во множество удаляемых sc-элементов для соответствующего процесса*
 - *снятие блокировки с заданного sc-элемента и при необходимости установка первой по приоритету планируемой блокировки или удаление данного sc-элемента, если он входит во множество удаляемых sc-элементов для некоторого процесса*
 - *поиск подпроцессов процесса и добавление их во множество отложенных действий в случае добавления самого процесса в данное множество*
 - *поиск подпроцессов процесса и удаление их из множества отложенных действий в случае удаления самого процесса из данного множества*

абстрактный программный sc-агент

- ⇒ *примечание**:

[При реализации *абстрактных программных sc-агентов* на языке SCP, соблюдение всех принципов синхронизации соответствующих этим sc-агентам процессов обеспечивается на уровне *sc-агентов интерпретации scr-программ*, т. е. средствами *платформы интерпретации sc-моделей*. При реализации *абстрактных программных sc-агентов* на уровне платформы, соблюдение всех принципов синхронизации возлагается,

во-первых, непосредственно на разработчика агентов, во-вторых, – на разработчика платформы. Так, например, платформа может предоставлять доступ к хранимым в *sc*-памяти элементам через некоторый программный интерфейс, уже учитывающий принципы работы с блокировками, что избавит разработчика агентов от необходимости учитывать все эти принципы вручную.]

⇒ *принципы работы**:

- { • [в результате появления в *sc*-памяти некоторой конструкции, удовлетворяющей условию инициирования какого-либо *abстрактного sc-агента*, реализованного при помощи *Языка SCP*, в *sc-памяти* генерируется и иницируется *scp-процесс*. В качестве шаблона для генерации используется *агентная scp-программа*, соответствующая данному *абстрактному sc-агенту*.]
- [каждый такой *scp-процесс*, соответствующий некоторой *агентной scp-программе*, может быть связан с набором структур, описывающих блокировки различных типов. Таким образом, синхронизация взаимодействия параллельно выполняемых *scp-процессов* осуществляется так же, как и в случае любых других *действий в sc-памяти*.]
- [несмотря на то что каждый *scp-оператор* представляет собой атомарное действие в *sc*-памяти, являющееся поддействием в рамках всего *scp-процесса*, блокировки, соответствующие одному оператору, не вводятся, чтобы избежать громоздкости и избытка дополнительных системных конструкций, создаваемых при выполнении некоторого *scp-процесса*. Вместо этого используются блокировки, общие для всего *scp-процесса*. Таким образом, *агенты интерпретации scp-программ* работают только с учетом блокировок, общих для всего интерпретируемого *scp-процесса*.]
- [процессы, описывающие деятельность агентов интерпретации *scp-программ*, как правило, не создаются, следовательно, и не вводятся соответствующие им блокировки. Поскольку такие агенты работают с уникальным *scp-процессом* и их число ограничено и известно, то использование блокировок для их синхронизации не требуется.]
- [в случае приостановки *scp-процесса* (добавления его во множество *отложенных действий*) в соответствии с общими правилами синхронизации все его дочерние процессы также должны быть приостановлены. В связи с этим все *scp-операторы*, которые в этот момент являются *настоящими сущностями*, становятся *отложенными действиями*.]
- [во избежание нежелательных изменений в самом теле *scp-процесса*, вся конструкция, сгенерированная на основе некоторой *scp-программы* (весь *sc-текст*, описывающий декомпозицию *scp-процесса* на *scp-операторы*), должна быть добавлена в *полную блокировку*, соответствующую данному *scp-процессу*.]
- [при необходимости разблокировать или заблокировать некоторую конструкцию каким-либо типом блокировки используются соответствующие *scp-операторы* класса *scp-оператор управления блокировками*.]
- [после завершения выполнения некоторого *scp-процесса* его текст, как правило, удаляется из *sc-памяти*, а все заблокированные конструкции освобождаются (разрушаются знаки структур, обозначающих блокировки).]
- [как правило, частный *класс действий*, соответствующий конкретной *scp-программе*, явно не вводится, а используется более общий класс *scp-процесс*, за исключением тех случаев, когда введение специального *класса действий* необходимо по каким-либо другим соображениям.]

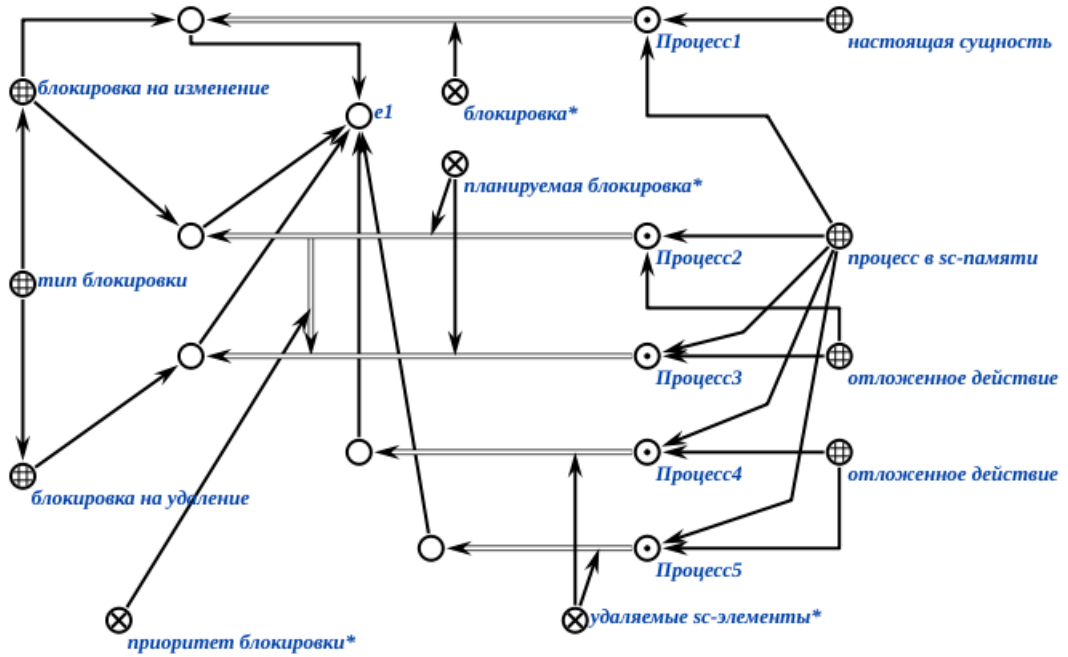
блокировка*

⇒ *примечание**:

[В общем случае весь механизм блокировок может описываться как на уровне *SC*-кода (для повышения уровня платформенной независимости), так и при необходимости может быть реализован на уровне *платформы интерпретации sc-моделей*, например для повышения производительности. Для этого каждому выполняемому в *sc*-памяти процессу на нижнем уровне может быть поставлена в соответствие некая уникальная таблица, в каждый момент времени содержащая перечень заблокированных элементов с указанием типа блокировки.]

⇒ *пример применения**:

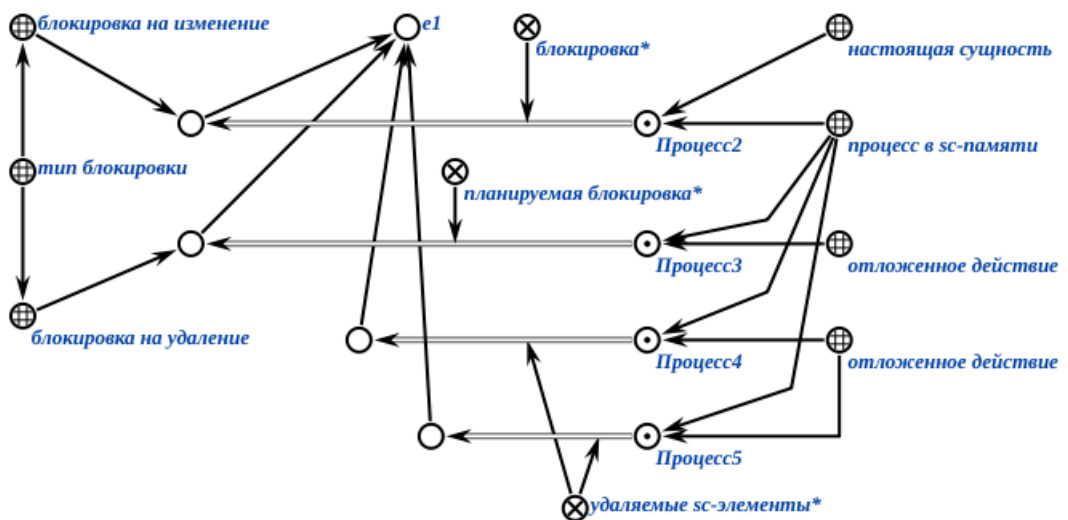
{ •



⇒

пояснение*:

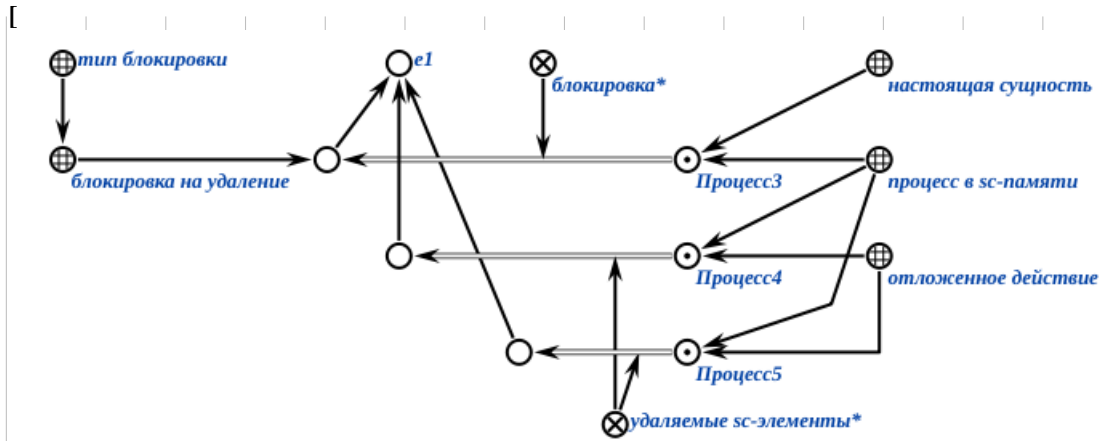
[В данном примере *Процесс1* непосредственно работает с *sc-элементом e1*, *Процесс2* и *Процесс3* планируют установить блокировку на любое изменение и блокировку на удаление соответственно, причем *Процесс2* попытался установить свою блокировку раньше, чем *Процесс3*, поэтому согласно направлению связки отношения *приоритет блокировки**, его блокировка будет установлена раньше. *Процесс4* и *Процесс5* ожидают снятия всех блокировок и планируемых блокировок, после чего *e1* будет удален и *Процесс1* и *Процесс2* продолжат свое выполнение. Никакие другие планируемые блокировки установлены быть уже не могут, поскольку *e1* попал во множество удаляемых *sc-элементов* как минимум одного процесса и, в соответствии с изложенным выше правилами, все остальные процессы кроме *Процесс1-Процесс5*, уже не смогут получить доступ к этому *sc-элементу*. Выполняемый процесс принадлежит множеству *настоящая сущность*, приостановленные – множеству *отложенное действие*.]



⇒

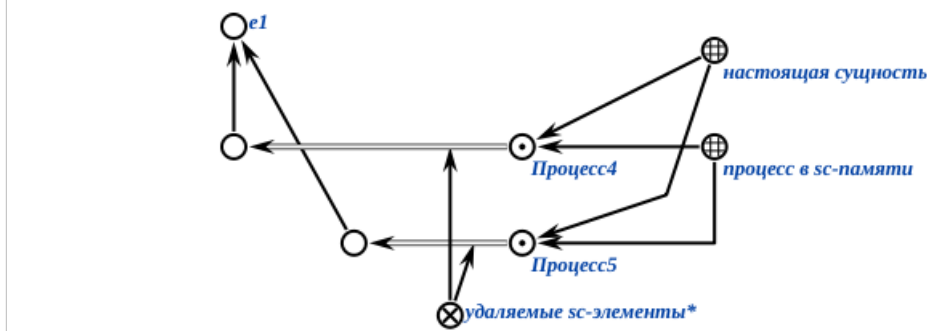
пояснение*:

[После того как *Процесс1* разблокировал *sc-элемент e1*, этот элемент будет заблокирован *Процессом2*, и *Процесс2* продолжит выполнение. *Планируемая блокировка**, установленная *Процессом2*, становится обычной *блокировкой**.]



⇒ *пояснение**:
 [После того как *Процесс2* разблокировал *sc-элемент e1*, этот элемент будет заблокирован *Процессом3*, и *Процесс3* продолжит выполнение.]

•



⇒ *пояснение**:
 [Когда все процессы снимут блокировки с *sc-элемента e1*, он может быть физически удален и *Процесс4* и *Процесс5* продолжат выполнение.]

)

взаимоблокировка*

⇒ *пояснение**:

[В зависимости от конкретных *типов блокировок* установленных параллельно выполняемыми процессами на некоторые *sc-элементы* и того, какие конкретно действия с этими *sc-элементами* предполагается выполнить далее в рамках выполнения этих процессов, возможны ситуации взаимоблокировки, когда каждый из указанных процессов будет ожидать снятия блокировки вторым процессом с нужного *sc-элемента*, не снимая при этом установленной им самим блокировки с *sc-элемента*, доступ к которому необходим второму процессу.]

В случае когда хотя бы одна из блокировок является *полной блокировкой*, ситуация взаимоблокировки возникнуть не может, поскольку *sc-элементы*, попавшие в *полную блокировку* некоторого *scr-процесса*, не доступны другим *scr-процессам* даже для чтения и, таким образом, остальные *scr-процессы* будут работать так, как будто заблокированные *sc-элементы* просто отсутствуют в текущем состоянии *sc-памяти*.

В случаях, когда ни одна из установленных блокировок не является *полной блокировкой*, возможно появление взаимоблокировок.]

⇒ *примечание**:

[Устранение *взаимоблокировки* невозможно без вмешательства специализированного *sc-метаагента*, который имеет право игнорировать блокировки, установленные другими процессами.]

В общем случае проблема конкретной взаимоблокировки может быть решена путем выполнения специализированным *sc-метаагентом* следующих шагов:

- откат нескольких операций, выполненных одним из участвующих в взаимоблокировке процессов настолько шагов назад, насколько это необходимо для того, чтобы второй процесс получил доступ к необходимым *sc-элементам* и смог продолжить выполнение;

- ожидание выполнения второго процесса вплоть до завершения или до снятия им всех блокировок с *sc-элементов*, доступ к которым необходимо получить первому процессу;
- повторное выполнение в рамках первого процесса отмененных операций и продолжение его выполнения, но уже с учетом изменений в памяти, внесенных вторым процессом.

]

sc-метаагент

⇒ *пояснение**:

[Для *sc-метаагентов* все *sc-элементы*, в том числе описывающие блокировки, планируемые блокировки и т. д. полностью эквивалентны между собой с точки зрения доступа к ним, т. е. любой *sc-метаагент* имеет доступ к любым *sc-элементам*, даже попавшим в полную блокировку для какого-либо другого процесса. Это необходимо для того, чтобы *sc-метаагенты* смогли выявлять и устранять различные проблемы, например, описанную выше проблему взаимоблокировки.

Таким образом, проблема синхронизации деятельности *sc-метаагентов* требует введения дополнительных правил.

Указанную проблему разделим на две более частные:

- обеспечение синхронизации деятельности *sc-метаагентов* между собой;
- обеспечение синхронизации деятельности *sc-метаагентов* и *программных sc-агентов*.

Первую проблему предлагается решить за счет запрета параллельного выполнения *sc-метаагентов*. Таким образом, в каждый момент времени в рамках одной *ostis-системы* может существовать только один процесс, соответствующий *sc-метаагенту* и являющийся *настоящей сущностью*.

Вторую проблему предлагается решить за счет введения дополнительных привилегий для *sc-метаагентов* при обращении к какому-либо *sc-элементу*. Для этого достаточно одного правила:

Если некоторый *sc-элемент* стал использоваться в рамках процесса, соответствующего *sc-метаагенту* (например, стал элементом хотя бы одного *scr-оператора*, входящего в данный процесс), то все процессы, в блокировки соответствующие которым попадает указанный *sc-элемент*, становятся отложенными действиями (приостанавливают выполнение). Как только указанный *sc-элемент* перестает использоваться в рамках процесса, соответствующего *sc-метаагенту*, все приостановленные по этой причине процессы продолжают выполнение.

Рассмотренные ограничения не ухудшают производительность *ostis-системы* существенно, поскольку *sc-метаагенты* предназначены для решения достаточно узкого класса задач, которые, как показал опыт практической разработки прототипов различных *ostis-систем*, возникают достаточно редко.]

⇒ *примечание**:

[Стоит отметить, что возможна ситуация, при которой выполнение некоторого процесса в *sc-памяти* прервано по причине возникновения какой-либо ошибки. В таком случае существует вероятность того, что блокировка, установленная данным процессом не будет снята до тех пор, пока этого не сделает *sc-метаагент*, обнаруживший подобную ситуацию. Однако указанная проблема на уровне *sc-модели* может быть решена лишь частично, для случаев, когда ошибка возникает при интерпретации *scr-программы*, отслеживается *scr-интерпретатором* и в памяти формируется соответствующая конструкция, сообщающая о проблеме *sc-метаагенту*. Случаи, когда возникла ошибка на уровне *scr-интерпретатора* или *sc-хранилища*, должны рассматриваться на уровне платформы интерпретации *sc-моделей*.]

} /* Завершили Сегмент “Принципы синхронизации деятельности *sc-агентов*” */

} /* Завершили Раздел “Предметная область и онтология действий, задач, планов, протоколов и методов, реализуемых *ostis-системой*, а также внутренних агентов, выполняющих эти действия” */

```

/* Раздел *****/
Предметная область и онтология Базового языка программирования ostis-систем
⊃=
{
⇒ дочерний раздел*:
  • Предметная область и онтология денотационной семантики Базового языка программирования ostis-систем
  • Предметная область и онтология операционной семантики Базового языка программирования ostis-систем

Предметная область Базового языка программирования ostis-систем (языка SCP – Semantic Code Programming)
:= [Предметная область Базового языка программирования ostis-систем]
:= [Предметная область Языка SCP]
⇒ примечание*:
  [В данную предметную область включаются все тексты программ Языка SCP. В ней исследуется типология операторов этих программ и заданные на них отношения.]
∈ предметная область
∋ максимальный класс объектов исследования':
  scr-программа
∋ класс объектов исследования':
  • агентная scr-программа
  • scr-процесс
  • scr-оператор
  • атомарный тип scr-оператора
∋ исследуемое отношение':
  • начальный оператор'
  • параметр scr-программы'
  • in-параметр'
  • out-параметр'
  • scr-операнд'

Язык SCP
:= часто используемый sc-идентификатор*:
  [scr-программа]
⇒ пояснение*:
  [В качестве базового языка для описания программ обработки текстов SC-кода предлагается Язык SCP.
  Язык SCP – это графовый язык процедурного программирования, предназначенный для эффективной обработки sc-текстов. Язык SCP является языком параллельного асинхронного программирования.
  Языком представления данных для текстов Языка SCP (scr-программ) является SC-код и, соответственно, любые варианты его внешнего представления. Язык SCP сам построен на основе SC-кода, вследствие чего scr-программы сами по себе могут входить в состав обрабатываемых данных для scr-программ, в т.ч. по отношению к самим себе. Таким образом, язык SCP предоставляет возможность построения реконфигурируемых программ. Однако для обеспечения возможности реконфигурирования программы непосредственно в процессе ее интерпретации необходимо на уровне интерпретатора Языка SCP (Абстрактной scr-машины) обеспечить уникальность каждой исполняемой копии исходной программы. Такую исполняемую копию, сгенерированную на основе scr-программы, будем называть scr-процессом. Включение знака некоторого действия в sc-памяти во множество scr-процессов гарантирует тот факт, что в декомпозиции данного действия будут присутствовать только знаки элементарных действий (scr-операторов), которые может интерпретировать реализация Абстрактной scr-машины (интерпретатора scr-программ).
  Язык SCP рассматривается как ассемблер для семантического компьютера.]

Базовая модель обработки sc-текстов
⇐ объединение*:
  {
  • Предметная область Базового языка программирования ostis-систем
  • Модель Абстрактной scr-машины
  }

```

- ⇒ *особенности**:
- { • [Тексты программ *Языка SCP* записываются при помощи тех же унифицированных семантических сетей, что и обрабатываемая информация, таким образом, можно сказать, что *Синтаксис языка SCP* на базовом уровне совпадает с *Синтаксисом SC-кода*.]
 - [Подход к интерпретации *scp-программ* предполагает создание при каждом вызове *scp-программы* уникального *scp-процесса*.]
- }
- ⇒ *достоинства**:
- { • [Одновременно в общей памяти могут выполняться несколько независимых *sc-агентов*, при этом разные копии *sc-агентов* могут выполняться на разных серверах, за счет распределенной реализации интерпретатора *sc-моделей* (*платформы реализации sc-моделей компьютерных систем*). Более того, *Язык SCP* позволяет осуществлять параллельные асинхронные вызовы подпрограмм с последующей синхронизацией, и даже параллельно выполнять операторы в рамках одной *scp-программы*.]
 - [Перенос *sc-агента* из одной системы в другую заключается в простом переносе фрагмента базы знаний, без каких-либо дополнительных операций, зависящих от платформы интерпретации.]
 - [Тот факт, что спецификации *sc-агентов* и их программы могут быть записаны на том же языке, что и обрабатываемые знания, существенно сокращает перечень специализированных средств, предназначенных для проектирования машин обработки знаний, и упрощает их разработку за счет использования более универсальных компонентов.]
 - [Тот факт, что для интерпретации *scp-программы* создается соответствующий ей уникальный *scp-процесс*, позволяет по возможности оптимизировать план выполнения перед его реализацией и даже непосредственно в процессе выполнения без потенциальной опасности испортить общий универсальный алгоритм всей программы. Более того, такой подход к проектированию и интерпретации программ позволяет говорить о возможности создания самореконфигурируемых программ.]
- }

Абстрактная scp-машина

⇒ *модель**:

Модель Абстрактной scp-машины

⇒ *примечание**:

[*Абстрактная scp-машина* представляет собой интерпретатор *scp-программ*, который должен являться частью *платформы интерпретации sc-моделей компьютерных систем* (хотя в общем случае могут существовать варианты платформы, не содержащие такого интерпретатора, что, однако, не позволит использовать достоинства предлагаемой базовой модели)]

scp-программа

С программа в sc-памяти

⇒ *пояснение**:

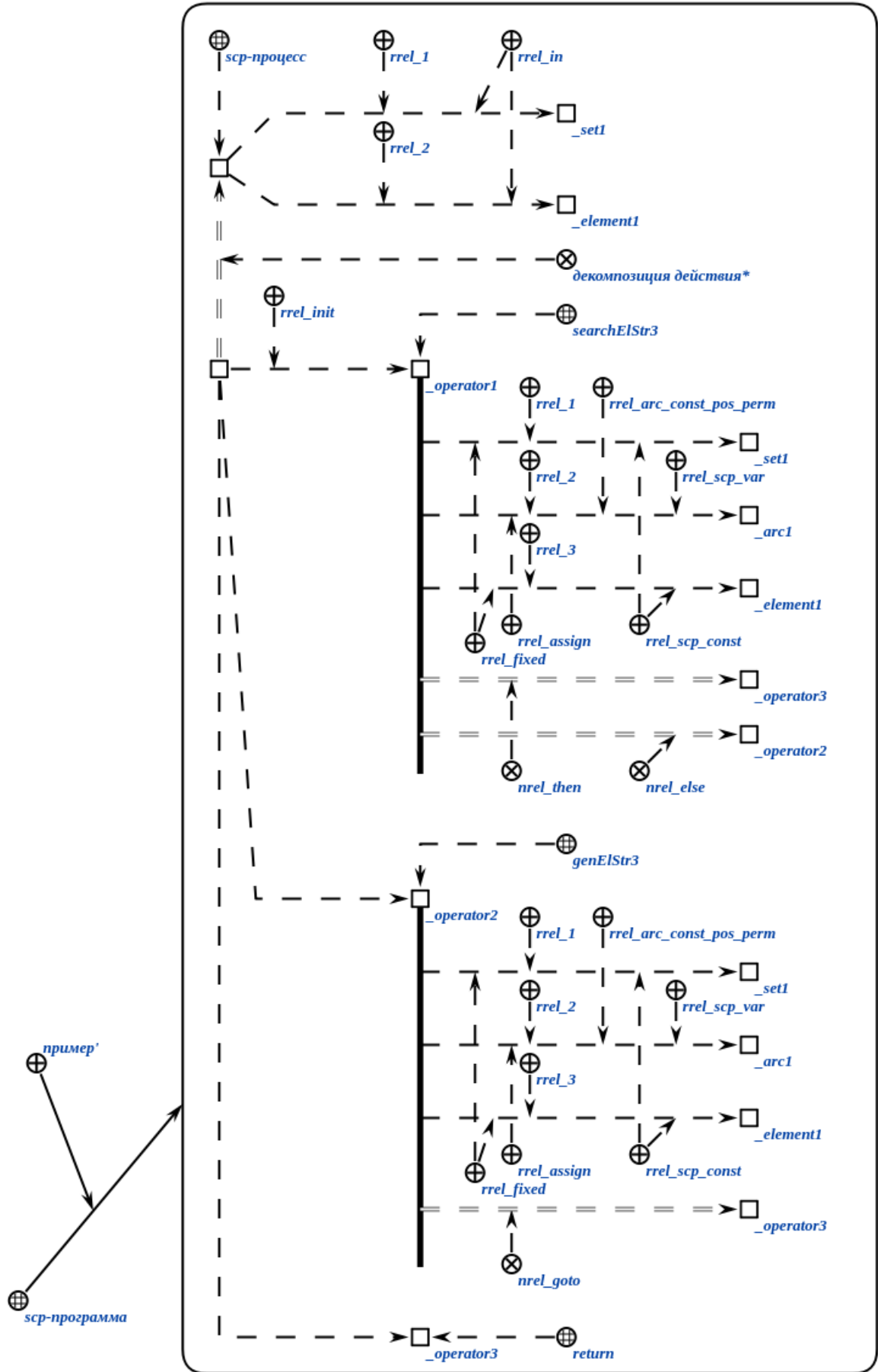
[Каждая *scp-программа* представляет собой *обобщенную структуру*, описывающую один из вариантов декомпозиции действий некоторого класса, выполняемых в *sc-памяти*. Знак *sc-переменной*, соответствующей конкретному декомпозируемому действию является в рамках *scp-программы* *ключевым sc-элементом*. Также явно указывается принадлежность данного знака множеству *scp-процессов*.

Принадлежность некоторого действия множеству *scp-процессов* гарантирует тот факт, что в декомпозиции данного действия будут присутствовать только знаки элементарных действий (*scp-операторов*), которые может интерпретировать реализация абстрактной *scp-машины*.

Таким образом, каждая *scp-программа* описывает в обобщенном виде декомпозицию некоторого *scp-процесса* на взаимосвязанные *scp-операторы*, с указанием, при их наличии, аргументов для данного *scp-процесса*.

По сути каждая *scp-программа* представляет собой описание последовательности элементарных операций, которые необходимо выполнить над семантической сетью, чтобы выполнить более сложное действие некоторого класса.]

⇒ *описание примера**:



\Rightarrow *пояснение*:*

[В приведенном примере показана *scp-программа*, состоящая из трех *scp-операторов*. Данная программа проверяет, содержится ли в заданном множестве (первый параметр) заданный элемент (второй параметр), и, если нет, то добавляет его в это множество.]

```

⇒ пример':
{
  _scp_process
  _∈ scp-процесс
  _∋ 1':: in-параметр':
    _set1
  _∋ 2':: in-параметр':
    _element1
  _⇐ декомпозиция действия*::
    ...
  _∋ 1'::
    _operator1
    _∈ searchElStr3
    _∋ 1':: scp-операнд с заданным значением':: scp-константа'::
      _set1
    _∋ 2':: scp-операнд со свободным значением':: scp-переменная':: sc-дуга основного
      вида'::
        _arc1
    _∋ 3':: scp-операнд с заданным значением':: scp-константа'::
      _element1
    _⇒ последовательность действий при отрицательном результате*::
      _operator2
    _⇒ последовательность действий при положительном результате*::
      _operator3
  _∋ _operator2
    _∈ genElStr3
    _∋ 1':: scp-операнд с заданным значением':: scp-константа'::
      _set1
    _∋ 2':: scp-операнд со свободным значением':: scp-переменная':: sc-дуга основного
      вида'::
        _arc1
    _∋ 3':: scp-операнд с заданным значением':: scp-константа'::
      _element1
    _⇒ следующий оператор*::
      _operator3
  _∋ _operator3
    _∈ return
}

```

агентная scp-программа

С *scp-программа*
 ⇒ *пояснение**:

[**агентные scp-программы** представляют собой частный случай *scp-программ* вообще, однако заслуживают отдельного рассмотрения, поскольку используются наиболее часто. *Scp-программы* данного класса представляют собой реализации программ агентов обработки знаний, и имеют жестко фиксированный набор параметров. Каждая такая программа имеет ровно два *in-параметра*'. Значение первого параметра является знаком бинарной ориентированной пары, являющейся вторым компонентом связки отношения *первичное условие инициирования** для абстрактного *sc-агента*, в множество *программ sc-агента** которого входит рассматриваемая **агентная scp-программа**, и, по сути, описывает класс событий, на которые реагирует указанный *sc-агент*.

Значением второго параметра является *sc-элемент*, с которым непосредственно связано событие, в результате возникновения которого был иницирован соответствующий *sc-агент*, т.е., например, сгенерированная либо удаляемая *sc-дуга* или *sc-ребро*.]

абстрактный sc-агент, реализуемый на Языке SCP

⇒ *принципы реализации**:

- { • [общие принципы организации взаимодействия *sc-агентов* и пользователей *ostis-системы* через общую *sc-память*]

- [в результате появления в *sc*-памяти некоторой конструкции, удовлетворяющей условию инициирования какого-либо *абстрактного sc-агента*, реализованного при помощи Языка SCP, в *sc*-памяти генерируется и иницируется *scp-процесс*. В качестве шаблона для генерации используется *агентная scp-программа*, указанная во множестве программ соответствующего *абстрактного sc-агента*]
- [каждый такой *scp-процесс*, соответствующий некоторой *агентной scp-программе*, может быть связан с набором структур, описывающих блокировки различных типов. Таким образом, синхронизация взаимодействия параллельно выполняемых *scp-процессов* осуществляется так же, как и в случае любых других *действий в sc-памяти*]
- [В рамках *scp-процесса* могут создаваться дочерние *scp-процессы*, однако синхронизация между ними при необходимости осуществляется посредством введения дополнительных внутренних блокировок. Таким образом, каждый *scp-процесс* с точки зрения *процессов в sc-памяти* является атомарным и законченным актом деятельности некоторого *sc-агента*]
- [во избежание нежелательных изменений в самом теле *scp-процесса*, вся конструкция, сгенерированная на основе некоторой *scp-программы* (весь текст *scp-процесса*), должна быть добавлена в *полную блокировку*, соответствующую данному *scp-процессу*]
- [все конструкции, сгенерированные в процессе выполнения *scp-процесса*, автоматически попадают в *полную блокировку*, соответствующую данному *scp-процессу*. Дополнительно следует отметить, что знак самой этой структуры и вся метаинформация о ней также включаются в эту структуру]
- [при необходимости можно вручную разблокировать или заблокировать некоторую конструкцию каким-либо типом блокировки, используя соответствующие *scp-операторы* класса *scp-оператор управления блокировками*]
- [после завершения выполнения некоторого *scp-процесса* его текст как правило, удаляется из *sc-памяти*, а все заблокированные конструкции освобождаются (разрушаются знаки структур, обозначавших блокировки)]
- [несмотря на то, что каждый *scp-оператор* представляет собой атомарное *действие в sc-памяти*, дополнительные блокировки, соответствующие одному оператору не вводятся, чтобы избежать громоздкости и избытка дополнительных системных конструкций, создаваемых при выполнении некоторого *scp-процесса*. Вместо этого используются блокировки, общие для всего *scp-процесса*. Таким образом, агенты *Абстрактной scp-машины* при интерпретации *scp-операторов* работают только с учетом блокировок, общих для всего интерпретируемого *scp-процесса*]
- [как правило, частный *класс действий*, соответствующий конкретной *scp-программе* явно не вводится, а используется более общий *класс scp-процесс*, за исключением тех случаев, когда введение специального *класса действий* необходимо по каким-либо другим соображениям]

}

scp-процесс⇒ *пояснение**:

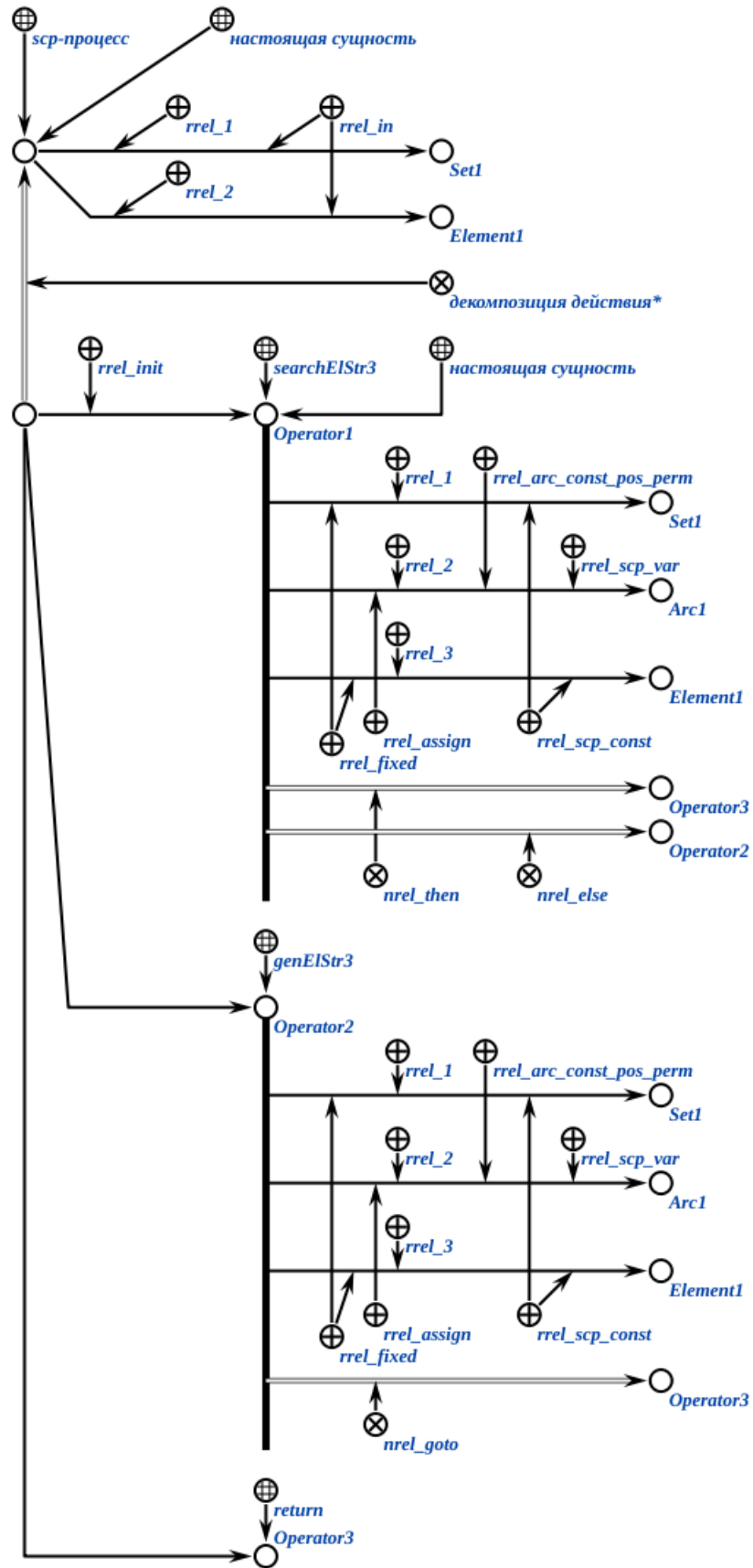
[Под *scp-процессом* понимается некоторое *действие в sc-памяти*, однозначно описывающее конкретный акт выполнения некоторой *scp-программы* для заданных исходных данных. Если *scp-программа* описывает алгоритм решения какой-либо задачи в общем виде, то *scp-процесс* обозначает конкретное действие, реализующее данный алгоритм для заданных входных параметров.

По сути, *scp-процесс* представляет собой уникальную копию, созданную на основе *scp-программы*, в которой каждой *sc-переменной*, за исключением *scp-переменных'*, соответствует сгенерированная *sc-константа*.

Принадлежность некоторого действия множеству *scp-процессов* гарантирует тот факт, что в декомпозиции данного действия будут присутствовать только знаки элементарных действий (*scp-операторов*), которые может интерпретировать реализация *Абстрактной scp-машины*.]

⇒ *пример выполнения**:

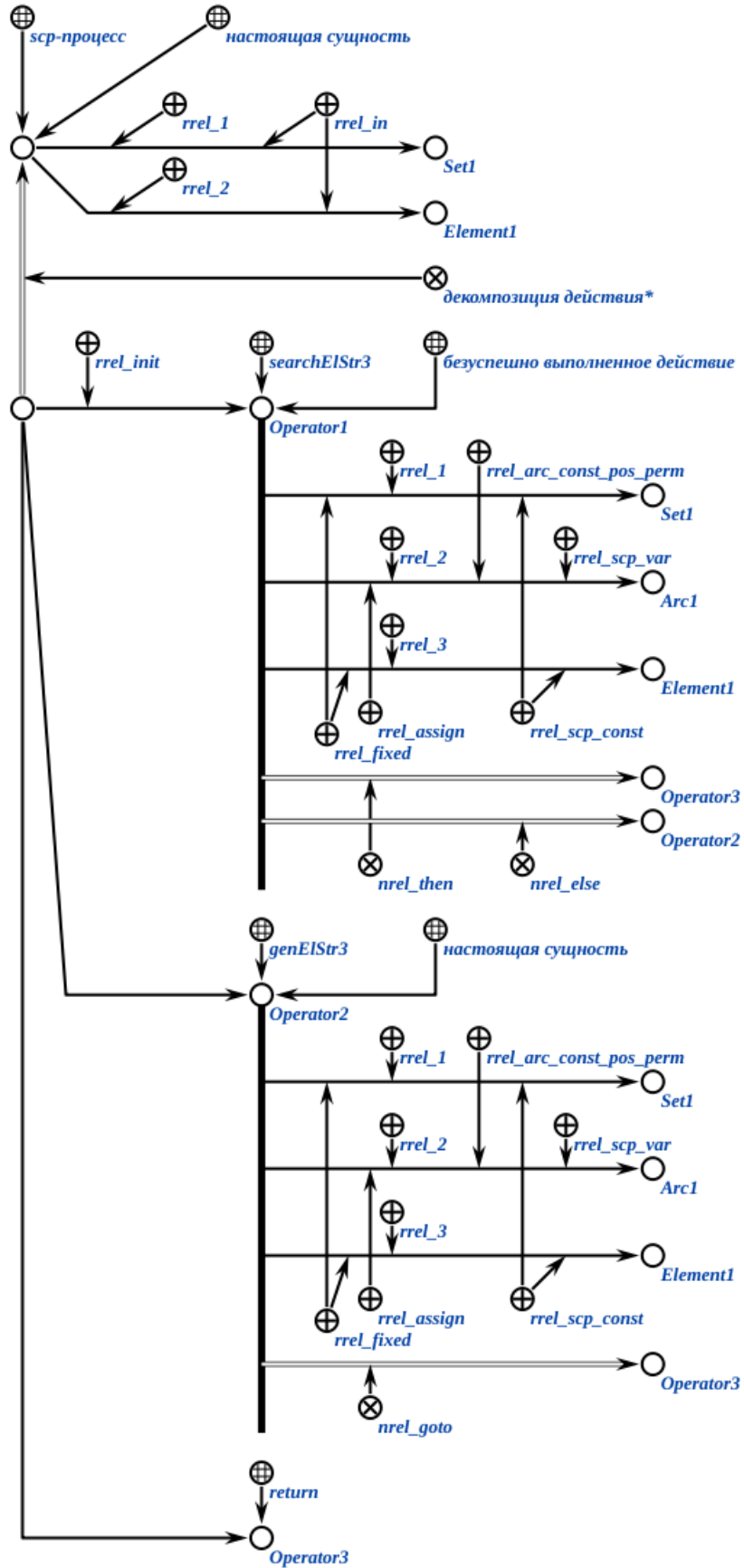
{•



⇒

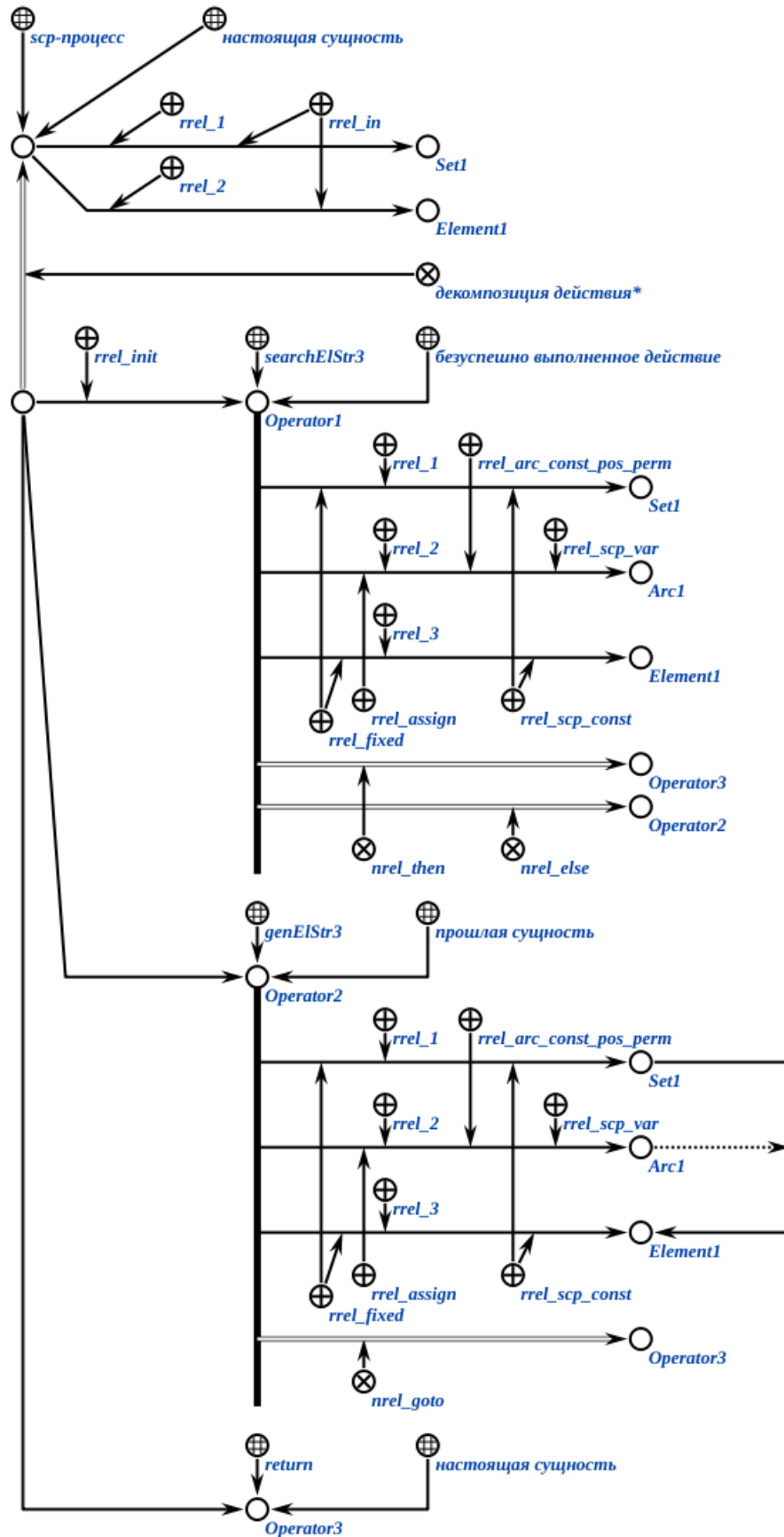
пояснение*:

[Осуществляется вызов *scp-программы*. Генерируется соответствующий *scp-процесс*. Происходит иницирование начального оператора *scp-процесса Operator1*]



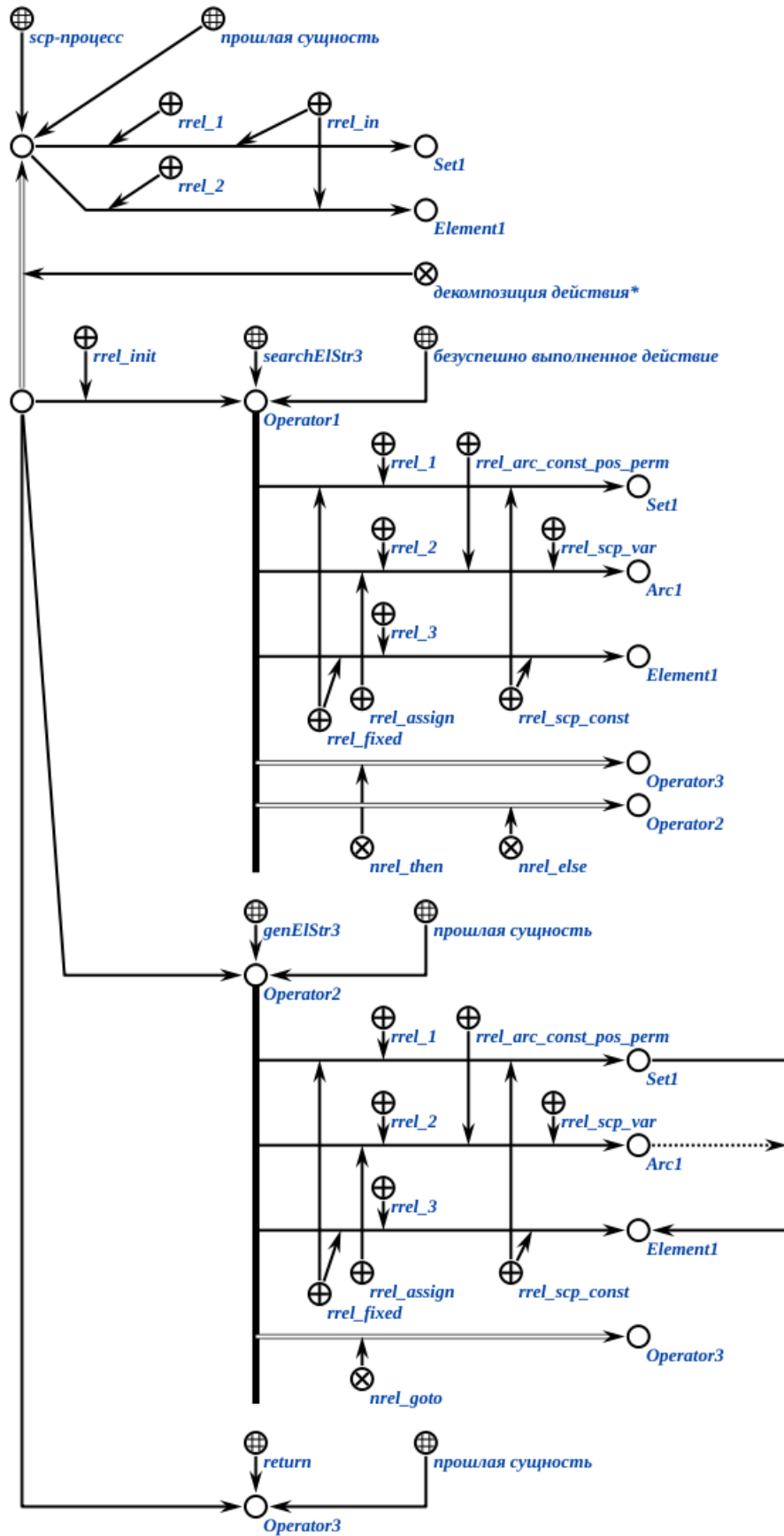
пояснение*:

[Оператор *Operator1* оказался безуспешно выполненным. Производится инициирование *scp-оператора генерации трёхэлементной конструкции Operator2*]



пояснение*:

[Оператор *Operator2* выполнен. Производится инициирование *scp-оператора* завершения выполнения программы *Operator3*]



] ⇒ пояснение*:
 [Оператор *Operator3* выполнен. Выполнение *scp-процесса* завершается.]

) ⇒ пояснение*:

[В приведенном примере последовательно показаны состояния *scp-процесса*, соответствующего *scp-программе*, добавляющей заданный элемент в заданное множество, если он там ранее не содержался. В примере предполагается, что рассматриваемый элемент (*Element1*) изначально не содержится во множестве (*Set1*).]

scp-оператор

С действие в *sc-памяти*
 ← семейство подмножеств*:
 атомарный тип *scp-оператора*
 ⇒ пояснение*:

[Каждый **scp-оператор** представляет собой некоторое элементарное действие в *sc-памяти*. Аргументы *scp-оператора* будем называть операндами. Порядок операндов указывается при помощи соответствующих ролевых отношений (*1'*, *2'*, *3'* и так далее). Операнд, помеченный ролевым отношением *1'*, будем называть первым операндом, помеченный ролевым отношением *2'* – вторым операндом, и т.д. Тип и смысл каждого операнда также уточняется при помощи различных подклассов отношения *scp-операнд'*. В общем случае операндом может быть любой *sc-элемент*, в том числе, знак какой-либо *scp-программы*, в том числе самой программы, содержащей данный оператор.

Каждый **scp-оператор** должен иметь один и более операнд, а также указание того **scp-оператора** (или нескольких), который должен быть выполнен следующим. Исключение их данного правила составляет *scp-оператор завершения выполнения программы*, который не содержит ни одного операнда и после выполнения которого никакие *scp-операторы* в рамках данной программы выполняться не могут.]

атомарный тип scp-оператора

⇒ пояснение*:

[Каждый **атомарный тип scp-оператора** представляет собой класс *scp-операторов*, который не разбивается на более частные, и, соответственно, интерпретируется реализацией *Абстрактной scp-машины*.]

начальный оператор'

С *1'*
 ⇒ пояснение*:

[Ролевое отношение **начальный оператор'** указывает в рамках декомпозиции соответствующего *scp-программе scp-процесса* те *scp-операторы*, которые должны быть выполнены в первую очередь, т.е. те, с которых собственно начинается выполнение *scp-процесса*.]

параметр scp-программы'

С аргумент действия'
 ⇒ разбиение*:
 { • *in-параметр'*
 • *out-параметр'*
 }
 ⇒ пояснение*:

[Ролевое отношение **параметр scp-программы'** связывает знак соответствующего *scp-программе scp-процесса* с его аргументами.]

in-параметр'

⇒ пояснение*:

[Параметры типа **in-параметр'** хоть и соответствуют *переменным scp-программы'*, не могут менять значение в процессе ее интерпретации. Фиксированное значение переменной устанавливается при создании уникальной копии *scp-программы (scp-процесса)* для ее интерпретации, и, таким образом, соответствующая *scp-переменная'* на момент начала ее интерпретации становится *scp-константой'* в рамках каждого *scp-оператора*, в котором встречалась данная *scp-переменная'*. Использование *in-параметров* можно рассматривать по аналогии с использованием варианта механизма передачи по значению в традиционных языках программирования, с тем условием, что значение локальной переменной в рамках дочерней программы не может быть изменено.]

out-параметр'

⇒ пояснение*:

[Параметры типа **out-параметр'** соответствуют *переменным scp-программы'* и обладают всеми теми же соответствующими свойствами. Чаще всего предполагается, что значение данного параметра необходимо родительской *scp-программе*, содержащей оператор вызова текущей *scp-программы*. При этом на мо-

мент начала интерпретации в качестве параметра дочернему процессу передается непосредственно узел, обозначающий переменную (а точнее, ее уникальную копию в рамках процесса) родительского процесса. Указанная переменная может при необходимости иметь значение, либо не иметь. После завершения и во время интерпретации дочернего процесса родительский процесс по-прежнему может работать с переменной, переданной в качестве *out-параметра*, при необходимости просматривая или изменяя ее значение. Использование *out-параметра* можно рассматривать по аналогии с использованием механизма передачи по ссылке в традиционных языках программирования.]

структура

⇒ *разбиение**:

Классификация структур с точки зрения Базовой модели обработки sc-текстов

```
= {
  • sc-конструкция нестандартного вида
  • sc-конструкция стандартного вида
    ⇒ разбиение*:
      {
        • одноэлементная sc-конструкция
        • трехэлементная sc-конструкция
        • пятиэлементная sc-конструкция
      }
}
```

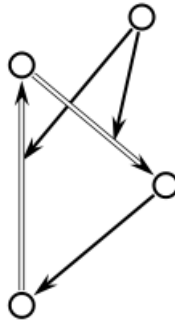
sc-конструкция нестандартного вида

⇒ *пояснение**:

[Каждая *sc-конструкция нестандартного вида* состоит из произвольного количества *sc-элементов* произвольного типа.]

⇒ *пример**:

[



]

sc-конструкция стандартного вида

⇒ *пояснение**:

[В свою очередь, каждый элемент *sc-конструкции стандартного вида* имеет свою условную строго фиксированную позицию в рамках этой *sc-конструкции* (первый элемент, второй элемент и т. д.). В зависимости от указанной позиции вводятся дополнительные ограничения на тип соответствующего *sc-элемента*.]

одноэлементная sc-конструкция

⇒ *пояснение**:

[Каждая *одноэлементная sc-конструкция* состоит из одного *sc-элемента* произвольного типа.]

⇒ *пример**:

[



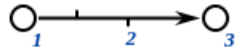
]

трехэлементная sc-конструкция

⇒ *пояснение**:

[Каждая *трехэлементная sc-конструкция* состоит из трех *sc-элементов*. Второй элемент всегда является *sc-коннектором*, остальные элементы могут быть произвольного типа.]

⇒ *пример**:

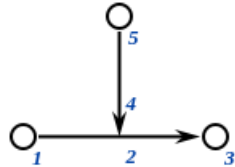


пятиэлементная sc-конструкция

⇒ *пояснение**:

[Каждая *пятиэлементная sc-конструкция* состоит из пяти *sc-элементов*. Второй и четвертый элементы обязательно являются *sc-коннекторами*, остальные элементы могут быть произвольного типа.]

⇒ *пример**:



} /* Завершили Раздел “Предметная область и онтология Базового языка программирования *ostis-систем*” */

```

/* Раздел *****/
Предметная область и онтология денотационной семантики
Базового языка программирования ostis-систем
⊃=
{
Предметная область денотационной семантики языка SCP
∈ предметная область
∋ максимальный класс объектов исследования':
  scp-оператор
∋ исследуемое отношение':
  • scp-операнд'
  • scp-константа'
  • scp-переменная'
  • scp-операнд с заданным значением'
  • scp-операнд со свободным значением'
  • формируемое множество'
  • удаляемый sc-элемент'

scp-оператор
← включение*:
  действие в sc-памяти
← семейство подмножеств*:
  атомарный тип scp-оператора
⇒ разбиение*:
  { • scp-оператор генерации конструкций
    ⇒ разбиение*:
      { • scp-оператор генерации конструкции по произвольному образцу
        • scp-оператор генерации пятиэлементной конструкции
        • scp-оператор генерации трехэлементной конструкции
        • scp-оператор генерации одноэлементной конструкции
      }
    • scp-оператор ассоциативного поиска конструкций
    ⇒ разбиение*:
      { • scp-оператор поиска конструкции по произвольному образцу
        • scp-оператор поиска пятиэлементной конструкции с формированием множеств
        • scp-оператор поиска трехэлементной конструкции с формированием множеств
        • scp-оператор поиска пятиэлементной конструкции
        • scp-оператор поиска трехэлементной конструкции
      }
    • scp-оператор удаления конструкций
    ⇒ разбиение*:
      { • scp-оператор удаления множества элементов трехэлементной конструкции
        • scp-оператор удаления одноэлементной конструкции
        • scp-оператор удаления пятиэлементной конструкции
        • scp-оператор удаления трехэлементной конструкции
      }
    • scp-оператор проверки условий
    ⇒ разбиение*:
      { • scp-оператор сравнения числовых содержимых файлов
        • scp-оператор проверки равенства числовых содержимых файлов
        • scp-оператор проверки совпадения значений операндов
        • scp-оператор проверки наличия содержимого у файла
        • scp-оператор проверки наличия значения у переменной
        • scp-оператор проверки типа sc-элемента
      }
    • scp-оператор управления значениями операндов
    ⇒ разбиение*:
      { • scp-оператор удаления значения переменной
        • scp-оператор присваивания значения переменной
      }
  }
}

```

- }
spr-оператор управления *spr*-процессами
 ⇒ разбиение*:
 - *spr*-оператор удаления значения переменной
 - *spr*-оператор завершения выполнения программы
 - конъюнкция предшествующих *spr*-операторов
 - *spr*-оператор ожидания завершения выполнения множества *spr*-программ
 - *spr*-оператор ожидания завершения выполнения *spr*-программы
 - *spr*-оператор асинхронного вызова подпрограммы
- }
spr-оператор управления событиями
 ⇐ разбиение*:
 - *spr*-оператор ожидания события
- }
spr-оператор обработки содержимых файлов
 ⇒ разбиение*:
 - *spr*-оператор вычисления арксинуса числового содержимого файла
 - *spr*-оператор вычисления аркосинуса числового содержимого файла
 - *spr*-оператор деления числовых содержимых файлов
 - *spr*-оператор умножения числовых содержимых файлов
 - *spr*-оператор вычитания числовых содержимых файлов
 - *spr*-оператор сложения числовых содержимых файлов
 - *spr*-оператор вычисления тангенса числового содержимого файла
 - *spr*-оператор вычисления косинуса числового содержимого файла
 - *spr*-оператор вычисления синуса числового содержимого файла
 - *spr*-оператор вычисления логарифма числового содержимого файла
 - *spr*-оператор возведения числового содержимого файла в степень
 - *spr*-оператор удаления содержимого файла
 - *spr*-оператор копирования содержимого файла
 - *spr*-оператор нахождения остатка от деления числовых содержимых файлов
 - *spr*-оператор нахождения целой части от деления числовых содержимых файлов
 - *spr*-оператор вычисления арктангенса числового содержимого файла
 - *spr*-оператор перевода в верхний регистр строкового содержимого файла
 - *spr*-оператор перевода в верхний регистр строкового содержимого файла
 - *spr*-оператор замены определенной части строкового содержимого файла на содержимое указанной файла
 - *spr*-оператор проверки совпадения конца строкового содержимого файла со строковым содержимым другого файла
 - *spr*-оператор проверки совпадения начальной части строкового содержимого файла со строковым содержимым другого файла
 - *spr*-оператор получения части строкового содержимого файла по индексам
 - *spr*-оператор поиска строкового содержимого файла в строковом содержимом другого файла
 - *spr*-оператор вычисления длины строкового содержимого файла
 - *spr*-оператор разбиения строки на подстроки
 - *spr*-оператор лексикографического сравнения строковых содержимых файлов
 - *spr*-оператор проверки равенства строковых содержимых файлов
- }
spr-оператор управления блокировками
 ⇒ разбиение*:
 - *spr*-оператор снятия всех блокировок данного *spr*-процесса
 - *spr*-оператор снятия блокировки с *sc*-элемента
 - *spr*-оператор установки полной блокировки на *sc*-элемент
 - *spr*-оператор установки блокировки на изменение *sc*-элемента
 - *spr*-оператор установки блокировки на удаление *sc*-элемента
 - *spr*-оператор снятия блокировки со структуры
 - *spr*-оператор установки полной блокировки на структуру
 - *spr*-оператор установки блокировки на изменение структуры
 - *spr*-оператор установки блокировки на удаление структуры
- }

scr-операнд'

- ← включение*:
аргумент действия'
 - ∈ неосновное понятие
 - ∈ ролевое отношение
 - ⇒ разбиение*:
 - {• *scr-константа'*
 - *scr-переменная'*
 - }
 - ⇒ разбиение*:
 - {• *scr-операнд с заданным значением'*
 - *scr-операнд со свободным значением'*
 - }
 - ⇒ разбиение*:
 - {• *константный sc-элемент'*
 - *переменный sc-элемент'*
 - }
 - ⇒ включение*:
 - *формируемое множество'*
 - ⇒ разбиение*:
 - {• *формируемое множество 1'*
 - *формируемое множество 2'*
 - *формируемое множество 3'*
 - *формируемое множество 4'*
 - *формируемое множество 5'*
 - }
 - *удаляемый sc-элемент'*
 - *тип sc-элемента'*
 - ⇒ разбиение*:
 - {• *sc-узел'*
 - ⇒ разбиение*:
 - {• *структура'*
 - *отношение'*
 - ⇒ включение*:
 - ролевое отношение'
 - *класс'*
 - }
 - *sc-дуга'*
 - ⇒ разбиение*:
 - {• *sc-дуга общего вида'*
 - *sc-дуга принадлежности'*
 - ⇒ включение*:
 - sc-дуга основного вида'*
 - ⇒ разбиение*:
 - {• *позитивная sc-дуга принадлежности'*
 - *негативная sc-дуга принадлежности'*
 - *нечеткая sc-дуга принадлежности'*
 - }
 - ⇒ разбиение*:
 - {• *временная sc-дуга принадлежности'*
 - *постоянная sc-дуга принадлежности'*
 - }
 - *sc-ребро'*
 - *файл'*
 - }
- ⇒ пояснение*:

[Ролевое отношение *scr-операнд'* является неосновным понятием и указывает на принадлежность аргументов *scr-оператору*. Помимо указания какого-либо класса *scr-операндов'* порядок аргументов *scr-оператора* дополнительно уточняется ролевыми отношениями 1', 2' и т. д.]

scr-константа'⇒ *пояснение**:

[В рамках *scr-программы scr-константы'* явно участвуют в *scr-операторах* в качестве элементов (в теоретико-множественном смысле) и напрямую обрабатываются при интерпретации *scr-программы*. Константами в рамках *scr-программы* могут быть *sc-элементы* любого типа, как *sc-константы*, так и *sc-переменные*. Константа в рамках *scr-программы* остается неизменной в течение всего срока интерпретации. Константа *scr-программы* может быть рассмотрена как переменная, значение которой совпадает с самой переменной в каждый момент времени, и изменено быть не может. Таким образом, далее будем считать, что *scr-константа'* и ее значение это одно и то же. Каждый *in-параметр'* при интерпретации каждой конкретной копии *scr-программы* становится *scr-константой'* в рамках всех ее операторов, хотя в исходном теле данной программы в каждом из этих операторов он является *scr-переменной'*.]

scr-переменная'⇒ *пояснение**:

[В рамках *scr-программы scr-переменные'* не обрабатываются явно при интерпретации, обрабатываются значения переменных. Каждая переменная *scr-программы* может иметь одно значение в каждый момент времени, т. е. представляет собой ситуативный *синглетон*, элементом которого является текущее значение *scr-переменной'*. Значение каждой *scr-переменной'* может меняться в ходе интерпретации *scr-программы*. При этом интерпретатор при обработке *scr-оператора* работает непосредственно со значениями *scr-переменных'*, а не самими *scr-переменными'* (которые также являются узлами той же семантической сети).]

scr-операнд с заданным значением'⇒ *пояснение**:

[Значение операндов, помеченных ролевым отношением *scr-операнд с заданным значением'*, считается заданным в рамках текущего *scr-оператора*. Данное значение учитывается при выполнении *scr-оператора* и остается неизменным после окончания выполнения *scr-оператора*. Каждая *scr-константа'* по умолчанию рассматривается как *scr-операнд с заданным значением'*, в связи с чем явное использование данного ролевого отношения в таком случае является избыточным. В таком случае в качестве значения рассматривается непосредственно сам операнд. В случае если отношением *scr-операнд с заданным значением'* помечена *scr-переменная'*, то осуществляется попытка поиска значения для данной *scr-переменной'* (ее элемента). Если попытка оказалась безуспешной, то возникает ошибка времени выполнения, которая должна быть обработана соответствующим образом.

Любой *scr-операнд с заданным значением'* независимо от конкретного типа *scr-оператора* может быть *scr-переменной'*.]

scr-операнд со свободным значением'⇒ *пояснение**:

[Значение операндов, помеченных ролевым отношением *scr-операнд со свободным значением'*, считается свободным (не заданным заранее) в рамках текущего *scr-оператора*. В начале выполнения *scr-оператора* связь между *scr-переменной'*, помеченной данным ролевым отношением, и ее элементом (значением) всегда удаляется. В результате выполнения данного оператора может быть либо сгенерировано новое значение *scr-переменной'*, либо не сгенерировано, тогда *scr-переменная'* будет считаться не имеющей значения. Ни одна *scr-константа'* не может быть помечена как *scr-операнд со свободным значением'*, поскольку константа не может изменять свое значение в ходе интерпретации *scr-программы*.]

тип sc-элемента'⇒ *пояснение**:

[Ролевое отношение *тип sc-элемента'* используется для уточнения типа *sc-элемента*, выступающего в роли значения некоторого операнда. *тип sc-элемента'* имеет смысл указывать только для операндов, помеченных как *scr-операнд со свободным значением'*, тогда данное уточнение типа *sc-элемента* будет использовано для сужения области поиска либо уточнения параметров генерации каких-либо конструкций. Значением *scr-операндов с заданным значением'* является конкретный, известный на момент начала выполнения *scr-оператора sc-элемент* с конкретным типом, не зависящим от указания *типа sc-элемента'*, в связи с чем использование ролевого отношения *тип sc-элемента'* в данном случае является некорректным.

Допускается использование комбинаций семантически непротиворечащих друг другу подмножеств указанного отношения. Например, допускается комбинация *константный sc-элемент'* и *sc-дуга общего вида'*, но не допускается комбинация *sc-узел'* и *sc-дуга'*.]

sc-дуга основного вида'

= (константный *sc-элемент'* \cap позитивная *sc-дуга принадлежности'* \cap постоянная *sc-дуга принадлежности'*)

формируемое множество'

⇒ пояснение*:

[Ролевое отношение *формируемое множество'* используется для указания того факта, что в результате выполнения *scr-оператора* должно быть сформировано либо дополнено некоторое множество *sc-элементов*, являющееся значением одного из операндов данного *scr-оператора*. При этом если данный операнд помечен как *scr-операнд со свободным значением'*, то множество будет сформировано с нуля (сгенерирован новый *sc-элемент*, обозначающий данное множество), в противном случае уже существующее множество может быть дополнено. Использование данного ролевого отношения предполагает, что при его отсутствии множество бы не формировалось, а значением указанного операнда стал бы произвольный *sc-элемент* из данного множества.

Ролевое отношение *формируемое множество'* без уточнения порядкового номера используется только в *scr-операторах обработки произвольных конструкций*. Для явного указания номера операнда, которому соответствует *формируемое множество'*, используются подмножества данного ролевого отношения, аналогичные ролевым отношениям, задающим порядок элементов в кортеже ($1'$, $2'$, $3'$ и т. д.), например *формируемое множество 1'*, *формируемое множество 2'* и т. д. Указанные ролевые отношения используются только в *scr-операторах поиска конструкций с формированием множеств*.]

удаляемый sc-элемент'

⇒ пояснение*:

[Ролевое отношение *удаляемый sc-элемент'* используется для указания тех операндов, значение которых должно быть удалено в процессе выполнения *scr-операторов удаления*. Данным ролевым отношением может быть помечен как *scr-операнд с заданным значением'*, так и *scr-операнд со свободным значением'*. При этом удаляемым *sc-элементом* может быть как *scr-константа'*, так и *scr-переменная'* (в случае *scr-переменной'* удаляется не только связка принадлежности между этой *scr-переменной'* и ее значением, но и непосредственно сам *sc-элемент*, являющийся значением).]

следует отличать*

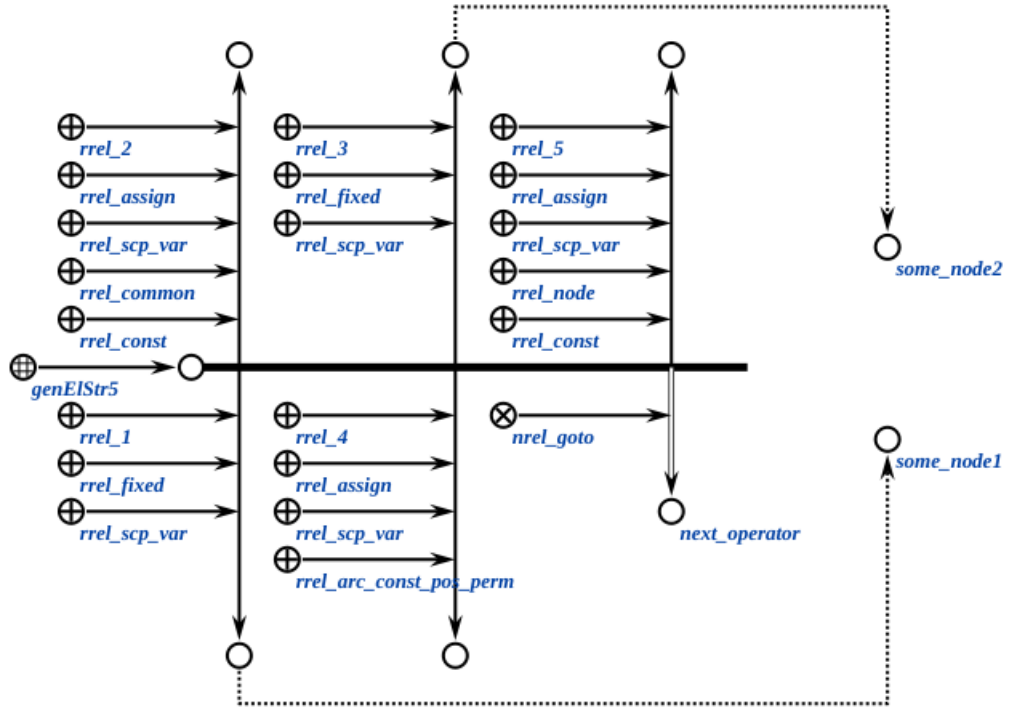
∋ { • *scr-переменная'*
• *sc-переменная*
}

∋ { • *scr-константа'*
• *sc-константа*
}

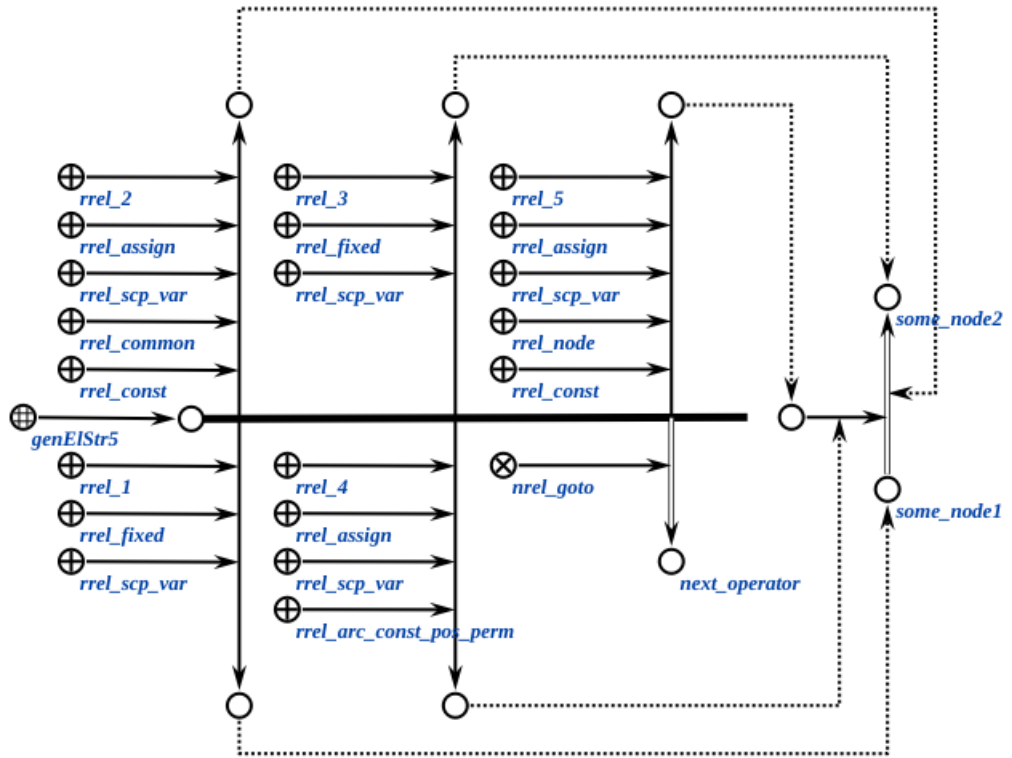
scr-оператор генерации пятиэлементной конструкции

⇒ пример выполнения*:

{ •



⇒ *пояснение**:
[Вызов scp-оператора.]



⇒ *пояснение**:
[Результат выполнения scp-оператора.]

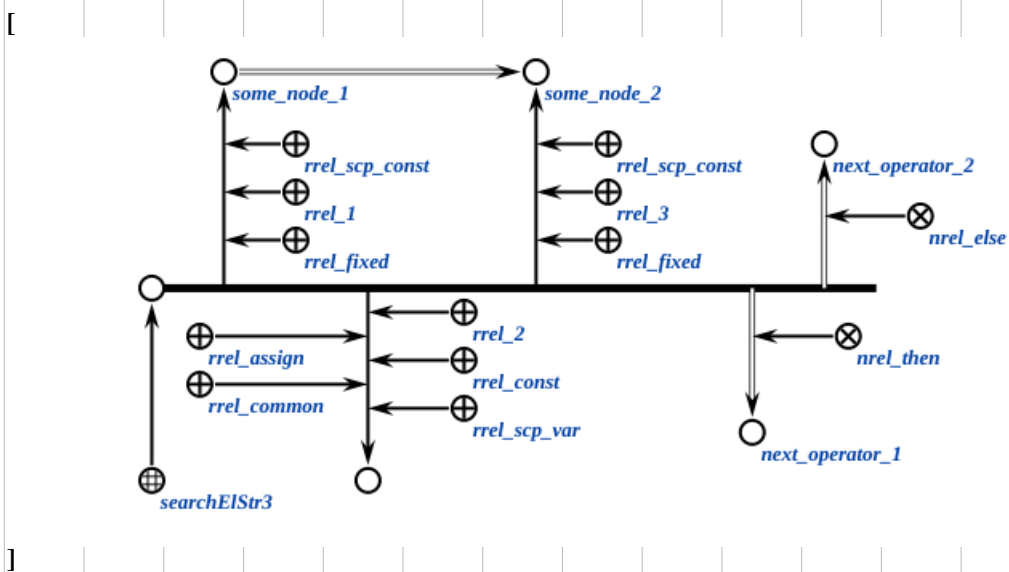
⇒ *пояснение**:

[В приведённом примере показан пример scp-оператора генерации пятиэлементной конструкции, которая имеет два scp-операнда с заданным значением. В примере предполагается, что рассматриваемые элементы (some_node1 и some_node2) изначально никак не связаны между собой.]

scp-оператор поиска трехэлементной конструкции

⇒ пример выполнения*:

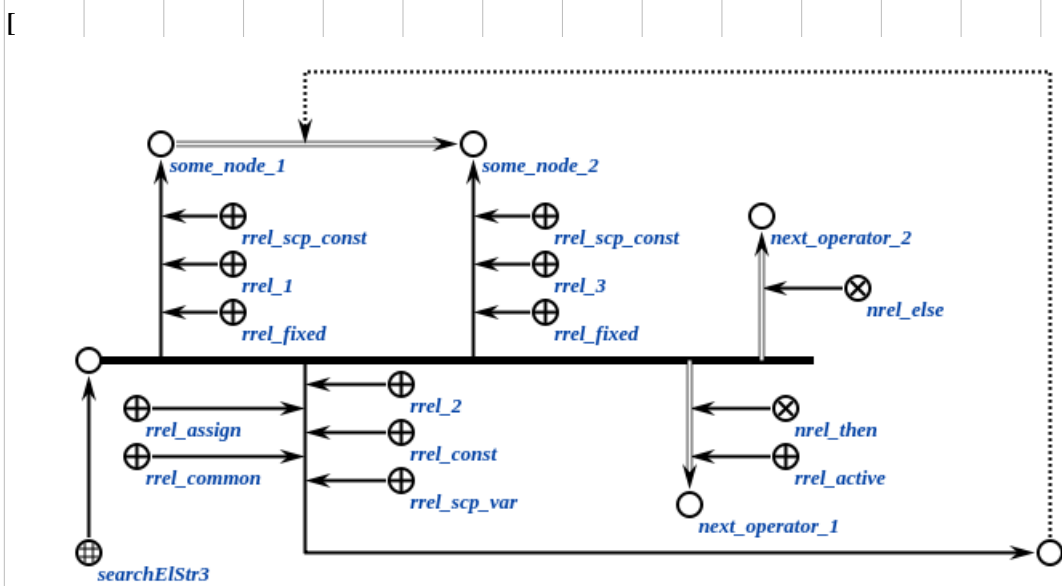
<•



] ⇒ пояснение*:

[Вызов scp-оператора.]

•



] ⇒ пояснение*:

[Результат выполнения scp-оператора.]

>

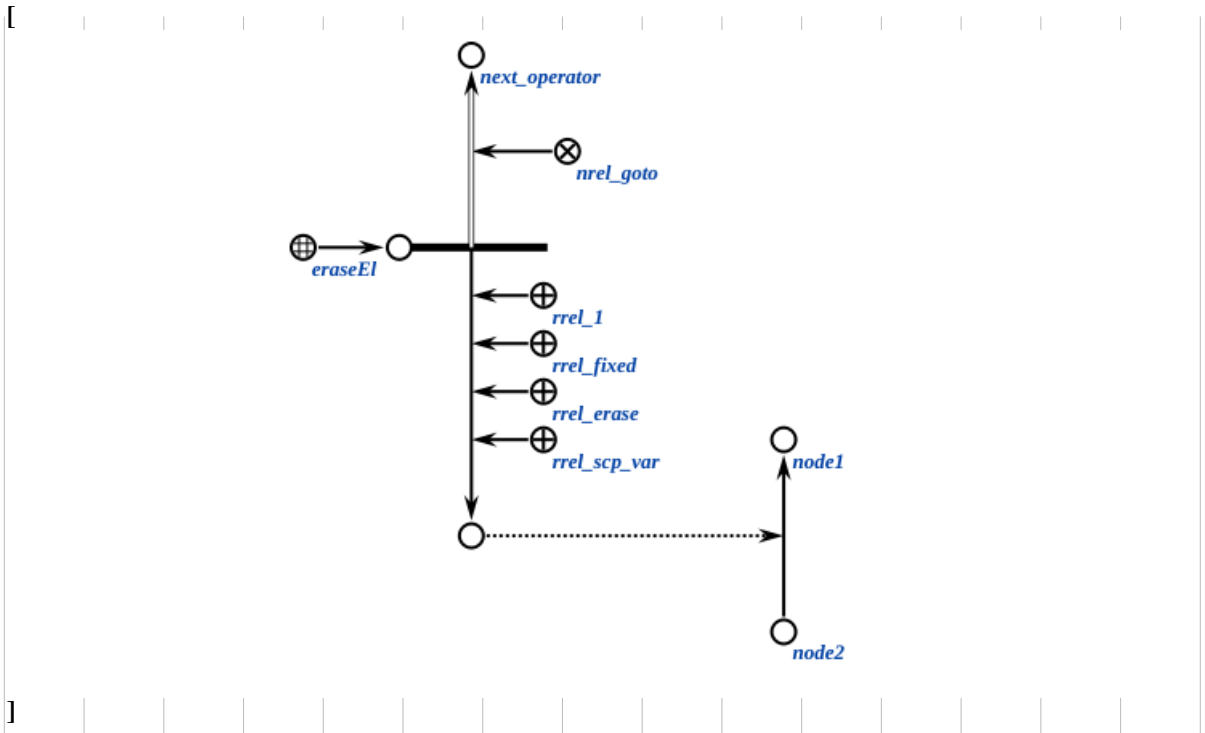
⇒ пояснение*:

[В приведённом примере показан пример scp-оператора поиска трехэлементной конструкции, которая имеет два scp-операнда с заданным значением. В примере предполагается, что рассматриваемые элементы (some_node1 и some_node2) изначально связаны между собой константной постоянной sc-дугой.]

scp-оператор удаления одноэлементной конструкции

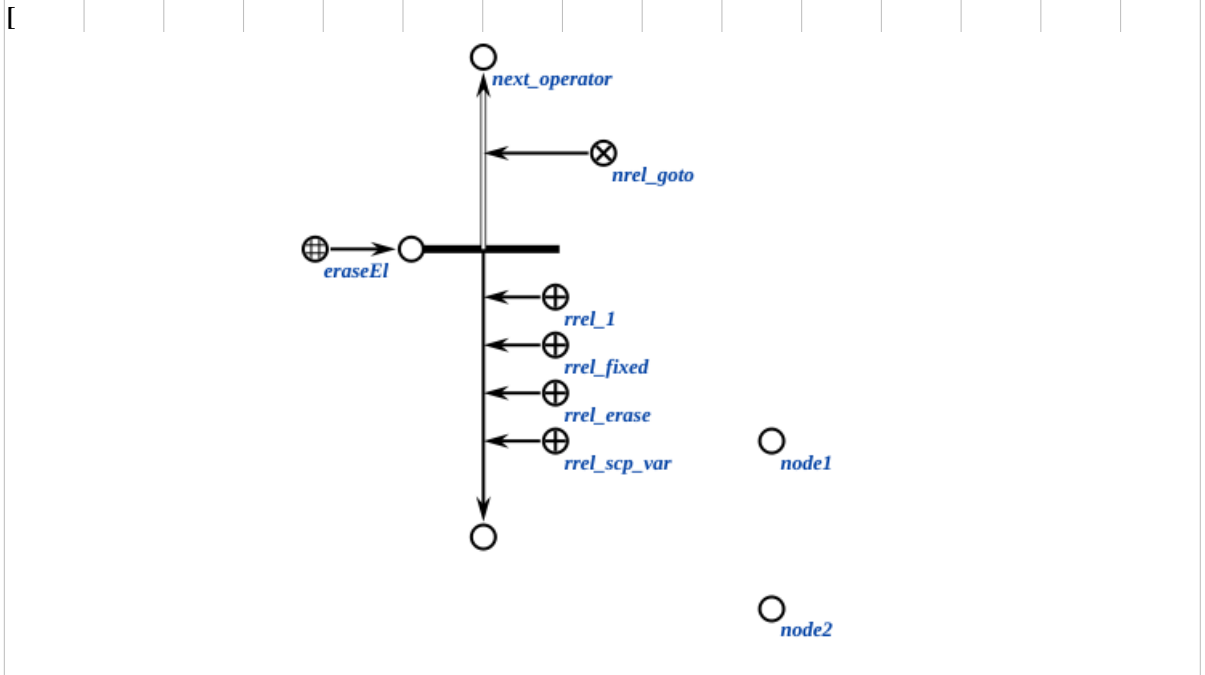
⇒ пример выполнения*:

<•



] ⇒ *пояснение**:
[Вызов scp-оператора.]

-



] ⇒ *пояснение**:
[Результат выполнения scp-оператора.]

) ⇒ *пояснение**:
[В приведённом примере показан пример scp-оператора удаления одноэлементной конструкции. В примере предполагается, что рассматриваемые элементы (node1 и node2) изначально связаны между собой базовой sc-дугой принадлежности.]

} /* Завершили Раздел “Предметная область и онтология денотационной семантики Базового языка программирования ostis-систем” */

/* Раздел *****/

**Предметная область и онтология операционной семантики
Базового языка программирования *ostis*-систем**

⊃=
{

Предметная область операционной семантики языка SCP

- ∈ предметная область
- ⊃ максимальный класс объектов исследования¹:
Абстрактная *scp*-машина
- ⊃ ключевой объект исследования¹:
 - Абстрактный *sc*-агент создания *scp*-процессов
 - Абстрактный *sc*-агент интерпретации *scp*-операторов
 - Абстрактный *sc*-агент синхронизации процесса интерпретации *scp*-программ
 - Абстрактный *sc*-агент уничтожения *scp*-процессов
 - Абстрактный *sc*-агент синхронизации событий в *sc*-памяти и ее реализации
 - Абстрактный *sc*-агент трансляции сформированной спецификации события в *sc*-памяти во внутреннее представление
 - Абстрактный *sc*-агент обработки события в *sc*-памяти, иницирующего агентную *scp*-программу

Абстрактная *scp*-машина

- ← декомпозиция абстрактного *sc*-агента*:
 - { • Абстрактный *sc*-агент создания *scp*-процессов
 - Абстрактный *sc*-агент интерпретации *scp*-операторов
 - Абстрактный *sc*-агент синхронизации процесса интерпретации *scp*-программ
 - Абстрактный *sc*-агент уничтожения *scp*-процессов
 - Абстрактный *sc*-агент синхронизации событий в *sc*-памяти и ее реализации
 - }

Абстрактный *sc*-агент создания *scp*-процессов

- ⇒ пояснение*:

[Задачей Абстрактного *sc*-агента создания *scp*-процессов является создание *scp*-процессов, соответствующих заданной *scp*-программе. Данный *sc*-агент активируется при появлении в *sc*-памяти иницированного действия, принадлежащего классу действие интерпретации *scp*-программы.

После проверки *sc*-агентом условия иницирования выполняется создание *scp*-процесса с учетом конкретных параметров интерпретации *scp*-программы, после чего осуществляется поиск начального оператора¹ *scp*-процесса и добавление его во множество настоящих сущностей.]

Абстрактный *sc*-агент интерпретации *scp*-операторов

- ⇒ пояснение*:

[Задачей Абстрактного *sc*-агента интерпретации *scp*-операторов является собственно интерпретация операторов *scp*-программы, то есть выполнение в *sc*-памяти действий, описываемых конкретным *scp*-оператором. Данный *sc*-агент активируется при появлении в *sc*-памяти *scp*-оператора, принадлежащего классу настоящих сущностей. После выполнения действия, описываемого *scp*-оператором, *scp*-оператор добавляется во множество прошлых сущностей. В случае когда семантика действия, описываемого *scp*-оператором, предполагает возможность ветвления *scp*-программы после выполнения данного *scp*-оператора, то используется одно из подмножеств класса выполненных действий – безуспешно выполненное действие или успешно выполненное действие.]

Абстрактный *sc*-агент синхронизации процесса интерпретации *scp*-программ

- ⇒ пояснение*:

[Задачей Абстрактного *sc*-агента синхронизации процесса интерпретации *scp*-программ является обеспечение переходов между *scp*-операторами в рамках одного *scp*-процесса. Данный *sc*-агент активизируется при добавлении некоторого *scp*-оператора во множество прошлых сущностей. Далее осуществляется переход по *sc*-дуге, принадлежащей отношению последовательность действий* (или более частным отношениям, в случае, если *scp*-оператор был добавлен во множество успешно выполненных действий или безуспешно выполненных действий). При этом очередной *scp*-оператор становится настоящей сущностью (активным *scp*-оператором) в том случае, если хотя бы один *scp*-оператор, связанный с ним входящими *sc*-дугами, принадлежащими отношению последовательность действий* (или более частным отношениям), стал

прошлой сущностью (или, соответственно, подмножеством прошлых сущностей). В случае, когда необходимо дождаться завершения выполнения всех предыдущих операторов, для синхронизации используется оператор класса *конъюнкция предшествующих операторов*.]

Абстрактный *sc*-агент уничтожения *scp*-процессов

⇒ *пояснение**:

[Задачей *Абстрактного sc-агента уничтожения scp-процессов* является уничтожение *scp-процесса*, т. е. удаление из *sc-памяти* всех *sc-элементов*, его составляющих. Данный *sc-агент* активируется при появлении в *sc-памяти scp-процесса*, принадлежащего множеству *прошлых сущностей*.

При этом уничтожаемый *scp-процесс* необязательно должен быть полностью сформирован. Необходимость уничтожения не до конца сформированного *scp-процесса* может возникнуть в случае, если при создании *scp-процесса* возникли проблемы, не позволяющие продолжить создание *scp-процесса* и его выполнение.]

Абстрактный *sc*-агент синхронизации событий в *sc-памяти* и ее реализации

⇒ *пояснение**:

[Задачей *Абстрактного sc-агента синхронизации событий в sc-памяти и ее реализации* является обеспечение работы *неатомарных sc-агентов*, реализованных на языке *SCP*.]

⇐ *декомпозиция абстрактного sc-агента**:

- { • *Абстрактный sc-агент трансляции сформированной спецификации события в sc-памяти во внутреннее представление*
- *Абстрактный sc-агент обработки события в sc-памяти, инициирующего агентную scp-программу*
- }

Абстрактный *sc*-агент трансляции сформированной спецификации события в *sc-памяти* во внутреннее представление

⇒ *пояснение**:

[Задачей *Абстрактного sc-агента трансляции сформированной спецификации события в sc-памяти во внутреннее представление* является трансляция ориентированных пар, описывающих *первичное условие инициирования** некоторого *sc-агента* во внутреннее представление элементарных событий на уровне *sc-хранилища*, при условии, что этот *sc-агент* реализован на платформенно-независимом уровне (с использованием языка *SCP*). Условием инициирования данного *sc-агента* является появление в *sc-памяти* нового элемента множества *активных sc-агентов*, для которого будет найдена и протранслирована соответствующая ориентированная пара.]

Абстрактный *sc*-агент обработки события в *sc-памяти*, инициирующего агентную *scp*-программу

⇒ *пояснение**:

[Задачей *Абстрактного sc-агента обработки события в sc-памяти, инициирующего агентную scp-программу*, является поиск *агентной scp-программы*, входящей во множество *программ sc-агента** для каждого *sc-агента*, первичное условие инициирования которого соответствует событию, произошедшему в *sc-памяти*, а также генерация и инициирование действия, направленного на интерпретацию этой программы. В результате работы данного *sc-агента* в *sc-памяти* появляется *инициированное действие*, принадлежащее классу *действие интерпретации scp-программы*.]

} /* Завершили Раздел “Предметная область и онтология операционной семантики Базового языка программирования *ostis-систем*” */

```

/* Раздел *****/
Предметная область и онтология искусственных нейронных сетей и соответствующая им предметная область и онтология действий по обработке искусственных нейронных сетей
⊃=
{
⇒ соавтор*:
• Головки В.А.
• Ковалёв М.В.
• Крощенко А.А.
• Михно Е.В.
⇐ конкатенация сегментов*:
{ • Предметная область и онтология искусственных нейронных сетей
  • Предметная область и онтология действий по обработке искусственных нейронных сетей
}
⇒ введение*:
[В последнее десятилетие обозначилась устойчивая тенденция широкого применения методов машинного обучения в самых разных областях человеческой деятельности, обусловленная в первую очередь развитием теории искусственных нейронных сетей (и.н.с.), а также аппаратных возможностей.

Преимущество и.н.с. заключается в том, что они могут работать с неструктурированными данными. Главный недостаток и.н.с. – это отсутствие понятной человеку обратной связи, которую можно было бы назвать цепочкой рассуждений, т.е. можно сказать, что и.н.с. работают как “черный ящик” (Castelvecchi.B.AIBlackBox-2016).

Сложность современных интеллектуальных систем, использующих нейросетевые модели, а также большой объём обрабатываемых ими данных обуславливают необходимость мониторинга, объяснения и понимания механизмов их работы с целью вербализации оценки и оптимизации их деятельности.

В связи с этим становится актуальна разработка нейросимволических подходов (описанных, в частности, в работе Tarek.R..NeuralSymbolic-2017), в частности, подходов по интеграции и.н.с и баз знаний, использующих онтологии. Такие интегрированные системы способны сочетать:

1) возможность семантической интерпретации обрабатываемых данных, используя представление решаемых и.н.с. прикладных задач, а так же спецификацию её входных и выходных данных;

2) с представлением самой структуры и.н.с., описанием её свойств и состояний, позволяющими упростить понимание её работы (Golovko.V..IntegrationANNKB-2018).

Можно выделить два основных направления интеграции и.н.с. с базами знаний:

1) построение интеллектуальных систем, способных использовать нейросетевые методы наравне с другими имеющимися в системе методами для решения задач или подзадач системы. Такие системы смогут учитывать семантику решаемых задач на более высоком уровне, что сделает решение этих задач более структурированными и прозрачными.

2) построение интеллектуальной среды по разработке, обучению и интеграции различных и.н.с., совместимых с базами знаний через представление и.н.с. с помощью онтологических структур и их интерпретацию средствами представления знаний. Такая среда предоставит возможность интроспекции и.н.с, возможность сохранения состояний и.н.с. после обучения и реконфигурации сети. Это позволит производить более глубокий анализ работы и.н.с. Так же формальное описание знаний рамках предметной области и.н.с. поможет уменьшить порог вхождения разработчиков в методы решения задач с помощью и.н.с.

Данный раздел посвящен предметной области и онтологии искусственных нейронных сетей и предметной области и онтологии действий по обработке искусственных нейронных сетей, которые являются основой развития обоих указанных направлений. ]

/* Сегмент *****/
Предметная область и онтология искусственных нейронных сетей
⊃=
{

Предметная область искусственных нейронных сетей
:= [Предметная область и.н.с.]
∈ предметная область

```

- Э *максимальный класс объектов исследования'*:
- *искусственная нейронная сеть*
- Э *класс объектов исследования'*:
- *искусственная нейронная сеть*
 - *искусственная нейронная сеть с прямыми связями*
 - *персептрон*
 - *персептрон Розенблатта*
 - *автоэнкодерная искусственная нейронная сеть*
 - *машина опорных векторов*
 - *искусственная нейронная сеть радиально-базисных функций*
 - *искусственная нейронная сеть с обратными связями*
 - *нейронная сеть Хопфилда*
 - *нейронная сеть Хэмминга*
 - *рекуррентная искусственная нейронная сеть*
 - *искусственная нейронная сеть Джордана*
 - *искусственная нейронная сеть Элмана*
 - *мультирекуррентная нейронная сеть*
 - *LSTM-элемент*
 - *GRU-элемент*
 - *полносвязная искусственная нейронная сеть*
 - *слабосвязная искусственная нейронная сеть*
 - *формальный нейрон*
 - *полносвязный формальный нейрон*
 - *сверточный формальный нейрон*
 - *рекуррентный формальный нейрон*
 - *синаптическая связь*
 - *параметр нейронной сети*
 - *настраиваемый параметр нейронной сети*
 - *весовой коэффициент*
 - *пороговое значение*
 - *ядро свертки*
 - *архитектурный параметр нейронной сети*
 - *количество слоев*
 - *количество формальных нейронов*
 - *количество синаптических связей*
 - *паттерн входной активности н.с.*
 - *признак*
 - *слой и.н.с.*
 - *полносвязный слой и.н.с*
 - *сверточный слой и.н.с*
 - *слой и.н.с. нелинейного преобразования*
 - *dropout слой и.н.с.*
 - *pooling слой и.н.с.*
- Э *исследуемое отношение'*:
- *формальный нейрон'*
 - *пороговый формальный нейрон'*
 - *синаптическая связь'*
 - *входное значение формального нейрона**
 - *выходное значение формального нейрона**
 - *функция активации**
 - *взвешенная сумма**
 - *распределяющий слой**
 - *обрабатывающий слой**
 - *выходной слой**

искусственная нейронная сеть

- :=*** [и.н.с.]
- :=*** [множество искусственных нейронных сетей]
- :=*** [нейронная сеть]
- ⇒*** *определение**:

[**искусственная нейронная сеть** – это совокупность нейронных элементов и связей между ними (Головкин В.А. НейроТехОбрДанных-2017кн).

Искусственная нейронная сеть состоит из **формальных нейронов**, которые связаны между собой посредством **синаптических связей**. Нейроны организованы в **слои**. Каждый нейрон слоя принимает сигналы со входящих в него синаптических связей, обрабатывает их единым образом с помощью заданной ему или всему слою **функции активации** и передает результат на выходящие из него синаптические связи.]

⇒ пояснение*:

[**искусственная нейронная сеть** – это биологически инспирированная математическая модель, обладающая обобщающей способностью после выполнения процедуры обучения. Под обобщающей способностью понимается способность модели выдавать корректные результаты для паттернов входной активности, не входящих в обучающую выборку.]

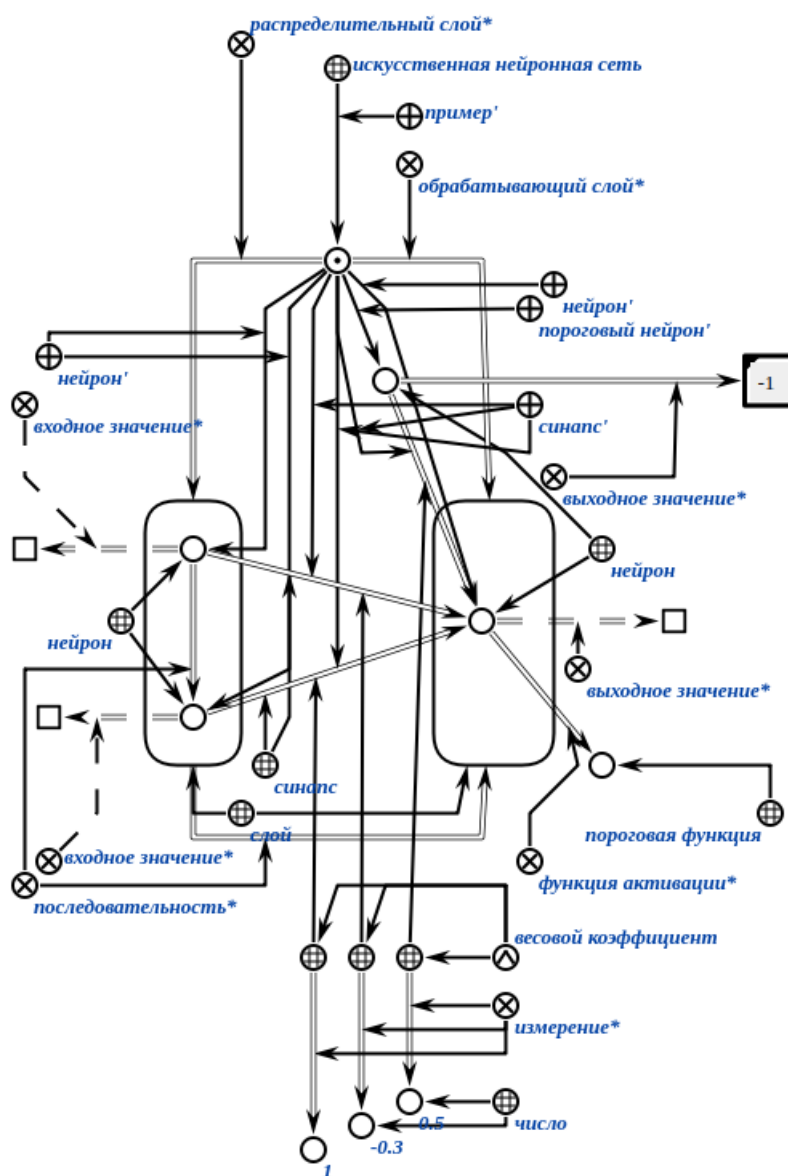
С математическая модель

⇒ пояснение*:

[**математическая модель** – это упрощенное описание объекта реального мира, выраженное с помощью математической символики]

⇒ описание примера*:

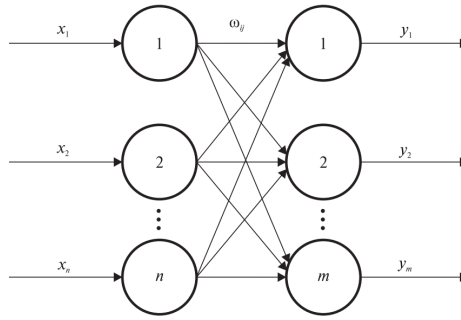
[



]

⇒ изображение*:

[



]

⇒ разбиение*:

Типология и.н.с. по признаку направленности связей

- = {
 - искусственная нейронная сеть с прямыми связями
 - ⇒ разбиение*:
 - {
 - перцептрон
 - ⇒ разбиение*:
 - {
 - перцептрон Розенблатта
 - автоэнкодерная искусственная нейронная сеть
 - машина опорных векторов
 - искусственная нейронная сеть радиально-базисных функций
 - искусственная нейронная сеть с обратными связями
 - ⇒ разбиение*:
 - {
 - нейронная сеть Хопфилда
 - нейронная сеть Хэмминга
 - рекуррентная искусственная нейронная сеть
 - ⇒ разбиение*:
 - {
 - искусственная нейронная сеть Джордана
 - искусственная нейронная сеть Элмана
 - мультирекуррентная нейронная сеть
 - LSTM-элемент
 - GRU-элемент

⇒ разбиение*:

Типология и.н.с. по признаку полноты связей

- = {
 - полносвязная искусственная нейронная сеть
 - слабосвязная искусственная нейронная сеть

⇒ решаемые задачи*:

задачи, которые могут быть решены с помощью и.н.с. с приемлемой точностью

- = {
 - задача классификации
 - ⊂ задача
 - ⇒ пояснение*:
 - [Задача построения классификатора, т.е. отображения $\tilde{c} : X \rightarrow C$, где $X \in \mathbb{R}^m$ – признаковое пространство п.в.а., $C = C_1, C_2, \dots, C_k$ – конечное и обычно небольшое множество меток классов.]
 - задача регрессии
 - ⊂ задача
 - ⇒ пояснение*:
 - [Задача построения оценочной функции по примерам $(x_i, f(x_i))$, где $f(x)$ – неизвестная функция]
 - ⇒ пояснение*:
 - [оценочная функция – отображение вида $\tilde{f} : X \rightarrow \mathbb{R}$, где $X \in \mathbb{R}^m$ – признаковое пространство п.в.а.]

- *задача кластеризация*
 С задача
 ⇒ *пояснение**:
 [Задача разбиения множества п.в.а. на группы (кластеры) по какой-либо метрике сходства.]
- *задача понижения размерности*
 С задача
 := [задача уменьшения размерности признакового пространства]
- *задача управления*
 С задача
- *задача фильтрации*
 С задача
- *задача детекции*
 С задача
 С задача классификации
 С задача регрессии
- *задача с ассоциативной памятью*
 С задача

формальный нейрон

:= [искусственный нейрон]

:= [нейрон]

:= [ф.н.]

:= [нейронный элемент]

:= [множество нейронов искусственных нейронных сетей]

:= [математическая модель реального биологического нейрона]

⇒ *примечание**:

[Отдельный формальный нейрон является искусственной нейронной сети с одним нейроном в единственном слое.]

С *искусственная нейронная сеть*

⇒ *пояснение**:

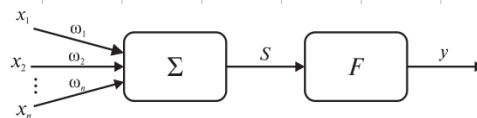
[**формальный нейрон** – это основной элемент *искусственной нейронной сети*, применяющий свою *функцию активации* (Головко.В.А..*НейроТехОбрДанных-2017кн*) к сумме произведений входных сигналов на весовые коэффициенты:

$$y = F \left(\sum_{i=1}^n w_i x_i - T \right) = F(WX - T)$$

где $X = (x_1, x_2, \dots, x_n)^T$ – вектор входного сигнала; $W = (w_1, w_2, \dots, w_n)$ – вектор весовых коэффициентов; T – пороговое значение; F – функция активации.]

⇒ *изображение**:

[



]

⇒ *примечание**:

[Формальные нейроны могут иметь полный набор связей с нейронами предшествующего слоя или неполный (разряженный) набор связей.]

⇒ *разбиение**:

{ • *полносвязный формальный нейрон*

:= [нейрон, у которого есть полный набор связей с нейронами предшествующего слоя]

⇒ *пояснение**:

[отдельный обрабатывающий элемент и.н.с., выполняющий функциональное преобразование взвешенной суммы элементов вектора входных значений с помощью функции активации]

• *сверточный формальный нейрон*

⇒ *пояснение**:

- [Отдельный обрабатывающий элемент и.н.с., выполняющий функциональное преобразование результата операции свертки матрицы входных значений с помощью функции активации.]
- ⇒ *примечание**:
[Сверточный формальный нейрон может быть представлен полносвязным формальным нейроном]
- *рекуррентный формальный нейрон*
⇒ *пояснение**:
[Формальный нейрон, имеющий обратную связь с самим собой или с другими нейронами и.н.с.]
- }

формальный нейрон'

- := [формальный нейронный элемент']
- := [нейронный элемент']
- := [нейрон']
- ∈ *ролевое отношение*
- ⇒ *первый домен**:
искусственная нейронная сеть
- ⇒ *второй домен**:
формальный нейрон
- ⇒ *область определения**:
искусственная нейронная сеть
- ⇒ *пояснение**:
[**формальный нейрон'** – ролевое отношение, связывающее искусственную нейронную сеть с ее нейроном.]

пороговый формальный нейрон'

- := [пороговый нейронный элемент']
- := [пороговый нейрон']
- ∈ *ролевое отношение*
- ⇒ *первый домен**:
искусственная нейронная сеть
- ⇒ *второй домен**:
формальный нейрон
- ⇒ *область определения**:
искусственная нейронная сеть
- ⇒ *пояснение**:
[**пороговый формальный нейрон'** – ролевое отношение, связывающее искусственную нейронную сеть с таким ее нейроном, выходное значение которого всегда равно -1.]
- ⇒ *пояснение**:
[Весовой коэффициент синаптической связи, выходящей из такого нейрона, является порогом для нейрона, в который данная синаптическая связь входит.]

синаптическая связь

- := [синапс]
- ⊂ *ориентированная пара*
- ⇒ *определение**:
[**синаптическая связь** – ориентированная пара, первым компонентом которой является нейрон, из которого исходит сигнал, а вторым компонентом – нейрон, который принимает этот сигнал.]

синаптическая связь'

- := [синапс']
- ∈ *ролевое отношение*
- ⇒ *первый домен**:
искусственная нейронная сеть
- ⇒ *второй домен**:
синаптическая связь
- ⇒ *область определения**:
{ }
- ← *объединение**:
{ • *искусственная нейронная сеть*
• *синапс*

}
 ⇒ *определение**:
 [синаптическая связь' – ролевое отношение, связывающее искусственную нейронную сеть с ее синапсом.]

параметр и.н.с.

С параметр

⇒ *разбиение**:

{ • *настраиваемый параметр и.н.с.*

:= [параметр и.н.с., значение которого изменяется в ходе обучения]

⇒ *разбиение**:

- { • *весовой коэффициент синаптической связи*
 • *пороговое значение*
 • *ядро свертки*

:= [квадратная матрица произвольного порядка, элементы которой изменяются в процессе обучения и.н.с.]

⇒ *примечание**:

[если сверточный формальный нейрон представить в виде полносвязного формального нейрона, соответствующее ядро свертки преобразуется в вектор весовых коэффициентов]

}

- *архитектурный параметр и.н.с.*

⇒ *примечание**:

[Параметр и.н.с., определяющий ее архитектуру.]

⇒ *разбиение**:

- { • *количество слоев*
 • *количество нейронов*
 • *количество синапсов*

}

}

весовой коэффициент синаптической связи

:= [вес синапса]

:= [сила синаптической связи]

С *настраиваемый параметр*

⇒ *пояснение**:

[*весовой коэффициент синаптической связи* – это числовой коэффициент, который ставится в соответствие каждому синапсу нейронной сети и изменяется в процессе обучения.]

⇒ *примечание**:

[Если сила синаптической связи отрицательна, то она называется *тормозящей*. В противном случае она является *усиливающей*.]

входное значение формального нейрона*

:= [входное значение нейрона*]

:= [входное значение*]

∈ *неролевое отношение*

∈ *бинарное отношение*

⇒ *первый домен**:

формальный нейрон

⇒ *второй домен**:

число

⇒ *область определения**:

{ }

⇐ *объединение**:

- { • *формальный нейрон*
 • *число*

}

⇒ *пояснение**:

[*входное значение формального нейрона** – неролевое отношение, связывающее нейрон входного слоя со значением признака п.в.а., который подается на вход нейронной сети.]

⇒ *теоретическая неточность**:

[Использование множества как формы представления входных данных является серьезным допущением, так как на практике входные данные структурированы более сложно – в многомерные массивы. Самым близким теоретическим аналогом здесь выступает тензор. К сожалению, описание теории нейронных сетей с помощью тензорного исчисления в литературе как таковое отсутствует, но активно используется на практике: например, во многих разрабатываемых нейросетевых фреймворках. Формализация нейронных сетей с помощью тензоров видится авторам наиболее вероятным направлением работы в ближайших изданиях *стандарта OSTIS*.]

паттерн входной активности и.н.с.

:= [п.в.а.]

∈ *мультимножество*

∈ *кортеж*

⇒ *пояснение**:

[**паттерн входной активности и.н.с.** – ориентированное мультимножество численных значений признаков некоторого объекта, которые могут выступать в качестве входных значений нейронов.]

⇒ *примечание**:

[В текущей версии *Стандарта OSTIS* предполагается, что п.в.а. содержит только преобработанные данные, то есть данные приведенные к численному виду и, возможно, преобразованные с помощью известных статистических методов (например, нормирования).]

признак

:= [feature]

:= [множество признаков]

⊂ *ролевое отношение*

⇒ *пояснение**:

[**признак** – множество ролевых отношений, каждое из которых связывает некоторый п.в.а. с численным значением, которое характеризует данный п.в.а. с какой-либо стороны.]

функция активации*

:= [функция активации нейрона*]

∈ *неролевое отношение*

∈ *бинарное отношение*

⇒ *пояснение**:

[**функция активации*** – неролевое отношение, связывающее формальный нейрон с функцией, результат применения которой к *взвешенной сумме нейрона* определяет его *выходное значение*.]

⇒ *область определения**:

{ }

← *объединение**:

- *формальный нейрон*
- *функция*

}

⇒ *первый домен**:

формальный нейрон

⇒ *второй домен**:

функция

⇒ *разбиение**:

- *линейная функция*

⇒ *формула**:

[

$$y = kS$$

где k – коэффициент наклона прямой, S – в.с.]

- *пороговая функция*

⇒ *формула**:

[

$$y = \text{sign}(S) = \begin{cases} 1, S > 0, \\ 0, S \leq 0 \end{cases}$$

]

- *сигмоидная функция*

⇒ *формула**:

[$y = \frac{1}{1 + e^{-cS}}$
	где $c > 0$ – коэффициент, характеризующий ширину сигмоидной функции по оси абсцисс, S – в.с.]
•	<i>функция гиперболического тангенса</i>
⇒	<i>формула*</i> :
[$y = \frac{e^{cS} - e^{-cS}}{e^{cS} + e^{-cS}}$
	где $c > 0$ – коэффициент, характеризующий ширину сигмоидной функции по оси абсцисс, S – в.с.]
•	<i>функция softmax</i>
⇒	<i>формула*</i> :
[$y_j = \text{softmax}(S_j) = \frac{e^{S_j}}{\sum_j e^{S_j}}$
	где S_j – в.с. j -го выходного нейрона]
•	<i>функция ReLU</i>
⇒	<i>формула*</i> :
[$y = F(S) = \begin{cases} S, S > 0, \\ kS, S \leq 0 \end{cases}$
	где $k = 0$ или принимает небольшое значение, например, 0.01 или 0.001.]
}	

взвешенная сумма*

:= [взвешенная сумма входных значений*]

:= [в.с.]

∈ нероловое отношение

∈ бинарное отношение

⇒ пояснение*:

[**взвешенная сумма*** – нероловое отношение, связывающее формальный нейрон с числом, являющимся суммой произведений входных сигналов на весовые коэффициенты входящих в нейрон синаптических связей.]

⇒ область определения*:

{

← объединение*:

• формальный нейрон

• число

}

⇒ первый домен*:

формальный нейрон

⇒ второй домен*:

число

⇒ формула*:

$$S = \sum_{i=1}^n w_i x_i - T$$

где n – размерность вектора входных значений, w_i – i -тый элемент вектора весовых коэффициентов, x_i – i -тый элемент вектора входных значений, T – пороговое значение.

выходное значение формального нейрона*

:= [выходное значение нейрона*]

:= [выходное значение*]

∈ нероловое отношение

∈ бинарное отношение

⇒ первый домен*:

⇒	<i>формальный нейрон</i>
⇒	<i>второй домен*</i> :
	<i>число</i>
⇒	<i>область определения*</i> :
	{ }
←	<i>объединение*</i> :
	{ • <i>формальный нейрон</i>
	• <i>число</i>
	}
⇒	<i>пояснение*</i> :
	[<i>входное значение*</i> – нерелевое отношение, связывающее нейрон с числом, являющимся результатом применения функции активации нейрона к его взвешенной сумме.]
⇒	<i>примечание*</i> :
	[Выходное значение нейрона является одним из входных сигналов для всех нейронов, в которые ведут выходящие из данного нейрона синапсы.]
слой и.н.с.	
:=	[слой]
:=	[слой искусственной нейронной сети]
:=	[множество слоев искусственных нейронных сетей]
⇒	<i>примечание*</i> :
	[отдельный слой является искусственной нейронной сетью с одним слоем]
С	<i>искусственная нейронная сеть</i>
⇒	<i>пояснение*</i> :
	[слой и.н.с – это множество нейронных элементов, на которые в каждый такт времени параллельно поступает информация от других нейронных элементов сети (<i>Головко.В.А..НейроТехОбрДанных-2017кн</i>)]
⇒	<i>пояснение*</i> :
	[слой и.н.с. – это множество формальных нейронов, осуществляющих параллельную независимую обработку вектора или матрицы входных значений]
⇒	<i>примечание*</i> :
	[функция активации слоя является функцией активации всех формальных нейронов этого слоя]
⇒	<i>примечание*</i> :
	[конфигурация слоя задается типом, количеством формальных нейронов, функцией активации]
⇒	<i>примечание*</i> :
	[описание последовательности слоев и.н.с. с конфигурацией каждого слоя задает архитектуру и.н.с.]
⇒	<i>разбиение*</i> :
	{ • <i>полносвязный слой и.н.с.</i>
	:= [слой, в котором каждый нейрон имеет связь с каждым нейроном предшествующего слоя]
	:= [слой, в котором каждый нейрон является полносвязным]
	• <i>сверточный слой и.н.с.</i>
	:= [слой, в котором каждый нейрон является сверточным]
	• <i>слой и.н.с. нелинейного преобразования</i>
	:= [слой, осуществляющий нелинейное преобразование входных данных]
⇒	<i>пояснение*</i> :
	[Как правило, выделяются в отдельные слои только в программных реализациях. Фактически рассматриваются как финальный этап расчета выходной активности любого нейрона – применение функции активации.]
⇒	<i>примечание*</i> :
	[не изменяет размерность входных данных]
	• <i>dropout слой и.н.с.</i>
	:= [слой, реализующий технику регуляризации dropout]
⇒	<i>примечание*</i> :
	[Данный тип слоя функционирует только во время обучения и.н.с.]
⇒	<i>пояснение*</i> :
	[Поскольку полносвязные слои имеют большое количество настраиваемых параметров, они подвержены эффекту переобучения. Один из способов устранить такой негативный эффект – выполнить частичный отсев результатов на выходе полносвязного слоя. На этапе обучения техника dropout позволяет отбросить выходную активность некоторых нейронов с опреде-

ленной, заданной вероятностью. Выходная активность “отброшенных” нейронов полагается равной нулю.]

- *pooling* слой и.н.с.
:= [подвыборочный слой]
:= [объединяющий слой]
:= [слой, осуществляющий уменьшение размерности входных данных]
- *слой и.н.с. батч-нормализации*

}

распределяющий слой*

:= [входной слой*]

∈ *нероловое отношение*

∈ *бинарное отношение*

⇒ *определение**:

[**распределяющий слой*** – нероловое отношение, связывающее искусственную нейронную сеть с ее слоем, нейроны которого принимают входные значения всей нейронной сети.]

⇒ *область определения**:

искусственная нейронная сеть

⇒ *первый домен**:

искусственная нейронная сеть

⇒ *второй домен**:

слой и.н.с.

обрабатывающий слой*

∈ *нероловое отношение*

∈ *бинарное отношение*

⇒ *определение**:

[**обрабатывающий слой*** – нероловое отношение, связывающее искусственную нейронную сеть с ее слоем, нейроны которого принимают на вход выходные значения нейронов предыдущего слоя.]

⇒ *область определения**:

искусственная нейронная сеть

⇒ *первый домен**:

искусственная нейронная сеть

⇒ *второй домен**:

слой и.н.с.

выходной слой*

∈ *нероловое отношение*

∈ *бинарное отношение*

⇒ *пояснение**:

[**выходной слой*** – нероловое отношение, связывающее искусственную нейронную сеть с ее слоем, выходные значения нейронов которого являются выходными значениями всей нейронной сети.]

⇒ *область определения**:

искусственная нейронная сеть

⇒ *первый домен**:

искусственная нейронная сеть

⇒ *второй домен**:

слой и.н.с.

} /* Завершили Сегмент “Предметная область и онтология искусственных нейронных сетей” */

/* Сегмент *****/

Предметная область и онтология действий по обработке искусственной нейронной сети

⊃=

{

Предметная область действий по обработке искусственных нейронных сетей

- := [Предметная область действий по обработке и.н.с.]
- ∈ предметная область
- ⊃ максимальный класс объектов исследования':
- действие по обработке искусственных нейронных сетей
- ⊃ класс объектов исследования':
- действие по обработке искусственных нейронных сетей
 - действие конфигурации весовых коэффициентов и.н.с.
 - действие конфигурации и.н.с.
 - действие интерпретации и.н.с.
 - метод обучения и.н.с.
 - метод обучения с учителем
 - метод обратного распространения ошибки
 - метод обучения без учителя
 - метод оптимизации
 - функция потерь
 - параметр обучения
 - скорость обучения
 - моментный параметр
 - параметр регуляризации
 - размер группы обучения
 - количество эпох обучения
 - выборка
- ⊃ исследуемое отношение':
- обучающая выборка'
 - тестовая выборка'
 - валидационная выборка'
 - метод обучения'
 - метод оптимизации'
 - функция потерь'

действие по обработке искусственной нейронной сети

- := [действие по обработке и.н.с.]
- := [действие с искусственной нейронной сетью]
- ⊂ действие
- ⇒ пояснение*:
- [В зависимости от того, является ли искусственная нейронная сеть знаком внешней по отношению к памяти системы сущности, элементы множества действие по обработке и.н.с. являются либо элементами множества **действие, выполняемое кибернетической системой в своей внешней среде**, либо элементом множества **действие, выполняемое кибернетической системой в собственной памяти. .**]
- ⇒ разбиение*:
- { • действие конфигурации и.н.с.
 - ⇒ разбиение*:
 - { • действие создания и.н.с.
 - действие редактирования и.н.с.
 - действие удаления и.н.с.
 - действие конфигурации слоя и.н.с.
 - ⇒ разбиение*:
 - { • действие добавления слоя в и.н.с.
 - действие редактирования слоя и.н.с.
 - действие удаления слоя и.н.с.
 - действие установки функции активации нейронов слоя и.н.с.
 - действие конфигурации нейрона в слое и.н.с.
 - ⇒ разбиение*:
 - { • действие добавления нейрона в слой и.н.с.
 - действие редактирования нейрона в слое и.н.с.
 - действие удаления нейрона из слоя и.н.с.
 - действие установки функции активации нейрона в слое и.н.с.

- действие конфигурации весовых коэффициентов и.н.с.
 - ⊃ действие обучения и.н.с.
 - ⊃ действие начальной инициализации весов и.н.с.
 - ⊃ действие начальной инициализации весов нейронов слоя и.н.с.
 - ⊃ действие начальной инициализации весов нейрона и.н.с.
- действие интерпретации и.н.с.

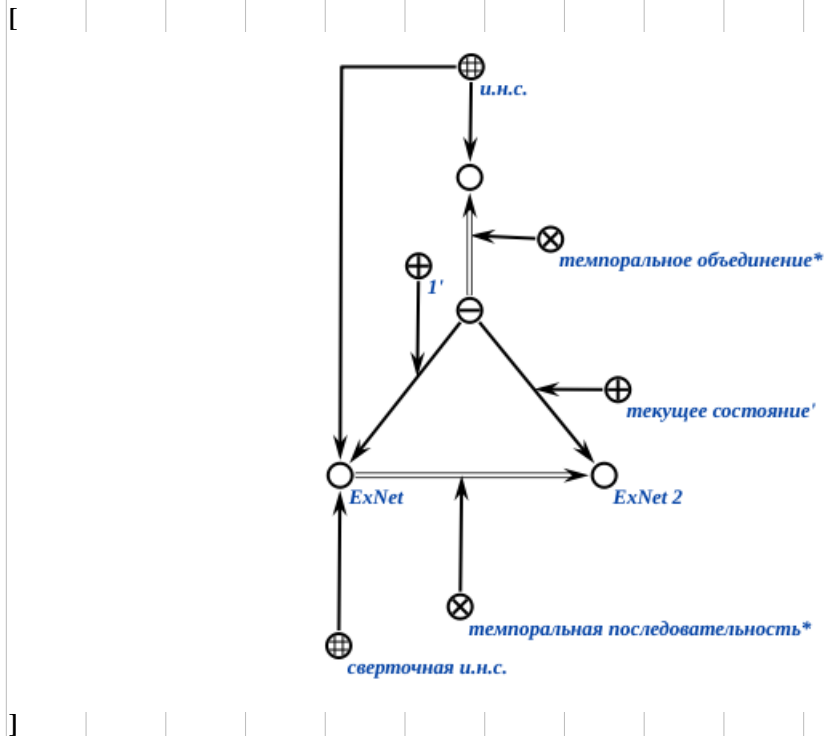
⇒ примечание*:

[Действия по обработке и.н.с осуществляет соответствующий коллектив агентов.]

⇒ пояснение*:

[Так как в результате действий по обработке и.н.с объект этих действий, конкретная и.н.с, может существенно меняться (меняется конфигурация сети, ее весовые коэффициенты), то и.н.с представляется в базе знаний как темпоральное объединение всех ее версий. Каждая версия является и.н.с. и темпоральной сущностью. На множестве этих темпоральных сущностей задается темпоральная последовательность с указанием первой и последней версии. Для каждой версии описываются специфичные знания. Общие для всех версий знания описываются для и.н.с, являющейся темпоральным объединением всех версий.]

⇒ пример*:



действие обучения и.н.с.

:= [действие обучения искусственной нейронной сети]

⊃ действие конфигурации весовых коэффициентов и.н.с.

⇒ определение*:

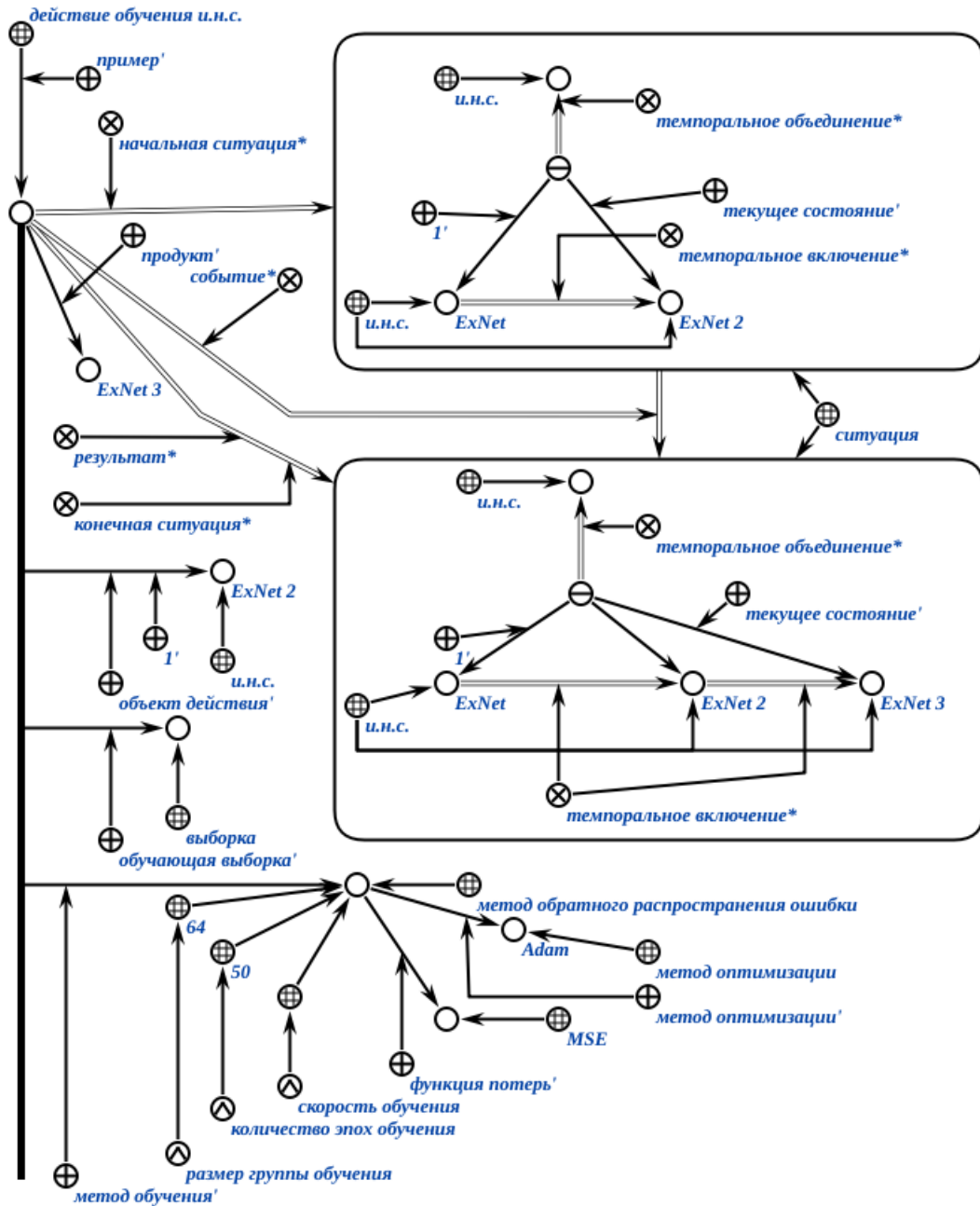
[**действие обучения и.н.с.** – действие, в ходе которого реализуется определенный метод обучения и.н.с. с заданными параметрами обучения и.н.с, методом оптимизации и функцией потерь.]

⇒ известные проблемы*:

- [Переобучение – проблема, возникающая при обучении и.н.с., заключающаяся в том, что сеть хорошо адаптируется к п.в.а. из обучающей выборки, при этом теряя способность к обобщению. Переобучение возникает из-за применения неоправданно сложной модели при обучении и.н.с. Это происходит, когда количество настраиваемых параметров и.н.с. намного больше размера обучающей выборки. Возможные варианты решения проблемы заключаются в упрощении модели, увеличении выборки, использовании регуляризации (параметр регуляризации, техника dropout и т.д.). Обнаружение переобученности сложнее, чем недообученности. Как правило, для этого применяется кросс-валидация на валидационной выборке, позволяющая оценить момент завершения процесса обучения. Идеальным вариантом является достижение баланса между переобученностью и недообученностью.]

- [Недообучение – проблема, возникающая при обучении и.н.с., заключающаяся в том, что сеть дает одинаково плохие результаты на обучающей и контрольной выборках. Чаще всего такого рода проблема возникает при недостаточном времени, затраченном на обучение модели. Однако это может быть вызвано и слишком простой архитектурой модели либо малым размером обучающей выборки. Соответственно решение, которое может быть принято ML-инженером, заключается в устранении этих недостатков: увеличение времени обучения, использование модели с большим числом настраиваемых параметров, увеличение размера обучающей выборки, а также уменьшение регуляризации и более тщательный отбор признаков для обучающих примеров.]

⇒ описание примера*:
[



выборка

С множество
⇒ пояснение*:

[выборка – множество п.в.а., используемых в процессе обучения, тестирования и архитектурной настройки и.н.с.]

обучающая выборка'

:= [training set']
 ∈ ролевое отношение
 ⇒ первый домен*:
 действие обучения и.н.с.
 ⇒ второй домен*:
 выборка
 ⇒ область определения*:
 {}
 ⇐ объединение*:
 { • действие обучения и.н.с.
 • выборка
 }
 ⇒ пояснение*:

[**обучающая выборка'** – ролевое отношение, связывающее действие обучения и.н.с. с выборкой, используемой для изменения настраиваемых параметров и.н.с. в процессе ее обучения.]

тестовая выборка'

:= [test set']
 ∈ ролевое отношение
 ⇒ первый домен*:
 действие обучения и.н.с.
 ⇒ второй домен*:
 выборка
 ⇒ область определения*:
 {}
 ⇐ объединение*:
 { • действие обучения и.н.с.
 • выборка
 }
 ⇒ пояснение*:

[**тестовая выборка'** – ролевое отношение, связывающее действие обучения и.н.с. с выборкой, используемой для проверки обобщающей способности обученной и.н.с.]

⇒ примечание*:
 [Элементы контрольной выборки не используются в процессе обучения.]

валидационная выборка'

∈ ролевое отношение
 ⇒ первый домен*:
 действие обучения и.н.с.
 ⇒ второй домен*:
 выборка
 ⇒ область определения*:
 {}
 ⇐ объединение*:
 { • действие обучения и.н.с.
 • выборка
 }
 ⇒ пояснение*:

[**валидационная выборка'** – ролевое отношение, связывающее действие обучения и.н.с. с выборкой, используемой для определения (настройки) архитектурных параметров и.н.с. и параметров обучения.]

⇒ примечание*:
 [Элементы валидационной выборки не используются в процессе обучения (не входят в обучающую выборку).]

метод обучения и.н.с.

С метод
 ⇒ пояснение*:

[**метод обучения и.н.с.** – метод итеративного поиска оптимальных значений настраиваемых параметров и.н.с., минимизирующих некоторую заданную функцию потерь.]

⇒ примечание*:

[Стоит отметить, что хотя целью применения метода обучения является минимизация функции потерь, “полезность” полученной после обучения модели можно оценить только по достигнутому уровню ее обобщающей способности.]

⊃ метод обучения с учителем

⇒ пояснение*:

[**метод обучения с учителем** – метод обучения с использованием заданных целевых переменных.]

⊃ метод обратного распространения ошибки

:= [м.о.р.о.]

⇒ алгоритм*:

[

Data: X – данные, E_t – желаемый отклик (метки), E_m – желаемая ошибка (в соответствии с выбранной функцией потерь)

Result: обученная нейронная сеть Net
инициализация весов W и порогов T ;

repeat

foreach $x \in X$ & $e \in E_t$ **do**

 фаза прямого распространения сигнала; вычисляются активации для всех слоев и.н.с.;

 фаза обратного распространения ошибки; вычисляются ошибки для последнего слоя и всех предшествующих слоев;

 изменение настраиваемых параметров и.н.с. в соответствии с вычисленными ошибками;

end

 вычисление общей ошибки E на данной эпохе;

until $E < E_m$;

]

⇒ примечание*:

[м.о.р.о. использует заданный метод оптимизации и заданную функцию потерь для реализации фазы обратного распространения ошибки и изменения настраиваемых параметров и.н.с. Одним из самых распространенных методов оптимизации является метод стохастического градиентного спуска. Приведенный м.о.р.о. используется для реализации последовательного варианта обучения.]

⇒ примечание*:

[Следует также отметить, что несмотря на то, что метод отнесен к методам обучения с учителем, в случае использования м.о.р.о. для обучения автокодировщиков в классических публикациях он рассматривается как метод обучения без учителя, поскольку в данном случае размеченные данные отсутствуют.]

⊃ метод обучения без учителя

⇒ пояснение*:

[**метод обучения без учителя** – метод обучения без использования заданных целевых переменных (в режиме самоорганизации)]

⇒ пояснение*:

[В ходе выполнения алгоритма метода обучения без учителя выявляются полезные структурные свойства набора. Неформально его понимают как метод для извлечения информации из распределения, выборка для которого не была вручную аннотирована человеком (*Гудфеллоу.Я..ГлубОбуч-2017кн*).]

метод обучения'

∈ ролевое отношение

⇒ первый домен*:

действие обучения и.н.с.

⇒ второй домен*:

метод обучения и.н.с.

⇒ область определения*:

{

 ← объединение*:

 { • действие обучения и.н.с.

 • метод обучения и.н.с.

 }

⇒ пояснение*:

[**метод обучения'** – ролевое отношение, связывающее действие обучения и.н.с. с методом обучения, используемым для обучения и.н.с. в рамках этого действия.]

метод оптимизации

С метод

⇒ определение*:

[метод оптимизации – метод для минимизации целевой функции потерь при обучении и.н.с.]

⇒ включение*:

• SGD

:= [стохастический градиентный спуск]

:= [с.г.с.]

:= [stochastic gradient descent]

⇒ примечание*:

[В методе стохастического градиентного спуска корректировка настраиваемых параметров и.н.с. выполняется в направлении максимального уменьшения функции стоимости, т.е. в направлении, противоположном вектору градиента функции потерь (Хайкин.С.НейрСети-2006кн)]

• Nesterov method

:= [метод Нестерова]

⇒ примечание*:

[Обучение методом с.г.с. иногда происходит очень медленно. Импульсный метод позволяет ускорить обучение, особенно в условиях высокой кривизны, небольших, но устойчивых градиентов или зашумленных градиентов. В импульсном методе вычисляется экспоненциально затухающее скользящее среднее прошлых градиентов и продолжается движение в этом направлении. Метод Нестерова является вариантом импульсного алгоритма, в котором градиент вычисляется после применения текущей скорости (Гудфеллоу.Я..ГлубОбуч-2017кн)]

• AdaGrad

:= [adaptive gradient]

⇒ примечание*:

[Данный метод по отдельности адаптирует скорости обучения всех настраиваемых параметров и.н.с., умножая их на коэффициент, обратно пропорциональный квадратному корню из суммы всех прошлых значений квадрата градиента (Duchi.J..AdaptiveSubgradMethods-2011cm)]

• RMSProp

:= [root mean square propagation]

⇒ примечание*:

[Данный метод является модификацией AdaGrad, которая позволяет улучшить его поведение в невыпуклом случае путем изменения способа агрегирования градиента на экспоненциально взвешенное скользящее среднее. Использование экспоненциально взвешенного скользящего среднего гарантирует повышение скорости сходимости после обнаружения выпуклой впадины, как если бы внутри этой впадины алгоритм AdaGrad был инициализирован заново (Гудфеллоу.Я..ГлубОбуч-2017кн)]

• Adam

:= [adaptive moments]

⇒ примечание*:

[Данный метод можно рассматривать как комбинацию RMSProp и AdaGrad (Kingma.D.Adam-2014cm). Помимо усредненного первого момента, данный метод использует усредненное значение вторых моментов градиентов]

⇒ примечание*:

[Успешность применения методов оптимизации зависит главным образом от знакомства пользователя с соответствующим алгоритмом (Гудфеллоу.Я..ГлубОбуч-2017кн).]

метод оптимизации'

∈ ролевое отношение

⇒ первый домен*:

метод обучения и.н.с.

⇒ второй домен*:

метод оптимизации

⇒ область определения*:

{

← объединение*:

{• метод обучения и.н.с.

	<ul style="list-style-type: none"> метод оптимизации
⇒	<p>пояснение*:</p> <p>[метод оптимизации['] – ролевое отношение, связывающее метод обучения и.н.с. с методом оптимизации, используемым для обучения и.н.с. с помощью данного метода.]</p>
	<p>функция потерь</p>
С	функция
⇒	<p>пояснение*:</p> <p>[функция потерь – функция, используемая для вычисления ошибки, рассчитываемой как разница между фактическим эталонным значением и прогнозируемым значением, получаемым и.н.с.]</p>
⇒	<p>включение*:</p> <ul style="list-style-type: none"> <p>MSE</p> <p>:= [mean square error]</p> <p>:= [средняя квадратичная ошибка]</p> <p>⇒ формула*:</p> <p>[</p> $MSE = \frac{1}{L} \sum_{l=1}^L \sum_{i=1}^m (y_i^l - e_i^l)^2$ <p>где y_i^l – прогноз модели, e_i^l – ожидаемый (эталонный) результат, m – размерность выходного вектора, L – объем обучающей выборки.]</p> <p>BCE</p> <p>:= [binary cross entropy]</p> <p>:= [бинарная кросс-энтропия]</p> <p>⇒ формула*:</p> <p>[</p> $BCE = - \sum_{l=1}^L (e^l \log(y^l) + (1 - e^l) \log(1 - y^l))$ <p>где y^l – прогноз модели, e^l – ожидаемый (эталонный) результат: 0 или 1, L – объем обучающей выборки.]</p> <p>⇒ примечание*:</p> <p>[для бинарной кросс-энтропии в выходном слое и.н.с. будет находиться один нейрон]</p> <p>MCE</p> <p>:= [multi-class cross entropy]</p> <p>:= [мультиклассовая кросс-энтропия]</p> <p>⇒ формула*:</p> <p>[</p> $MCE = - \sum_{l=1}^L \sum_{i=1}^m e_i^l \log(y_i^l)$ <p>где y_i^l – прогноз модели, e_i^l – ожидаемый (эталонный результат), m – размерность выходного вектора]</p> <p>⇒ примечание*:</p> <p>[для мультиклассовой кросс-энтропии количество нейронов в выходном слое и.н.с. совпадает с количеством классов]</p>
⇒	<p>примечание*:</p> <p>[Для решения задачи классификации рекомендуется использовать бинарную или мультиклассовую кросс-энтропийную функцию потерь, для решения задачи регрессии рекомендуется использовать среднюю квадратичную ошибку.]</p>
	<p>функция потерь[']</p>
∈	ролевое отношение
⇒	<p>первый домен*:</p> <p>метод обучения и.н.с.</p>
⇒	<p>второй домен*:</p> <p>функция потерь</p>
⇒	<p>область определения*:</p> <p>{ }</p>

←	<i>объединение*</i> : { <ul style="list-style-type: none"> • <i>метод обучения и.н.с.</i> • <i>функция потерь</i> }
⇒	<i>пояснение*</i> : [функция потерь – ролевое отношение, связывающее метод обучения и.н.с. с функцией потерь, используемым для обучения и.н.с. с помощью данного метода.]
	параметр обучения
:=	[группа наиболее общих параметров метода обучения и.н.с.]
⇒	<i>состав группы параметров обучения*</i> : { <ul style="list-style-type: none"> • <i>скорость обучения</i> ⇒ <i>пояснение*</i>: [скорость обучения – параметр, определяющий скорость изменения параметров и.н.с. в процессе обучения.] • <i>моментный параметр</i> := [момент] := [momentum] ⇒ <i>пояснение*</i>: [моментный параметр – параметр, используемый в процессе обучения для устранения проблемы “застревания” алгоритма обучения в локальных минимумах минимизируемой функции потерь.] ⇒ <i>пояснение*</i>: [При обучении и.н.с. частой является ситуация остановки процесса в определенной точке локального минимума без достижения желаемого уровня обобщающей способности и.н.с. Для устранения такого нежелательного явления вводится дополнительный параметр (момент) позволяющий алгоритму обучения “перескочить” через локальный минимум и продолжить процесс.] • <i>параметр регуляризации</i> ⇒ <i>пояснение*</i>: [параметр регуляризации – параметр, применяемый для контроля уровня переобучения и.н.с.] ⇒ <i>пояснение*</i>: [регуляризация – добавление дополнительных ограничений к правилам изменения настраиваемых параметров и.н.с. с целью предотвратить переобучение.] • <i>размер группы обучения</i> ⇒ <i>пояснение*</i>: [размер группы обучения – размер группы п.в.а., которая используется для изменения параметров и.н.с. на каждом элементарном шаге обучения.] • <i>количество эпох обучения</i> ⇒ <i>пояснение*</i>: [эпоха обучения – одна итерация алгоритма обучения, в ходе которой все обучающие п.в.а. из обучающей выборки были однократно использованы.] }
}	/* Завершили Сегмент “Предметная область и онтология действий по обработке искусственной нейронной сети” */
}	/* Завершили Раздел “Предметная область и онтология искусственных нейронных сетей и соответствующая им предметная область и онтология действий по обработке искусственных нейронных сетей” */

/ Раздел *****/*

Предметная область и онтология интерфейсов *ostis-систем*

⊃=

{

⇒

соавтор:*

Садовский М.Е.

⇒

дочерний раздел:*

- *Предметная область и онтологий интерфейсных действий пользователей *ostis-систем**
- *Предметная область и онтология естественных языков*

Предметная область интерфейсов *ostis-систем*

∈ *предметная область*

⊃ *максимальный класс объектов исследования¹:*

пользовательский интерфейс

⊃

класс объектов исследования¹:

- *командный пользовательский интерфейс*
- *графический пользовательский интерфейс*
- *WIMP-интерфейс*
- *SILK-интерфейс*
- *естественно-языковой интерфейс*
- *речевой интерфейс*
- *пользовательский интерфейс *ostis-системы**
- *компонент пользовательского интерфейса*
- *атомарный компонент пользовательского интерфейса*
- *неатомарный компонент пользовательского интерфейса*
- *визуальная часть пользовательского интерфейса *ostis-системы**
- *компонент пользовательского интерфейса для представления*
- *компонент вывода*
- *компонент выполнения*
- *параграф*
- *декоративный компонент пользовательского интерфейса*
- *контейнер*
- *меню*
- *строка меню*
- *панель инструментов*
- *панель вкладок*
- *окно*
- *модальное окно*
- *немодальное окно*
- *интерактивный компонент пользовательского интерфейса*
- *флаговая кнопка*
- *радиокнопка*
- *переключатель*
- *кнопка-счетчик*
- *полоса прокрутки*
- *кнопка*

пользовательский интерфейс

⊃ *командный пользовательский интерфейс*

⊃ *графический пользовательский интерфейс*

⊃ *WIMP-интерфейс*

⊃ *пользовательский интерфейс *ostis-системы**

⊃ *Пользовательский интерфейс *Метасистемы IMS.ostis**

⊃ *Пользовательский интерфейс *ИСС по геометрии**

:= [Пользовательский интерфейс интеллектуальной справочной системы по геометрии]

⊃ *Пользовательский интерфейс *ИСС по дискретной математике**

:= [Пользовательский интерфейс интеллектуальной справочной системы по дискретной математике]

- ⇒ *Пользовательский интерфейс ИСС по географии*
:= [Пользовательский интерфейс интеллектуальной справочной системы по географии]
- ⇒ *Пользовательский интерфейс ИСС по искусственным нейронным сетям*
:= [Пользовательский интерфейс интеллектуальной справочной системы по искусственным нейронным сетям]
- ⇒ *Пользовательский интерфейс ИСС по лингвистике*
:= [Пользовательский интерфейс интеллектуальной справочной системы по лингвистике]
- ⊃ *SILK-интерфейс*
:= [(Speech – речь, Image – образ, Language – язык, Knowledge – знание)]
 - ⊃ *естественно-языковой интерфейс*
 - ⊃ *речевой интерфейс*

пользовательский интерфейс⇒ *пояснение**:

[*пользовательский интерфейс* – один из наиболее важных компонентов компьютерной системы. Представляет собой совокупность аппаратных и программных средств, обеспечивающих обмен информацией между пользователем и компьютерной системой.]

командный пользовательский интерфейс⇒ *пояснение**:

[*командный пользовательский интерфейс* – пользовательский интерфейс, при котором обмен информацией между компьютерной системой и пользователем осуществляется путем написания текстовых инструкций или команд.]

графический пользовательский интерфейс⇒ *пояснение**:

[*графический пользовательский интерфейс* – пользовательский интерфейс, при котором обмен информацией между компьютерной системой и пользователем осуществляется при помощи графических компонентов компьютерной системы.]

WIMP-интерфейс⇒ *пояснение**:

[*WIMP-интерфейс* – пользовательский интерфейс, при котором обмен информацией между компьютерной системой и пользователем осуществляется в форме диалога при помощи окон, меню и других элементов управления.]

SILK-интерфейс⇒ *пояснение**:

[*SILK-интерфейс* – пользовательский интерфейс, наиболее приближенный к естественной для человека форме общения. Компьютерная система находит для себя команды, анализируя человеческую речь и находя в ней ключевые фразы. Результат выполнения команд преобразуется в понятную человеку форму, например, в естественно-языковую форму или изображение.]

естественно-языковой интерфейс⇒ *пояснение**:

[*естественно-языковой интерфейс* – SILK-интерфейс, обмен информацией между компьютерной системой и пользователем в котором происходит за счёт диалога. Диалог ведётся на одном из естественных языков.]

речевой интерфейс⇒ *пояснение**:

[*речевой интерфейс* – SILK-интерфейс, обмен информацией в котором происходит за счёт диалога, в процессе которого компьютерная система и пользователь общаются с помощью речи. Данный вид интерфейса наиболее приближен к естественному общению между людьми.]

пользовательский интерфейс ostis-системы⊃ *ostis-система*⇒ *пояснение**:

[*пользовательский интерфейс ostis-системы* представляет собой специализированную *ostis-систему*, ориентированную на решение интерфейсных задач, и имеющую в своем составе базу знаний и решатель задач пользовательского интерфейса *ostis-системы*.

Для решения задачи построения пользовательского интерфейса в базе знаний *пользовательского интерфейса ostis-системы* необходимо наличие *sc-модели компонентов пользовательского интерфейса, интерфейсных действий пользователей*, а также классификации *пользовательских интерфейсов* в целом. При проектировании интерфейса используется компонентный подход, который предполагает представление всего интерфейса приложения в виде отдельных специфицированных компонентов, которые могут разрабатываться и совершенствоваться независимо.]

компонент пользовательского интерфейса

⇒ *пояснение**:

[*компонент пользовательского интерфейса* – знак фрагмента базы знаний, имеющий определённую форму внешнего представления на экране.]

⇒ *разбиение**:

{

- *атомарный компонент пользовательского интерфейса*
- *неатомарный компонент пользовательского интерфейса*

}

атомарный компонент пользовательского интерфейса

⇒ *пояснение**:

[*атомарный компонент пользовательского интерфейса* – компонент пользовательского интерфейса, не содержащий в своём составе других компонентов пользовательского интерфейса.]

неатомарный компонент пользовательского интерфейса

⇒ *пояснение**:

[*неатомарный компонент пользовательского интерфейса* – компонент пользовательского интерфейса, состоящий из других компонентов пользовательского интерфейса.]

визуальная часть пользовательского интерфейса ostis-системы

⊂ *неатомарный компонент пользовательского интерфейса*

⇒ *пояснение**:

[*визуальная часть пользовательского интерфейса ostis-системы* – часть базы знаний пользовательского интерфейса *ostis-системы*, содержащая необходимые для отображения пользовательского интерфейса компоненты.]

компонент пользовательского интерфейса

:= [user interface component]

⊃ *компонент пользовательского интерфейса для отображения*

:= [presentation user interface component]

⊃ *компонент вывода*

:= [output]

⊃ *компонент вывода изображения*

:= [image-output]

⊃ *компонент вывода графической информации*

:= [graphical-output]

⊃ *диаграмма*

:= [chart]

⊃ *карта*

:= [map]

⊃ *индикатор выполнения*

:= [progress-bar]

⊃ *компонент вывода видео*

:= [video-output]

⊃ *компонент вывода звука*

:= [sound-output]

⊃ *компонент вывода текста*

:= [text-output]

⊃ *заголовок*

		:= [headline]	
	▷	<i>параграф</i>	
		:= [paragraph]	
	▷	<i>сообщение</i>	
		:= [message]	
▷	<i>декоративный компонент пользовательского интерфейса</i>		
	:=	[decorative user interface component]	
	▷	<i>разделитель</i>	
		:= [separator]	
	▷	<i>пустое пространство</i>	
		:= [blank-space]	
▷	<i>контейнер</i>		
	:=	[container]	
	▷	<i>меню</i>	
		:= [menu]	
	▷	<i>строка меню</i>	
		:= [menu-bar]	
	▷	<i>панель инструментов</i>	
		:= [tool-bar]	
	▷	<i>строка состояния</i>	
		:= [status-bar]	
	▷	<i>таблично-строковый контейнер</i>	
		:= [table-row-container]	
	▷	<i>списковый контейнер</i>	
		:= [list-container]	
	▷	<i>таблично-клеточный контейнер</i>	
		:= [table-cell-container]	
	▷	<i>древовидный контейнер</i>	
		:= [tree-container]	
	▷	<i>панель вкладок</i>	
		:= [tab-pane]	
	▷	<i>панель вращения</i>	
		:= [spin-pane]	
	▷	<i>узловой контейнер</i>	
		:= [tree-node-container]	
	▷	<i>панель прокрутки</i>	
		:= [scroll-pane]	
	▷	<i>окно</i>	
		:= [window]	
	▷	<i>модальное окно</i>	
		:= [modal-window]	
	▷	<i>немодальное окно</i>	
		:= [non-modal-window]	
▷	<i>интерактивный компонент пользовательского интерфейса</i>		
	:=	[interactive user interface component]	
	▷	<i>компонент ввода данных</i>	
		:= [data-input-component]	
	▷	<i>компонент ввода данных с прямой ответной реакцией</i>	
		:= [data-input-component-with-direct-feedback]	
	▷	<i>компонент ввода текста с прямой ответной реакцией</i>	
		:= [text-input-component-with-direct-feedback]	
	▷	<i>многострочное текстовое поле</i>	
		:= [multi-line-text-field]	
	▷	<i>однострочное текстовое поле</i>	
		:= [single-line-text-field]	
	▷	<i>ползунок</i>	

	:= [slider]
▷	<i>область рисования</i>
	:= [drawing-area]
▷	<i>компонент выбора</i>
	:= [selection-component]
▷	<i>компонент выбора нескольких значений</i>
	:= [selection-component-multiple-values]
▷	<i>компонент выбора одного значения</i>
	:= [selection-component-single-values]
▷	<i>компонент выбора данных</i>
	:= [selectable-data-representation]
▷	<i>флаговая кнопка</i>
	:= [check-box]
▷	<i>радиокнопка</i>
	:= [radio-button]
▷	<i>переключатель</i>
	:= [toggle-button]
▷	<i>выбираемый элемент</i>
	:= [selectable-item]
▷	<i>компонент ввода данных без прямой ответной реакции</i>
	:= [data-input-component-without-direct-feedback]
▷	<i>кнопка-счётчик</i>
	:= [spin-button]
▷	<i>компонент речевого ввода</i>
	:= [speech-input]
▷	<i>компонент ввода движений</i>
	:= [motion-input]
▷	<i>компонент для представления и взаимодействия с пользователем</i>
	:= [presentation-manipulation-component]
▷	<i>активирующий компонент</i>
	:= [activating-component]
▷	<i>компонент непрерывной манипуляции</i>
	:= [continuous-manipulation-component]
▷	<i>полоса прокрутки</i>
	:= [scrollbar]
▷	<i>компонент редактирования размера</i>
	:= [resizer]
▷	<i>компонент запроса действий</i>
	:= [operation-trigger-component]
▷	<i>компонент выбора команд</i>
	:= [command-selection-component]
▷	<i>кнопка</i>
	:= [button]
▷	<i>пункт меню</i>
	:= [menu-item]
▷	<i>компонент ввода команд</i>
	:= [command-input-component]

компонент пользовательского интерфейса для представления

⇒ *пояснение**:

[компонент пользовательского интерфейса для представления – компонент пользовательского интерфейса, не подразумевающий взаимодействия с пользователем.]

компонент вывода

⇒ *пояснение**:

[*компонент вывода* – компонент пользовательского интерфейса, предназначенный для представления информации.]

индикатор выполнения

⇒ *пояснение**:

[*индикатор выполнения* – компонент пользовательского интерфейса, предназначенный для отображения процента выполнения какой-либо задачи.]

параграф

⇒ *пояснение**:

[*параграф* – компонент пользовательского интерфейса, предназначенный для отображения блоков текста. Он отделяется от других блоков пустой строкой или первой строкой с отступом.]

декоративный компонент пользовательского интерфейса

⇒ *пояснение**:

[*декоративный компонент пользовательского интерфейса* – компонент пользовательского интерфейса, предназначенный для стилизации интерфейса.]

контейнер

⇒ *пояснение**:

[*контейнер* – компонент пользовательского интерфейса, задача которого состоит в размещении набора компонентов, включённых в его состав.]

меню

⇒ *пояснение**:

[*меню* – компонент пользовательского интерфейса, содержащий несколько вариантов для выбора пользователем.]

строка меню

⇒ *пояснение**:

[*строка меню* – горизонтальная полоса, содержащая ярлыки меню. Строка меню предоставляет пользователю место в окне, где можно найти большинство основных функций программы.]

панель инструментов

⇒ *пояснение**:

[*панель инструментов* – компонент пользовательского интерфейса, на котором размещаются элементы ввода или вывода данных.]

панель вкладок

⇒ *пояснение**:

[*панель вкладок* – контейнер, который может содержать несколько вкладок (секций) внутри, которые могут быть отображены, нажав на вкладке с названием в верхней части панели. Одновременно отображается только одна вкладка.]

окно

⇒ *пояснение**:

[*окно* – обособленная область экрана, содержащая различные элементы пользовательского интерфейса. Окна могут располагаться поверх друг друга.]

модальное окно

⇒ *пояснение**:

[*модальное окно* – окно, которое блокирует работу пользователя с системой до тех пор, пока пользователь окно не закроет.]

немодальное окно

⇒ *пояснение**:

[*немодальное окно* – окно, которое позволяет выполнять переключение между данным окном и другим окном без необходимости закрытия окна.]

интерактивный компонент пользовательского интерфейса⇒ *пояснение**:

[*интерактивный компонент пользовательского интерфейса* – компонент пользовательского интерфейса, с помощью которого осуществляется взаимодействие с пользователем.]

флаговая кнопка⇒ *пояснение**:

[*флаговая кнопка* – компонент пользовательского интерфейса, позволяющий пользователю управлять параметром с двумя состояниями — включено и отключено.]

радиокнопка⇒ *пояснение**:

[*радиокнопка* – компонент пользовательского интерфейса, который позволяет пользователю выбрать одну опцию из predetermined набора.]

переключатель⇒ *пояснение**:

[*переключатель* – компонент пользовательского интерфейса, который позволяет пользователю переключаться между двумя состояниями.]

кнопка-счетчик⇒ *пояснение**:

[*кнопка-счетчик* – компонент пользовательского интерфейса, как правило, ориентированный вертикально, с помощью которого пользователь может изменить значение в прилегающем текстовом поле, в результате чего значение в текстовом поле увеличивается или уменьшается.]

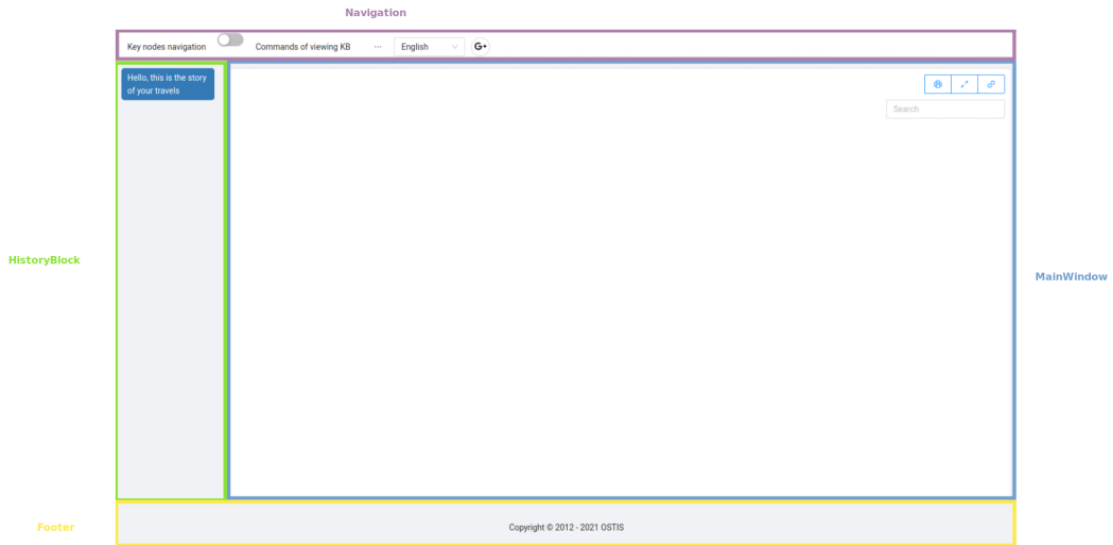
полоса прокрутки⇒ *пояснение**:

[*полоса прокрутки* – компонент пользовательского интерфейса, который используется для отображения компонентов пользовательского интерфейса, больших по размеру, чем используемый для их отображения контейнер.]

кнопка⇒ *пояснение**:

[*кнопка* – компонент пользовательского интерфейса, при нажатии на который происходит программно связанное с этим нажатием действие либо событие.]

Стартовая страница пользовательского интерфейса Метасистемы IMS.ostis€ *Визуальная часть пользовательского интерфейса Метасистемы IMS.ostis*С *визуальная часть пользовательского интерфейса ostis-системы*← *часть**:*Пользовательский интерфейс Метасистемы IMS.ostis*€ *окно*⇒ *иллюстрация**:



- ⇒] декомпозиция*:
- Панель навигации
 - ∈ неатомарный компонент пользовательского интерфейса
 - ⇒ декомпозиция*:
 - Главное меню
 - ∈ меню
 - ⇒ декомпозиция*:
 - Пункт меню для навигации по ключевым понятиям
 - ∈ пункт меню
 - Пункт меню для выполнения команд просмотра базы знаний
 - ∈ пункт меню
 - Компонент перехода в экспертный режим
 - ∈ переключатель
 - Компонент выбора языка
 - ∈ компонент выбора одного значения
 - Компонент авторизации
 - ∈ кнопка
- Блок истории запросов пользователя
 - ∈ неатомарный компонент пользовательского интерфейса
- Основной блок
 - ∈ неатомарный компонент пользовательского интерфейса
 - ⇒ декомпозиция*:
 - Главное окно
 - ∈ окно
 - Панель инструментов
 - ∈ неатомарный компонент пользовательского интерфейса
 - ⇒ декомпозиция*:
 - Кнопка отправки содержимого главного окна на печать
 - ∈ кнопка
 - Кнопка управления видимостью блока истории запросов пользователя
 - ∈ кнопка
 - Кнопка отображения ссылки на текущий запрос пользователя
 - ∈ кнопка
 - Поле поиска
 - ∈ однострочное текстовое поле
- Панель отображения информации об авторских правах

€ *неатомарный компонент пользовательского интерфейса*

}

}

/ Завершили Раздел "Предметная область и онтология интерфейсов ostis-систем" */*

/* Раздел *****/	
Предметная область и онтология интерфейсных действий пользователей ostis-системы	
⊃=	
{	
⇒	соавтор*: Садовский М.Е.
Предметная область интерфейсных действий пользователей	
∈	предметная область
⊃	максимальный класс объектов исследования':
•	интерфейсное действие пользователя
⊃	класс объектов исследования':
•	действие мышью
•	прокрутка мышью
•	прокрутка мышью вверх
•	прокрутка мышью вниз
•	наведение мышью
•	отпускание мышью
•	нажатие мыши
•	одиночное нажатие мыши
•	двойное нажатие мыши
•	жест мышью
•	отведение мышью
•	перетаскивание мышью
•	действие голосом
•	действие клавиатурой
•	нажатие функциональной клавиши
•	нажатие клавиши набора текста
•	действие осязанием
•	действие сенсором
•	нажатие сенсора
•	одиночное нажатие сенсора
•	двойное нажатие сенсора
•	жест по сенсору
•	жест по сенсору одним пальцем
•	жест по сенсору несколькими пальцами
•	отпускание сенсором
•	перетаскивание сенсором
•	действие пером
•	нажатие функциональной клавиши пером
•	рисование пером
•	написание текста пером
⊃	исследуемое отношение':
•	инициируемое пользовательским интерфейсом действие*
⇒	частная предметная область*: Предметная область интерфейсных действий пользователей ostis-системы
интерфейсное действие пользователя	
:=	[user interface action]
⇒	пояснение*: [Действие, выполняемое пользователем над некоторым компонентом пользовательского интерфейса. Для связи данного действия с компонентом пользовательского интерфейса и необходимым к выполнению внутренним действием системы используется отношение инициируемое пользовательским интерфейсом действие*]
⊃	действие мышью
:=	[mouse-action]
⊃	прокрутка мышью
:=	[mouse-scroll]
⊃	прокрутка мышью вверх

	:= [mouse-scroll-up]
▷	<i>прокрутка мышью вниз</i>
	:= [mouse-scroll-down]
▷	<i>наведение мышью</i>
	:= [mouse-hover]
▷	<i>отпускание мышью</i>
	:= [mouse-drop]
▷	<i>нажатие мыши</i>
	:= [mouse-click]
▷	<i>одиночное нажатие мыши</i>
	:= [mouse-single-click]
▷	<i>двойное нажатие мыши</i>
	:= [mouse-double-click]
▷	<i>жест мышью</i>
	:= [mouse-gesture]
▷	<i>отведение мышью</i>
	:= [mouse-unhover]
▷	<i>перетаскивание мышью</i>
	:= [mouse-drag]
▷	<i>действие голосом</i>
	:= [speech-action]
▷	<i>действие клавиатурой</i>
	:= [keyboard-action]
▷	<i>нажатие функциональной клавиши</i>
	:= [press-function-key]
▷	<i>нажатие клавиши набора текста</i>
	:= [type-text]
▷	<i>действие осязанием</i>
	:= [tangible-action]
▷	<i>действие сенсором</i>
	:= [touch-action]
▷	<i>нажатие сенсора</i>
	:= [touch-click]
▷	<i>одиночное нажатие сенсора</i>
	:= [touch-single-click]
▷	<i>двойное нажатие сенсора</i>
	:= [touch-double-click]
▷	<i>жест по сенсору</i>
	:= [touch-gesture]
▷	<i>жест по сенсору одним пальцем</i>
	:= [one-fingure-gesture]
▷	<i>жест по сенсору несколькими пальцами</i>
	:= [multiple-finger-gesture]
▷	<i>отпускание сенсором</i>
	:= [touch-drop]
▷	<i>перетаскивание сенсором</i>
	:= [touch-drag]
▷	<i>действие пером</i>
	:= [pen-base-action]
▷	<i>нажатие функциональной клавиши пером</i>
	:= [touch-function-key]
▷	<i>рисование пером</i>
	:= [draw]
▷	<i>написание текста пером</i>

:= [write-text]

прокрутка мышью

⇒ *пояснение**:

[*прокрутка мышью* – интерфейсное действие пользователя, соответствующее прокрутке содержимого некоторого компонента пользовательского интерфейса при помощи мыши.]

наведение мышью

⇒ *пояснение**:

[*наведение мышью* – интерфейсное действие пользователя, соответствующее появлению курсора мыши в рамках компонента пользовательского интерфейса.]

отпускание мышью

⇒ *пояснение**:

[*отпускание мышью* – интерфейсное действие пользователя, соответствующее отпусанию некоторого компонента пользовательского интерфейса в рамках другого компонента пользовательского интерфейса при помощи мыши.]

нажатие мыши

⇒ *пояснение**:

[*нажатие мыши* – интерфейсное действие пользователя, соответствующее выполнению нажатия мыши в рамках некоторого компонента пользовательского интерфейса.]

отведение мышью

⇒ *пояснение**:

[*отведение мышью* – интерфейсное действие пользователя, соответствующее выходу курсора мыши за рамки компонента пользовательского интерфейса.]

перетаскивание мышью

⇒ *пояснение**:

[*перетаскивание мышью* – интерфейсное действие пользователя, соответствующее перетаскиванию компонента пользовательского интерфейса при помощи мыши.]

нажатие сенсора

⇒ *пояснение**:

[*нажатие сенсора* – интерфейсное действие пользователя, соответствующее выполнению нажатия сенсора в рамках некоторого компонента пользовательского интерфейса.]

жест по сенсору

⇒ *пояснение**:

[*жест по сенсору* – интерфейсное действие пользователя, соответствующее выполнению некоторого жеста, выполняемого при помощи движения пальцев на экране сенсора.]

отпускание сенсором

⇒ *пояснение**:

[*отпускание сенсором* – интерфейсное действие пользователя, соответствующее отпусанию некоторого компонента пользовательского интерфейса в рамках другого компонента пользовательского интерфейса при помощи сенсора.]

перетаскивание сенсором

⇒ *пояснение**:

[*перетаскивание сенсором* – интерфейсное действие пользователя, соответствующее перетаскиванию компонента пользовательского интерфейса при помощи сенсора.]

действие пером

⇒ *пояснение**:

[*действие пером* – интерфейсное действие пользователя, осуществляемое при помощи пера на графическом планшете.]

класс интерфейсных действий пользователя

⊂ класс действий

⊆ семейство подмножеств*:
интерфейсное действие пользователя

⇒ пояснение*:

[класс интерфейсных действий пользователя – множество, элементами которого являются классы интерфейсных действий пользователя.]

иницируемое пользовательским интерфейсом действие*

⇒ пояснение*:

[При взаимодействии пользователя с компонентом пользовательского интерфейса могут быть произведены различные интерфейсные действия. В зависимости от выполненного интерфейсного действия и компонента, над которым оно было выполнено, происходит иницирование некоторого внутреннего действия системы. Для задания такого иницируемого при взаимодействии с пользовательским интерфейсом действия и используется указанное отношение. Первым компонентом связи отношения иницируемое пользовательским интерфейсом действие* является связка, элементами которой являются элемент множества компонентов пользовательского интерфейса и элемент множества класс интерфейсных действий пользователя. Вторым компонентом является элемент множества класс внутренних действий системы.]

∈ квазибинарное отношение

∈ ориентированное отношение

⇒ первый домен*:

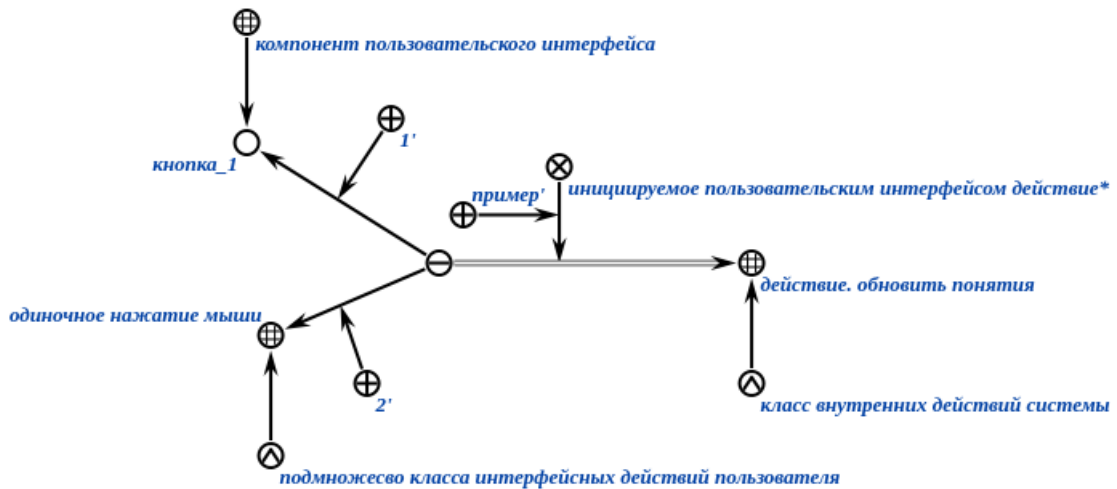
компонент пользовательского интерфейса ∪ класс интерфейсных действий пользователя

⇒ второй домен*:

класс внутренних действий системы

⇒ описание примера*:

[



]

} /* Завершили Раздел “Предметная область и онтология интерфейсных действий пользователей ostis-системы” */

/* Раздел *****/	
Предметная область и онтология естественных языков	
⊇	
{	
⇒	соавтор*:
	• Гордей А.Н.
	• Никифоров С.А.
	• Бобёр Е.С.
	• Святощик М.И.
Предметная область естественных языков	
∈	предметная область
⊃	максимальный класс объектов исследования ¹ :
	язык
⊃	класс объектов исследования ¹ :
	• плановый язык
	• язык общения
	• лексема
	• номинативная единица
	• комбинаторный вариант лексемы
	• естественный язык
	• тайген
	• ёген
⊃	исследуемое отношение ¹ :
	• морфологическая парадигма*
	• член предложения ¹
язык	
⇒	разбиение*:
{	• естественный язык
	⇒ пояснение*:
	[Естественный язык представляет собой язык, который не был создан целенаправленно]
	• искусственный язык
	⇒ пояснение*:
	[Искусственный язык представляет собой язык, специально разработанный для достижения определённых целей]
	⊃ Эсперанто
	⊃ Python
	⊃ сконструированный язык
	⇒ пояснение*:
	[Сконструированный язык представляет собой искусственный язык, предназначенный для общения людей]
	⊃ Эсперанто
}	
⊃	международный язык
⇒	пояснение*:
	[Международный язык представляет собой естественный или искусственный язык, использующийся для общения людей разных из стран]
	⊃ Английский язык
	⊃ Русский язык
плановый язык	
←	пересечение*:
{	• сконструированный язык
	• международный язык
}	
язык общения	
←	объединение*:
{	• естественный язык

	• <i>сконструированный язык</i>
	}
Э	<i>Английский язык</i>
Э	<i>Русский язык</i>
Э	<i>Эсперанто</i>
←	<i>объединение*</i> :
	{• <i>корневой язык</i>
	⇒ <i>пояснение*</i> :
	[Корневой язык представляет собой язык, для которого характерно полное отсутствие словоизменения и наличие грамматической значимости порядка слов, состоящих только из корня.]
	Э <i>Английский язык</i>
	• <i>агглютинативный язык</i>
	⇒ <i>пояснение*</i> :
	[Агглютинативный язык характеризуется развитой системой употребления суффиксов, приставок, добавляемых к неизменяемой основе слова, которые используются для выражения категорий числа, падежа, рода и др.]
	Э <i>Английский язык</i>
	• <i>флективный язык</i>
	⇒ <i>пояснение*</i> :
	[Для флективного языка характерно развитое употребление окончаний для выражения категорий рода, числа, падежа, сложная система склонения глаголов, чередование гласных в корне, а также строгое различие частей речи.]
	Э <i>Русский язык</i>
	• <i>профлективный язык</i>
	⇒ <i>пояснение*</i> :
	[Для профлективного языка характерны агглютинация (в случае именного словоизменения), флексия и чередование гласных (аблаут)(в случае глагольного словоизменения).]
	}

лексема

С	<i>файл</i>
⇒	<i>пояснение*</i> :
	[<i>Лексема</i> – тайген или ёген конкретного естественного языка.]
⇒	<i>источник*</i> :
	<i>Гордей.А.Н.ОснованияКС-2005ст</i>

номинативная единица

С	<i>файл</i>
⇒	<i>пояснение*</i> :
	[<i>Номинативная единица</i> – устойчивая последовательность комбинаторных вариантов лексем, в которой один вариант лексемы (модификатор) определяет другой (актуализатор), например: ‘записная книжка’, ‘бежать галопом’.]
⇒	<i>источник*</i> :
	<i>Гордей.А.Н.ОснованияКС-2005ст</i>

комбинаторный вариант лексемы

С	<i>файл</i>
⇒	<i>пояснение*</i> :
	[<i>Комбинаторный вариант лексемы</i> – вариант лексемы в упорядоченном наборе её вариантов (парадигме).]
⇒	<i>источник*</i> :
	<i>Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст</i>

морфологическая парадигма*

Є	<i>квазибинарное отношение</i>
⇒	<i>пояснение*</i> :
	[<i>Морфологическая парадигма*</i> – квазибинарное отношение, связывающее лексему с её комбинаторными вариантами.]
⇒	<i>первый домен*</i> :
	<i>словоформа</i>

⇒ второй домен*:
лексема

естественный язык

⇒ разбиение*:

- { • часть языка
 - ⇒ разбиение*:
 - { • тайген
 - ёген
 - }
- знак алфавита синтаксиса

⇒ пояснение*:

[Знаки алфавита синтаксиса – вспомогательные средства синтаксиса (на макроуровне – предлоги, послелого, союзы, частицы и др., на микроуровне – флексии, префиксы, постфиксы, инфиксы и др.), служащие для соединения составных частей языковых структур и образования морфологических парадигм.]

⇒ источник*:

Гордей.А.Н.ОснованияКС-2005ст

}

тайген

⇒ пояснение*:

[Тайген – часть языка, обозначающая индивида.]

⇒ источник*:

Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст

⇒ источник*:

Гордей.А.Н.ОКЧЗКЯ-2007ст

⇒ разбиение*:

- { • развёрнутый тайген
 - ⇒ разбиение*:
 - { • составной тайген
 - сложный тайген
 - }

- свёрнутый тайген

⇒ разбиение*:

- { • сокращённый тайген
- сжатый тайген

⇒ разбиение*:

- { • информационный тайген

⇒ пояснение*:

[Информационный тайген – тайген, обозначающий индивида в информационном фрагменте модели мира.]

⇒ источник*:

Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст

⇒ источник*:

Гордей.А.Н.ОКЧЗКЯ-2007ст

- физический тайген

⇒ пояснение*:

[Физический тайген – тайген, обозначающий индивида в физическом фрагменте модели мира.]

⇒ источник*:

Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст

⇒ источник*:

Гордей.А.Н.ОКЧЗКЯ-2007ст

⇒ разбиение*:

- { • постоянный тайген

⇒ пояснение*:

[Постоянный тайген – физический тайген, обозначающий постоянного индивида.]

⇒ источник*:

Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст

									⇒ источник*: Гордей.А.Н.ОКЧЗКЯ-2007ст
									• переменный тайген
									⇒ пояснение*: [Переменный тайген – физический тайген, обозначающий переменного индивида.]
									⇒ источник*: Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст
									⇒ источник*: Гордей.А.Н.ОКЧЗКЯ-2007ст
									}
									⇒ разбиение*: {• качественный тайген • количественный тайген }
									⇒ разбиение*: {• одноместный тайген • многоместный тайген ⊃ интенсивный тайген ⊃ экстенсивный тайген }
									}
									}
									}
ёген									
⇒									пояснение*: [Ёген – часть языка, обозначающая признак индивида.]
									⇒ источник*: Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст
									⇒ источник*: Гордей.А.Н.ОКЧЗКЯ-2007ст
⇒									разбиение*: {• развёрнутый ёген ⇒ разбиение*: {• составной ёген • сложный ёген }
									• свёрнутый ёген ⇒ разбиение*: {• сокращённый ёген ⇒ разбиение*: {• информационный ёген ⇒ пояснение*: [Информационный ёген – еген, обозначающий признак индивида в информационном фрагменте модели мира.] ⇒ источник*: Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст ⇒ источник*: Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст
									• физический ёген ⇒ пояснение*: [Физический ёген – еген, обозначающий признак индивида в физическом фрагменте модели мира.] ⇒ источник*: Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст ⇒ источник*: Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст
									⇒ разбиение*: {• постоянный ёген ⇒ пояснение*:

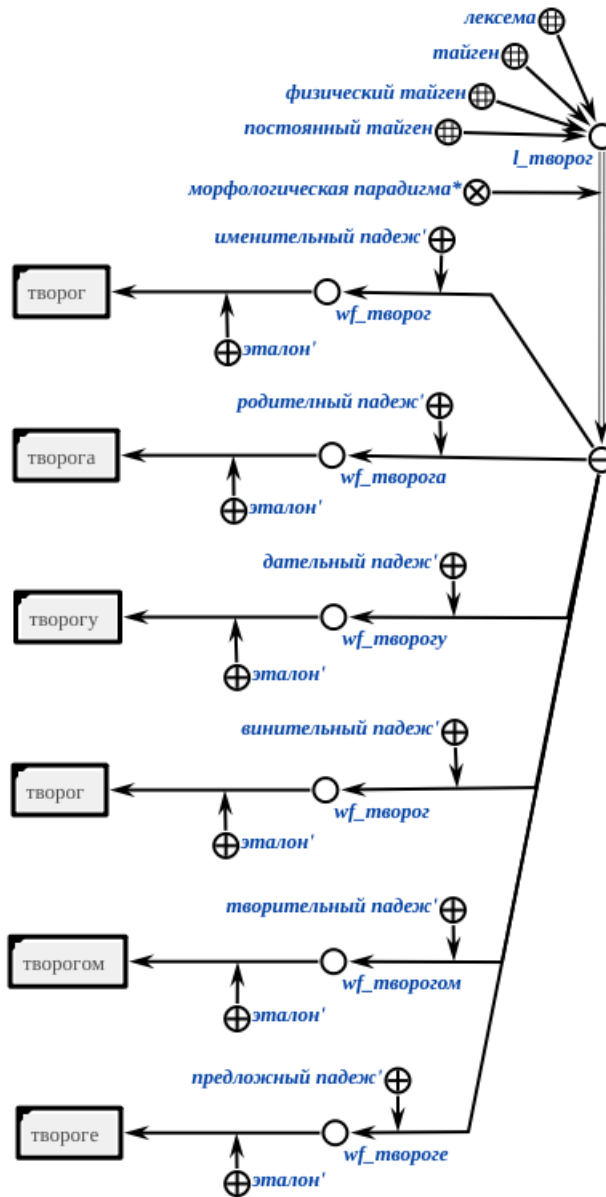
		⇒	источник*: Гордей.А.Н. <i>Plagiarism PSBCS-2020</i> ст
	}		
•	второстепенный член предложения'	⇒	разбиение*:
	{	•	косвенное дополнение'
		•	определение'
		⇒	пояснение*: [<i>Определение'</i> – это одно из второстепенных ролевых отношений, связывающее декомпозицию текста с файлом, содержимое которого обозначает модификацию подлежащего, дополнения, обстоятельства места и времени.]
		⇒	источник*: Гордей.А.Н. <i>Виртуальная ЦКСКП-2007</i> ст
		⇒	источник*: Гордей.А.Н. <i>Динамический ССП-2017</i> ст
		⇒	источник*: Гордей.А.Н. <i>Теоретическая ГВЯЛК-2007</i> эл
	•	обстоятельство'	
		⇒	пояснение*: [<i>Обстоятельство'</i> – это одно из второстепенных ролевых отношений, связывающее декомпозицию текста с файлом, содержимое которого обозначает либо модификацию, либо локализацию сказуемого.]
		⇒	источник*: Гордей.А.Н. <i>Виртуальная ЦКСКП-2007</i> ст
		⇒	источник*: Гордей.А.Н. <i>Динамический ССП-2017</i> ст
		⇒	источник*: Гордей.А.Н. <i>Теоретическая ГВЯЛК-2007</i> эл
		⇒	разбиение*:
		{	• обстоятельство степени'
			⇒ пояснение*: [Обстоятельство степени – обстоятельство, обозначающее модификацию сказуемого.]
		•	обстоятельство образа действия'
			⇒ пояснение*: [Обстоятельство образа действия – обстоятельство, обозначающее модификацию сказуемого.]
		•	обстоятельство места'
			⇒ пояснение*: [Обстоятельства места – обстоятельство, обозначающее пространственную локализацию сказуемого.]
			⇒ разбиение*:
			{
			• динамическое обстоятельство места'
			• статическое обстоятельство места'
			}
		•	обстоятельство времени'
			⇒ пояснение*: [Обстоятельство времени – обстоятельство, обозначающее временную локализацию сказуемого.]
			⇒ разбиение*:
			{
			• динамическое обстоятельство времени'
			• статическое обстоятельство времени'
			}
	}		
	}		

Пример sc.g-текста, описывающего лексему

∈ sc.g-текст

⇒ пояснение*:

[Здесь представлено описание лексемы с указанием ее принадлежности определённой части речи. Также описание содержит морфологическую парадигму данной лексемы, связывающую ее с ее словоформами.]



] **Пример этапов разбора текста естественного языка**

⊃=
{

⊕ Русский язык

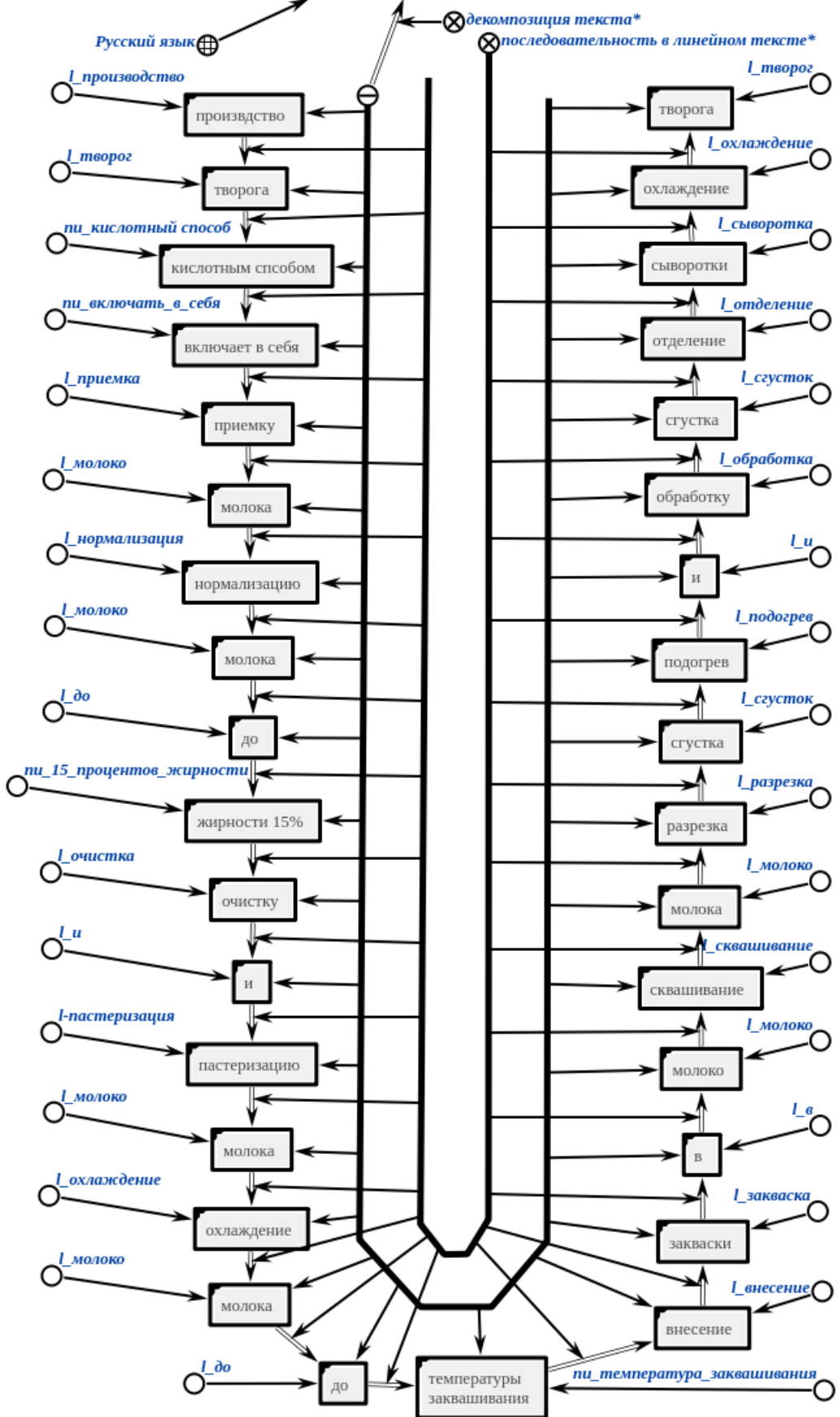
Производство творога кислотным способом включает в себя: приёмку молока, нормализацию молока до жирности 15%, очистку и пастеризацию молока, охлаждение молока до температуры заквашивания, внесение закваски в молоко, сквашивание молока, разрезку сгустка, подогрев и обработку сгустка, отделение сыворотки, охлаждение творога.

] ⇒ пояснение*:

[с точки зрения ostis-системы, любой естественно-языковой текст является *файлом*.]

⇒ лексическая структура*:

Производство творога кислотным способом включает в себя: приёмку молока, нормализацию молока до жирности 15%, очистку и пастеризацию молока, охлаждение молока до температуры заквашивания, внесение закваски в молоко, сквашивание молока, разрезку сгустка, подогрев и обработку сгустка, отделение сыворотки, охлаждение творога.

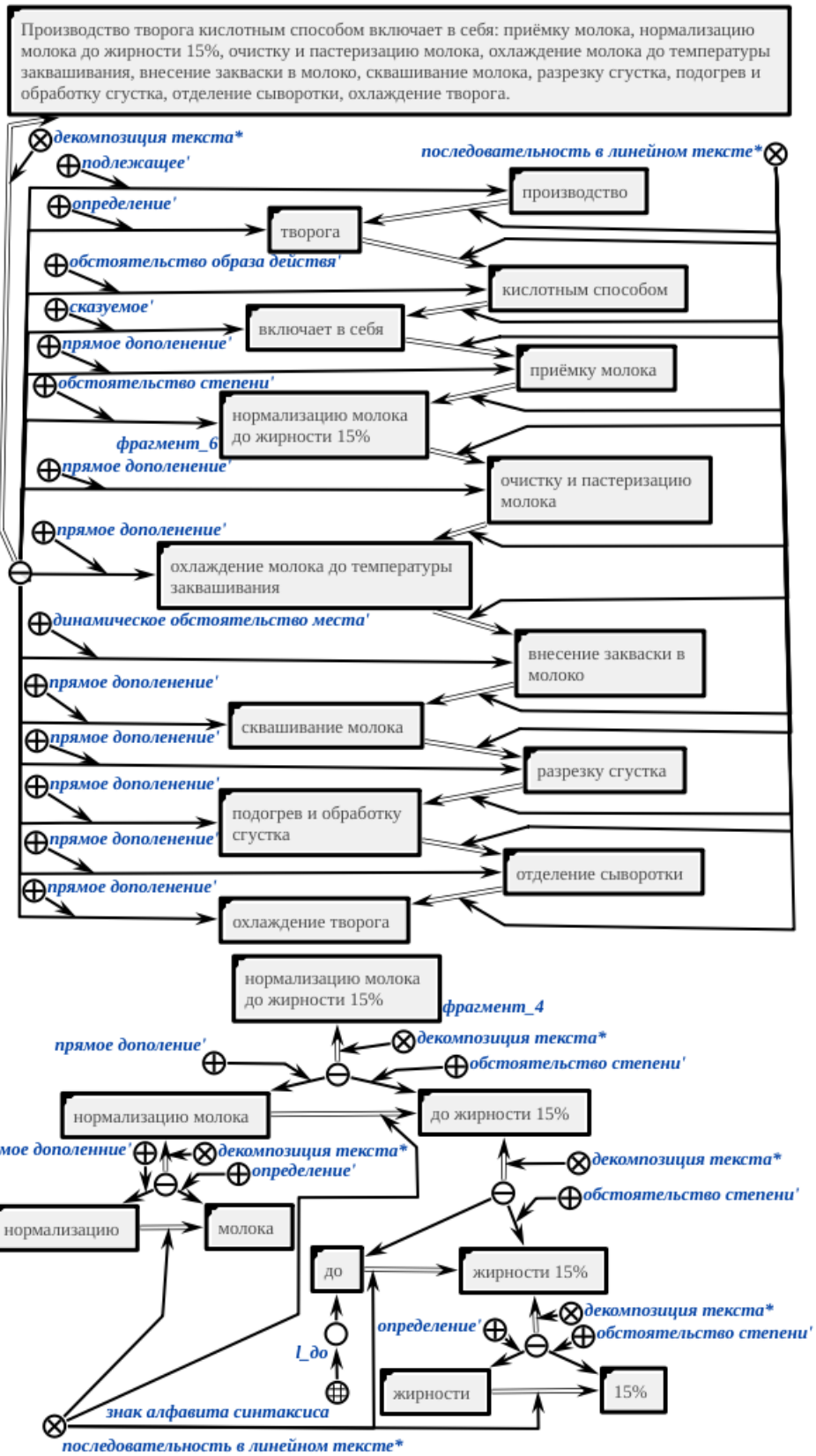


⇒ пояснение*:

[Данная конструкция описывает декомпозицию исходного текста на фрагменты с указанием их принадлежности определённой *номинативной единице* или *знаку алфавита синтаксиса*.]

⇒ синтаксическая структура*:

[



]

/ Раздел *****/*

Предметная область и онтология базовых интерпретаторов логико-семантических моделей ostis-систем

$\supset=$
{

$:=$ [Предметная область и онтология платформ интерпретации sc-моделей компьютерных систем]

\in раздел базы знаний

\Rightarrow дочерний раздел*:

- Предметная область и онтология программных вариантов реализации базового интерпретатора логико-семантических моделей ostis-систем на современных компьютерах
- Предметная область и онтология семантических ассоциативных компьютеров для ostis-систем

Предметная область базовых интерпретаторов логико-семантических моделей ostis-систем

\in предметная область

\ni максимальный класс объектов исследования':

платформа интерпретации sc-моделей компьютерных систем

логико-семантическая модель кибернетической системы

$:=$ [формальная модель (формальное описание) функционирования кибернетической системы, состоящая из:

- формальной модели информации, хранимой в памяти кибернетической системы;
- формальной модели коллектива агентов, осуществляющих обработку указанной информации.

]

\supset sc-модель кибернетической системы

$:=$ [логико-семантическая модель кибернетической системы, представленная в SC-коде]

$:=$ [логико-семантическая модель ostis-системы, которая, в частности, может быть функционально эквивалентной моделью какой-либо кибернетической системы, не являющейся ostis-системой]

кибернетическая система

\supset компьютерная система

$:=$ [искусственная кибернетическая система]

\supset o

stis-система

$:=$ [компьютерная система, построенная по технологии OSTIS на основе интерпретации спроектированной логико-семантической sc-модели этой системы]

платформа интерпретации sc-моделей компьютерных систем

$:=$ [базовый интерпретатор логико-семантических моделей ostis-систем]

$:=$ [Семейство платформ интерпретации sc-моделей компьютерных систем]

$:=$ [платформа реализации sc-моделей компьютерных систем]

$:=$ часто используемый sc-идентификатор*:

[универсальный интерпретатор sc-моделей компьютерных систем]

$:=$ [универсальный интерпретатор унифицированных логико-семантических моделей компьютерных систем]

\subset встроенная ostis-система

$:=$ [встроенная пустая ostis-система]

$:=$ [универсальный интерпретатор sc-моделей ostis-систем]

$:=$ [универсальная базовая ostis-система, обеспечивающая имитацию любой ostis-системы путем интерпретации sc-модели имитируемой ostis-системы]

\Rightarrow примечание*:

[соотношение между имитируемой и универсальной ostis-системой в известной мере аналогично соотношению между машиной Тьюринга и универсальной машиной Тьюринга]

\Rightarrow пояснение*:

[Под платформой интерпретации sc-моделей компьютерных систем понимается реализация платформы интерпретации sc-моделей, которая в общем случае включает в себя:

Реализация платформы интерпретации sc-моделей компьютерных систем (универсального интерпретатора sc-моделей компьютерных систем) может иметь большое число вариантов – как программно, так и

аппаратно реализованных. Логическая архитектура *платформы интерпретации sc-моделей компьютерных систем* обеспечивает независимость проектируемых компьютерных систем от многообразия вариантов реализации интерпретатора их моделей и в общем случае включает в себя:

- хранилище *sc-текстов* (*sc-хранилище*, хранилище знаковых конструкций, представленных SC-коде);
- файловую память *sc-машины*;
- средства, обеспечивающие взаимодействие *sc-агентов* над общей памятью (*sc-память*);
- базовые средства интерфейса для взаимодействия системы с внешним миром (пользователем или другими системами). Указанные средства включают в себя, как минимум, редактор, транслятор (в *sc-память* и из нее) и визуализатор для одного из базовых универсальных вариантов представления *SC-кода* (*SCg-код*, *SCs-код*, *SCn-код*), средства, позволяющие задавать системе вопросы из некоторого универсального класса (например, запрос семантической окрестности некоторого объекта);
- реализацию *Абстрактной scr-машины*, то есть интерпретатор *scr-программ* (программ Языка SCP).

При необходимости, в *платформу интерпретации sc-моделей компьютерных систем* могут быть заранее на платформенно-зависимом уровне включены какие-либо компоненты машин обработки знаний или баз знаний, например, с целью упрощения создания первой версии *прикладной ostis-системы*.

Реализация платформы может осуществляться на основе произвольного набора существующих технологий, включая аппаратную реализацию каких-либо ее частей. С точки зрения компонентного подхода любая *платформа интерпретации sc-моделей компьютерных систем* является **платформенно-зависимым многократно используемым компонентом.**

⊃ программный вариант реализации платформы интерпретации *sc-моделей компьютерных систем*
 ⊃ семантический ассоциативный компьютер
 ⇒ разбиение*:

- { ● *однопользовательский вариант реализации платформы интерпретации sc-моделей компьютерных систем*
 := [вариант реализации платформы интерпретации *sc-моделей компьютерных систем*, рассчитанный на то, что с конкретной *ostis-системой* взаимодействует только один пользователь (владелец)]
 ⇒ *примечание**:
 [При таком варианте реализации платформы оказывается невозможным реализовать некоторые важные принципы *Технологии OSTIS*, например, коллективную согласованную разработку базы знаний системы в процессе ее эксплуатации. При этом могут использоваться различные сторонние средства, например для разработки базы знаний на уровне исходных текстов.]
- *многопользовательский вариант реализации платформы интерпретации sc-моделей компьютерных систем*
 := [вариант реализации платформы интерпретации *sc-моделей компьютерных систем*, рассчитанный на то, что с конкретной *ostis-системой* одновременно или в разное время могут взаимодействовать разные пользователи, в общем случае обладающие разными правами, сферами ответственности, уровнем опыта, и имеющие свою конфиденциальную часть хранимой в базе знаний информации]

платформа интерпретации sc-моделей компьютерных систем

⇒ разбиение*:

- { ● *программный вариант реализации платформы интерпретации sc-моделей компьютерных систем*
 := [программная платформа интерпретации *sc-моделей ostis-систем*]
 := [программный базовый интерпретатор *sc-моделей ostis-систем*]
- *семантический ассоциативный компьютер*
 := [аппаратная платформа интерпретации *sc-моделей ostis-систем*]
 := [аппаратно реализованный базовый интерпретатор *sc-моделей ostis-систем*]

} /* Завершили Раздел “Предметная область и онтология базовых интерпретаторов логико-семантических моделей *ostis-систем*” */

```

/* Раздел *****/
Предметная область и онтология программных вариантов реализации базового интерпретатора логико-семантических моделей ostis-систем на современных компьютерах
⊃=
{
⇒ соавтор*:
   Корончик Д.Н.

Предметная область программных вариантов реализации базового интерпретатора логико-семантических моделей ostis-систем на современных компьютерах
∈ предметная область
∋ максимальный класс объектов исследования':
  программный вариант реализации платформы интерпретации sc-моделей компьютерных систем
∋ ключевой объект исследования':
  • Программный вариант реализации платформы интерпретации sc-моделей компьютерных систем
  • Программная модель sc-памяти
  • Реализация sc-хранилища и средств доступа к нему
  • Реализация sc-хранилища
  • sc-адрес
  • элемент sc-хранилища
  • метка синтаксического типа sc-элемента
  • метка уровня доступа sc-элемента
  • sc-итератор
  • sc-шаблон
  • контекст процесса в рамках программной модели sc-памяти
  • блокировка sc-элемента в рамках программной модели sc-памяти
  • подписка на событие в sc-памяти в рамках программной модели sc-памяти
  • Реализация файловой памяти ostis-системы
  • Реализация базового набора платформенно-зависимых sc-агентов и их компонентов
  • Реализация подсистемы взаимодействия с внешней средой с использованием сетевых протоколов SCTP
  • sctp-команда
  • Реализация подсистемы взаимодействия с внешней средой с использованием протоколов SCTP
  • Реализация подсистемы взаимодействия с внешней средой с использованием протоколов на основе формата JSON
  • Реализация вспомогательных инструментальных средств для работы с sc-памятью
  • Реализация интерпретатора sc-моделей пользовательских интерфейсов
  • Реализация scr-интерпретатора

программный вариант реализации платформы интерпретации sc-моделей компьютерных систем
:= [программная реализация платформы интерпретации sc-моделей компьютерных систем ]
:= [программный вариант реализации базового интерпретатора логико-семантических моделей компьютерных систем]
:= [программный вариант реализации базового интерпретатора логико-семантических моделей ostis-систем на современных компьютерах]
:= [вариант реализации базового интерпретатора логико-семантических моделей компьютерных систем на традиционных компьютерах с архитектурой фон Неймана]
⇒ пояснение*:
  [Одним из путей, позволяющих осуществлять апробацию, развитие, а в ряде случаев и внедрение новых моделей и технологий вне зависимости от наличия соответствующих аппаратных средств является разработка программных моделей этих аппаратных средств, которые были бы функционально эквивалентны этим аппаратным средствам, но при этом интерпретировались на базе традиционной аппаратной архитектуры (в данной работе традиционной архитектурой будем считать архитектуру фон Неймана, как доминирующую в настоящее время). Очевидно, что производительность таких программных моделей в общем случае будет ниже, чем самих аппаратных решений, однако в большинстве случаев она оказывается достаточной для того, чтобы развивать соответствующую технологию параллельно с разработкой аппаратных средств и осуществления постепенного перевода уже работающих систем с программной модели на аппаратные средства.]

```


- ⊃ *web-ориентированный вариант реализации платформы интерпретации sc-моделей компьютерных систем*
- := [вариант реализации платформы интерпретации sc-моделей компьютерных систем предполагающий взаимодействие пользователей с системой посредством сети Интернет]
- ⊃ *многопользовательский вариант реализации платформы интерпретации sc-моделей компьютерных систем*
- ⊃ *Программный вариант реализации платформы интерпретации sc-моделей компьютерных систем*

Программный вариант реализации платформы интерпретации sc-моделей компьютерных систем

⇒ *принципы реализации**:

[Поскольку sc-тексты представляют собой семантические сети, то есть, по сути, графовые конструкции определенного вида, то на нижнем уровне задача разработки программного варианта реализации платформы интерпретации sc-моделей сводится к разработке средств хранения и обработки таких графовых конструкций.

К настоящему времени разработано большое количество простейших моделей представления графовых конструкций в линейной памяти, таких как матрицы смежности, списки смежности и другие (*DiscreteMathSys-эл*). Однако, при разработке сложных систем как правило приходится использовать более эффективные модели, как с точки зрения объема информации, требуемого для представления, так и с точки зрения эффективности обработки графовых конструкций, хранимых в той или иной форме.

К наиболее распространенным программным средствам, ориентированным на хранение и обработку графовых конструкций относятся графовые СУБД (*Neo4j Neo4j-эл*, *ArangoDB ArangoDB-эл*, *OrientDB OrientDB-эл*, *Grakn Grakn-эл* и др.), а также так называемые rdf-хранилища (*Virtuoso Virtuoso-эл*, *Sesame Sesame-эл* и др.), предназначенные для хранения конструкций, представленных в модели RDF. Для доступа к информации, хранимой в рамках таких средств, могут использоваться как языки, реализуемые в рамках конкретного средства (например, язык Cypher в Neo4j), так и языки, являющиеся стандартами для большого числа систем такого класса (например, SPARQL для rdf-хранилищ).

Популярность и развитость такого рода средств приводит к тому, что на первый взгляд целесообразным и эффективным кажется вариант реализации *программного варианта реализации платформы интерпретации sc-моделей* на базе одного из таких средств. Однако, существует ряд причин, по которым было принято решение о реализации *программного варианта реализации платформы интерпретации sc-моделей* с нуля. К ним относятся следующие:

- для обеспечения эффективности хранения и обработки информационных конструкций определенного вида (в данном случае – конструкций SC-кода, sc-конструкций), должна учитываться специфика этих конструкций. В частности, описанные в работе *Корончик.Д.Н. УСМПИИСиТИКП-2013ст* эксперименты показали значительный прирост эффективности собственного решения по сравнению с существующими на тот момент;
- в отличие от классических графовых конструкций, где дуга или ребро могут быть инцидентны только вершине графа (это справедливо и для rdf-графов) в SC-коде вполне типичной является ситуация, когда sc-коннектор инцидентен другому sc-коннектору или даже двум sc-коннекторам. В связи с этим существующие средства хранения графовых конструкций не позволяют в явном виде хранить sc-конструкции (sc-графы). Возможным решением данной проблемы является переход от sc-графа к орграфу инцидентности, пример которого описан в работе *Ивашенко.В.П..ПредставССАОСОВСМП-2015ст*, однако такой вариант приводит к увеличению числа хранимых элементов в несколько раз и значительно снижает эффективность алгоритмов поиска из-за необходимости делать большое количество дополнительных итераций;
- в основе обработки информации в рамках Технологии OSTIS лежит многоагентный подход, в рамках которого агенты обработки информации, хранимой в sc-памяти (sc-агенты) реагируют на события, происходящие в sc-памяти и обмениваются информацией посредством спецификации выполняемых ими действий в sc-памяти *Shunkevich.D.V.AgentOMMTCPSPDIS-2018ст*. В связи с этим одной из важнейших задач является реализация в рамках *программного варианта реализации платформы интерпретации sc-моделей* возможности подписки на события, происходящие в программной модели sc-памяти, которая на данный момент практически не поддерживается в рамках современных средств хранения и обработки графовых конструкций;
- SC-код позволяет описывать также внешние информационные конструкции любого рода (изображения, текстовые файлы, аудио- и видеофайлы и т.д.), которые формально трактуются как содержимое *sc-элементов*, являющихся знаками *внешних файлов ostis-системы*. Таким образом, компонентом *программного варианта реализации платформы интерпретации sc-моделей* должна быть реализация файловой памяти, которая позволяет хранить указанные конструкции в каких-либо общепринятых форматах. Реализация такого компонента в рамках современных средств хранения и обработки графовых конструкций также не всегда представляется возможной.

По совокупности перечисленных причин было принято решение о реализации *программного варианта реализации платформы интерпретации sc-моделей "с нуля"* с учетом особенностей хранения и обработки информации в рамках Технологии OSTIS.]

⇒ *декомпозиция программной системы**:

- Программная модель sc-памяти
- Реализация интерпретатора sc-моделей пользовательских интерфейсов

⇒ *пояснение**:

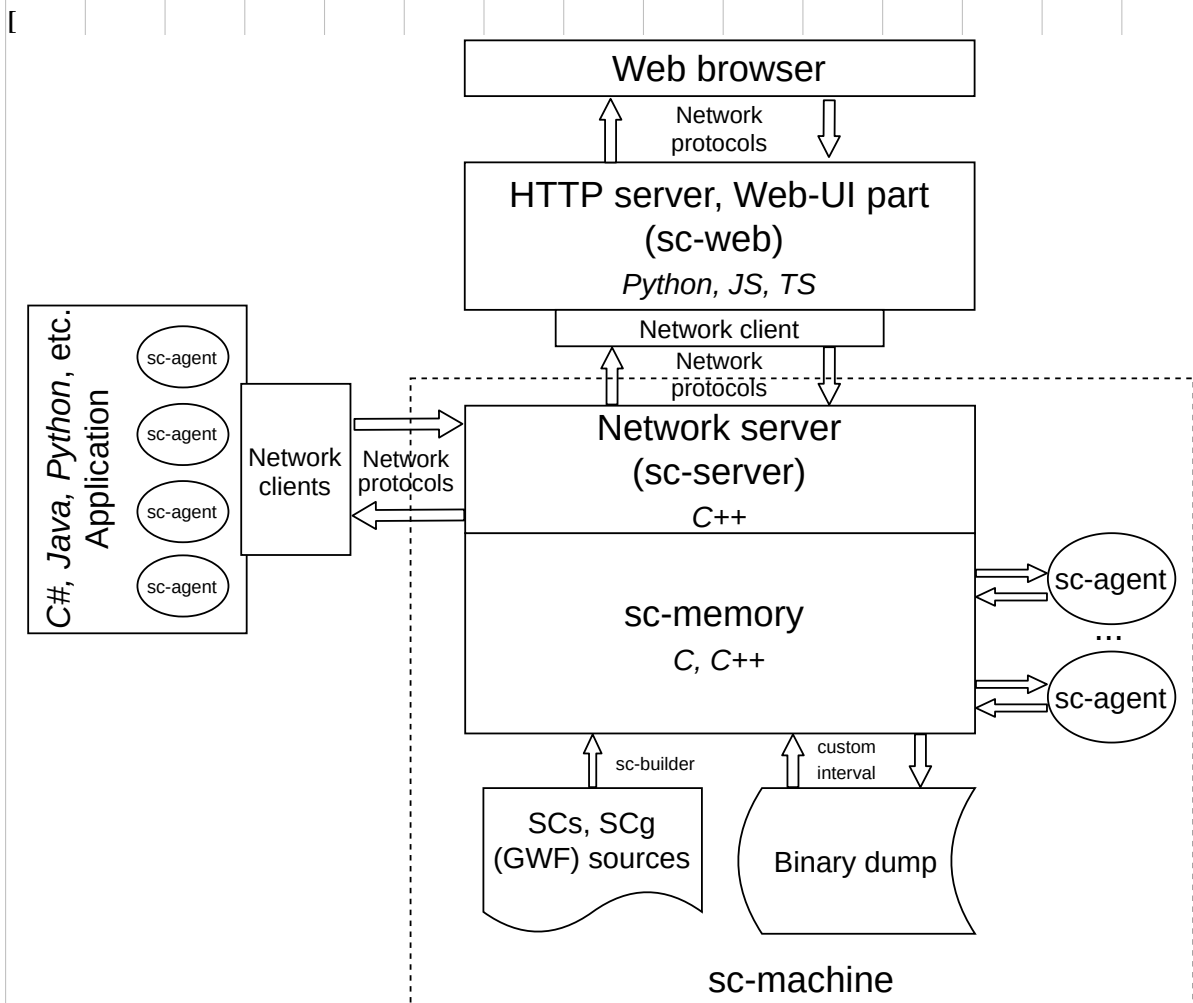
[Текущий Программный вариант реализации платформы интерпретации sc-моделей компьютерных систем является web-ориентированным, то есть с точки зрения современной архитектуры каждая ostis-система представляет собой web-сайт доступный онлайн посредством обычного браузера. Такой вариант реализации обладает очевидным преимуществом – доступ к системе возможен из любой точки мира, где есть Интернет, при этом для работы с системой не требуется никакого специализированного программного обеспечения. С другой стороны, такой вариант реализации обеспечивает возможность параллельной работы с системой нескольких пользователей.]

В то же время, взаимодействие клиентской и серверной части организовано таким образом, что web-интерфейс может быть легко заменен на настольный или мобильный интерфейс, как универсальный, так и специализированный.

Данный вариант реализации распространяется под open-source лицензией, для хранения исходных текстов используется хостинг Github и коллективная учетная запись ostis-dev.

Реализация является кроссплатформенной и может быть собрана из исходных текстов в различных операционных системах.]

⇒ *иллюстрация**:



⇒ *пояснение**:

[На приведенной иллюстрации видно, что ядром платформы является Программная модель sc-памяти (sc-machine), которая одновременно может взаимодействовать как с Реализацией интерпретатора sc-моделей пользовательских интерфейсов (sc-web ScWeb-эл), так и с любыми сторонними

приложениями по соответствующим сетевым протоколам. С точки зрения общей архитектуры *Реализация интерпретатора sc-моделей пользовательских интерфейсов* выступает как один из множества возможных внешних компонентов, взаимодействующих с *Программной моделью sc-памяти* по сети.]

Программная модель sc-памяти

:= [sc-machine]

:= [Программная модель семантической памяти, реализованная на основе традиционной линейной памяти и включающая средства хранения sc-конструкций и базовые средства для обработки этих конструкций, в том числе удаленного доступа к ним посредством соответствующих сетевых протоколов]

⇐ программная модель*:
sc-память

∈ программная модель sc-памяти на основе линейной памяти

⇒ основной репозиторий исходных текстов*:

[<https://github.com/ostis-dev/sc-machine.git>]

⇒ компонент программной системы*:

- Реализация sc-хранилища и средств доступа к нему
⇒ пояснение*:

[В рамках текущей *Программной модели sc-памяти* под *sc-хранилищем* понимается компонент программной модели, осуществляющий хранение sc-конструкций и доступ к ним через программный интерфейс. В общем случае *sc-хранилище* может быть реализовано по-разному. Кроме собственно *sc-хранилища* *Программная модель sc-памяти* включает также *Реализацию файловой памяти ostis-системы*, предназначенную для хранения содержимого *внутренних файлов ostis-систем*. Стоит отметить, что при переходе с *Программной модели sc-памяти* на ее аппаратную реализацию файловую память *ostis-системы* целесообразно будет реализовывать на основе традиционной линейной памяти (во всяком случае, на первых этапах развития *семантического компьютера*).]

- Реализация базового набора платформенно-зависимых sc-агентов и их общих компонентов
- Реализация подсистемы взаимодействия с внешней средой с использованием сетевых протоколов
- Реализация вспомогательных инструментальных средств для работы с sc-памятью
- Реализация scr-интерпретатора

⇒ программная документация*:

[<http://ostis-dev.github.io/sc-machine/>]

⇒ используемый язык программирования*:

- C
- C++
- Python

⇒ примечание*:

[Текущий вариант *Программной модели sc-памяти* предполагает возможность сохранения состояния (слепка) памяти на жесткий диск и последующей загрузки из ранее сохраненного состояния. Такая возможность необходима для перезапуска системы, в случае возможных сбоев, а также при работе с исходными текстами базы знаний, когда сборка из исходных текстов сводится к формированию слепка состояния памяти, который затем помещается в *Программную модель sc-памяти*.]

Реализация sc-хранилища и средств доступа к нему

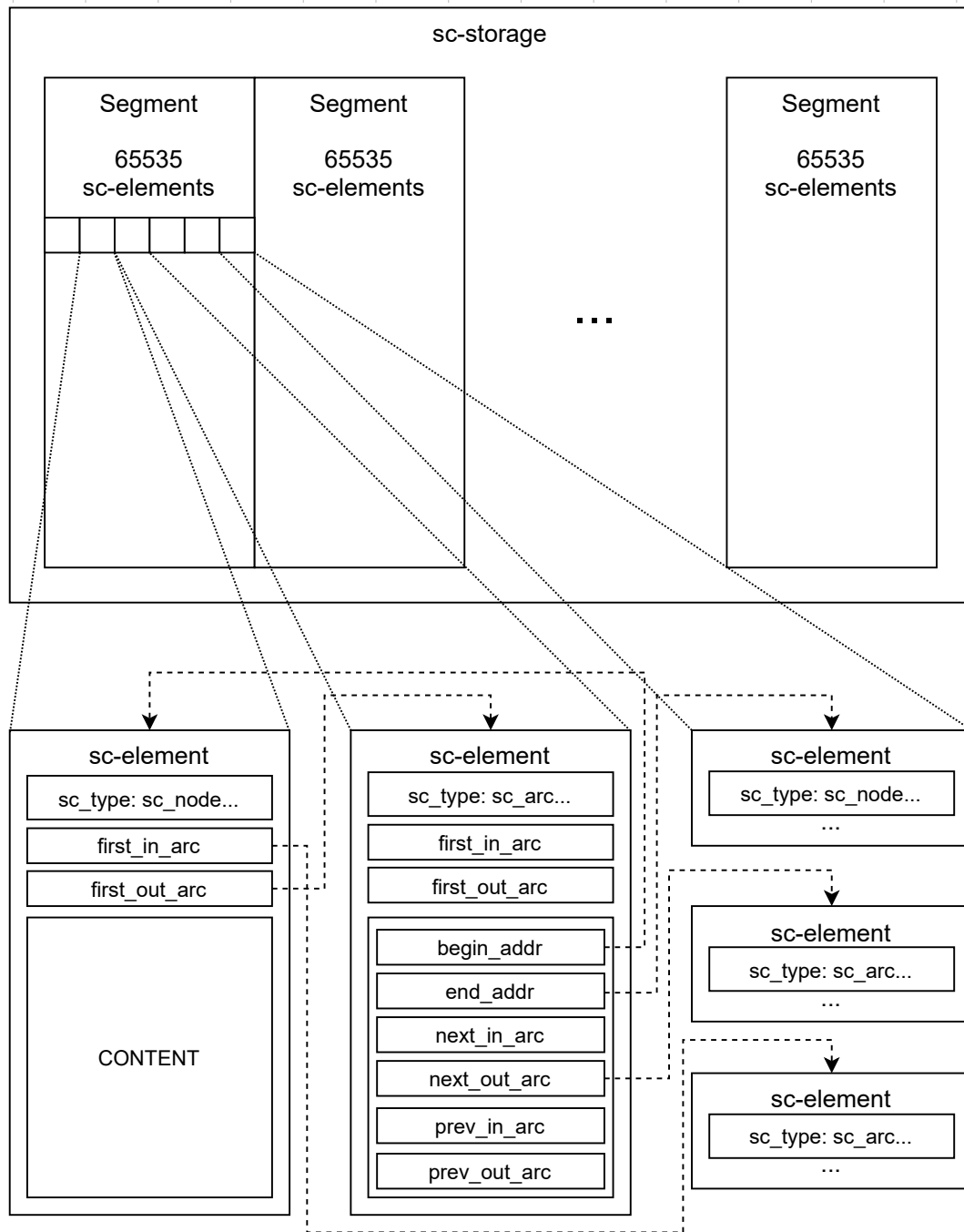
⇒ компонент программной системы*:

- Реализация sc-хранилища
- Реализация файловой памяти *ostis-системы*

Реализация sc-хранилища

∈ реализация sc-хранилища на основе линейной памяти

⇒ иллюстрация*:



⇒] класс объектов программной системы*:
 сегмент *sc*-хранилища

:= [страница *sc*-хранилища]

⇒ пояснение*:

[В рамках данной реализации *sc*-хранилища *sc*-память моделируется в виде набора сегментов, каждый из которых представляет собой фиксированного размера упорядоченную последовательность элементов *sc*-хранилища, каждый из которых соответствует конкретному *sc*-элементу. В настоящее время каждый сегмент состоит из $2^{16} - 1 = 65535$ элементов *sc*-хранилища. Выделение сегментов *sc*-хранилища позволяет, с одной стороны, упростить адресный доступ к элементам *sc*-хранилища, с другой стороны – реализовать возможность выгрузки части *sc*-памяти из оперативной памяти на файловую систему при необходимости. Во втором случае сегмент *sc*-хранилища становится минимальной (атомарной) выгружаемой частью *sc*-памяти. Механизм выгрузки сегментов реализуется в соответствии с существующими принципами организации виртуальной памяти в современных операционных системах.]

⇒ примечание*:

[Максимально возможное число сегментов ограничивается настройками программной реализации sc-хранилища (в настоящее время по умолчанию установлено количество $2^{16} - 1 = 65535$ сегментов, но в общем случае оно может быть другим). Таким образом, технически максимальное количество хранимых sc-элементов в текущей реализации составляет около 4.3×10^9 sc-элементов.]

⇒ примечание*:

[По умолчанию все сегменты физически располагаются в оперативной памяти, если объема памяти не хватает, то предусмотрен механизм выгрузки части сегментов на жесткий диск (механизм виртуальной памяти).]

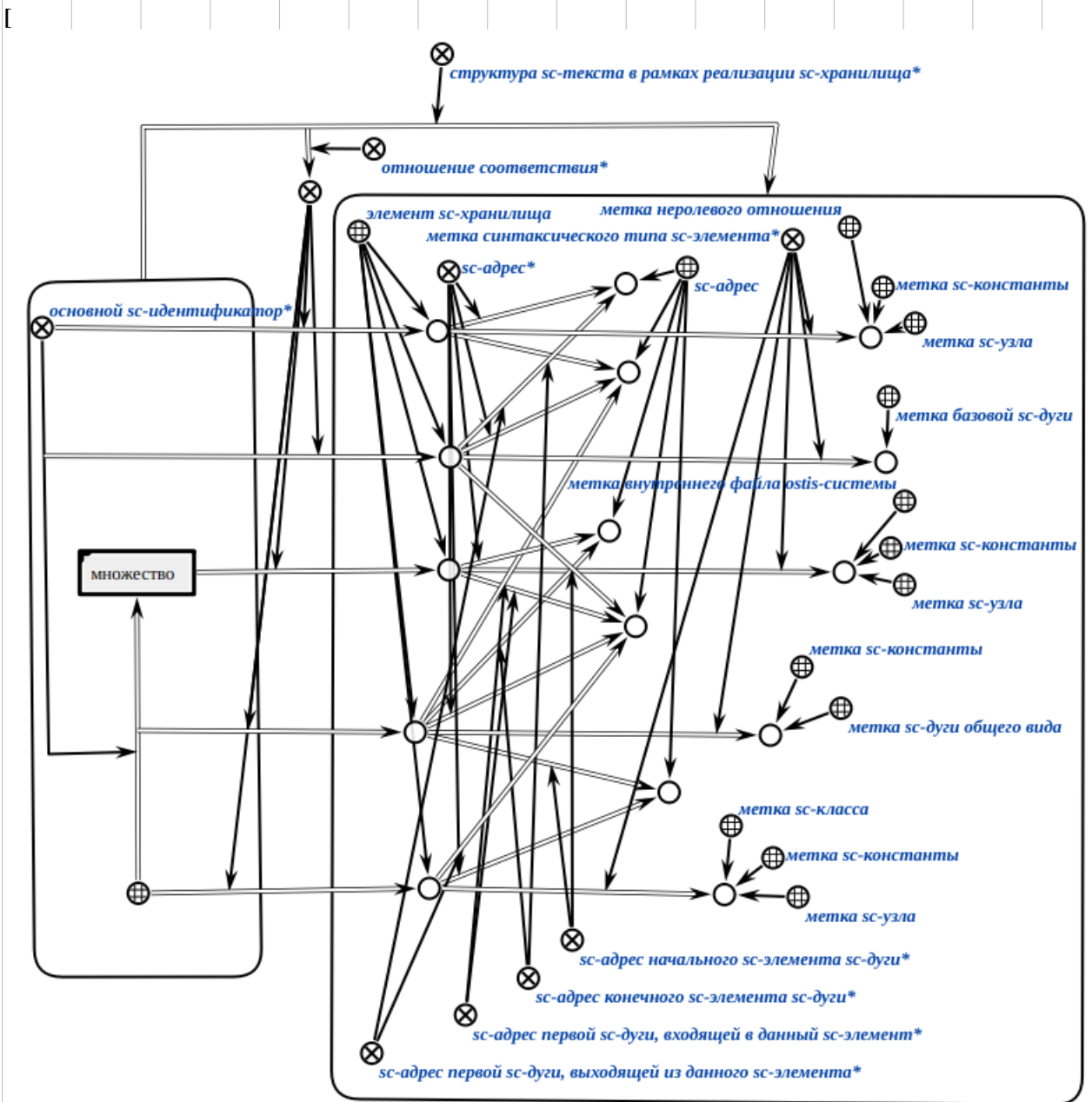
⇒ класс объектов программной системы*:

элемент sc-хранилища

⇒ пояснение*:

[Каждый сегмент состоит из набора структур данных, описывающих конкретные sc-элементы (элементы sc-хранилища). Независимо от типа описываемого sc-элемента каждый элемент sc-хранилища имеет фиксированный размер (в текущий момент – 48 байт), что обеспечивает удобство их хранения. Таким образом, максимальный размер базы знаний в текущей программной модели sc-памяти может достигнуть 223 Гб (без учета содержимого внутренних файлов ostis-системы, хранимого на внешней файловой системе).]

⇒ пример*:



] ⇒ пояснение*:

[Для наглядности в данном примере опущены *метки уровня доступа*]

sc-адрес

- := [адрес элемента *sc-хранилища*, соответствующего заданному *sc-элементу*, в рамках текущего состояния реализации *sc-хранилища* в составе программной модели *sc-памяти*]
- ⇒ *пояснение**:
[Каждый элемент *sc-хранилища* в текущей реализации может быть однозначно задан его адресом (*sc-адресом*), состоящим из номера сегмента и номера *элемента sc-хранилища* в рамках сегмента. Таким образом, *sc-адрес* служит уникальными координатами *элемента sc-хранилища* в рамках *Реализации sc-хранилища*.]
- ⇒ *примечание**:
[*Sc-адрес* никак не учитывается при обработке базы знаний на семантическом уровне и необходим только для обеспечения доступа к соответствующей структуре данных, хранящейся в линейной памяти на уровне *Реализации sc-хранилища*.]
- ⇒ *примечание**:
[В общем случае *sc-адрес* элемента *sc-хранилища*, соответствующего заданному *sc-элементу*, может меняться, например, при пересборке базы знаний из исходных текстов и последующем перезапуске системы. При этом *sc-адрес* элемента *sc-хранилища*, соответствующего заданному *sc-элементу*, непосредственно в процессе работы системы в текущей реализации меняться не может.]
- ⇒ *примечание**:
[Для простоты будем говорить "*sc-адрес sc-элемента*", имея в виду *sc-адрес элемента sc-хранилища*, однозначно соответствующего данному *sc-элементу*.]
- ⇒ *семейство отношений, однозначно задающих структуру заданной сущности**:
- *номер сегмента sc-хранилища**
 - *номер элемента sc-хранилища в рамках сегмента**

элемент sc-хранилища

- := [ячейка *sc-хранилища*]
- := [элемент *sc-хранилища*, соответствующий *sc-элементу*]
- := [образ *sc-элемента* в рамках *sc-хранилища*]
- := [структура данных, каждый экземпляр которой соответствует одному *sc-элементу* в рамках *sc-хранилища*]
- ⇒ *пояснение**:
[Каждый элемент *sc-хранилища*, соответствующий некоторому *sc-элементу*, описывается его синтаксическим типом (меткой), а также независимо от типа указывается *sc-адрес* первой входящей в данный *sc-элемент sc-дуги* и первой выходящей из данного *sc-элемента sc-дуги* (могут быть пустыми, если таких *sc-дуг* нет).
Оставшиеся байты в зависимости от типа соответствующего *sc-элемента* (*sc-узел* или *sc-дуга*) могут использоваться либо для хранения содержимого внутреннего файла *ostis-системы* (может быть пустым, если *sc-узел* не является знаком файла), либо для хранения спецификации *sc-дуги*.]
- ⇒ *разбиение**:
- { ● *элемент sc-хранилища, соответствующий sc-узлу*
 - ⇒ *семейство отношений, однозначно задающих структуру заданной сущности**:
 - { ● *метка синтаксического типа sc-элемента**
 - *метка уровня доступа sc-элемента**
 - *sc-адрес первой sc-дуги, выходящей из данного sc-элемента**
 - *sc-адрес первой sc-дуги, входящей в данный sc-элемент**
 - *содержимое элемента sc-хранилища**
 - ⇒ *второй домен**:
содержимое элемента sc-хранилища
 - := [содержимое элемента *sc-хранилища*, соответствующего внутреннему файлу *ostis-системы*]
 - ⇒ *пояснение**:
[Каждый *sc-узел* в текущей реализации может иметь содержимое (может стать *внутренним файлом ostis-системы*). В случае, если размер содержимого внутреннего файла *ostis-системы* не превышает 48 байт (размер *спецификации sc-дуги в рамках sc-хранилища*, например *небольшой строковый sc-идентификатор*), то это содержимое явно хранится в рамках элемента *sc-хранилища* в виде последовательности байт. В противном случае оно помещается в специальном образом организованную файловую память (за ее организацию отвечает отдельный модуль платформы, который в общем случае может быть устроен по-разному), а в рамках элемента *sc-хранилища* хранится

уникальный адрес соответствующего файла, позволяющий быстро найти его на файловой системе.]

}

⇒

*примечание**:

[*sc-адрес первой sc-дуги, выходящей из данного sc-элемента**, *sc-адрес первой sc-дуги, входящей в данный sc-элемент** и *содержимое элемента sc-хранилища** в общем случае могут отсутствовать (быть нулевыми, "пустыми"), но размер элемента в байтах останется тем же.]

- элемент *sc-хранилища, соответствующий sc-дуге*

⇒

*семейство отношений, однозначно задающих структуру заданной сущности**:

{

- *метка синтаксического типа sc-элемента**
- *метка уровня доступа sc-элемента**
- *sc-адрес первой sc-дуги, выходящей из данного sc-элемента**
- *sc-адрес первой sc-дуги, входящей в данный sc-элемент**
- *спецификация sc-дуги в рамках sc-хранилища**

⇒ *второй домен**:

спецификация sc-дуги в рамках sc-хранилища

⇒ *семейство отношений, однозначно задающих структуру заданной сущности**:

{

- *sc-адрес начального sc-элемента sc-дуги**
- *sc-адрес конечного sc-элемента sc-дуги**
- *sc-адрес следующей sc-дуги, выходящей из того же sc-элемента**
- *sc-адрес следующей sc-дуги, входящей в тот же sc-элемент**
- *sc-адрес предыдущей sc-дуги, выходящей из того же sc-элемента**
- *sc-адрес предыдущей sc-дуги, входящей в тот же sc-элемент**

}

}

⇒

*примечание**:

[*sc-ребра* в текущий момент хранятся так же, как *sc-дуги*, то есть имеют начальный и конечный *sc-элементы*, отличие заключается только в *метке синтаксического типа sc-элемента*. Это приводит к ряду неудобств при обработке, но *sc-ребра* используются в настоящее время достаточно редко.]

}

⇒

*примечание**:

[С точки зрения программной реализации структура данных для хранения *sc-узла* и *sc-дуги* остается та же, но в ней меняется список полей (компонентов).

Кроме того, как можно заметить каждый элемент *sc-хранилища* (в том числе, *элемент sc-хранилища, соответствующий sc-дуге*) не хранит список *sc-адресов* связанных с ним *sc-элементов*, а хранит *sc-адреса* одной выходящей и одной входящей дуги, каждая из которых в свою очередь хранит *sc-адреса* следующей и предыдущей дуг в списке исходящих и входящих *sc-дуг* для соответствующих элементов.

Все перечисленное позволяет:

- сделать размер такой структуры фиксированным (в настоящее время 48 байт) и не зависящим от синтаксического типа хранимого *sc-элемента*;
- обеспечить возможность работы с *sc-элементами* без учета их синтаксического типа в случаях, когда это необходимо (например, при реализации поисковых запросов вида "Какие *sc-элементы* являются элементами данного множества", "Какие *sc-элементы* непосредственно связаны с данным *sc-элементом*" и т.д.);
- обеспечить возможность доступа к *элементу sc-хранилища* за константное время;
- обеспечить возможность помещения *элемента sc-хранилища* в процессорный кэш, что в свою очередь, позволяет ускорить обработку *sc-конструкций*;

]

⇒

*примечание**:

[Текущая Программная модель *sc-памяти* предполагает, что вся *sc-память* физически расположена на одном компьютере. Для реализации распределенного варианта Программной модели *sc-памяти* предполагается расширить *sc-адрес* указанием адреса того физического устройства, где хранится соответствующий элемент *sc-хранилища*.]

метка синтаксического типа sc-элемента

:= [уникальный числовой идентификатор, однозначно соответствующий заданному типу sc-элементов и приписываемый соответствующему элементу sc-хранилища на уровне реализации]

⇒ *примечание**:

[Очевидно, что тип (класс, вид) sc-элемента в sc-памяти может быть задан путем явного указания принадлежности данного sc-элемента соответствующему классу (sc-узел, sc-дуга и т.д.).

Однако, в рамках *платформы интерпретации sc-моделей компьютерных систем* должен существовать какой-либо набор *меток синтаксического типа sc-элемента*, которые задают тип элемента на уровне платформы и не имеют соответствующей sc-дуги принадлежности (а точнее – базовой sc-дуги), явно хранимой в рамках sc-памяти (ее наличие подразумевается, однако она не хранится явно, поскольку это приведет к бесконечному увеличению числа sc-элементов, которые необходимо хранить в sc-памяти). Как минимум, должна существовать метка, соответствующая классу *базовая sc-дуга*, поскольку явное указание принадлежности sc-дуги данному классу порождает еще одну *базовую sc-дугу*.

Таким образом, *базовые sc-дуги*, обозначающие принадлежность sc-элементов некоторому известному ограниченному набору классов представлены *неявно*. Этот факт необходимо учитывать в ряде случаев, например, при проверке принадлежности sc-элемента некоторому классу, при поиске всех выходящих sc-дуг из заданного sc-элемента и т.д.

При необходимости некоторые из таких неявно хранимых sc-дуг могут быть представлены явно, например, в случае, когда такую sc-дугу необходимо включить в какое-либо множество, то есть провести в нее другую sc-дугу. В этом случае возникает необходимость синхронизации изменений, связанных с данной sc-дугой (например, ее удалении), в явном и неявном ее представлении. В текущей *Реализации sc-хранилища* данный механизм не реализован.

Таким образом, полностью отказаться от *меток синтаксического типа sc-элементов* невозможно, однако увеличение их числа хоть и повышает производительность платформы за счет упрощений некоторых операций по проверке типов sc-элемента, но приводит к увеличению числа ситуаций, в которых необходимо учитывать явное и неявное представление sc-дуг, что, в свою очередь, усложняет развитие платформы и разработку программного кода для обработки хранимых sc-конструкций.]

⇐ *второй домен**:

*метка синтаксического типа sc-элемента**

▷ *метка sc-узла*

⇒ *числовое выражение в шестнадцатеричной системе**:
[0x1]

▷ *метка внутреннего файла ostis-системы*

⇒ *числовое выражение в шестнадцатеричной системе**:
[0x2]

▷ *метка sc-ребра общего вида*

⇒ *числовое выражение в шестнадцатеричной системе**:
[0x4]

▷ *метка sc-дуги общего вида*

⇒ *числовое выражение в шестнадцатеричной системе**:
[0x8]

▷ *метка sc-дуги принадлежности*

⇒ *числовое выражение в шестнадцатеричной системе**:
[0x10]

▷ *метка sc-константы*

⇒ *числовое выражение в шестнадцатеричной системе**:
[0x20]

▷ *метка sc-переменной*

⇒ *числовое выражение в шестнадцатеричной системе**:
[0x40]

▷ *метка позитивной sc-дуги принадлежности*

⇒ *числовое выражение в шестнадцатеричной системе**:
[0x80]

▷ *метка негативной sc-дуги принадлежности*

⇒ *числовое выражение в шестнадцатеричной системе**:
[0x100]

▷ *метка нечеткой sc-дуги принадлежности*

- ⇒ числовое выражение в шестнадцатеричной системе*:
[0x200]
- ⊃ метка *постоянной sc-дуги*
- ⇒ числовое выражение в шестнадцатеричной системе*:
[0x400]
- ⊃ метка *временной sc-дуги*
- ⇒ числовое выражение в шестнадцатеричной системе*:
[0x800]
- ⊃ метка *небинарной sc-связки*
- ⇒ числовое выражение в шестнадцатеричной системе*:
[0x80]
- ⊃ метка *sc-структуры*
- ⇒ числовое выражение в шестнадцатеричной системе*:
[0x100]
- ⊃ метка *ролевого отношения*
- ⇒ числовое выражение в шестнадцатеричной системе*:
[0x200]
- ⊃ метка *неролевого отношения*
- ⇒ числовое выражение в шестнадцатеричной системе*:
[0x400]
- ⊃ метка *sc-класса*
- ⇒ числовое выражение в шестнадцатеричной системе*:
[0x800]
- ⊃ метка *абстрактной сущности*
- ⇒ числовое выражение в шестнадцатеричной системе*:
[0x1000]
- ⊃ метка *материальной сущности*
- ⇒ числовое выражение в шестнадцатеричной системе*:
[0x2000]
- ⊃ метка *константной позитивной постоянной sc-дуги принадлежности*
- := [метка базовой sc-дуги]
- := [метка sc-дуги основного вида]
- ⇐ *пересечение**:
- метка *sc-дуги принадлежности*
 - метка *sc-константы*
 - метка *позитивной sc-дуги принадлежности*
 - метка *постоянной sc-дуги*
- ⇒ *примечание**:
[метки синтаксических типов *sc-элементов* могут комбинироваться между собой для получения более частных классов меток. С точки зрения программной реализации такая комбинация выражается операцией побитового сложения значений соответствующих меток.]
- ⊃ метка *переменной позитивной постоянной sc-дуги принадлежности*
- ⇐ *пересечение**:
- метка *sc-дуги принадлежности*
 - метка *sc-переменной*
 - метка *позитивной sc-дуги принадлежности*
 - метка *постоянной sc-дуги*
- ⇒ *примечание**:
[Числовые выражения некоторых классов меток могут совпадать. Это сделано для уменьшения размера элемента *sc-хранилища* за счет уменьшения максимального размера метки. Конфликт в данном случае не возникает, поскольку такие классы меток не могут комбинироваться, например *метка ролевого отношения* и *метка нечеткой sc-дуги принадлежности*.]
- ⇒ *примечание**:
[Важно отметить, что каждому из выделенных классов меток (кроме классов, получаемых путем комбинации других классов) однозначно соответствует порядковый номер бита в линейной памяти, что можно заметить, глядя на соответствующие числовые выражения классов меток. Это означает, что классы меток не включают друг в друга, например, указание *метки позитивной sc-дуги принадлежности* не означает

автоматическое указание *метки sc-дуги принадлежности*. Это позволяет сделать операции комбинирования и сравнения меток более эффективными.]

⇐ *недостатки текущего состояния**:

- { • [На данный момент число *меток синтаксического типа sc-элемента* достаточно велико, что приводит к возникновению достаточно большого числа ситуаций, в которых нужно учитывать явное и неявное хранение *sc-дуг принадлежности* соответствующим классам. С другой стороны, изменение набора меток с какой-либо целью в текущем варианте реализации представляет собой достаточно трудоемкую задачу (с точки зрения объема изменений в программном коде платформы и *sc-агентов*, реализованных на уровне платформы), а расширение набора меток без увеличения объема элемента *sc-хранилища* в байтах оказывается и вовсе невозможным.]

⇒ *вариант решения**:

[Решением данной проблемы является максимально возможная минимизация числа меток, например, до числа меток, соответствующих *Алфавиту SC-кода*. В таком случае принадлежность *sc-элементов* любым другим классам будет записываться явно, а число ситуаций, в которых необходимо будет учитывать неявное хранение *sc-дуг*, будет минимальным.]

- [Некоторые метки из текущего набора *меток синтаксического типа sc-элемента* используются достаточно редко (например, *метка sc-ребра общего вида* или *метка негативной sc-дуги принадлежности*), в свою очередь, в *sc-памяти* могут существовать классы, имеющие достаточно много элементов (например, *бинарное отношение* или *число*). Данный факт не позволяет в полной мере использовать эффективность наличия меток.]

⇒ *вариант решения**:

[Решением данной проблемы является отказ от заранее известного набора меток и переход к динамическому набору меток (при этом их число может оставаться фиксированным). В этом случае набор классов, выражаемых в виде меток будет формироваться на основании каких-либо критериев, например, числа элементов данного класса или частоты обращений к нему.]

}

метка уровня доступа sc-элемента

⇐ *второй домен**:

*метка уровня доступа sc-элемента**

⇒ *обобщенная структура**:

- { • *метка уровня доступа sc-элемента на чтение*
- *метка уровня доступа sc-элемента на запись*

}

⇒ *пояснение**:

[В текущей *Реализации sc-хранилища* *метки уровня доступа* используются для того, чтобы обеспечить возможность ограничения доутспа некоторых процессов в *sc-памяти* к некоторым *sc-элементам*, хранимым в *sc-памяти*.

Каждому элементу *sc-хранилища* соответствует *метка уровня доступа sc-элемента на чтение* и *метка уровня доступа sc-элемента на запись*, каждая из которых выражается числом от 0 до 255.

В свою очередь, каждому процессу (чаще всего, соответствующему некоторому *sc-агенту*), который пытается получить доступ к данному элементу *sc-хранилища* (прочитать или изменить его) соответствует уровень доступа на чтение и запись, выраженный в том же числовом диапазоне. Указанный уровень доступа для процесса является частью *контекста процесса*. Доступ на чтение или запись к элементу *sc-хранилища* не разрешается, если уровень доступа соответственно на чтение или запись у процесса ниже, чем у элемента *sc-хранилища*, к которому осуществляется доступ.

Таким образом нулевое значение *метки уровня доступа sc-элемента на чтение* и *метки уровня доступа sc-элемента на запись* означает, что любой процесс может получить неограниченный доступ к данному элементу *sc-хранилища*.]

sc-итератор

:= [ScIterator]

⇐ *класс компонентов**:

Реализация sc-хранилища

⇒ *пояснение**:

[С функциональной точки зрения *sc-итераторы* как часть *Реализации sc-хранилища* представляют собой базовое средство доступа к конструкциям, хранимым в *sc-памяти*, которое позволяет осуществить чтение (просмотр) конструкций, изоморфных простейшим шаблонам – *трехэлементным sc-конструкциям* и

пятиэлементным *sc*-конструкциям (см. Раздел Предметная область и онтология решателей задач *ostis-систем*).

С точки зрения реализации *sc-итератор* представляет собой структуру данных, которая соответствует определенному дополнительно уточняемому классу *sc*-конструкций и позволяет при помощи соответствующего набора функций последовательно осуществлять просмотр всех *sc*-конструкций данного класса, представленных в текущем состоянии *sc*-памяти (итерацию по *sc*-конструкциям).

Каждому классу *sc-итераторов* соответствует некоторый известный класс (шаблон, образец) *sc*-конструкций. При создании *sc*-итератора данный шаблон уточняется, то есть некоторым (как минимум одному) элементам шаблона ставится в соответствие конкретный заранее известный *sc-элемент* (отправная точка при поиске), а другим элементам шаблона (тем, которые нужно найти) ставится в соответствие некоторый тип *sc*-элемента из числа типов, соответствующих *меткам синтаксического типа sc-элемента*.

Далее путем вызова соответствующей функции (или метода класса в ООП) осуществляется последовательный просмотр всех *sc*-конструкций, соответствующих полученному шаблону (с учетом указанных типов *sc*-элементов и заранее заданных известных *sc*-элементов), то есть *sc-итератор* последовательно "переключается" с одной конструкции на другую до тех пор, пока такие конструкции существуют. Проверка существования следующей конструкции проверяется непосредственно перед переключением. В общем случае конструкций, соответствующих указанному шаблону, может не существовать, в этом случае итерирование происходить не будет (будет 0 итераций).

На каждой итерации в *sc*-итератор записываются *sc*-адреса *sc*-элементов, входящих в соответствующую *sc*-конструкцию, таким образом найденные элементы могут быть обработаны нужным образом в зависимости от задачи.]

⊃ *трехэлементный sc-итератор*
⇒ *класс sc-конструкций**:
трехэлементная sc-конструкция

⊃ *пятиэлементный sc-итератор*
⇒ *класс sc-конструкций**:
пятиэлементная sc-конструкция
⇒ *примечание**:

[В настоящее время *пятиэлементный sc-итератор* реализуется на основе *трехэлементных sc-итераторов* и в этом смысле не является атомарным. Однако, введение *пятиэлементных sc-итераторов* целесообразно с точки зрения удобства разработчика программ обработки *sc*-конструкций.]

sc-шаблон

:= [ScTemplate]

:= [структура данных в линейной памяти, описывающая обобщенную *sc*-структуру, которая в свою очередь может быть либо явно представлена *sc*-памяти, либо не представлена в ее текущем состоянии, но может быть представлена при необходимости]

⇐ *класс компонентов**:
Реализация sc-хранилища

⇒ *пояснение**:

[*Sc-итераторы* позволяют осуществлять поиск только *sc*-конструкций простейшей конфигурации. Для реализации поиска *sc*-конструкций более сложной конфигурации, а также генерации сложных *sc*-конструкций используются *sc-шаблоны*, на основе которых затем осуществляется поиск или генерация конструкций. *Sc-шаблон* представляет собой структуру данных, соответствующую некоторой *обобщенной структуре*, т.е. *структуре*, содержащей *sc-переменные*. При помощи соответствующего набора функций можно осуществлять

- поиск в текущем состоянии *sc*-памяти всех *sc*-конструкций, изоморфных заданному шаблону. В качестве параметров поиска можно указать значения для каких-либо из *sc*-переменных в составе шаблона. После осуществления поиска будет сформировано множество результатов поиска, каждый из которых представляет собой множество пар вида “*sc*-переменная из шаблона – соответствующая ей *sc*-константа”. Данное множество может быть пустым (в текущем состоянии *sc*-памяти нет конструкций, изоморфных заданному образцу) или содержать один или более элементов. Подстановка значений *sc*-переменных может осуществляться как по *sc*-адресу, так и по системному *sc*-идентификатору;
- генерацию *sc*-конструкции, изоморфной заданному шаблону. Параметры и результаты генерации формируются так же, как в случае поиска, за исключением того, что в случае генерации результат всегда один и множество результатов не формируется;

Таким образом, каждый *sc-шаблон* фактически задает множество шаблонов, формируемых путем указания значений для *sc-переменных*, входящих в исходный шаблон.

Важно отметить, что *sc-шаблон* представляет собой структуру данных в линейной памяти, соответствующую некоторой *обобщенной структуре* в *sc-памяти*, но не саму эту *обобщенную структуру*. Это означает, что *sc-шаблон* может быть автоматически сформирован на основе *обобщенной структуры*, явно представленной в *sc-памяти*, а также сформирован на уровне программного кода путем вызова соответствующих функций (методов). Во втором случае *sc-шаблон* будет существовать только в линейной памяти и соответствующая *обобщенная структура* не будет явно представлена в *sc-памяти*. В этом случае подстановка значений *sc-переменных* будет возможна только по системному *sc-идентификатору*, поскольку *sc-адресов* у соответствующих элементов шаблона существовать не будет.]

⇒ *примечание**:

[При поиске *sc-конструкций*, изоморфных заданному шаблону, крайне важно с точки зрения производительности с какого *sc-элемента* начинать поиск. Как известно, в общем случае задача поиска в графе представляет собой NP-полную задачу, однако поиск в *sc-графе* позволяет учитывать семантику обрабатываемой информации, что, в свою очередь, позволяет существенно снизить время поиска.

Одним из возможных вариантов оптимизации алгоритма поиска, реализованным на данный момент, является упорядочение трехэлементных *sc-конструкций*, входящих в состав *sc-шаблона*, по очередности поиска по этим *sc-конструкциям* по критерию снижения числа возможных вариантов поиска, которые порождает та или иная трехэлементная *sc-конструкция*, содержащая *sc-переменные*. Так, в первую очередь при поиске выбираются те трехэлементные *sc-конструкции*, которые изначально содержат две *sc-константы*, затем те, которые изначально содержат одну *sc-константу*. После выполнения шага поиска приоритет *sc-конструкций* изменяется с учетом результатов, полученных на предыдущем шаге.

Другой вариант оптимизации основывается на той особенности формализации в SC-коде, что в общем случае число *sc-дуг*, входящих в некоторый *sc-элемент*, как правило значительно меньше числа выходящих из него *sc-дуг*. Таким образом, целесообразным оказывается осуществлять поиск вначале по входящим *sc-дугам*.]

⇒ *примечание**:

[Можно предположить, что возможности, предоставляемые *sc-шаблонами* позволяют полностью исключить использование *sc-итераторов*. Однако это не совсем так по следующим причинам:

- функции поиска и генерации по шаблону реализуются на основе *sc-итераторов*, как базового средства поиска *sc-конструкций* в рамках *Реализации sc-хранилища*.
- *sc-итераторы* дают возможность более гибко организовать процесс поиска с учетом семантики конкретных *sc-элементов*, участвующих в поиске. Так например, можно учесть тот факт, что для некоторых *sc-элементов* число входящих *sc-дуг* значительно меньше, чем выходящих (или наоборот) таким образом, при поиске конструкций, содержащих такие *sc-элементы* более эффективно начать перебор с тех участков, где дуг потенциально меньше.

]

контекст процесса в рамках программной модели *sc-памяти*

:= [ScContext]

:= [контекст процесса, выполняемого на уровне программной модели *sc-памяти*]

:= [метаописание процесса в *sc-памяти*, выполняемого на уровне программной модели *sc-памяти*]

:= [структура данных, содержащая метаинформацию о процессе, выполняемом в *sc-памяти* на уровне платформы]

← *класс компонентов**:

Реализация sc-хранилища

⇒ *пояснение**:

[Каждому процессу, выполняемому в *sc-памяти* на уровне *платформы интерпретации sc-моделей компьютерных систем* (и чаще всего соответствующего некоторому *sc-агенту*, реализованному на уровне платформы) ставится в соответствие *контекст процесса*, который является структурой данных, описывающей метаинформацию о данном процессе. На текущий момент контекст процесса содержит сведения об уровне доступа на чтение и запись для данного процесса (См. *метка уровня доступа sc-элемента*).

При вызове в рамках процесса любых функций (методов), связанных с доступом к хранимым в *sc-памяти* конструкциям одним из параметров обязательно является *контекст процесса*.]

блокировка *sc-элемента* в рамках программной модели *sc-памяти*

:= [ScLock]

← *класс компонентов**:

Реализация sc-хранилища⇒ *смотри*:**Предметная область и онтология действий, задач, планов, протоколов и методов, реализуемых ostis-системой, а также внутренних агентов, выполняющих эти действия***подписка на событие в sc-памяти в рамках программной модели sc-памяти**

:= [ScEvent]

:= [структура данных, описывающая в рамках программной модели sc-памяти соответствие между классом событий в sc-памяти и действиями, которые должно быть совершены при возникновении в sc-памяти событий данного класса]

⇐ *класс компонентов*:**Реализация sc-хранилища*⇒ *пояснение*:*

[Для того, чтобы обеспечить возможность создания sc-агентов в рамках *платформы интерпретации sc-моделей компьютерных систем* реализована возможность создать подписку на событие, принадлежащее одному из классов *элементарных событий в sc-памяти** (см. Раздел “*Предметная область и онтология темпоральных сущностей базы знаний ostis-системы*”), уточнив при этом sc-элемент, с которым должно быть связано событие данного класса (например, sc-элемент, для которого должна появиться входящая или исходящая sc-дуга). Подписка на событие представляет собой структуру данных, описывающую класс ожидаемых событий и функцию в программном коде, которая должна быть вызвана при возникновении данного события.

Все подписки на события регистрируются в рамках таблицы событий. При любом изменении в sc-памяти происходит просмотр данной таблицы и запуск функций, соответствующих произошедшему событию.

В текущей реализации обработка каждого события осуществляется в отдельном потоке операционной системы, при этом на уровне реализации задается параметр, описывающий число максимальных потоков, которые могут выполняться параллельно.

Таким образом оказывается возможным реализовать sc-агенты, реагирующие на события в sc-памяти, а также при выполнении некоторого процесса в sc-памяти приостановить его работу и дождаться возникновения некоторого события (например, создать подзадачу некоторому коллективу sc-агентов и дождаться ее решения).]

Реализация файловой памяти ostis-системы⇒ *пояснение*:*

[Для хранения содержимого внутренних файлов ostis-систем, размер которого превышает 48 байт, используются файлы, явно хранимые на файловой системе, доступ к которой осуществляется средствами операционной системы, на которой работает *Программный вариант реализации платформы интерпретации sc-моделей компьютерных систем*.

В общем случае множество различных внутренних файлов ostis-системы могут иметь одинаковое содержимое. Было бы разумно не хранить содержимое одинаковых файлов дважды. Для этого при создании соответствующего sc-узла и указании файла на файловой системе, который является содержимым данного sc-узла, вычисляется hash-сумма содержимого с помощью алгоритма SHA256. В результате получается строка из 32 символов, которая и выступает в качестве *содержимого элемента sc-хранилища**. Само же содержимое копируется в файл на файловой системе, путь к которому строится на основании hash-суммы. Рядом с этим файлом создается файл, в котором хранятся sc-адреса всех sc-узлов, имеющих одно и то же ранее указанное содержимое. Таким образом, для того, чтобы найти все sc-узлы, имеющие указанное содержимое, необходимо вычислить hash-сумму искомого содержимого-образца и проверить наличие файла на файловой системе по пути, вычисляемому из hash-суммы и если он существует, то вернуть список хранящихся sc-адресов.

Кроме того, для реализации быстрого поиска sc-элементов по их строковым sc-идентификаторам или их фрагментам (подстрокам) используется дополнительное хранилище вида ключ-значение, которое ставит в соответствие *строковому sc-идентификатору* *sc-адрес* того *sc-элемента*, идентификатором которого является данная строка (в случае основного и системного sc-идентификатора) или *sc-элемента*, который является знаком *внутреннего файла ostis-системы* (в случае неосновного sc-идентификатора).]

Реализация базового набора платформенно-зависимых sc-агентов и их общих компонентов

:= [sc-kpm]

⇒ *компонент программной системы*:*

- *Реализация базового набора поисковых sc-агентов*
⇒ *используемый язык программирования*:*

- *C*
- ⇒ *компонент программной системы**:
 - *Реализация Абстрактного sc-агента поиска семантической окрестности заданной сущности*
 - *Реализация Абстрактного sc-агента поиска всех сущностей, частных по отношению к заданной*
 - *Реализация Абстрактного sc-агента поиска всех сущностей, общих по отношению к заданной*
 - *Реализация Абстрактного sc-агента поиска всех sc-идентификаторов, соответствующих заданной сущности*
 - *Реализация Абстрактного sc-агента поиска базовых sc-дуг, инцидентных заданному sc-элементу*
 - ⇒ *компонент программной системы**:
 - *Реализация Абстрактного sc-агента поиска базовых sc-дуг, входящих в заданный sc-элемент*
 - *Реализация Абстрактного sc-агента поиска базовых sc-дуг, выходящих из заданного sc-элемента*
 - *Реализация Абстрактного sc-агента поиска базовых sc-дуг, входящих в заданный sc-элемент, с указанием множеств, которым принадлежат эти sc-дуги*
 - *Реализация Абстрактного sc-агента поиска базовых sc-дуг, выходящих из заданного sc-элемента, с указанием множеств, которым принадлежат эти sc-дуги*
- *Реализация базового механизма сборки информационного мусора*
- ⇒ *используемый язык программирования**:
 - *C*
- ⇒ *примечание**:

[Текущая реализация механизма сборки информационного мусора содержит один sc-агент, реагирующий на явное добавление какого-либо sc-элемента во множество “информационный мусор” и осуществляющий физическое удаление этого sc-элемента из sc-памяти]
- *Реализация базового набора интерфейсных sc-агентов*
- ⇒ *используемый язык программирования**:
 - *C++*
- ⇒ *компонент программной системы**:
 - *Реализация Абстрактного sc-агента обработки команд пользовательского интерфейса*
 - *Реализация Абстрактного sc-агента трансляции из внутреннего представления знаний во промежуточный транспортный формат*
 - ⇒ *примечание**:

[В настоящее время используется подход, при котором независимо от формы внешнего представления информации, информация хранимая в sc-памяти вначале транслируется в промежуточный транспортный формат на базе JSON, который затем обрабатывается sc-агентами пользовательского интерфейса, входящими в состав *Реализации интерпретатора sc-моделей пользовательских интерфейсов*]

Реализация подсистемы взаимодействия с внешней средой с использованием сетевых протоколов

- ⇒ *компонент программной системы**:
 - *Реализация подсистемы взаимодействия с внешней средой с использованием протокола SCTP*
 - *Реализация подсистемы взаимодействия с внешней средой с использованием протоколов на основе формата JSON*
- ⇒ *пояснение**:

[Взаимодействие программной модели sc-памяти с внешними ресурсами может осуществляться посредством специализированного программного интерфейса (API), однако этот вариант неудобен в большинстве случаев, поскольку:

 - поддерживается только для очень ограниченного набора языков программирования (C, C++, Python);
 - требует того, чтобы клиентское приложение, обращающееся к программной модели sc-памяти, фактически составляло с ней единое целое, таким образом исключается возможность построения распределенного коллектива *ostis*-систем;

- как следствие предыдущего пункта, исключается возможность параллельной работы с sc-памятью нескольких клиентских приложений.

Для того, чтобы обеспечить возможность удаленного доступа к sc-памяти не учитывая при этом языки программирования, с помощью которых реализовано конкретное клиентское приложение, было принято решение о реализации возможности доступа к sc-памяти с использованием универсальных протоколов, не зависящих от средств реализации того или иного компонента или системы. В качестве таких протоколов были разработаны бинарный протокол SCTP и текстовый протокол на базе JSON.]

SCTP

:= [Semantic Code Transfer Protocol]

⇔ *аналогия**:

HTTP

⇒ *пояснение**:

[SCTP представляет собой *бинарный протокол*, позволяющий осуществлять операции чтения (поиска) и редактирования конструкций, хранящихся в sc-памяти, а также отслеживать события, происходящие в sc-памяти.

Взаимодействие между клиентом и сервером на протоколе SCTP осуществляется путем обмена *sctp-командами*, каждая из которых представляет собой набор байт, предназначенный для машинной обработки (но не восприятия человеком).]

*следует отличать**

- ⇒ {
 - *SCTP*
 - := [Semantic Code Transfer Protocol]
 - *Stream Control Transmission Protocol*
 - := [Протокол передачи с управлением потоком]
 - ⇒ *примечание**:
[Протокол транспортного уровня в компьютерных сетях, разработанный в 2000 году.]

sctp-команда

⇒ *обобщенная декомпозиция**:

- {
 - *заголовок sctp-команды*
 - := [часть sctp-команды, в которой указан её тип и некоторая дополнительная информация о ней]
 - *аргументы sctp-команды*
 - := [часть sctp-команды, которая содержит её аргументы и размер которой может быть разным в зависимости от типа команды.]

⇒ *включение*: пример**:

- *sctp-команда удаления sc-элемента с указанным sc-адресом*
- *sctp-команда создания нового sc-узла указанного типа*
- *sctp-команда получения начального и конечного элемента sc-дуги*

⇒ *примечание**:

[Выполнение каждой sctp-команды предполагает наличие sctp-результата, однозначно соответствующего данной команде.]

SCTP

⇒ *программная документация**:

[<http://ostis-dev.github.io/sc-machine/net/sctp/>]

⇒ *недостаток**:

- [Команды протокола SCTP являются низкоуровневыми (ориентированы на работу с единичными sc-элементами или простейшими sc-конструкциями из 3 или 5 элементов). Это приводит к тому, что выполнение даже несложного преобразования в базе знаний или ассоциативный поиск по набору взаимосвязанных конструкций выражаются в виде достаточно большого набора sctp-команд. С учетом того, что для каждой команды существует sctp-результат, также пересылаемый по сети, это излишне нагружает сеть и сильно ухудшает производительность системы в целом. Кроме того, производительность системы начинает сильно зависеть от пропускной способности сети.]
- [Протокол SCTP не предназначен для восприятия человеком]

⇒ *достоинство**:

-

[Протокол SCTP является кросс-платформенным]

- [Протокол SCTP может быть достаточно просто реализован практически на любом языке программирования]

⇒ *обобщенная реализация**:

- *sctp-сервер*

⇒ *пояснение**:

[Sctp-сервер обрабатывает sctp-команды, приходящие от разных sctp-клиентов, и обеспечивает их интерпретацию в sc-памяти.]

- *sctp-клиент*

⇒ *пояснение**:

[Sctp-клиенты в общем случае могут быть реализованы на разных языках программирования и иметь разный программный интерфейс. По сути задачей sctp-клиента является преобразование высокоуровневых команд представленных в форме, удобной программисту, в одну или более низкоуровневых sctp-команд, отправка их на сервер, ожидание sctp-результата и его интерпретация.]

Реализация подсистемы взаимодействия с внешней средой с использованием протокола SCTP

⇒ *компонент программной системы**:

- *Реализация sctp-сервера*
- *Реализация sctp-клиента*

⇒ *примечание**:

[Реализация подсистемы взаимодействия с внешней средой с использованием протокола SCTP включает в себя Реализацию sctp-клиента на языке C++, в то же время есть другие реализации sctp-клиентов в рамках того же программного варианта реализации платформы, например, в рамках Реализации интерпретатора sc-моделей пользовательских интерфейсов.]

Реализация подсистемы взаимодействия с внешней средой с использованием протоколов на основе формата JSON

⇒ *пояснение**:

[В связи с большим числом недостатков протокола SCTP было принято решение о разработке другого протокола на основе какого-либо общепринятого текстового транспортного формата. В качестве такого формата был выбран формат JSON.]

← *реализация**:

Протокол взаимодействия с sc-памятью на основе JSON

⇒ *примечание**:

[Данный протокол пока не имеет собственного названия]

⇒ *программная документация**:

[<http://ostis-dev.github.io/sc-machine/http/websocket/>]

⇒ *пояснение**:

[В рамках *Протокола взаимодействия с sc-памятью на основе JSON* каждая команда представляет собой json-объект, в котором указываются идентификатор команда, тип команды и ее аргументы. В свою очередь ответ на команду также представляет собой json-объект, в котором указываются идентификатор команды, ее статус (выполнена успешно/безуспешно) и результаты. Структура аргументов и результатов команды определяется типом команды.]

⇒ *достоинство**:

- [JSON является общепринятым открытым форматом, для работы с которым существует большое количество библиотек для популярных языков программирования. Это, в свою очередь, упрощает реализацию клиента и сервера для протокола, построенного на базе JSON.]
- [Реализация протокола на базе JSON не накладывает принципиальных ограничений на объем (длину) каждой команды, в отличие от бинарного протокола. Таким образом, появляется возможность использования неатомарных команд, позволяющих, например, за один акт пересылки такой команды по сети создать сразу несколько sc-элементов. Важными примерами таких команд являются *Команда генерации по произвольному образцу* и *Команда поиска по произвольному образцу*.]

⇒ *примечание**:

[Можно сказать, что протокол на базе JSON является следующим шагом на пути к созданию мощного и универсального языка запросов, аналогичного языку SQL для реляционных баз данных и предназначенному для работы с sc-памятью. Следующий шагом станет реализация такого протокола на основе одного из стандартов внешнего отображения sc-конструкций, например, *SCs-кода*, что, в

свою очередь, позволит передавать в качестве команд целые программы обработки sc-конструкций, например на языке SCP.]

Реализация вспомогательных инструментальных средств для работы с sc-памятью

⇒ компонент программной системы*:

Реализация сборщика базы знаний из исходных текстов, записанных в SCs-коде

:= [sc-builder]

⇒ используемый язык*:

SCs-код

⇒ пояснение*:

[Сборщик базы знаний из исходных текстов позволяет осуществить сборку базы знаний из набора исходных текстов, записанных в SCs-коде с ограничениями (см. *Раздел **про исходные тексты***) в бинарный формат, воспринимаемый *Программной моделью sc-памяти*. При этом возможна как сборка "с нуля" (с уничтожением ранее созданного слепка памяти), так и аддитивная сборка, когда информация, содержащаяся в заданном множестве файлов, добавляется к уже имеющемуся слепку состояния памяти.

В текущей реализации сборщик осуществляет "склеивание" ("слияние") sc-элементов, имеющих на уровне исходных текстов одинаковые *системные sc-идентификаторы*.]

Реализация интерпретатора sc-моделей пользовательских интерфейсов

:= [sc-web]

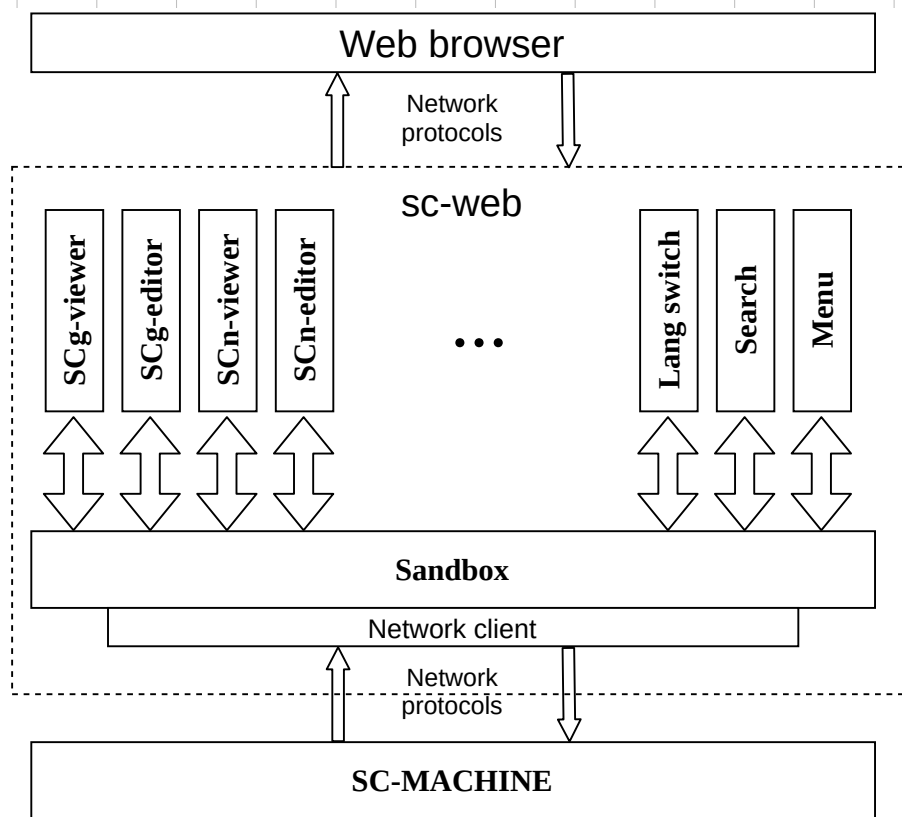
⇒ пояснение*:

[Наряду с реализацией *Программной модели sc-памяти* важной частью *Программного варианта реализации платформы интерпретации sc-моделей компьютерных систем* является *Реализация интерпретатора sc-моделей пользовательских интерфейсов*, которая предоставляет базовые средства просмотра и редактирования базы знаний пользователем, средства для навигации по базе знаний (задания вопросов к базе знаний) и может дополняться новыми компонентами в зависимости от задач, решаемых каждой конкретной *ostis-системой*.]

⇒ используемый язык программирования*:

- JavaScript
- TypeScript
- Python

⇒ иллюстрация*:



⇒ *пояснение**:

[На данной иллюстрации показан планируемый вариант архитектуры *Реализация интерпретатора sc-моделей пользовательских интерфейсов*, важным принципом которой является простота и однотипность подключения любых компонентов пользовательского интерфейса (редакторов, визуализаторов, переключателей, команд меню и т.д.). Для этого реализуется программная прослойка Sandbox, в рамках которой реализуются низкоуровневые операции взаимодействия с серверной частью и которая обеспечивает более удобный программный интерфейс для разработчиков компонентов.]

⇒ *недостатки текущей реализации**:

- { • [Отсутствие единого унифицированного механизма клиент-серверного взаимодействия. Часть компонентов (визуализатор sc-текстов в SCn-коде, команды меню и др.) работают по протоколу HTTP, часть по протоколу SCTP с использованием технологии WebSocket, это приводит к значительным трудностям при развитии платформы.]
- [Протокол HTTP предполагает четкое разделение активного клиента и пассивного сервера, который отвечает на запросы клиентов. Таким образом, сервер (в данном случае – sc-память) практически не имеет возможности по своей инициативе отправить сообщение клиенту, что повышает безопасность системы, но значительно снижает ее интерактивность. Кроме того, такой вариант реализации затрудняет реализацию принятого в Технологии OSTIS многоагентного подхода, в частности, затрудняет реализацию sc-агентов на стороне клиента. Указанные проблемы могут быть решены путем постоянного мониторинга определенных событий со стороны клиента, однако такой вариант неэффективен. Кроме того, часть интерфейса фактически работает напрямую с sc-памятью с использованием технологии WebSocket, а часть – через прослойку на базе библиотеки tornado для языка программирования Python, что приводит к дополнительным зависимостям от сторонних библиотек.]
- [Часть компонентов (например, поле поиска по идентификатору) реализована сторонними средствами и практически никак не связана с sc-памятью. Это затрудняет развитие платформы.]
- [Текущая *Реализация интерпретатора sc-моделей пользовательских интерфейсов* ориентирована только на ведение диалога с пользователем (в стиле вопрос пользователя – ответ системы). Не поддерживаются такие очевидно необходимые ситуации, как выполнение команды, не предполагающей ответа; возникновение ошибки или отсутствие ответа; необходимость задания вопроса системой пользователю и т.д.]
- [Ограничена возможность взаимодействия пользователя с системой без использования специальных элементов управления. Например, можно задать вопрос системе, нарисовав его в SCg-коде, но ответ

пользователь не увидит, хотя в памяти он будет сформирован соответствующим агентом.; Большая часть технологий, использованных при реализации платформы, к настоящему моменту устарела, что затрудняет развитие платформы.]

- [Идея платформенной независимости пользовательского интерфейса (построения sc-модели пользовательского интерфейса) реализована не в полной мере. Полностью описать sc-модель пользовательского интерфейса (включая точное размещение, размеры, дизайн компонентов, их поведение и др.) в настоящее время скорее всего окажется затруднительно из-за ограничений производительности, однако вполне возможно реализовать возможность задания вопросов ко всем компонентам интерфейса, изменить их расположение и т.д., однако эти возможности нельзя реализовать в текущей версии реализации платформы.]
- [Интерфейсная часть работает медленно из-за недостатков протокола SCTP и некоторых недостатков реализации серверной части на языке Python.]
- [Не реализован механизм наследования при добавлении новых внешних языков. Например, добавление нового языка даже очень близкого к SCg-коду требует физического копирования кода компонента и внесение соответствующих изменений, при этом получаются два никак не связанных между собой компонента, которые начинают развиваться независимо друг от друга.]
- [Слабый уровень задокументированности текущей *Реализации интерпретатора sc-моделей пользовательских интерфейсов*.]

}

⇒ требования к будущей реализации*:

{

- [Унифицировать принципы взаимодействия всех компонентов интерфейса с *Программной моделью sc-памяти*, независимо от того, к какому типу относится компонент. Например, список команд меню должен формироваться через тот же механизм, что и ответ на запрос пользователя, и команда редактирования, сформированная пользователем, и команда добавления нового фрагмента в базу знаний и т.д.]
- [Унифицировать принципы взаимодействия пользователей с системой независимо от способа взаимодействия и внешнего языка. Например, должна быть возможность задания вопросов и выполнения других команд прямо через SCg/SCn интерфейс. При этом необходимо учитывать принципы редактирования базы знаний, чтобы пользователя не мог под видом задания вопроса внести новую информацию в согласованную часть базы знаний.]
- [Унифицировать принципы обработки событий, происходящих при взаимодействии пользователя с компонентами интерфейса – поведение кнопок и других интерактивных компонентов должно задаваться не статически сторонними средствами, а реализовываться в виде агента, который, тем не менее, может быть реализован произвольным образом (не обязательно на платформенно-независимом уровне). Любое действие, совершаемое пользователем, на логическом уровне должно трактоваться и обрабатываться как инициирование агента.]
- [Обеспечить возможность выполнять команды (в частности, задавать вопросы) с произвольным количеством аргументов, в том числе – без аргументов.]
- [Обеспечить возможность отображения ответа на вопрос по частям, если ответ очень большой и для отображения требуется много времени.]
- [Каждый отображаемый компонент интерфейса должен трактоваться как изображение некоторого sc-узла, описанного в базе знаний. Таким образом, пользователь должен иметь возможность задания произвольных вопросов к любым компонентам интерфейса.]
- [Максимально упростить и задокументировать механизм добавления новых компонентов.]
- [Обеспечить возможность добавления новых компонентов на основе имеющихся без создания независимых копий. Например, должна быть возможность создать компонент для языка, расширяющего язык SCg новыми примитивами, переопределять принципы размещения sc-текстов и т.д.]
- [Свести к минимуму зависимость от сторонних библиотек.]
- [Свести к минимуму использование протокола HTTP (начальная загрузка общей структуры интерфейса), обеспечить возможность равноправного двустороннего взаимодействия серверной и клиентской части.]
- [Полностью отказаться от протокола SCTP, перейти на протокол на базе JSON, задокументировать его.]

}

⇒ примечание*:

[Очевидно, что реализация большинства из приведенных требований связана не только с собственно вариантом реализации платформы, но и требует развития теории логико-семантических моделей пользовательских интерфейсов и уточнения в рамках нее общих принципов организации пользова-

- ⇒ тельских интерфейсов ostis-систем. Однако, принципиальная возможность реализации таких моделей должна быть учтена в рамках реализации платформы.]
- ⇒ компонент программной системы*:
- *Панель меню команд пользовательского интерфейса*
⇒ *пояснение**:
[*Панель меню команд пользовательского интерфейса* содержит изображения классов команд (как атомарных, так и неатомарных), имеющихся на данный момент в базе знаний и входящих в декомпозицию *Главного меню пользовательского интерфейса* (имеется в виду полная декомпозиция, которая в общем случае может включать несколько уровней неатомарных классов команд).
Взаимодействие с изображением неатомарного класса команд инициирует команду изображения классов команд, входящих в декомпозицию данного неатомарного класса команд.
Взаимодействие с изображением атомарного класса команд инициирует генерацию команды данного класса с ранее выбранными аргументами на основе соответствующей *обобщенной формулировки класса команд* (шаблона класса команд).]
 - *Компонент переключения языка идентификации отображаемых sc-элементов*
⇒ *пояснение**:
[*Компонент переключения языка идентификации отображаемых sc-элементов* является изображением множества имеющихся в системе естественных языков. Взаимодействие пользователя с данным компонентом переключает пользовательский интерфейс в режим общения с конкретным пользователем с использованием *основных sc-идентификаторов*, принадлежащих данному *естественному языку*. Это значит, что при изображении sc-идентификаторов sc-элементов на каком-либо языке, например, SCg-коде или SCn-коде будут использоваться *основные sc-идентификаторы*, принадлежащие данному *естественному языку*. Это касается как sc-элементов, отображаемых в рамках *Панели визуализации и редактирования знаний*, так и любых других sc-элементов, например, классов команд и даже самих *естественных языков*, изображаемых в рамках самого *Компонента переключения языка идентификации отображаемых sc-элементов*.]
 - *Компонент переключения внешнего языка визуализации знаний*
⇒ *пояснение**:
[*Компонент переключения внешнего языка визуализации знаний* служит для переключения языка визуализации знаний в текущем окне, отображаемом на *Панели визуализации и редактирования знаний*. В текущей реализации в качестве таких языков по умолчанию поддерживаются SCg-код и SCn-код, а также любые другие языки, входящие во множество *внешних языков визуализации SC-кода*.]
 - *Поле поиска sc-элементов по идентификатору*
⇒ *пояснение**:
[*Поле поиска sc-элементов по идентификатору* позволяет осуществлять поиск sc-идентификаторов, содержащих подстроку, введенную в данное поле (с учетом регистра). В результате поиска отображается список sc-идентификаторов, содержащих указанную подстроку, при взаимодействии с которыми осуществляется автоматическое задание вопроса “Что это такое?”, аргументом которого является либо для сам sc-элемент, имеющий данный sc-идентификатор (в случае, если указанный sc-идентификатор является основным или системным, и, таким образом, указанный sc-элемент может быть определен однозначно), либо для самого внутреннего файла ostis-системы, являющегося sc-идентификатором (в случае, если данный sc-идентификатор является неосновным).]
 - *Панель отображения диалога пользователя с ostis-системой*
⇒ *пояснение**:
[*Панель отображения диалога пользователя с ostis-системой* отображает упорядоченный по времени список sc-элементов, являющихся знаками действий, которые инициировал пользователь в рамках диалога с ostis-системой путем взаимодействия с изображениями соответствующих классов команд (то есть, если действие было инициировано другим способом, например, путем его явного инициирования через создание дуги принадлежности множеству *иницированных действий* в sc.g-редакторе, то на данной панели оно отображено не будет). При взаимодействии пользователя с любым из изображенных знаков действий на *Панели визуализации и редактирования знаний* отображается окно, содержащее результат выполнения данного действия на том языке визуализации, на котором он был отображен, когда пользователь просматривал его в последний (предыдущий) раз. Таким образом, в текущей реализации данная панель может работать только в том случае, если инициированное пользователем действие предполагает явно представленный в памяти результат данного действия. В свою очередь, из этого следует, что в настоящее время данная панель, как и в

целом *Реализация интерпретатора sc-моделей пользовательских интерфейсов*, позволяет работать с системой только в режиме диалога "вопрос-ответ".]

- *Панель визуализации и редактирования знаний*

- ⇒ *пояснение**:

[*Панель визуализации и редактирования знаний* отображает окна, содержащие sc-текст, представленный на некотором языке из множества *внешних языков визуализации SC-кода* и, как правило, являющийся результатом некоторого действия, инициированного пользователем. Если соответствующий визуализатор поддерживает возможность редактирования текстов соответствующего естественного языка, то он одновременно является также и редактором.]

- ⇒ *компонент программной системы**:

- *Визуализатор sc.n-текстов*
- *Визуализатор и редактор sc.g-текстов*

- ⇒ *примечание**:

[При необходимости пользовательский интерфейс каждой конкретной *ostis-системы* может быть дополнен визуализаторами и редакторами различных внешних языков, которые в текущей версии *Реализации интерпретатора sc-моделей пользовательских интерфейсов* будут также располагаться на *Панели визуализации и редактирования знаний*.]

Реализация scr-интерпретатора

- ← *программная реализация**:

Абстрактная scr-машина

- ⇒ *примечание**:

[Важнейшей особенностью Языка SCP является тот факт, что его программы записываются таким же образом, что и обрабатываемые ими знания, то есть в SC-коде. Это, с одной стороны, дает возможность сделать *ostis-системы* платформенно-независимыми (четко разделить *sc-модель компьютерной системы* и платформу интерпретации таких моделей), а с другой стороны требует наличия в рамках платформы *Реализации scr-интерпретатора*, то есть интерпретатора программ Языка SCP.]

- ⇒ *используемый язык программирования**:

- C++

- ⇒ *компонент программной системы**:

- *Реализация Абстрактного sc-агента создания scr-процессов*
- *Реализация Абстрактного sc-агента интерпретации scr-операторов*

- ⇒ *компонент программной системы**:

- *Реализация Абстрактного sc-агента интерпретации scr-операторов генерации конструкций*
- *Реализация Абстрактного sc-агента интерпретации scr-операторов ассоциативного поиска конструкций*
- *Реализация Абстрактного sc-агента интерпретации scr-операторов удаления конструкций*
- *Реализация Абстрактного sc-агента интерпретации scr-операторов проверки условий*
- *Реализация Абстрактного sc-агента интерпретации scr-операторов управления значениями операндов*
- *Реализация Абстрактного sc-агента интерпретации scr-операторов управления scr-процессами*
- *Реализация Абстрактного sc-агента интерпретации scr-операторов управления событиями*
- *Реализация Абстрактного sc-агента интерпретации scr-операторов обработки содержимых числовых файлов*
- *Реализация Абстрактного sc-агента интерпретации scr-операторов обработки содержимых строковых файлов*

- *Реализация Абстрактного sc-агента синхронизации процесса интерпретации scr-программ*

- *Реализация Абстрактного sc-агента уничтожения scr-процессов*

- *Реализация Абстрактного sc-агента синхронизации событий в sc-памяти и ее реализации*

- ⇒ *примечание**:

[Текущая *Реализация scr-интерпретатора* не включает в себя специализированных средств для работы с блокировками, поскольку механизм блокировок элементов *sc-памяти* реализован на более низком уровне в рамках *Реализация sc-хранилища и механизма доступа к нему*]

} /* Завершили Раздел "Предметная область и онтология программных вариантов реализации базового интерпретатора логико-семантических моделей *ostis-систем* на современных компьютерах" */

```

/* Раздел *****/
Предметная область и онтология семантических
ассоциативных компьютеров для ostis-систем
⊃=
{
Предметная область семантических ассоциативных компьютеров для ostis-систем
∈ предметная область
∋ максимальный класс объектов исследования':
  семантический ассоциативный компьютер
∋ класс объектов исследования':
  • информационно-логическая задача
  • машина, ориентированная на решение информационно-логических задачи

семантический ассоциативный компьютер
:= [аппаратно реализованный интерпретатор семантических моделей (sc-моделей) компьютерных систем]
:= [семантический ассоциативный компьютер, управляемый знаниями]
:= [компьютер с нелинейной структурно перестраиваемой (графодинамической) ассоциативной памятью,
  переработка информации в которой сводится не к изменению состояния элементов памяти, а к изменению
  конфигурации связей между ними]
:= [sc-компьютер]
:= [scp-компьютер]
:= [универсальный компьютер нового поколения, специально предназначенный для реализации семантически
  совместимых гибридных интеллектуальных компьютерных систем]
:= [универсальный компьютер нового поколения, ориентированный на аппаратную интерпретацию логико-
  семантических моделей интеллектуальных компьютерных систем]
:= [универсальный компьютер нового поколения, ориентированный на аппаратную интерпретацию ostis-систем]
:= [ostis-компьютер]
:= [компьютер для реализации ostis-систем]
:= [компьютер, управляемый знаниями, представленными в SC-коде]
:= [компьютер, ориентированный на обработку текстов SC-кода]
⇒ принципы, лежащие в основе*:
{
• [нелинейная память – каждый элементарный фрагмент хранимого в памяти текста может быть
  инцидентен неограниченному числу других элементарных фрагментов этого текста]
• [структурно перестраиваемая (реконфигурируемая) память – процесс отработки хранимой в памяти
  информации сводится не только к изменению состояния элементов, но и к реконфигурации связей
  между ними]
• [в качестве внутреннего способа кодирования знаний, хранимых в памяти семантического ассоциатив-
  ного компьютера, используется универсальный (!) способ нелинейного (графоподобного) смыслового
  представления знаний, названный нами SC-кодом (семантическим, смысловым компьютерным
  кодом)]
• [обработка информации осуществляется коллективом агентов, работающих над общей памятью.
  Каждый из них реагирует на соответствующую ему ситуацию или событие в памяти (компьютер,
  управляемый хранимыми знаниями)]
• [есть программно реализуемые агенты, поведение которых описывается хранимыми в памяти агентно-
  ориентированными программами, которые интерпретируются соответствующими коллективами
  агентов; есть базовые агенты, которые не могут быть реализованы программно (в частности, это агенты
  интерпретации агентных программ, базовые рецепторные агенты-датчики, базовые эффекторные
  агенты)]
• [все агенты работают над общей памятью одновременно. Более того, если для какого-либо агента
  в некоторый момент времени в различных частях памяти возникает сразу несколько условий его
  применения, разные акты указанного агента в разных частях памяти могут выполняться одновременно
  (акт агента – это неделимый, целостный процесс деятельности агента)]
• [для того, чтобы акты агентов, параллельно выполняемые в общей памяти не "мешали" друг другу,
  для каждого акта в памяти фиксируется и постоянно актуализируется его текущее состояние. То есть
  каждый акт сообщает всем остальным о своих намерениях и пожеланиях, которым остальные агенты

```

не должны препятствовать (например, это различного рода блокировки используемых элементов семантической памяти)]

- [кроме того, агенты (точнее, выполняемые ими акты) должны соблюдать "этику", стараясь не в ущерб себе создавать максимально благоприятные условия для других агентов (актов), например, не жадничать, быстрее возвращать, не захватывать (не блокировать) лишние элементы памяти, как можно скорее освобождать (деблокировать) заблокированные элементы памяти]
- [процессор и память семантического ассоциативного компьютера глубоко интегрированы и составляют единую процессоро-память. Процессор семантического ассоциативного компьютера равномерно "распределен" по его памяти так, что процессорные элементы одновременно являются и элементами памяти компьютера. Обработка информации в семантическом ассоциативном компьютере сводится к реконфигурации каналов связи между процессорными элементами, следовательно память такого компьютера есть не что иное, как коммутатор (!) указанных каналов связи. Таким образом, текущее состояние конфигурации этих каналов связи и есть текущее состояние обрабатываемой информации]

}

⇒ *основная цель**:

[Повысить производительность компьютерных систем, основанных на знаниях, представленных в виде семантических сетей]

⇒ *проблема создания**:

[Традиционная архитектура компьютерных систем не позволяет эффективно использовать сложно-структурированные знания в реальном масштабе времени.

Решение проблемы функционирования компьютерных систем обработки знаний в реальном масштабе времени трудно достижимо без использования принципов параллельной обработки и ассоциативного доступа к структурам данных. Разработка подобных систем чрезвычайно трудоемка, поэтому для обеспечения ее экономической целесообразности жизненный цикл функционирования подобных систем должен быть достаточно продолжительным (от десятка лет и более). Сменяемость аппаратных и программных средств вычислительной техники в настоящее время достигает от нескольких лет до нескольких месяцев. Это обуславливает необходимость наличия у *интеллектуальных компьютерных систем* таких качеств, как:

- открытость (в плане модифицируемости и добавления новых методов представления и переработки знаний);
- интегрируемость (в смысле возможности функционирования в составе комплексов различных вычислительных и исполнительных средств).

Анализ современных систем обработки знаний показывает, что ни одна из архитектур, на базе которых они реализованы, не обладает в совокупности всеми указанными выше свойствами.]

⇒ *пояснение**:

[Особенности логической организации вычислительных систем, которые существенно затрудняют создание машин переработки знаний на основе традиционных ЭВМ:

- последовательная обработка, ограничивающая эффективность ЭВМ физическими возможностями элементной базы;
- низкий уровень доступа к памяти, т.е. сложность и громоздкость выполнения процедуры ассоциативного поиска нужного фрагмента знаний.
- линейная организация памяти и чрезвычайно простой вид конструктивных объектов, непосредственно хранимых в памяти. Это приводит к тому, что в интеллектуальных системах, построенных на базе современных ЭВМ, манипулирование знаниями осуществляется с большим трудом. Во-первых, приходится оперировать не самими структурами, а их громоздкими линейными представлениями (списками, матрицами смежности, матрицами инцидентности); во-вторых, линеаризация сложных структур разрушает локальность их преобразований;
- представление информации в памяти современных ЭВМ имеет уровень весьма далекий от семантического, что делает переработку знаний довольно громоздкой, требующей учета большого количества деталей, касающейся не смысла перерабатываемой информации, а способа ее представления в памяти;
- в современных ЭВМ имеет место весьма низкий уровень аппаратно реализуемых операций над нечисловыми данными и полностью отсутствует аппаратная поддержка логических операций над фрагментами знаний, имеющих сложную структуру, что делает манипулирование такими фрагментами весьма сложным.

Основным проявлением перечисленных недостатков традиционных ЭВМ является гипертрофированное развитие программного обеспечения, сложность его создания и невысокая надежность.

⇒ Трудности, связанные с использованием универсальных многопроцессорных ЭВМ, послужили толчком к развитию теоретических исследований в области параллельного программирования.]
 ⇒ *формальная основа решения проблемы**:

SC-код

:= [Предлагаемое в рамках *Технологии OSTIS* уточнение понятия *семантической сети*]

:= [Semantic Computer Code]

⇒ *примечание**:

[*SC-код* обладает фиксированными синтаксисом и денотационной семантикой, но нефиксированной операционной семантикой. Следовательно, позволяет на его основе создать семейство языков представления знаний различного вида.

SC-код изначально разрабатывался как язык кодирования информации в памяти *семантических ассоциативных компьютеров*, таким образом в нем изначально заложены такие принципы, как универсальность (возможность представить знания любого рода) и унификация (единообразие) представления, а также минимизацию *Алфавита SC-кода*, которая, в свою очередь, позволяет облегчить создание аппаратной платформы, позволяющей хранить и обрабатывать тексты *SC-кода*.]

информационно-логическая задача

:= [информационно-логическая или комбинаторная задачи по переработке сложноструктурированных баз данных]

:= [класс задач, предполагающих переработку нечисловой сложноструктурированной информации и допускающих при этом отсутствие точного алгоритма их решения]

⇒ *примечание**:

[Понятие *информационно-логических задач* фактически совпадает по смыслу с широко используемым в последнее время понятием задач искусственного интеллекта (*интеллектуальных задач*), что позволяет использовать оба эти термина.]

⇒ *примечание**:

[Разработка средств решения задач того или иного класса в настоящее время обычно осуществляется путем создания *языка программирования высокого уровня*, ориентированного на этот класс задач, и путем реализации такого языка на современных компьютерах, т.е. путем создания транслятора. Поскольку логика решения задач искусственного интеллекта плохо согласуется с современными языками программирования, более целесообразной является разработка принципиально новых языков, отражающих на уровне их элементарных операций основные логические механизмы решения задач рассматриваемого класса. Такие языки программирования обычно называют языками сверх-высокого уровня или непроцедурными языками. Реализация языков сверх-высокого уровня на современных компьютерах представляется весьма сложной в силу большого разрыва между языками этого класса и внутренними языками современных компьютеров, для преодоления которого создание эффективного транслятора оказывается практически невозможным.]

⇒ *пояснение**:

[Таким образом, состояние проблемы автоматизации решения задач искусственного интеллекта (информационно-логических задач) в настоящее время можно охарактеризовать тем, что эта проблема входит в противоречие с принципами логической организации современных компьютеров и, в первую очередь, с используемыми в современных компьютерах внутренними языками.

Причины плохой приспособленности современных компьютеров к решению информационно-логических задач:

- в современных компьютерах при работе со сложноструктурированными базами данных время, затрачиваемое на информационный поиск, на 2-3 порядка превышает время, затрачиваемое на собственно переработку;
- в современных компьютерах имеет место весьма низкий уровень аппаратно реализуемых операций над нечисловыми данными;
- представление информации в памяти современных компьютеров имеет уровень весьма далекий от семантического, что делает переработку информации довольно громоздкой, требующей учета большого количества деталей, касающихся не смысла перерабатываемой информации, а способа ее представления в памяти;
- в современных компьютерах громоздко реализуются даже простейшие процедуры логического вывода.

Перечисленные причины, по существу, не устраняются также и в развиваемых в настоящее время подходах к построению нетрадиционных высокопроизводительных компьютеров, ибо, в основном, все эти подходы (даже если они достаточно далеко отходят от предложенных фон Нейманом

принципов организации вычислительных машин) неявно сохраняют точку зрения на компьютер как на большой арифмометр и тем самым сохраняют ее ориентацию на задачи числового характера. Очевидно, что эффективность этих машин будет прежде всего определяться степенью близости их внутреннего языка к языкам непроцедурного типа (к языкам сверх-высокого уровня). Учитывая указанное назначение таких машин, их естественно назвать не вычислительными машинами, а математическими машинами или даже "мыслящими" машинами.]

машина, ориентированная на решение информационно-логических задач

⇒ *класс решаемых задач**:

информационно-логическая задача

:= [машина, реализующая стремление так организовать процесс переработки информации, чтобы он был наиболее близок к семантическому (содержательному) уровню]

⇒ *задачи разработки**:

- { • [разработка семантического способа представления перерабатываемой информации в памяти машины]
- [разработка такого внутреннего языка, запись программ на котором была бы максимально близка к тому, что называют записью алгоритма на содержательном уровне]
- [разработка и исследование способов семантического представления информации различного вида]
- [разработка и исследование принципов организации развитой ассоциативной памяти для непосредственного хранения семантического представления информации]
- [разработка и исследование языка программирования высокого уровня, который (1) ориентирован на решение информационно-логических задач; (2) обеспечивает непосредственную реализацию простых процедур логического вывода. (3) согласован с выбранным способом семантического представления перерабатываемой информации, (4) приспособлен к использованию в качестве внутреннего языка параллельной однородной структуры, имеющей распределенную ассоциативную память для хранения сложноструктурированных данных]
- [разработка и исследование принципов построения и, принципов параллельного взаимодействия функциональных средств, обеспечивающих непосредственную переработку семантического представления информации в распределенной ассоциативной памяти и реализующих управление потоком словноструктурированных данных]
- [экспериментальная проверка полученных результатов]

⇒ *примечание**:

[Исследования по системам искусственного интеллекта убедительно показали, что способ представления знаний в их памяти, точнее степень его близости к семантическому, является фактором, во многом определяющим эффективность таких систем.]

⇒ *принципы построения**:

- { • [такие машины целесообразно строить как машины, манипулирующие графовыми структурами непосредственно на физическом уровне]

⇒ *пояснение**:

[Предполагается создание структурно-перестраиваемой ("графовой") памяти. Такая память состоит из ячеек, однозначно соответствующих вершинам хранимого в памяти графа, но, в отличие от обычной памяти, эти ячейки связываются не фиксированными условными связями, задающими фиксированную последовательность (порядок) ячеек в памяти, а реально (физически) проводимыми связями произвольной конфигурации. Эти связи соответствуют дугам, ребрам, гиперребрам записанного в памяти графа. Очевидно, что в ходе переработки информации в структурно-перестраиваемой памяти меняются не только и даже не столько состояния ячеек памяти, как это имеет место в обычной памяти, сколько конфигурация связей между этими ячейками. Т.е. в структурно-перестраиваемой памяти в ходе переработки информации не только перераспределяются метки на вершинах записанного в памяти графа, но и меняется структура самого этого графа.]

- [в качестве внутреннего языка таких машин целесообразно использовать язык типа PROLOG.]

⇒ *пояснение**:

[Разработка языка типа PROLOG, предназначенного к использованию в качестве внутреннего языка программирования для машин со структурно-перестраиваемой памятью, требует решения нетривиальной задачи согласования графового способа представления данных в структурно-перестраиваемой памяти и способа записи в этой же памяти самих программ, описывающих переработку этих данных. Переход на "графовый" способ кодирования программ и данных в структурно-перестраиваемой памяти обеспечивает компактность их представления и существенно упрощает аппаратную реализацию операций над сложными структурами.]

Говоря об аппаратной интерпретации языка типа PROLOG, необходимо подчеркнуть следующее. На уровне любого языка типа PROLOG, т.е. на уровне абстрактной PROLOG – машины, естественным образом реализуется эффективное распараллеливание процесса переработки сложных структур, организованное по принципу управления потоком запросов или управления потоком перерабатываемых сложноструктурированных данных. Управление потоком сложноструктурированных данных при этом основывается на использовании развитой формы ассоциативного доступа, а именно, доступа к произвольному фрагменту перерабатываемых данных (фрагменту, имеющему произвольный размер и произвольную структуру). Из вышесказанного следует, что создание машины, аппаратно интерпретирующей язык типа PROLOG, есть не что иное, как создание параллельной машины, управляемой потоком сложноструктурированных данных и имеющей развитую ассоциативную память.]

- [принцип организации переработки информации непосредственно в памяти]

⇒ *пояснение**:

[Этот принцип обеспечивает значительное ускорение переработки информации благодаря исключению этапов передачи информации из памяти в процессор и обратно, но оплачивается ценой большой избыточности функциональных (процессорных) средств, равномерно распределяемых по памяти. При распределении функциональных средств по структурно-перестраиваемой памяти каждая ячейка дополняется функциональным (процессорным) элементом, а перестраиваемые связи между ячейками становятся коммутируемыми каналами связи между функциональными элементами. Каждый функциональный элемент при этом имеет свою специальную внутреннюю регистровую память, отражающую важные для данного функционального элемента аспекты текущего состояния процесса выполнения элементарных операций внутреннего языка.]

}

⇒ *формальная основа и направления исследований**:

- [сочетание теории множеств, теории алгебраических систем, теории графов, математической логики, теории вычислений, методов машинного моделирования]
- [теория наиболее общего вида структур данных (квази-графов), используемых в задачах искусственного интеллекта и являющихся обобщением классических алгебраических моделей]
- [ориентированный на аппаратную интерпретацию способ кодирования сложноструктурированных данных (квази-графы) в виде однородных семантических сетей специального вида]
- [новый логический язык, являющийся модификацией классического и обеспечивающий описание квази-графов]
- [графовые варианты логического языка описания квази-графов и, в частности, ориентированный на аппаратную интерпретацию способ кодирования логических формул в виде однородных семантических сетей]
- [непроцедурный язык программирования типа PROLOG, отличающийся удобством работы со сложными структурами данных (квази-графами) и развитостью средств управления вычислительным процессом]
- [предлагаемый в качестве внутреннего аппаратно интерпретируемого языка программирования способ записи непроцедурных программ в виде однородных семантических сетей]
- [архитектура и принципы организации однородной ассоциативной параллельной структуры, ориентированной на переработку семантических сетей и обеспечивающей аппаратную интерпретацию непроцедурного языка программирования типа PROLOG]

}

⇒ *задачи исследований**:

[Исследовать пути построения параллельных машин, управляемых потоком сложноструктурированных данных. В качестве памяти таких машин рассмотреть структурно-перестраиваемые запоминающие среды, обеспечивающие непосредственное хранение графовых структур и манипулирование ими, а также обеспечивающие ассоциативный доступ к произвольным фрагментам перерабатываемых графовых структур (фрагментам, имеющим произвольный вид и произвольный размер). Исследовать пути и принципы аппаратной интерпретации непроцедурных языков программирования типа PROLOG. В качестве интерпретируемого (внутреннего) языка исследовать язык программирования графового типа, являющийся способом записи программ в виде однородных семантических сетей.]

⇒ *достоинства**:

[Разработка машин, ориентированных на решение информационно-логических задач позволит:

- существенно расширить класс аппаратно интерпретируемых (непосредственно перерабатываемых) структур данных;

- обеспечить высокую скорость переработки сложноструктурированных данных, благодаря (1) "укрупнению" аппаратно реализуемых операций преобразования структур данных, (2) глубокому распараллеливанию процесса переработки сложных структур как на программном, так и на микропрограммном уровне, (3) организации переработки информации непосредственно в памяти;
- существенно расширить логические возможности компьютеров благодаря использованию логического языка в качестве основы внутреннего языка программирования;
- обеспечить достаточно высокую технологичность компьютеров рассматриваемого класса благодаря их организации как однородных структур.

1

}

/ Завершили Раздел "Предметная область и онтология семантических ассоциативных компьютеров для osts-систем" */*

/ Раздел *****/*

Предметная область и онтология Экосистемы OSTIS

⊃=

{

⇒

дочерний раздел:*

- *Предметная область и онтология методов и средств реализации целенаправленного и персонифицированного процесса обучения пользователей для каждой ostis-системы, входящей в состав Экосистемы OSTIS*
- *Предметная область и онтология ostis-систем, являющихся персональными ассистентами пользователей в рамках Экосистемы OSTIS*
- *Предметная область и онтология семантически совместимых интеллектуальных порталов научно-технических знаний*
- *Предметная область и онтология семантически совместимых ostis-систем управления рецептурным производством*

Предметная область Экосистемы OSTIS

∈ *предметная область*

⊃ *максимальный класс объектов исследования¹:*

Экосистема OSTIS

⊃ *класс объектов исследования¹:*

- *ostis-система*
- *самостоятельная ostis-система*
- *поддержка совместимости между компьютерными системами и их пользователями в Экосистеме OSTIS*

Экосистема OSTIS

:= [Социотехническая экосистема, представляющая собой коллектив взаимодействующих семантических компьютерных систем и осуществляющая перманентную поддержку эволюции и семантической совместимости всех входящих в нее систем, на протяжении всего их жизненного цикла]

:= [Неограниченно расширяемый коллектив постоянно эволюционируемых семантических компьютерных систем, которые взаимодействуют между собой и с пользователями для корпоративного решения сложных задач и для постоянной поддержки высокого уровня совместимости и взаимопонимания во взаимодействии как между собой, так и с пользователями]

⇒ *пояснение*:*

[Поскольку *Технология OSTIS* ориентирована на разработку *семантических компьютерных систем*, обладающих высоким уровнем *обучаемости* и, в частности, высоким уровнем *семантической совместимости*, и поскольку *обучаемость* и *совместимость* есть только способность к обучению (т.е. к высоким темпам расширения и совершенствования своих знаний и навыков), а также способность к обеспечению высокого уровня взаимопонимания (согласованности), необходима некая среда, социотехническая инфраструктура, в рамках которой были бы созданы максимально комфортные условия для реализации указанных выше способностей. Такая среда названа нами *Экосистемой OSTIS*, которая представляет собой коллектив взаимодействующих (через сеть Интернет):

- самих ostis-систем;*
- пользователей указанных ostis-систем (как конечных пользователей, так и разработчиков);*
- некоторых компьютерных систем, не являющихся ostis-системами, но рассматриваемых ими в качестве дополнительных информационных ресурсов или сервисов.*

]

⇒ *основная задача*:*

[Обеспечить постоянную поддержку совместимости компьютерных систем, входящих в *Экосистему OSTIS* как на этапе их разработки, так и в ходе их эксплуатации. Проблема здесь заключается в том, что в ходе эксплуатации систем, входящих в *Экосистему OSTIS*, они могут изменяться из-за чего совместимость может нарушаться.

Задачами *Экосистемы OSTIS* являются:

- оперативное внедрение всех согласованных изменений стандарта ostis-систем (в том числе, и изменений систем используемых понятий и соответствующих им терминов);*

- перманентная поддержка высокого уровня взаимопонимания всех систем, входящих в *Экосистему OSTIS*, и всех их пользователей;
- корпоративное решение различных сложных задач, требующих координации деятельности нескольких (чаще всего, априори неизвестных) *ostis-систем*, а также, возможно, некоторых пользователей.

]

⇒ *примечание**:

[*Экосистема OSTIS* – это переход от самостоятельных (автономных, отдельных, целостных) *ostis-систем* к коллективам самостоятельных *ostis-систем*, т.е. к распределенным *ostis-системам*]

ostis-система

⇒ *разбиение**:

- { • *атомарная встроенная ostis-система*
:= [ostis-система, интегрированная в состав самостоятельной ostis-системы, но не в состав другой встроенной ostis-системы]
- *неатомарная встроенная ostis-система*
:= [ostis-система, которая интегрирована в состав самостоятельной ostis-системы, и включает в себя некоторые другие встроенные ostis-системы]
- ⊃ *интерфейс ostis-системы*
- *самостоятельная ostis-система*
:= [целостная ostis-система, которая должна самостоятельно решать соответствующее множество задач и, в частности, взаимодействовать с внешней средой (вербально – с пользователями и другими компьютерными системами, так и невербально)]
- *коллектив ostis-систем*
:= [группа общающихся ostis-систем, в состав которой могут входить не только самостоятельные ostis-системы, но и коллективы ostis-систем]
:= [распределенная ostis-система]
- *Экосистема OSTIS*
∈ *максимальный коллектив ostis-систем*
∈ *коллектив ostis-систем, не являющийся частью другого коллектива ostis-систем*

}

самостоятельная ostis-система

⇒ *пояснение**:

[Подчеркнем, что к *самостоятельным ostis-системам*, входящим в состав *Экосистемы OSTIS*, предъявляются особые требования:

- они должны обладать всеми необходимыми знаниями и навыками для обмена сообщениями и целенаправленной организации взаимодействия с другими *ostis-системами*, входящими в *Экосистему OSTIS*;
- в условиях постоянного изменения и эволюции *ostis-систем*, входящих в *Экосистему OSTIS*, каждая из них должна сама следить за состоянием своей совместимости (согласованности) со всеми остальными *ostis-системами*, т.е. должна самостоятельно поддерживать эту совместимость, согласовывая с другими ostis-системами все требующие согласования изменения, происходящие у себя и в других системах.
- каждая система, входящая в состав *Экосистемы OSTIS*, должна:
 - интенсивно, активно и целенаправленно обучаться (как с помощью учителей-разработчиков, так и самостоятельно);
 - сообщать всем другим системам о предлагаемых или окончательно утвержденных изменениях в онтологиях и, в частности, в наборе используемых понятий;
 - принимать от других *ostis-систем* предложения об изменениях в онтологиях (в том числе в наборе используемых понятий) для согласования или утверждения этих предложений;
 - реализовывать утвержденные изменения в онтологиях, хранимых в ее базе знаний;
 - способствовать поддержанию высокого уровня семантической совместимости не только с другими *ostis-системами*, входящими в *Экосистему OSTIS*, но и со своими пользователями (т.е. обучать их, информировать их об изменениях в онтологиях).

]

Экосистема OSTIS

⇒ *пояснение**:

[*Экосистема OSTIS* является формой реализации, совершенствования и применения *Технологии OSTIS* и, следовательно, является формой создания, развития, самоорганизации рынка семантически совместимых компьютерных систем и включает в себя все необходимые для этого ресурсы – информационные, технологические, кадровые, организационные, инфраструктурные.

Экосистеме OSTIS ставится в соответствие ее **объединенная база знаний**, которая представляет собой **виртуальное объединение баз знаний** всех *ostis-систем*, входящих в состав *Экосистемы OSTIS*. Качество этой *базы знаний* (полнота, непротиворечивость, чистота) является постоянной заботой всех самостоятельных *ostis-систем*, входящих в состав *Экосистемы OSTIS*. Соответственно этому каждой указанной *ostis-системе* ставится в соответствие своя *база знаний* и своя иерархическая система *sc-агентов*.

По назначению *ostis-системы*, входящие в *Экосистему OSTIS*, могут быть:

- ассистентами конкретных пользователей или конкретных пользовательских коллективов;
- типовыми встраиваемыми подсистемами *ostis-систем*;
- системами информационной и инструментальной поддержки проектирования различных компонентов и различных классов *ostis-систем*;
- системами информационной и инструментальной поддержки проектирования или производства различных классов технических и других искусственно создаваемых систем;
- порталами знаний по самым различным научным дисциплинам;
- системами автоматизации управления различными сложными объектами (производственными предприятиями, учебными заведениями, кафедрами вузов, конкретными обучаемыми);
- интеллектуальными справочными и help-системами;
- интеллектуальными обучающими системами, семантическими электронными учебными пособиями;
- интеллектуальными робототехническими системами.

]

поддержка совместимости между компьютерными системами и их пользователями в *Экосистеме OSTIS*

⇒ *пояснение**:

[Есть три аспекта поддержки совместимости и взаимопонимания в *Экосистеме OSTIS*

- поддержка совместимости между самими *ostis-системами*, входящими в *Экосистему OSTIS* в процессе их эволюции;
- поддержка совместимости между каждой *ostis-системой* и текущим состоянием *Технологии OSTIS* в процессе эволюции этой технологии;
- поддержка совместимости и взаимопонимания между *ostis-системами*, входящими в *Экосистему OSTIS*, и их пользователями при активном стимулировании со стороны *Экосистемы OSTIS* того, чтобы каждый пользователь *Экосистемы OSTIS* был одновременно не только активным ее конечным пользователем, но и активным ее разработчиком.

Таким образом, для обеспечения высокой эффективности эксплуатации и высоких темпов эволюции *Экосистемы OSTIS*, необходимо постоянно повышать уровень информационной совместимости (уровень взаимопонимания) не только между компьютерными системами, входящими в состав *Экосистемы OSTIS*, но также между этими системами и их пользователями. Одним из направлений обеспечения такой совместимости является стремление к тому, чтобы *база знаний* (картина мира) каждого пользователя стала частью (фрагментом) **Объединенной базы знаний *Экосистемы OSTIS***. Это значит, что каждый пользователь должен знать, как устроена структура каждой научно-технической дисциплины (объекты исследования, предметы исследования, определения, закономерности и т.д.), как могут быть связаны между собой различные дисциплины.

Формирование таких навыков системного построения картины Мира необходимо начинать со средней школы. Для этой цели необходимо создать комплекс совместимых интеллектуальных обучающих систем по всем дисциплинам среднего образования с четко описанными междисциплинарными связями (*Баишмаков.А.И..РазработкаКУИОС-2003кн, Таранчук.В.Б.WolframMСТРИОС-2015ст*). Благодаря этому можно предотвратить формирование у пользователей "мозаичной" картины Мира как множества слабо связанных между собой дисциплин. А это, в свою очередь, означает существенное повышение качества образования, которое абсолютно необходимо для качественной эксплуатации компьютерных систем следующего поколения – *семантических компьютерных систем*.

Пользователи и, первую очередь, разработчики *Экосистемы OSTIS* должны иметь высокий уровень:

- математической культуры (культуры формализации) при построении формальной модели среды, в которой функционирует интеллектуальная система, формальных моделей решаемых ею задач и формальных моделей различных используемых ею способов решения задач;
- системной культуры, позволяющей адекватно оценивать качество разрабатываемых систем с точки зрения общей теории систем и, в частности, оценивать общий уровень автоматизации, реализуемый с помощью этих систем. Системная культура предполагает стремление и умение избегать эклектики, стремление и умение обеспечить качественную стратифицированность, гибкость, рефлексивность, а также качественное сопровождение, высокий уровень обучаемости и комфортный пользовательский интерфейс разрабатываемых систем;
- технологической культуры, обеспечивающей совместимость разрабатываемых систем и их компонентов, а также постоянное расширение библиотеки многократно используемых компонентов создаваемых систем и предполагающей высокий уровень проектной дисциплины;
- умения работать в команде разработчиков наукоемких систем, что предполагает высокий уровень умения работать на междисциплинарных стыках, высокий уровень коммуникабельности и договороспособности, т.е. способности не столько отстаивать свою точку зрения, сколько согласовывать ее с точками зрения других разработчиков в интересах развития *Экосистемы OSTIS*;
- активности и ответственности за общий результат – высокие темпы эволюции *Экосистемы OSTIS* в целом.

Таким образом высокие темпы эволюции *Экосистемы OSTIS* обеспечиваются не только профессиональной квалификацией пользователей (знаниями о *Технологии OSTIS*, о текущем состоянии и проблемах *Экосистемы OSTIS* и навыками использования *Технологии OSTIS* и интеллектуальных систем, входящих в *Экосистему OSTIS*), но и соответствующими человеческими качествами. Очевидно, что современный уровень договороспособности, активности и ответственности не может быть основой для эволюции таких систем, как *Экосистема OSTIS*.

Поддержка совместимости *Экосистемы OSTIS* с ее пользователями осуществляется следующим образом:

- в каждую *ostis-систему* включаются встроенные *ostis-системы*, ориентированные
 - на перманентный мониторинг деятельности конечных пользователей и разработчиков этой *ostis-системы*,
 - на анализ качества и, в первую очередь, корректности этой деятельности,
 - на перманентное ненавязчивое персонафицированное обучение, направленное на повышение качества деятельности пользователей, т.е. на повышение их квалификации;
- в состав *Экосистемы OSTIS* включаются *ostis-системы*, специально предназначенные для обучения пользователей *Экосистемы OSTIS* базовым общепризнанным знаниям и навыкам решения соответствующих классов задач. Сюда входят и знания, соответствующие уровню среднего образования, и знания соответствующие базовым дисциплинам высшего образования в области информатики (и, в том числе, в области искусственного интеллекта), и базовые знания по *Технологии OSTIS* и об *Экосистеме OSTIS*.

1

Экосистема OSTIS

⇒ обоснование*:

[Проблема создания рынка совместимых компьютерных систем – **вызов современной науке и технике**. От ученых, работающих в области искусственного интеллекта требуется умение коллективно работать над решением междисциплинарных проблем и доводить эти решения до общей интегрированной теории интеллектуальных систем, предполагающей интеграцию всех направлений искусственного интеллекта, и до технологий, доступных широкому кругу инженеров. От инженеров интеллектуальных систем требуется активное участие в развитии соответствующих технологий и существенное повышение уровня математической, системной, технологической и организационно-психологической культуры.

Но главной задачей здесь является снижение барьера между научными исследованиями в области искусственного интеллекта и инженерией в области разработки интеллектуальных систем. Для этого наука должна стать конструктивной и ориентированной на интеграцию своих результатов в форме комплексной технологии разработки интеллектуальных систем, а инженерия, осознав наукоемкость своей деятельности, должна активно участвовать в разработке технологий.

Особый акцент в *Экосистеме OSTIS* делается на постоянный процесс согласования *онтологий* (и, в первую очередь, на согласование семейства всех используемых понятий и терминов, соответствующих этим понятиям) между всеми (!) активными субъектами *Экосистемы OSTIS* – между всеми *ostis-системами* и всеми пользователями.

При наличии *ostis-систем*, являющихся персональными ассистентами пользователей во взаимодействии с *Экосистемой OSTIS*, вся эта Экосистема будет восприниматься пользователями как единая интеллектуальная система, объединяющая все имеющиеся в *Экосистеме OSTIS* информационные ресурсы и сервисы.

Принципы организации *Экосистемы OSTIS* создают все необходимые условия для привлечения к разработке и совершенствованию *Технологии OSTIS* научные, организационные и финансовые ресурсы, которые будут направлены на развитие методов и средств искусственного интеллекта и на формирование рынка семантически совместимых интеллектуальных систем.]

} /* Завершили Раздел “Предметная область и онтология *Экосистемы OSTIS*” */

/* Раздел *****/

Предметная область и онтология методов и средств реализации целенаправленного и персонафицированного процесса обучения пользователей для каждой ostis-системы, входящей в состав Экосистемы OSTIS

⊃=
{

Предметная область методов и средств реализации целенаправленного и персонафицированного процесса обучения пользователей для каждой ostis-системы, входящей в состав Экосистемы OSTIS

∈ предметная область
 ∋ максимальный класс объектов исследования':
 обучаемость
 ∋ класс объектов исследования':
 • обучаемость пользователя компьютерной системы
 • интеллектуальная компьютерная система
 • подсистема обучения пользователей интеллектуальных систем
 • интеллектуальная обучающая система
 • интеллектуальная обучающая ostis-система

обучаемость

:= [способность системы приобретать новые знания и навыки]
 ⊃ неограниченная обучаемость
 ⇐ требование*:
 интеллектуальная компьютерная система
 := [степень обучаемости, при которой не накладывается никаких ограничений на типологию приобретенных знаний и навыков]
 ⇒ пояснение*:
 [Говоря другими словами, система, обладающая неограниченной обучаемостью, при необходимости может с течением времени приобрести любое знание и способность решать любую задачу]
 ⇒ примечание*:
 [Уточним, что это не означает, что одна конкретная система будет уметь решать любую задачу, это означает, что система может приобрести способность решать нужную ей задачу, при этом нет принципиальных ограничений на класс таких задач.]

обучение пользователя компьютерной системы

⊂ процесс
 ⇒ разбиение*:
 {
 • обучение пользователя компьютерной системы принципам работы с этой компьютерной системой
 := [обучение конечного пользователя компьютерной системы]
 • обучение пользователя компьютерной системы принципам устройства и эволюции этой компьютерной системы
 := [обучение разработчика компьютерной системы]
 • обучение пользователя компьютерной системы знаниям из некоторой предметной области, заложенным в эту компьютерную систему
 }
 ⇒ примечание*:

[Для реализации третьего из перечисленных аспектов обучения существует отдельный класс систем, называемых обучающие системы. В то же время первые два из перечисленных аспектов не менее важны, поскольку неумение конечного пользователя работать с компьютерной системой приводит к ее неэффективному использованию, а незнание разработчиком принципов работы системы приводит к дополнительным накладным расходам при ее эволюции, а иногда и к невозможности или нецелесообразности такой эволюции – проще переделать систему, чем разобраться в том, как она устроена и как ее улучшить. В особенности это актуально для интеллектуальных систем, принципы работы с которыми и принципы функционирования которых сложнее, чем у традиционных компьютерных систем.]

интеллектуальная компьютерная система

:= [сложная техническая система, разработка и даже использование которой требует высоких профессиональных качеств]

⇒ *Проблемы текущего состояния**:

- { • [недостаточная эффективность использования современных интеллектуальных систем, трудоемкость их внедрения и сопровождения, которые в значительной мере определяются высоким порогом вхождения конечных пользователей в интеллектуальные системы.]
- [Пользователь часто не использует значительную часть функций даже традиционных компьютерных систем просто по той причине, что не знает об их наличии и не имеет простого механизма, позволяющего о них узнать. Для интеллектуальных систем данная проблема стоит еще более остро.]
- [Высоки затраты на обучение разработчиков интеллектуальных систем, на их адаптацию под особенности устройства конкретной интеллектуальной системы.]

}
⇒ *примечание**:

[Перечисленные трудности связаны не только с естественной сложностью интеллектуальных компьютерных систем по сравнению с традиционными компьютерными системами, но с низким уровнем документации для таких систем, неудобством использования такой документации, трудоемкостью локализации средств и области решения той или иной задачи, как для конечного пользователя, так и для разработчика.]

⇒ *предлагаемый подход**:

[Для решения указанных проблем, предлагается подход, предполагающий дополнение каждой интеллектуальной системы модулем, представляющим собой интеллектуальную обучающую подсистему, целью которой является обучение конечного пользователя и разработчика основной системы принципам работы с ней, принципам ее функционирования и развития.]

⇒ *основная идея**:

[Независимо от того, для решения каких задач разрабатывается интеллектуальная система, она должна обладать некоторыми функциями обучающей системы, даже если система изначально не является обучающей.]

⇒ *следствие**:

- [пользователь должен иметь возможность обучаться как принципам работы с интеллектуальной системой, так и иметь возможность получать новые знания о той предметной области, для которой создается интеллектуальная система]
- [разработчик интеллектуальных систем должен иметь возможность обучаться принципам внутреннего устройства системы, принципам ее функционирования, назначению конкретных компонентов системы, иметь возможность локализовать ту часть системы, в которой он должен разобраться для внесения изменений в функциональные возможности системы.]

⇒ *примечание**:

[Для реализации данной идеи интеллектуальная система должна содержать не только знания о той предметной области, для которой она разработана, но и:

- знания о самой себе, своей архитектуре, компонентах, функциях, принципах работы и т.д.;
- знания о пользователе, его опыте, навыках, предпочтениях, интересах;
- знания о задачах, которые решает сама система в текущий момент и задачах которые планируются к решению в будущем;
- знания об актуальных задачах по развитию системы и ее сопровождению.

] ⇒ *реализация**:

подсистема обучения пользователей интеллектуальных компьютерных систем

:= [подсистема обучения конечных пользователей и разработчиков интеллектуальных компьютерных систем]

⇒ *реализуемая функция**:

- *обучение пользователя компьютерной системы принципам работы с этой компьютерной системой*
- *обучение пользователя компьютерной системы принципам устройства и эволюции этой компьютерной системы*

{
подсистема обучения конечных пользователей и разработчиков интеллектуальных компьютерных систем

⇒ технологическая основа*:
Технология OSTIS

}

⇒ преимущества*:

- { • [В основе Технологии OSTIS лежит SC-код, который позволяет в унифицированном (одинаковом) виде представить любую информацию, что позволит сделать предлагаемый подход универсальным и подходящим для любого класса интеллектуальных систем.]
- [Технология OSTIS и, в частности, SC-код, легко интегрируется с любыми современными технологиями, что позволит применить предлагаемый подход для большого числа уже разработанных интеллектуальных систем.]
- [SC-код позволяет хранить и описывать в базе знаний ostis-системы любую внешнюю (инородную) по отношению к SC-коду информацию в виде внутренних файлов ostis-систем. Таким образом, база знаний обучающей подсистемы может содержать в явном виде фрагменты уже имеющейся документации к системе, представленной в любой форме.]
- [В рамках Технологии OSTIS уже разработаны модели баз знаний ostis-систем, решателей задач ostis-систем и пользовательских интерфейсов ostis-систем, предполагающие полное их описание в базе знаний системы. Таким образом для ostis-систем предлагаемый подход к обучению конечных пользователей и разработчиков реализуется значительно проще и дает дополнительные преимущества.]
- [Одним из основных принципов Технологии OSTIS является обеспечение гибкости (модифицируемости) систем, разрабатываемых на ее основе. Таким образом, использование Технологии OSTIS обеспечит возможность эволюции самой интеллектуальной обучающей подсистемы.]

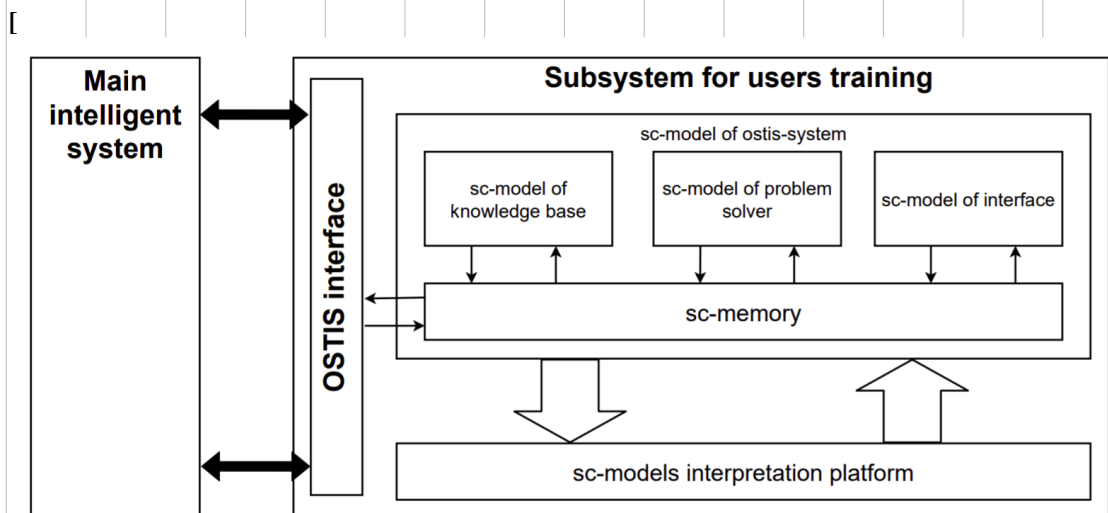
}

подсистема обучения пользователей интеллектуальных систем

⇒ пояснение*:

[Для реализации взаимодействия подсистемы обучения пользователей интеллектуальных систем, реализуемой на основе Технологии OSTIS с основной интеллектуальной системой, которая в общем случае может быть реализована на основе какой-либо другой технологии, предполагается разработка интерфейсного компонента, который также является частью подсистемы. Важно отметить, что для разных интеллектуальных систем такие компоненты будут в значительной степени пересекаться, что, в свою очередь, позволит снизить затраты на интеграцию подсистемы обучения и основной интеллектуальной системы.]

⇒ иллюстрация*:



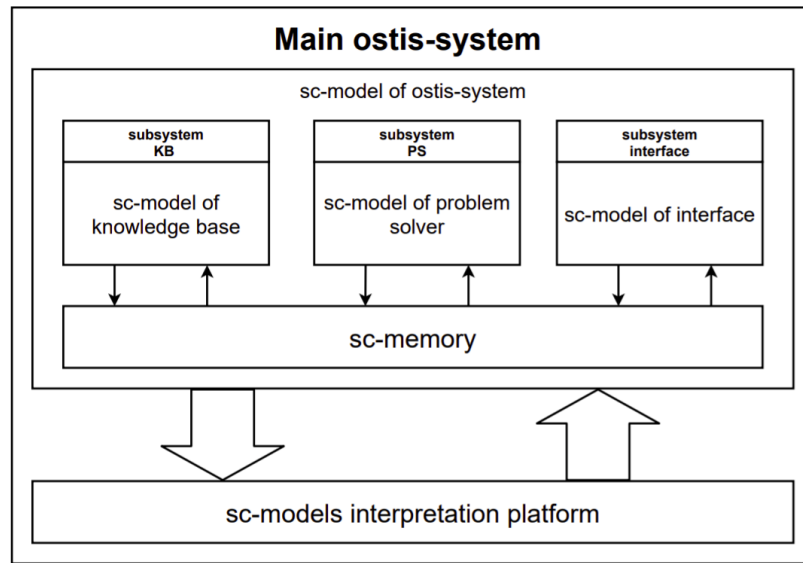
] примечание*:

[В случае, если рассматриваемая интеллектуальная система является ostis-системой, ее интеграция с подсистемой обучения пользователей интеллектуальных систем осуществляется более глубоко. Компоненты подсистемы обучения пользователей интеллектуальных систем просто дополняют уже существующие в основной ostis-системе компоненты, что позволяет максимально снизить

затраты на интеграцию *подсистемы обучения пользователей интеллектуальных систем* и основной *ostis-системы*.]

⇒ иллюстрация*:

[



]

интеллектуальная обучающая система

:= [ИОС]

С интеллектуальная система

⇒ примечание*:

[Такого рода системы по сравнению с традиционными системами электронного обучения (например, электронными учебниками) предоставляют обладают рядом существенных преимуществ.]

С интеллектуальная справочная система

⇒ пояснение*:

[Каждая *интеллектуальная обучающая система* в качестве простейшего средства изучения учебного материала предполагает наличие средств навигации по этому материалу и средств задания по нему различных вопросов. Системы, обладающие только таким ограниченным набором возможностей, названы *интеллектуальными справочными системами*. Таким образом, можно сказать, что *интеллектуальная обучающая система* обязательно реализует в себе функции *интеллектуальной справочной системы*.]

⊃ интеллектуальная обучающая ostis-система

:= [интеллектуальная обучающая система, построенная на основе Технологии OSTIS]

интеллектуальная обучающая ostis-система

⇒ обобщенная часть*:

- база знаний интеллектуальной обучающей ostis-системы

⇒ преимущества*:

- [SC-код позволяет представлять знания любого рода, в том числе конкретные факты, логические утверждения (аксиомы, теоремы, определения), текстовые и мультимедийные иллюстрации и комментарии, примеры конкретных задач с решениями, в том числе доказательства и т.д.]
- [Пользователю становятся доступны достаточно полные сведения об изучаемой предметной области, отражены все ее аспекты, благодаря явному помещению в базу знаний всех предметных закономерностей и взаимосвязей понятий.]
- [База знаний системы рассматривается как иерархия предметных областей и соответствующих им онтологий, то есть позволяет произвести семантическую структуризацию предлагаемого учащемуся материала, что существенно облегчает процесс обучения за счет систематизации знаний на основе именно их семантики, а не каких-либо других сторонних факторов. Кроме этого, знания в базе могут делиться на логические разделы, каждый из которых соответствует какому-либо фрагменту излагаемого материала. База знаний позволяет осуществлять свободную навигацию по любым ассоциативным связям, изучая таким образом материал в той последовательности, которая кажется

более логичной для самого обучаемого. С другой стороны, такой подход позволяет указать рекомендуемую последовательность изучения материала. При необходимости структура предметных областей может быть легко перестроена.]

- [Пользователю в явном виде представляется семантическая структура изучаемого учебного материала и изучаемой предметной области. При этом обеспечивается наглядная визуализация любого уровня указанной семантической структуры.]
- [Знания из различных областей представляются в сходном виде, что позволяет говорить не о семействе не связанных между собой обучающих систем по различным предметным областям, а о глобальном смысловом пространстве, объединяющем в себе знания всего семейства разрабатываемых систем. В свою очередь, наличие такого смыслового пространства обеспечивает ряд дополнительных возможностей:
 - каждая система при необходимости может использовать знания, относящиеся к другим системам, что позволяет задавать не только вопросы, касающиеся конкретной предметной области, но и вопросы, носящие междисциплинарный характер;
 - в рамках глобального смыслового пространства можно выделить часть знаний, которые имеют отношение ко многим системам из всего комплекса, например базовые знания из области математики, логики и т.д. Концепция глобального смыслового пространства позволяет записывать такие фрагменты знаний только в одной из систем, а затем использовать их во всех остальных, что существенно уменьшает количество дублирований, сокращает сроки разработки систем и снижает накладные расходы.
- [Унифицированное представление знаний позволяет не ограничивать номенклатуру пользовательских запросов только специально выделенными для этого командами, а задавать произвольный запрос системе с использованием универсального языка отображения знаний, что делает перечень возможных запросов зависящим только от количества и разнообразия знаний, внесенных в базу знаний системы.]

- } *решатель задач интеллектуальной обучающей ostis-системы*
⇒ *преимущество**:

[Пользователю предоставляется возможность задавать системе любые вопросы и задачи по изучаемой предметной области. Это достигается включением в ИОС решателя задач, способного решать задачи по их формулировкам, в том числе, введенным пользователем. При этом указанный решатель задач может находить путь решения задачи даже, если соответствующий способ решения (например, алгоритм) ему неизвестен.]

- { *пользовательский интерфейс интеллектуальной обучающей ostis-системы*
⇒ *преимущества**:

- [Унификация моделей пользовательских интерфейсов позволяет отображать знания различного рода в унифицированном виде независимо от предметной области, к которой эти знания относятся. Таким образом, все разрабатываемые системы будут обладать пользовательским интерфейсом, построенным по одним и тем же принципам, что позволит существенно сократить срок ознакомления учащегося со всем семейством систем. Данный факт не отрицает возможность и необходимость разработки отдельных компонентов интерфейса, ориентированных на конкретную предметную область, например, редактора геометрических чертежей, виртуальной лаборатории для проведения химических опытов и т.д.]
- [ИОС имеет интеллектуальный пользовательский интерфейс с компьютерными (виртуальными) моделями различных объектов изучаемой предметной области, что позволяет системе "понимать" смысл (анализировать семантику) пользовательских действий по преобразованию этих объектов. Все это существенно повышает уровень интерактивной виртуальной лабораторной среды электронного учебника.]
- [Каждый компонент пользовательского интерфейса также является отображением определенного элемента из базы знаний, что позволяет, во-первых, легко менять интерфейс системы даже во время ее работы, а, во-вторых, позволяет пользователю задавать системе вопросы не только касательно предметной области, которой посвящена данная система, но и касательно любого из компонентов интерфейса и других частей системы. Таким образом, пользователю достаточно научиться задавать системе

несколько простейших вопросов, чтобы в дальнейшем изучить все тонкости работы системой уже в процессе общения с ней.]

- [При общении с системой пользователю предоставляется свобода в выборе любого из множества синонимичных терминов (идентификаторов), зарегистрированных в базе знаний системы. При этом указанные термины могут принадлежать различным естественным языкам.]
- [Появляется принципиальная возможность реализации естественно-языкового интерфейса с пользователем (благодаря широким возможностям семантического анализа пользовательских сообщений и возможностям синтеза на семантическом уровне сообщений, адресуемых пользователям).]
- [Достаточно легко осуществляется переориентация ИОС на обслуживание пользователей с другим естественным языком (т.к. основная часть базы знаний ИОС, непосредственно описывающая семантику соответствующей предметной области, абсолютно не зависит от внешнего языка, в т.ч. от естественного).]

}

⇒ преимущества*:

- { • [Помимо возможности чтения текстов и иллюстративных материалов учебника предоставляется возможность навигации по семантическому пространству предметной области.]
- [Пользователю предоставляется возможность под контролем системы тренироваться (приобретать практические навыки) в решении самых различных задач по изучаемой предметной области. При этом система
 - осуществляет семантический анализ правильности решения задач как по свободно конструируемым ответам (результатам), так и по протоколам решения;
 - локализует допущенные пользователем ошибки в решении задач, определяет их причину и выдает соответствующие рекомендации пользователю.
-]
 - [Пользователю предоставляется полная свобода в выборе последовательности изучения учебного материала (маршрута навигации по учебному материалу), но соответствующие рекомендации выдаются.]
 - [Пользователю предоставляется полная свобода в выборе решаемых им задач (в сборнике задач и лабораторных работ), но соответствующие рекомендации выдаются. Эти рекомендации направлены на то, чтобы минимизировать число решаемых задач, обеспечивающих приобретение требуемых практических навыков.]
 - [Достаточно легко осуществляется интеграция нескольких самостоятельных ИОС по смежным дисциплинам в единый учебник, что, в частности, предоставляет возможность задавать вопросы и задачи на стыке этих дисциплин.]
 - [Пользователь ИОС работает под наблюдением и контролем интеллектуального help-а, который помогает пользователю быстро и эффективно освоить возможности системы. По сути это не что иное, как руководство пользователя ИОС, оформленное как семантический электронный учебник.]
 - [При проектировании базы знаний ИОС появляется уникальная возможность проверять семантическую корректность формируемого информационного ресурса:
 - корректность определений и утверждений;
 - корректность использования различных понятий;
 - корректность алгоритмов;
 - корректность доказательств теорем;
 - и т.д.

} ⇒ примечание*:

[Часть из перечисленных возможностей (а в предельном случае и все их них) могут быть реализованы в рамках подсистемы обучения пользователей интеллектуальных систем.]

}

/* Завершили Раздел “Предметная область и онтология методов и средств реализации целенаправленного и персонализированного процесса обучения пользователей для каждой *ostis*-системы, входящей в состав Экосистемы *OSTIS*” */

```

/* Раздел *****/
Предметная область и онтология ostis-систем, являющихся
персональными ассистентами пользователей в рамках
Экосистемы OSTIS
⊃=
{
Предметная область ostis-систем, являющихся персональными ассистентами пользователей в рамках
Экосистемы OSTIS
∈ предметная область
∋ максимальный класс объектов исследования':
персональный ostis-ассистент

персональный ostis-ассистент
:= [ostis-система, являющаяся персональным ассистентом пользователя в рамках Экосистемы OSTIS]
⇒ возможности*:
{
• [Возможность анализа деятельности пользователя и формирования рекомендаций по ее оптимизации.]
• [Возможность адаптации под настроение пользователя, его личностные качества, общую окружающую
обстановку, задачи, которые чаще всего решает пользователь.]
• [Перманентное обучение самого ассистента в процессе решения новых задач, при этом обучаемость
потенциально не ограничена.]
• [Возможность вести диалог с пользователем на естественном языке, в том числе в речевой форме.]
• [Возможность отвечать на вопросы различных классов, при этом если системе что-то не понятно, то
она сама может задавать встречные вопросы.]
• [Возможность автономного получения информации от всей окружающей среды, а не только от
пользователя (в текстовой или речевой форме). При этом система может как анализировать до-
ступные информационные источники (например, в интернете), так и анализировать окружающий ее
физический мир, например, окружающие предметы или внешний вид пользователя.]
}
⇒ достоинства*:
{
• [Пользователю нет необходимости хранить разную информацию в разной форме в разных местах, вся
информация хранится в единой базе знаний компактно и без дублирований.]
• [Благодаря неограниченной обучаемости ассистенты могут потенциально автоматизировать практи-
чески любую деятельность, а не только самую рутинную.]
• [Благодаря базе знаний, ее структуризации и средствам поиска информации в базе знаний пользователь
может получить более точную информацию более быстро.]
}
⊃ персональный ostis-ассистент учебного назначения
:= [персональный ассистент-учитель]
⊃ персональный ostis-ассистент по здоровому образу жизни и здоровому питанию
:= [персональный фитнес-тренер]
⊃ персональный ostis-ассистент для ухода за пациентом
⊃ персональный секретарь-референт
}
/* Завершили Раздел "Предметная область и онтология ostis-систем, являющихся персональными асси-
стентами пользователей в рамках Экосистемы OSTIS" */

```

```

/* Раздел *****/
Предметная область и онтология семантически совместимых
ostis-систем управления рецептурным производством
⊃=
{
⇒ соавтор*:
• Таберко В.В.
• Иванюк Д.С.
• Касьяник В.В.
• Пупена А.Н.
∈ раздел базы знаний

Предметная область семантически совместимых ostis-систем управления рецептурным производством
∈ предметная область
∋ максимальный класс объектов исследования':
система автоматизации производства
∋ класс объектов исследования':
• разработка системы автоматизации производства
• стандарт предприятия

система автоматизации производства
:= [система автоматизации производственной деятельности предприятия]
:= [система автоматизации деятельности производственного предприятия]
С система автоматизации рецептурного производства
⇒ проблемы текущего состояния*:
{• [Существующие средства автоматизации деятельности предприятия имеют высокую стоимость,
трудны в освоении и адаптации к конкретному производству. Как правило, такие средства, с одной
стороны, жестко ориентированы на решение некоторого ограниченного класса задач, с другой
стороны, разработчики стремятся сделать такого рода средства как можно более универсальными,
наращивая их частными решениями, что приводит к сложности и громоздкости таких систем.]
⇒ следствие*:
[Как следствие подобного подхода к наращиванию функционала, существующие средства
автоматизации деятельности предприятия имеют низкий уровень гибкости (возможности
внесения изменений), что приводит к существенным накладным расходам при адаптации
таких средств к новым требованиям. Как правило, внесение изменений в указанные средства
требует вмешательства разработчиков (часто сторонних с точки зрения предприятия), что
влечет значительные временные и финансовые затраты. Как следствие двух указанных
проблем, далеко не всякое предприятие может обеспечить высокий уровень автоматизации
своей деятельности, даже в случае наличия на рынке подходящих решений.]
• [Отсутствие общих унифицированных моделей и средств построения систем автоматизации деятель-
ности предприятия приводит к большому количеству дублированных аналогичных решений как в
рамках различных предприятий, так и в рамках разных подразделений одного предприятия. При этом
часто возникает ситуация, когда некоторые частные системы, решающие различные задачи в рамках
одного предприятия, оказываются несовместимы между собой, что приводит к дополнительным
расходам на реализацию механизмов согласования, (например, преобразование форматов данных)]
⇒ следствие*:
[Отсутствие такого рода моделей препятствует дальнейшему повышению уровня автоматизи-
зации предприятия, в частности, в области автоматизации принятия решений в нештатных
ситуациях, прогнозирования дальнейшего развития событий.]
• [Высокий уровень зависимости системы автоматизации предприятия от разработчиков приводит к
проблемам внедрения и сопровождения такой системы при смене разработчика]
• [Отсутствие формальных моделей различных стандартов, регламентирующих деятельность предприя-
тия, приводит к возможным трудностям в трактовке тех или иных положений стандарта, обучении
соответствующего персонала и решении текущих вопросов, поскольку поиск необходимой инфор-
мации в документе большого объема может оказаться затруднительным. Кроме того, данный факт
затрудняет процесс проверки предприятия или его подразделений на соответствие необходимым
стандартам.]
}

```


- ⇒ *принципы устранения**:
- { • [Предприятие рассматривается как распределенная, интеллектуальная социотехническая система, в основе которой лежит хорошо структурированная общая база знаний предприятия.]
 - ⇒ *детализация**:
 - [Все знания предприятия объединяются в единое информационное пространство – базу знаний предприятия, которая хранится в семантической памяти.]
 - [В рамках базы знаний предприятия интегрируются все модели предприятия различного уровня детализации, включая модели управления знаниями, онтологические модели бизнес-процессов и модели реинжиниринга этих бизнес-процессов.]
 - [Предприятие рассматривается как иерархическая многоагентная система. В качестве агентов выступают как сотрудники предприятия, так и программные (программно-аппаратные) агенты. Иерархичность многоагентной системы означает то, что агенты могут быть неатомарными, т.е. коллективами взаимодействующих между собой агентов, причем такая структура может быть многократно вложенной. Например, группа роботизированных систем может быть логически (или даже физически) объединена в целый роботизированный комплекс, способный решать задачи определенного класса.]
 - ⇒ *детализация**:
 - [Все участники процесса (люди, роботизированные системы, различного рода производственные комплексы и т.д.) трактуются как агенты над этой общей базой знаний. Это означает, что они (а) отслеживают интересующие их ситуации в базе знаний и реагируют на них (б) описывают результаты своей деятельности в базе знаний, для того чтобы эта информация была доступна другим агентам и они могли ее анализировать.]
 - ⇒ *следствие**:
 - [В конечном итоге весь менеджмент процессов производства при таком подходе сводится к грамотной спецификации задач в такой общей базе знаний, указания их приоритетов, исполнителей, сроков выполнения и т.д.]
 - [Весь комплекс средств (как информационных, так и материальных) обеспечивающих деятельность предприятия оформляется в виде интегрированной распределенной интеллектуальной системы, которую будем называть *интеллектуальной корпоративной системой предприятия*. Основными пользователями этой системы являются сотрудники предприятия.]
 - [Проектирование онтологической модели предприятия сводится к проектированию онтологической модели его интеллектуальной корпоративной системы, которая далее может интерпретироваться имеющимся набором материальных ресурсов. При этом онтологическая модель предприятия является одновременно и объектом, и результатом проектирования.]
 - [В качестве основы для реализации *интеллектуальной корпоративной системы предприятия* предлагается использовать Технологию OSTIS.]
 - ⇒ *следствие**:
 - [Разработка системы сводится к разработке ее модели, описанной средствами SC-кода (sc-модели), которая затем интерпретируется одной из платформ интерпретации.]
 - [База знаний имеет иерархическую структуру, позволяющую рассматривать хранящиеся знания на различных уровнях детализации (прежде всего это иерархия предметных областей и соответствующих им онтологий).]
 - [Модель обработки знаний основана на многоагентном подходе, позволяющем строить параллельные асинхронные решатели задач, интегрировать различные модели решения задач в рамках одной системы.]
- } ⇒ *источник**:
Golenkov.V.V..ODBME-2017cm
- ⇒ *преимущества**:
- { • [Отсутствие необходимости разработки средств непосредственного взаимодействия компонентов системы (человек-роботизированная система, человек-человек и т.д.) за счет их взаимодействия посредством общей памяти.]
 - [За счет того, что все агенты взаимодействуют посредством общей памяти, в общем случае для системы не важно, как физически устроен тот или иной агент. Таким образом, постепенная замена ручного труда автоматизированными системами или совершенствование таких систем не требует внесения изменений в общую систему автоматизации.]

- [За счет использования общей единой базы знаний и широких возможностей ассоциативного поиска в такой базе знаний любой участник процесса производства в любой момент времени имеет доступ ко всей необходимой ему информации, а не к каким-либо заранее предусмотренным ее фрагментам, расширение числа которых может быть связано с дополнительными накладными расходами. Таким образом, существенно облегчается процесс мониторинга различных процессов и ускоряется поиск ответов на интересующие пользователя вопросы. При этом запросы пользователя к системе могут уточняться различными способами]
- [Одна и та же информация, хранящаяся в базе знаний может по-разному визуализироваться для различных категорий пользователей, при этом сама информация будет оставаться неизменной, будут меняться только средства её отображения. Таким образом, отсутствует необходимость дублирования информации]
- [Поскольку все производственные процессы специфицируются в базе знаний и управляются ей, внесение изменений в такие процессы в общем случае сводится к внесению изменений в базу знаний и, при необходимости, замене соответствующего оборудования. При этом существенно снижаются накладные расходы на перепрограммирование компонентов системы, налаживание взаимодействия между ними.]
- [Спецификация всех производственных процессов в единой базе знаний предоставляет широкие возможности для их автоматического анализа, в том числе – постоянного мониторинга текущих процессов, автоматического выявления и устранения нештатных ситуаций, оптимизации текущих процессов, автоматического планирования будущих процессов и т.д.]

}

⇒

источник:**Taberko.V..DBMECI-2018cm***разработка системы автоматизации производства**

С процесс

⇒ класс продуктов*:

система автоматизации производства

D формализация стандартов предприятия

⇒ пояснение*:

[Основой онтологического подхода к проектированию предприятия является формализация стандартов. Каждый стандарт рассматривается как онтология соответствующей предметной области, являющаяся основой для автоматизированного решения ряда задач, включая информационное обслуживание сотрудников, формальную оценку соответствия предприятия этим стандартам и т.д.

Одной из важных проблем внедрения стандарта на предприятии является возможность неоднозначной трактовки некоторых положений стандарта, а также необходимость постоянной коррекции такой трактовки с целью приближения ее к смыслу оригинала. Кроме того, существуют особенности применения стандарта на каждом предприятии, необходимость актуализации используемого стандарта (т.к. любой стандарт постоянно эволюционирует), с последующим внесением изменений в структуру и организацию деятельности предприятия для обеспечения соответствия стандарту. Одним из путей решения такого рода проблем является построение его формальной семантической модели, которая могла бы одинаково интерпретироваться как компьютерной системой, так и человеком. Однако, представление стандарта в виде его формальной семантической модели является не самим стандартом, а субъективной трактовкой этого стандарта разработчиком указанной модели. Но при этом формальное семантическое представление стандарта создает конструктивную почву для его согласования, а также для обеспечения четкости и однозначности его трактовки. Кроме того, формальное семантическое представление стандарта обеспечивает значительное упрощение внесения изменений в такое представление стандарта, которые могут быть вызваны либо уточнением трактовки (понимания) этого стандарта, либо эволюцией самого стандарта (его исходного документа). Указанное упрощение обусловлено тем, что в семантическом представлении любых знаний, в том числе и стандартов, знаки всех описываемых сущностей и связей между ними представлены однократно, т.е. их изменения в рамках базы знаний локализованы, в то время как прослеживание таких связей в естественно-языковом представлении стандарта затруднено.

Формальное семантическое представление стандарта позволяет без внесения каких-либо изменений в структуру такого представления дополнять его различного рода дидактической информацией (примерами, пояснениями, аналогиями и т.д.), которая способствует более быстрому пониманию стандарта сотрудниками предприятия.

Построение формальной модели стандарта сводится к построению интегрированной формальной онтологии, специфицирующей соответствующую предметную область. Для этого необходимо отобразить структуру и содержание исходного текста документа стандарта на иерархию предметных областей и соответствующих им онтологий. Выделение предметных областей позволяет локализовать область решения задач, решаемых в рамках базы знаний, в том числе, проектных. Иными словами, поиск путей решения задачи из некоторой предметной области ограничен рамками этой предметной области.

Использование онтологического подхода к построению формальной модели стандарта позволяет путем добавления интеллектуальных агентов обработки знаний построить на его основе интеллектуальную справочную систему, предоставляющую широкий спектр информационных услуг пользователям, в том числе, способную отвечать на широкий спектр вопросов, ответ на которые может быть в явном виде не представлен в тексте стандарта, или поиск его в таком тексте затруднителен.]

стандарт предприятия

← *класс объектов действия*:*

формализация стандартов предприятия

⇒ *класс продуктов*:*

формальная модель стандарта предприятия

⇒ *проблемы применения*:*

- { • [Дублирование информации в рамках документа, описывающего стандарт]
- [Трудоемкость сопровождения самого стандарта, обусловленная в том числе дублированием информации, в частности, трудоемкость изменения терминологии]
- [Проблема интернационализации стандарта – фактически перевод стандарта на несколько языков приводит к необходимости поддержки и согласования независимых версий стандарта на разных языках]
- [При переводе теряется смысл из-за неудачно написанного текста, неудачного перевода, трудности понимания]
- [Неудобство применения стандарта, в частности, трудоемкость поиска необходимой информации. Как следствие – трудоемкость изучения стандарта]
- [несогласованность формы различных стандартов между собой, как следствие – трудоемкость автоматизации процессов развития и применения стандартов]
- [Трудоемкость автоматизации проверки соответствия объектов или процессов требования того или иного стандарта]

}

⇒ *источник*:*

- *Серенков.П.С..КонцепцияИСКБЗНОО-2004ст*
- *Углева.В.А.АктуализацияССПСТООП-2012ст*

⇒ *принцип устранения*:*

[В качестве основы для автоматизации процессов создания, развития и применения стандартов предлагается использовать Технологию OSTIS и соответствующий набор моделей, методов и средств разработки семантически совместимых интеллектуальных систем, в частности, SC-код как основу формализации стандарта.]

⇒ *преимущества*:*

- { • [Автоматизация процессов согласования стандартов распределенным коллективом авторов.]
- [Возможность фиксации противоречивых точек зрения на одну и ту же проблему в процессе обсуждения и даже в процессе применения разрабатываемого стандарта.]
- [Возможность эволюции стандарта непосредственно в процессе его применения.]
- [Отсутствие дублирования информации на семантическом уровне.]
- [Независимость системы понятий от терминологии, как следствие – от естественного языка, на котором изначально создавался стандарт.]
- [Возможность автоматизации процессов верификации стандартов, в том числе выявления противоречий, информационных дыр, логических дублирований.]
- [Повышение эффективности использования стандарта, обеспечение возможности решать различные задачи без необходимости преобразования стандарта в какой-либо другой формат, в частности, возможность автоматизации процесса проверки соответствия чего-либо необходимым стандартам.]

}

▷ *стандарт рецептурного производства*

Э *ISA-88*

⇒ *пояснение**:

[ISA 88 – основополагающий стандарт для партионного производства. Стандарт широко используется на предприятиях в Америке и Европе, активно внедряется и на территории Республики Беларусь]

⇒ *недостатки современного состояния**:

- { • [Американская версия стандарта – ANSI/ISA- 88.00.01-2010 – является уже обновленной, третьей редакцией от 2010 года, в то же время европейская версия, принятая в 1997 – IEC 61512-1 – основывается на старой версии ISA-88.01-1995]
- [Русский вариант стандарта – ГОСТ Р МЭК 61512- 1-2016 – идентичен IEC 61512-1, то есть также уже устарел]
- [Русский вариант стандарта – ГОСТ Р МЭК 61512- 1-2016 – также вызывает целый ряд вопросов, связанный с не очень удачным переводом оригинальных английских терминов на русский язык]

⇒ *официальный сайт**:

[<https://www.isa.org/isa88/>]

⇒ *пояснение**:

[Основным достоинством ISA-88 является разделение предметной области рецептурного производства на максимально независимые предметные области рецептов, оборудования и управления *Parshall.J..AS88BCFUP-1999кн*. Такое разделение позволяет специалистам предприятия при решении своих задач осуществлять поиск их решения строго в своей предметной области, абстрагируясь от остальных. Этот факт закладывает фундамент в обеспечение гибкости рецептурного производства и сам по себе является основой для построения онтологической модели предприятия.

Стандарт ISA-88 в целом состоит из четырех частей, однако особую ценность для построения формальной онтологии имеет первая часть, поскольку она предоставляет терминологию и целостный набор понятий и моделей, используемых в управлении рецептурным производством.]

⇒ *формализация**:

Формальная модель стандарта ISA-88

С *Предметная область предприятий рецептурного производства*

Предметная область предприятий рецептурного производства

⇒ *частная предметная область**:

- *Предметная область физических моделей рецептурных производств*
- *Предметная область процессных моделей рецептурных производств*
- *Предметная область моделей процедурного управления оборудованием рецептурных производств*
- *Предметная область деятельности по управлению рецептурным производством*

Предметная область физических моделей рецептурных производств

Э *максимальный класс объектов исследования'*:

equipment entity

Э *немаксимальный класс объектов исследования'*:

- *area*
- *site*
- *process cell*
- *unit*
- *equipment module*
- *control module*
- *enterprise*
- *связь оборудования*

:= [equipment relation]

Э *исследуемое отношение'*:

*содержит**

:= [contains*]

Предметная область процессных моделей рецептурных производств

Э *максимальный класс объектов исследования'*:

процессный элемент

- := [process element]
- Э *немаксимальный класс объектов исследования'*:
 - *этап процесса*
:= [process stage]
 - *операция процесса*
:= [process operation]
 - *действие процесса*
:= [process action]
- Э *исследуемое отношение'*:
*связь процессных элементов**
:= [process element link*]

Предметная область моделей процедурного управления оборудованием рецептурных производств

- Э *максимальный класс объектов исследования'*:
процедурный элемент
:= [procedural element]
- Э *немаксимальный класс объектов исследования'*:
 - *процедура процессной ячейки*
:= [process cell procedure]
 - *юнит-процедура*
:= [unit procedure]
 - *операция*
:= [operation]
 - *фаза*
:= [phase]
 - *рецептурный процедурный элемент*
:= [recipe procedural element]
 - *процедурный элемент оборудования*
:= [equipment procedural element]
 - *процедура процессной ячейки рецепта*
:= [recipe process cell procedure]
 - *рецептурная юнит-процедура*
:= [recipe unit procedure]
 - *рецептурная операция*
:= [recipe operation]
 - *рецептурная фаза*
:= [recipe phase]
 - *процедура процессной ячейки оборудования*
:= [equipment process cell procedure]
 - *юнит-процедура*
:= [equipment unit procedure]
 - *операция оборудования*
:= [equipment operation]
 - *фаза оборудования*
:= [equipment phase]
- Э *исследуемое отношение'*:
*порядок исполнения**
:= [execution order*]

фаза

- := [атомарный процедурный элемент]
- ⊃ *рецептурная фаза*
- ⊃ *фаза оборудования*
- ⊂ *процедурный элемент*
- ⇒ *пояснение**:
[Фаза – нижний уровень процедурного элемента в модели технологического управления.]

порядок исполнения*

- ⇒ *первый домен**: *второй домен**:
процедурный элемент

- ⇒ *пояснение**:
[Отношение **порядок исполнения*** связывает *процедурный элемент* с процедурным элементом, начинающим выполнение по его завершении.]
- }
- /* Завершили Раздел “Предметная область и онтология семантически совместимых ostis-систем управления рецептурным производством” */*

/* Раздел *****/

Предметная область и онтология семантически совместимых интеллектуальных порталов научно-технических знаний

⊃=
{
⇒ дочерний раздел*:
 Логико-семантическая модель Метасистемы IMS.ostis

Предметная область семантически совместимых интеллектуальных порталов научно-технических знаний

∈ предметная область
 ∋ максимальный класс объектов исследования':
 портал научных знаний

портал научных знаний

⇒ пояснение*:
 [Целями интеллектуального **портала научных знаний** являются:

- Ускорение погружения каждого человека в новые для него научные области при постоянном сохранении общей целостной картины Мира (образовательная цель);
- Фиксация в систематизированном виде новых научных результатов так, чтобы все основные связи новых результатов с известными были четко обозначены;
- Автоматизация координации работ по рецензированию новых результатов;
- Автоматизация анализа текущего состояния базы знаний.

Создание интеллектуальных **порталов научных знаний**, обеспечивающих повышение темпов интеграции и согласования различных точек зрения, – это способ существенного повышения темпов эволюции научно-технической деятельности.

Совместимые **порталы научных знаний**, реализованные в виде *ostis-систем*, входящих в *Экосистему OSTIS*, являются основой новых принципов организации научной деятельности, в которой

- результатами этой деятельности являются не статьи, монографии, отчеты и другие научно-технические документы, а фрагменты глобальной базы знаний, разработчиками которых являются свободно формируемые научные коллективы, состоящие из специалистов в соответствующих научных дисциплинах,
- с помощью **порталов научных знаний** осуществляется
 - координация процесса рецензирования новой научно-технической информации, поступающей от научных работников в базы знаний этих порталов,
 - процесс согласования различных точек зрения ученых (в частности, введению и семантической корректировке понятий, а также введению и корректировке терминов, соответствующих различным сущностям).

Реализация семейства семантически совместимых порталов научных знаний в виде совместимых *ostis-систем*, входящих в состав *Экосистемы OSTIS*, предполагает разработку иерархической системы семантически согласованных формальных онтологий, соответствующих различным научно-техническим дисциплинам, с четко заданным наследованием свойств описываемых сущностей и с четко заданными междисциплинарными связями, которые описываются связями между соответствующими формальными онтологиями и специфицируемыми ими предметными областями.

Реализация **порталов научных знаний** в виде семейства семантически совместимых *ostis-систем* означает также попытку преодолеть "вавилонское столпотворение" многообразия научно-технических языков, не меняя сути научно-технических знаний, а сводя эти знания к единой универсальной форме смыслового представления знаний в памяти порталов научных знаний, т.е. к форме которая в достаточной степени понятна как *ostis-системам*, так и любым потенциальным их пользователям.

Примером **портала научных знаний**, построенного в виде *ostis-системы* является *Метасистема IMS.ostis*, содержащая все известные на текущий момент знания и навыки, входящие в состав *Технологии OSTIS*.]

} /* Завершили Раздел “Предметная область и онтология семантически совместимых интеллектуальных порталов научно-технических знаний” */

```

/* Раздел *****/
Логико-семантическая модель Метасистемы IMS.ostis
⊃=
{
Метасистема IMS.ostis
⇒ назначение*:
[Эффективность любой технологии, в том числе и Технологии OSTIS определяется не только сроками создания искусственных систем соответствующего класса, но и темпами совершенствования самой технологии (темпами совершенствования средств автоматизации и темпами совершенствования системы стандартов, лежащих в основе технологии).

Для фиксации текущего состояния Технологии OSTIS, а также для организации ее эффективного использования и ее перманентного совершенствования с участием ученых, работающих в области искусственного интеллекта, и инженеров, разрабатывающих семантические компьютерные системы различного назначения, в состав Экосистемы OSTIS вводится Метасистема IMS.ostis (МетасистемаIMS-2016эл), назначение которой делает ее ключевой ostis-системой в рамках Экосистемы OSTIS.]
:= [Интеллектуальная метасистема комплексной информационной и инструментальной поддержки проектирования совместимых семантических компьютерных систем, которая является формой реализации общей теории и технологии проектирования семантических компьютерных систем и которая поддерживает высокий темп эволюции указанной теории и технологии]
:= [интеллектуальная система, предназначенная для комплексной информационной и инструментальной поддержки проектирования семантически совместимых компьютерных систем, на назначение которых не накладывается никаких ограничений]
:= [Intelligent MetaSystem for intelligent systems design]
:= [IMS.ostis]
:= [Фреймворк интеллектуальных систем]
:= [Интеллектуальная метасистема комплексной поддержки проектирования совместимых семантических компьютерных систем по Технологии OSTIS]
:= [Фреймворк ostis-систем]
:= [Фреймворк IMS.ostis]
⇒ url*:
[http://ims.ostis.net]
⇒ назначение*:
[Метасистема IMS.ostis является в Экосистеме OSTIS ключевой интеллектуальной системой, которая поддерживает не только проектирование новых интеллектуальных систем и не только замену устаревших компонентов в интеллектуальных системах, входящих в состав Экосистемы OSTIS, но и включение (интеграцию) в состав Экосистемы OSTIS новых создаваемых интеллектуальных систем.

Метасистема IMS.ostis ориентирована на разработку и практическое внедрение методов и средств компонентного проектирования семантически совместимых интеллектуальных систем, которая предоставляет возможность быстрого создания интеллектуальных приложений различного назначения. Подчеркнем при этом, что сферы практического применения методики компонентного проектирования семантически совместимых интеллектуальных систем ничем не ограничены.]
:= [реализация технологии проектирования семантически совместимых компьютерных систем в виде метасистемы, построенной по той же технологии и обеспечивающей комплексную информационную и инструментальную поддержку проектирования семантически совместимых компьютерных систем]
⇒ декомпозиция*:
[
□ полное описание самой Технологии OSTIS;
□ история эволюции Технологии OSTIS;
□ описание правил использования Технологии OSTIS;
□ описание организационной инфраструктуры, направленной на развитие Технологии OSTIS;
□ библиотека многократно используемых и семантически совместимых компонентов ostis-систем;
□ методы и инструментальные средства проектирования различного вида компонентов ostis-систем;
□ технические средства координации деятельности участников проекта, направленные на постоянное совершенствование Технологии OSTIS.
]

```


- ⇒]
новизна*:
- [Новизна *Метасистемы IMS.ostis* заключается в унификации представления различного вида информации в памяти компьютерных систем на основе смыслового (семантического) представления этой информации, что обеспечивает:
- устранение дублирования одной и той же информации в разных интеллектуальных системах и в разных компонентах одной и той же системы;
 - семантическую совместимость различных компонентов интеллектуальных систем и различных интеллектуальных систем в целом;
 - существенное расширение библиотек совместимых многократно используемых компонентов компьютерных систем за счет "крупных" компонентов и, в частности, типовых подсистем.
- ⇐]
продукт*:
Проект IMS.ostis
- Проект IMS.ostis***
- ⇒ подзадачи*:
- { • [Разработать *Метасистему IMS.ostis*, обеспечивающую быстрое компонентное проектирование семантически совместимых компьютерных систем различного назначения.]
 - [Разработать методы и средства, обеспечивающие интенсивное развитие рынка семантически совместимых прикладных интеллектуальных систем, созданных на основе *Метасистемы IMS.ostis*.]
 - [Разработать методы и средства, обеспечивающие стимулирование интенсивного развития самой *Метасистемы IMS.ostis*.]
- ⇒]
принципы организации*:
- [Организация *Проекта IMS.ostis* реализуется в форме взаимодействия *Метасистемы IMS.ostis* с его пользователями и основана на следующих принципах:
- Решатель задач и пользовательский интерфейс *Метасистемы IMS.ostis* обеспечивают поддержку всего комплекса проектных задач, решаемых разработчиками прикладных интеллектуальных систем, а также разработчиками самой *Метасистемы IMS.ostis*.
 - Для стимулирования развития рынка совместимых прикладных интеллектуальных систем, разработанных с помощью *Метасистемы IMS.ostis* и развития самой этой метасистемы используются технические средства анализа и оценки объекта и значимости персонального вклада каждого разработчика в специальных условных единицах.
 - Для стимулирования развития рынка совместимых прикладных интеллектуальных систем, разработанных с помощью *Метасистемы IMS.ostis*, за каждую такую интеллектуальную систему, зарегистрированную и специфицированную в рамках *Метасистемы IMS.ostis*, разработчикам выделяется вознаграждение в используемых условных единицах после того, как эта прикладная система будет протестирована на предмет семантической совместимости с другими системами, разработанными с помощью *Метасистемы IMS.ostis*. При этом *Метасистема IMS.ostis* становится площадкой для рекламы и распространения интеллектуальных систем, разработанных с ее помощью.
 - Стимулирование развития самой *Метасистемы IMS.ostis* осуществляется следующим образом. Участие в развитии *Метасистемы IMS.ostis* носит открытый характер, для чего достаточно соответствующим образом зарегистрироваться. Авторские права каждого разработчика *Метасистемы IMS.ostis* защищаются и каждый его вклад в зависимости от его ценности автоматически измеряется и фиксируется в используемых условных единицах.
 - Участие в развитии *Метасистемы IMS.ostis* может иметь самые различные формы (в простейшем случае, это может быть указание на конкретные ошибки, на конкретные трудности, с которыми пользователь столкнулся, формулировка конкретных пожеланий; более сложным вкладом является добавление в базу знаний метасистемы новых знаний, новых компонентов в библиотеку многократно используемых компонентов). При этом автор нового многократно используемого компонента, включенного в библиотеку *Метасистемы IMS.ostis*, может выбрать любую лицензию для его распространения и, в том числе, назначить ему любую цену.
 - Ознакомление зарегистрированными пользователями с *Метасистемой IMS.ostis* носит бесплатный открытый характер. При коммерческой разработке прикладных интеллектуальных систем стоимость каждого обращения к библиотекам *Метасистемы IMS.ostis* вполне доступна, но существенно снижается в зависимости от степени активности пользователя в развитии *Метасистемы IMS.ostis*. Это еще один механизм стимулирования участия в развитии *Метасистемы IMS.ostis*.

Таким образом, указанные принципы организации *Метасистемы IMS.ostis* обеспечивают на постоянной основе привлечение к разработке *Метасистемы IMS.ostis* и к формированию рынка семантически совместимых прикладных интеллектуальных систем неограниченные научные, технические и финансовые ресурсы и, в частности, привлечение любых специалистов, желающих участвовать в этом открытом проекте.]

Метасистема IMS.ostis

⇒ *принципы реализации**:

[Принципы технической реализации *Метасистемы IMS.ostis* полностью совпадают с принципами технической реализации прикладных интеллектуальных систем, разрабатываемых с помощью этой метасистемы.]

⇒ *декомпозиция ostis-системы**:

{ • *Sc-модель Метасистемы IMS.ostis*

⇒ *декомпозиция sc-модели ostis-системы**:

- *База знаний Метасистемы IMS.ostis*
- *Решатель задач Метасистемы IMS.ostis*
- *Пользовательский интерфейс Метасистемы IMS.ostis*

• *Программный вариант реализации платформы интерпретации sc-моделей компьютерных систем*

⇒ *подсистема**:

• *Библиотека многократно используемых компонентов OSTIS*

• *Средства разработки компонентов ostis-систем*

⇒ *декомпозиция**:

- *Средства поддержки проектирования баз знаний ostis-систем*
- *Средства поддержки проектирования решателей задач ostis-систем*
- *Средства поддержки проектирования пользовательских интерфейсов ostis-систем*

База знаний Метасистемы IMS.ostis

⇒ *декомпозиция**:

{ • *Стандарт OSTIS*

• *Раздел Проект OSTIS. История, текущее состояние и перспективы эволюции и применения Технологии OSTIS*

• *Документация Метасистемы IMS.ostis*

• *История и текущие процессы эксплуатации Метасистемы IMS.ostis*

• *Раздел Проект IMS.ostis. История, текущие процессы и план развития Метасистемы IMS.ostis*

}

Решатель задач Метасистемы IMS.ostis

⇒ *примечание**:

[Решатель задач собственно *Метасистемы IMS.ostis*, без учета подсистем, включает в себя набор агентов информационного поиска, реализующих базовые механизмы навигации по базе знаний.]

⇒ *декомпозиция**:

{ • *Абстрактный sc-агент поиска всех входящих константных позитивных стационарных sc-дуг принадлежности*

• *Абстрактный sc-агент поиска всех идентификаторов заданного sc-элемента*

• *Абстрактный sc-агент поиска полной семантической окрестности заданного элемента*

• *Абстрактный sc-агент поиска связей декомпозиции для заданного sc-элемента*

• *Абстрактный sc-агент поиска всех известных сущностей, являющихся общими по отношению к заданной*

• *Абстрактный sc-агент поиска определения или пояснения для заданного объекта*

• *Абстрактный sc-агент поиска всех известных сущностей, являющихся частными по отношению к заданной*

• *Абстрактный sc-агент поиска всех выходящих константных позитивных стационарных sc-дуг принадлежности с их ролевыми отношениями*

• *Абстрактный sc-агент поиска всех выходящих константных позитивных стационарных sc-дуг принадлежности*

• *Абстрактный sc-агент поиска всех входящих константных позитивных стационарных sc-дуг принадлежности с их ролевыми отношениями*

}

Библиотека многократно используемых компонентов OSTIS

:= [Библиотека OSTIS]

⇒ *примечание**:

[Библиотека многократно используемых компонентов OSTIS включает в себя как сами многократно используемые компоненты, так и средства их спецификации, а также средства поиска компонентов по различным критериям, и, таким образом, представляет собой целую подсистему *Метасистемы IMS.ostis* со своей базой знаний и решателем задач.]

⇒ *декомпозиция**:

- {
 - Библиотека платформ интерпретации *sc-моделей компьютерных систем*
 - Библиотека многократно используемых компонентов баз знаний *ostis-систем*
 - Библиотека многократно используемых внутренних агентов *ostis-систем*
 - ⇒ *декомпозиция**:
 - {
 - Библиотека *sc-агентов информационного поиска*
 - Библиотека *sc-агентов погружения интегрируемого знания в базу знаний*
 - Библиотека *sc-агентов выравнивания онтологии интегрируемого знания с основной онтологией текущего состояния базы знаний*
 - Библиотека *sc-агентов планирования решения явно сформулированных задач*
 - Библиотека *sc-агентов логического вывода*
 - Библиотека *sc-агентов обнаружения и удаления информационного мусора*
 - Библиотека *координирующих sc-агентов*
 - Библиотека *sc-моделей языков программирования высокого уровня и соответствующих им интерпретаторов*
 - Библиотека многократно используемых методов, интерпретируемых *ostis-системами*
 - ⇒ *подсистема**:
 - Библиотека многократно используемых программ обработки *sc-текстов*
 - ⇒ *подсистема**:
 - Библиотека многократно используемых *scp-программ*
 - Библиотека многократно используемых компонентов интерфейсов *ostis-систем*
 - ⇒ *декомпозиция**:
 - {
 - Библиотека описаний внешних языков
 - Библиотека редакторов внешних информационных конструкций
 - Библиотека трансляторов в *sc-память*
 - Библиотека трансляторов из *sc-памяти* во внешнее представление
 - Библиотека визуализаторов
 - Библиотека сред общения
 - Библиотека элементов управления пользовательским интерфейсом
 - Библиотека многократно используемых встраиваемых *ostis-систем*

Средства поддержки проектирования баз знаний *ostis-систем*:= [Встраиваемая *ostis-система* комплексной поддержки проектирования баз знаний *ostis-систем*]

:= [Встраиваемая типовая интеллектуальная система комплексной автоматизации проектирования, а также управления процессом коллективного проектирования и совершенствования баз знаний интеллектуальных систем на всех этапах их жизненного цикла]

:= [Интеллектуальная система автоматизированного проектирования баз знаний]

:= [Встраиваемая интеллектуальная система, поддержки проектирования и совершенствования баз знаний интеллектуальных систем на всех этапах их жизненного цикла]

:= [Интеллектуальный компьютерный фреймворк баз знаний интеллектуальных систем, разрабатываемых по Технологии OSTIS]

⇒ *пояснение**:

[Известно, что разработка базы знаний интеллектуальной системы является весьма трудоемким процессом, во много определяющим качество интеллектуальной системы. Очевидно также, что сокращение сроков разработки базы знаний возможно путем организации коллективной разработки, но при условии решения ряда задач, например:

- Как в рамках коллектива разработчиков одной и той же базы знаний предотвратить синдром "лебедя, рака и щуки" или синдром "семи нянек" и как снизить накладные расходы на согласование их деятельности по созданию качественной базы знаний.

- Как обеспечить возможность включения любых уже формализованных знаний в базу знаний любой интеллектуальной системы (если они там необходимы) без какой-либо "ручной" корректировки этих знаний и тем самым полностью исключить повторную разработку и адаптацию этих знаний.

]

⇒ *назначение**:

[*Средства поддержки проектирования баз знаний ostis-систем* осуществляют:

- мониторинг деятельности каждого участника процесса проектирования баз знаний, что необходимо для защиты его авторских прав, для оценки объема и значимости его вклада в проектную деятельность, для оценки его профессиональной квалификации, для качественного распределения новых проектных работ с учетом его текущей квалификации и планируемого направления ее повышения, для реализации откатов, то есть отмены ошибочных решений, принятых администраторами или менеджерами проектируемой базы знаний;
- контроль версий проектируемой базы знаний, реализацию необходимых возвратов к предшествующим версиям;
- контроль исполнительской дисциплины;
- анализ текущего состояния и динамики процесса проектирования, выявление критических ситуаций;
- семантический анализ корректности результатов проектных работ всех участников;
- оценку объема и значимости деятельности каждого участника проекта;
- оценку текущего состояния и динамики развития квалификационного портрета каждого участника проекта;
- формирование рекомендаций по повышению квалификации каждого участника проекта;
- контроль качества (непротиворечивости, целостности, полноты, чистоты) текущего состояния проектируемой и совершенствуемой базы знаний.

]

⇒ *принципы функционирования**:

[Каждый участник процесса проектирования базы знаний может выполнять различные виды проектных работ:

- предложить новый фрагмент в согласованную часть базы знаний или некоторую корректировку (удаление, изменение) в этой части базы знаний;
- высказать согласие или несогласие с предложенной кем-то корректировкой или добавлением в согласованную часть базы знаний;
- провести верификацию, тестирование, рецензирование предложенной кем-то корректировки или добавления в согласованную часть базы знаний и написать замечания к доработке этого предложения;
- предложить формулировку нового проектного задания, например, на устранение указываемого противоречия (ошибки), на заполнение указываемой информационной дыры;
- высказать конструктивные критические замечания к формулировке нового проектного задания;
- предложить исполнителя или группу исполнителей для выполнения пока не исполняемого проектного задания;
- высказать конструктивные критические замечания к предложенным исполнителям некоторого свободного проектного задания.

]

⇒ *декомпозиция**:

{ • *База знаний средств поддержки проектирования баз знаний ostis-систем*

• *Решатель задач средств поддержки проектирования баз знаний ostis-систем*

⇒ *декомпозиция**:

{ • *Абстрактный sc-агент верификации баз знаний*

• *Абстрактный sc-агент редактирования баз знаний*

• *Абстрактный sc-агент автоматизации деятельности разработчиков баз знаний*

• *Абстрактный sc-агент автоматизации деятельности администраторов баз знаний*

• *Абстрактный sc-агент автоматизации деятельности менеджеров баз знаний*

• *Абстрактный sc-агент автоматизации деятельности экспертов базы знаний*

• *Абстрактный sc-агент оценки качества базы знаний*

}

• *Пользовательский интерфейс средств поддержки проектирования баз знаний ostis-систем*

}

⇒ *источник**:

Средства поддержки проектирования решателей задач ostis-систем

- ⇒ декомпозиция*:
- {
 - Средства поддержки проектирования программ Базового языка программирования ostis-систем
 - Средства поддержки проектирования коллективов внутренних агентов ostis-систем
 - Интеллектуальная среда проектирования искусственных нейронных сетей, семантически совместимых с базами знаний ostis-систем

Средства поддержки проектирования коллективов внутренних агентов ostis-систем

- ⇒ декомпозиция*:
- {
 - База знаний средств поддержки проектирования коллективов внутренних агентов ostis-систем
 - Решатель задач средств поддержки проектирования коллективов внутренних агентов ostis-систем
 - ⇒ декомпозиция*:
 - {
 - Абстрактный sc-агент верификации sc-агентов
 - ⇒ декомпозиция*:
 - {
 - Абстрактный sc-агент верификации спецификации sc-агента
 - Абстрактный sc-агент проверки неатомарного sc-агента на непротиворечивость его спецификации спецификациям более частных sc-агентов в его составе
 - Абстрактный sc-агент отладки коллективов sc-агентов
 - ⇒ декомпозиция*:
 - {
 - Абстрактный sc-агент поиска всех выполняющихся процессов, соответствующих заданному sc-агенту
 - Абстрактный sc-агент иницирования заданного sc-агента на заданных аргументах
 - Абстрактный sc-агент активации заданного sc-агента
 - Абстрактный sc-агент деактивации заданного sc-агента
 - Абстрактный sc-агент установки блокировки заданного типа для заданного процесса на заданный sc-элемент
 - Абстрактный sc-агент снятия всех блокировок заданного процесса
 - Абстрактный sc-агент снятия всех блокировок с заданного sc-элемента
 - Пользовательский интерфейс средств поддержки проектирования коллективов внутренних агентов ostis-систем
- ⇒ источник*:
Shunkevich.D.V.AgentOMMTCPSPDIS-2018cm

Средства поддержки проектирования программ Базового языка программирования ostis-систем

- ⇒ декомпозиция*:
- {
 - База знаний средств поддержки проектирования программ Базового языка программирования ostis-систем
 - ⇒ пояснение*:
[База знаний средств поддержки проектирования программ Базового языка программирования ostis-систем включает в себя, в частности, типологию некорректностей в scr-программах, способов их выявления и устранения.]
 - Решатель задач средств поддержки проектирования программ Базового языка программирования ostis-систем
 - ⇒ декомпозиция*:
 - {
 - Абстрактный sc-агент верификации scr-программ
 - Абстрактный sc-агент отладки scr-программ
 - ⇒ декомпозиция*:
 - {
 - Абстрактный sc-агент запуска заданной scr-программы для заданного множества входных данных
 - Абстрактный sc-агент запуска заданной scr-программы для заданного множества входных данных в режиме пошагового выполнения

- *Абстрактный sc-агент поиска всех scr-операторов в рамках scr-программы*
- *Абстрактный sc-агент поиска всех точек останова в рамках scr-процесса*
- *Абстрактный sc-агент добавления точки останова в scr-программу*
- *Абстрактный sc-агент удаления точки останова из scr-программы*
- *Абстрактный sc-агент добавления точки останова в scr-процесс*
- *Абстрактный sc-агент удаления точки останова из scr-процесса*
- *Абстрактный sc-агент продолжения выполнения scr-процесса на один шаг*
- *Абстрактный sc-агент продолжения выполнения scr-процесса до точки останова или завершения*
- *Абстрактный sc-агент просмотра информации об scr-процессе*
- *Абстрактный sc-агент просмотра информации об scr-операторе*

}

- *Пользовательский интерфейс средств поддержки проектирования программ Базового языка программирования ostis-систем*

}

}

}

/ Завершили Раздел "Логико-семантическая модель Метасистемы IMS.ostis" */*

/* Раздел ***/**

Библиографический раздел Стандарта OSTIS

⊃=
{

Abrams.M..UIML-AIXUIL-1995cm

⇒ *библиографическая ссылка**:

[M. Abrams, C. Phanouriou, A. L. Batongbacal и др., “UIML: an appliance-independent XML user interface language”, *Computer Networks*, т. 31, № 11, с. 1695—1708, 1999, issn: 1389-1286. doi: [https://doi.org/10.1016/S1389-1286\(99\)00044-4](https://doi.org/10.1016/S1389-1286(99)00044-4). url: <https://www.sciencedirect.com/science/article/pii/S1389128699000444>]

ИскусИнтелМоделМетод-1990кн

⇒ *библиографическая ссылка**:

[*Искусственный интеллект: Модели и методы*. М.: Радио и связь, 1990]

ИскусИнтелПрограмАппаратСредства-1990кн

⇒ *библиографическая ссылка**:

[*Искусственный интеллект: Программные и аппаратные средства*. М.: Радио и связь, 1990]

ИскусИнтелСОиЭС-1990кн

⇒ *библиографическая ссылка**:

[*Искусственный интеллект: Системы общения и экспертные системы*. М.: Радио и связь, 1990]

Butt.A..ActiveRaUL-AWFBUITCAMRD-2013cm

⇒ *библиографическая ссылка**:

[A. Butt, A. Haller, S. Liu и др., “ActiveRaUL: A Web form-based User Interface to Create and Maintain RDF data”, т. 1035, окт. 2013, с. 117—120]

ArangoDB-эл

⇒ *библиографическая ссылка**:

[(сент. 2021). ArangoDB, the multi-model database for graph and beyond. англ., url: <https://www.arangodb.com>]

Арнольд.В.И.ЧтоТМ-2012кн

⇒ *библиографическая ссылка**:

[В. Арнольд, *Что такое математика?* М.: МЦНМО, 2012]

Башмаков.А.И..РазработкаКУИОС-2003кн

⇒ *библиографическая ссылка**:

[А. Башмаков и И. Башмаков, *Разработка компьютерных учебников и обучающих систем*. М.: Филинь, 2003]

Бирюков.Б.В.ТеориСГФ-1960ст

⇒ *библиографическая ссылка**:

[Б. Бирюков, “Теория смысла Готлоба Фреге”, М: АН СССР, 1960]

Van den Bergh.J..CAP3-CSAUIS-2011cm

⇒ *библиографическая ссылка**:

[J. Van den Bergh, K. Luyten и K. Coninx, “CAP3: Context-sensitive abstract user interface specification”, авг. 2011, с. 31—40. doi: [10.1145/1996461.1996491](https://doi.org/10.1145/1996461.1996491)]

Davydenko.I.T.SemanticММТКВСДВРС-2018cm

⇒ *библиографическая ссылка**:

[I. Davydenko, “Semantic models, method and tools of knowledge bases coordinated development based on reusable components”, англ., в *Open semantic technologies for intelligent systems*, V. Golenkov, ред., сер. 2, BSUIR, Minsk, 2018, с. 99—118]

DiscreteMathSys-эл

- ⇒ *библиографическая ссылка**:
 [(сент. 2021). Open Semantic Technology for Intelligent Systems (OSTIS). ostis-discrete-math. англ., url: <https://github.com/ostis-apps/ostis-discrete-math>]
- Duchi.J..AdaptiveSubgradMethods-2011cm**
 ⇒ *библиографическая ссылка**:
 [J. Duchi, E. Hazan и Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization”, *Journal of Machine Learning Research*, 2011]
- Fillmore.Ch.J.TheCCR-1977cm**
 ⇒ *библиографическая ссылка**:
 [C. J. Fillmore, “The Case for Case Reopened. Syntax and Semantics”, англ., в *Syntax and Semantics. vol. 8*, Academic Press, New-York, 1977, с. 32—35]
- Fillmore.Ch.J.FrameS-2005cm**
 ⇒ *библиографическая ссылка**:
 [C. J. Fillmore, “Frame Semantics”, англ., в *Linguistics in the Morning Calm. Papers Prepared for the 1981 Seoul International Conference on Linguistics.*, Hanshin Publishing Co., Seoul, 1982, с. 111—137]
- Финн.В.К.ПринципКонстрИнтелСистем-2008cm**
 ⇒ *библиографическая ссылка**:
 [В. Финн, “Принципы конструирования интеллектуальных систем”, сер. Вып. 4, 2008]
- Глушков.В.М.Кибернетика-1979cm**
 ⇒ *библиографическая ссылка**:
 [В. Глушков, “Кибернетика”, 1979, с. 850—856]
- Головко.В.А.НейроТехОбрДанных-2017кн**
 ⇒ *библиографическая ссылка**:
 [В. А. Головко и В. В. Краснопрошин, *Нейросетевые технологии обработки данных*. Минск: Издательство БГУ, 2017, isbn: 9789855664674]
- Гудфеллоу.Я..ГлубОбуч-2017кн**
 ⇒ *библиографическая ссылка**:
 [Я. Гудфеллоу, И. Бенджио и А. Курвилль, *Глубокое обучение*. Москва: ДМК Пресс, 2017, isbn: 9785970605547]
- Grakn-эл**
 ⇒ *библиографическая ссылка**:
 [(сент. 2021). Vaticle | Home. англ., url: <https://vaticle.com/>]
- Гордей.А.Н.ОснованияКС-2005cm**
 ⇒ *библиографическая ссылка**:
 [А. Н. Гордей, “Основания комбинаторной семантики”, в *Слово и словарь: сборник научных статей по лексикографии*, Гродно, ГрГУ; отв. ред. Л. В. Рычкова [и др.], Гродно, 2005, с. 32—35]
- Гордей.А.Н.ВиртуальнаяЦКСКП-2007cm**
 ⇒ *библиографическая ссылка**:
 [А. Н. Гордей, “Parts of Language and the Procedures of Its Delineation”, в *Пути поднебесной*, Минск, БГУ, Минск, 2006, с. 69—75]
- Гордей.А.Н.ВиртуальнаяЦКСКП-2007cm**
 ⇒ *библиографическая ссылка**:
 [А. Н. Гордей, “Виртуальная цепь как синтаксический код предложения (на примере китайского языка)”, в *Язык, общество и проблемы межкультурной коммуникации: Материалы Междунар. науч. конф.*, ГрГУ, Гродно, 2007, с. 349—358]
- Гордей.А.Н.ТеоретическаяГВЯЛК-2007эл**
 ⇒ *библиографическая ссылка**:

[А. Н. Гордей. (2007). Теоретическая грамматика восточных языков: лекционный курс]

Гордей.А.Н.ВиртуальнаяЦКСКП-2007ст

⇒ *библиографическая ссылка**:

[А. Н. Гордей, “Виртуальная цепь как синтаксический код предложения (на примере китайского языка)”, в *Язык, общество и проблемы межкультурной коммуникации: Материалы Междунар. науч. конф., Гродно, 22–23 нояб. 2007 г. в 2 ч.*, Гродно, ГрГУ, Гродно, 2007, с. 349—358]

Гордей.А.Н.ОКЧЗКЯ-2007ст

⇒ *библиографическая ссылка**:

[А. Н. Гордей, “О комбинаторике числовых знаков в китайском языке”, в *Научные чтения, посвящённые Виктору Владимировичу Мартынову: сб. науч. тр.*, РИВШ, Минск, 2015, с. 12—14]

Гордей.А.Н.ДинамическийССП-2017ст

⇒ *библиографическая ссылка**:

[А. Н. Гордей, “Динамический синтаксис в семантическом представлении”, в *Иностранные языки в высшей школе. – Вып. 4(43)*, Рязань : Редакц.-изд. центр РГУ им. С. А. Есенина, Рязань, 2017, с. 26—34]

Гордей.А.Н.TheoryAGK-2017эл

⇒ *библиографическая ссылка**:

[А. Н. Гордей. (). Theory for Automatic Generation of Knowledge Architecture: TAPAZ-2. англ. Mode of access: <http://tapaz.by>. — Date of access: accessed 18.04.2021]

Гордей.А.Н.PlagiarismPSBCS-2020ст

⇒ *библиографическая ссылка**:

[А. Н. Гордей, “Plagiarism Problem Solving Based on Combinatory Semantics”, англ., в. окт. 2020, с. 176—197, isbn: 978-3-030-60446-2. doi: 10.1007/978-3-030-60447-9_11]

Гордей.А.Н.СемантическаяРСЕОСКРЯ-2021ст

⇒ *библиографическая ссылка**:

[А. Н. Гордей, “Семантическая разметка события и её отображение средствами китайского и русского языков”, в *Иностранные языки в высшей школе. No 2 (57)*, Рязань : Редакц.-изд. центр РГУ им. С. А. Есенина, 2021, с. 5—26. doi: 10.37724/RSU.2021.57.2.001]

Хайкин.С.НейрСети-2006кн

⇒ *библиографическая ссылка**:

[С. Хайкин, *Нейронные сети: полный курс, 2-е издание*. Москва: Издательский дом “Вильямс”, 2006, isbn: 5845908906]

МетасистемаIMS-2016эл

⇒ *библиографическая ссылка**:

[*Метасистема IMS.ostis [Электронный ресурс]*, Режим доступа: <http://ims.ostis.net>. — Дата доступа: 24.11.2016]

ИнформЭнциклопедСловарь-2008кн

⇒ *библиографическая ссылка**:

[*Информатика: Энциклопедический словарь*. М.: Физмалит]

Ивашенко.В.П..ПредставССАОСОВСМП-2015ст

⇒ *библиографическая ссылка**:

[В. П. Ивашенко, Н. Л. Вереник, А. И. Гирель и др., “Представление семантических сетей и алгоритмы их организации и семантической обработки на вычислительных системах с массовым параллелизмом”, в *Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015) : материалы V междунар. науч.-техн. конф., Минск, 19–21 февр. 2015 г.*, Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: В. В. Голенков (отв. ред.) [и др.], Минск, 2015, с. 133—140]

Balzert.H..The-JANUS-ADEGMTTUI-1996ст

⇒ *библиографическая ссылка**:

[H. Balzert, F. Hofmann, V. Kruschinski и др., “The JANUS Application Development Environment—Generating More than the User Interface”, июнь 1996]

Kingma.D.Adam-2014cm⇒ *библиографическая ссылка**:[D. Kingma и J. Ba, “Adam: A Method for Stochastic Optimization”, англ., 2014. url: <https://arxiv.org/pdf/1412.6980.pdf>]**Корончик.Д.Н.УСМПИИСuТИКП-2013cm**⇒ *библиографическая ссылка**:[Д. Н. Корончик, “Унифицированные семантические модели пользовательских интерфейсов интеллектуальных систем и технология их компонентного проектирования”, в *Открытые семантические технологии проектирования интеллектуальных систем*, В. Голенков, ред., БГУИР, Минск, 2013, с. 403—406]**Курбацкий.А.Н.НоваяСЯТЭВНА-2020cm**⇒ *библиографическая ссылка**:[А. Н. Курбацкий, “Новая семантика языка IT - это вам не английский...”, дек. 2020, (accessed 2021, Sep). url: http://vsbel.by/Portico/2020/6/080_%D0%9A%D1%83%D1%80%D0%B1%D0%B0%D1%86%D0%BA%D0%B8%D0%B9.pdf]**Лукашевич.Н.В.ТезауВЗИП-2011кн**⇒ *библиографическая ссылка**:[Н. Лукашевич, *Тезаурусы в задачах информационного поиска*. М.: Москва, 2011]**Paternò.F..MARIA-AUDMALLFSOAIUE-2009cm**⇒ *библиографическая ссылка**:[F. Paternò, C. Santoro и L. Spano, “MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments”, *ACM Trans. Comput.-Hum. Interact.*, т. 16, нояб. 2009. doi: 10.1145/1614390.1614394]**Мартынов.В.В.УниверсКУСК-1984кн**⇒ *библиографическая ссылка**:[В. В. Мартынов, *Универсальный семантический код: УСК-3*, под ред. А. Е. Михневича, ред. Минск: Наука и техника, 1984]**Мартынов.В.В.СемиологОИ-1974кн**⇒ *библиографическая ссылка**:[В. Мартынов, *Семиологические основы информатики*. М.: Наука и техника, 1974]**Puerta.A..Beyond-DMFAUIG-1994cm**⇒ *библиографическая ссылка**:

[A. Puerta, H. Eriksson, J. Gennari и др., “Beyond Data Models for Automated User Interface Generation.”, янв. 1994, с. 353—366]

Мельчук.И.А.ОпытТЛМСТ-1999кн⇒ *библиографическая ссылка**:[И. А. Мельчук, *Опыт теории лингвистических моделей Смысл - текст*. М.: Школа «Языки русской культуры», 1999]**Мельчук.И.А.КакНМЛ-1998cm**⇒ *библиографическая ссылка**:

[И. А. Мельчук, “Как начиналась математическая лингвистика”, 1998, с. 358—370]

Моррис.Ч.У.ОсновТЗ-2001cm⇒ *библиографическая ссылка**:

[Ч. Моррис, “Основания теории знаков”, сер. Вып. 2, Екатеринбург, 2001]

Neo4j-эл⇒ *библиографическая ссылка**:[(сент. 2021). Graph Database Platform | Graph Database Management System | Neo4j. англ., url: <https://neo4j.com/>]

OrientDB-эл

⇒ *библиографическая ссылка**:
 [(сент. 2021). Home | OrientDB Community Edition. англ., url: <https://orientdb.org/>]

Осипов.Г.С.МетодыИИИ-2015кн

⇒ *библиографическая ссылка**:
 [Г. С. Осипов, *Методы искусственного интеллекта*, 2-е. М. : Физматлит, 2015, isbn: 978-5-9221-1655-8]

Parshall.J..AS88BCFUP-1999кн

⇒ *библиографическая ссылка**:
 [L. L. Parshall J., *Applying S88: Batch Control from a user's perspective*. англ. ISA, 1999]

Пирс.Ч.С.ЧтоТЗ-2009ст

⇒ *библиографическая ссылка**:
 [Ч. Пирс, “Что такое знак?”, сер. Т.7, вып. 3, Томск, 2009]

Поспелов.Д.А.МоделированиеРОАМА-1989кн

⇒ *библиографическая ссылка**:
 [Д. А. Поспелов, *Моделирование рассуждений: опыт анализа мыслительных актов*. М.: Радио и связь, 1989]

Редько.В.Г.МоделКЭ-2019кн

⇒ *библиографическая ссылка**:
 [В. Редько, *Моделирование когнитивной эволюции: на пути к теории эволюционного происхождения мышления*. М.: URSS: ЛЕНАНД]

Golenkov.V.V.ODBME-2017ст

⇒ *библиографическая ссылка**:
 [V. V. Golenkov и other, “Ontology-based Design of Batch Manufacturing Enterprises”, 2017, с. 265—280]

Taberko.V..DBMESI-2018ст

⇒ *библиографическая ссылка**:
 [V. Taberko и other, “Design of batch manufacturing enterprises in the context of Industry 4.0”, 2018, с. 307—320]

Серенков.П.С..КонцепцияИСКБЗНОО-2004ст

⇒ *библиографическая ссылка**:
 [П. С. Серенков, В. Л. Соломахо, В. А. Нифагин и др., “Концепция инфраструктуры стандартизации как базы знаний на основе онтологий”, 2004, с. 25—29]

Sesame-эл

⇒ *библиографическая ссылка**:
 [(сент. 2021). Welcome · Eclipse RDF4J™ | The Eclipse Foundation. англ., url: <https://rdf4j.org/>]

Shunkevich.D.V.AgentOMMTCPSPDIS-2018ст

⇒ *библиографическая ссылка**:
 [D. Shunkevich, “Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems”, англ., в *Open semantic technologies for intelligent systems*, V. Golenkov, ред., сер. 2, BSUIR, Minsk, 2018, с. 119—132]

Стефанюк.В.Л.ЛокалОргИС-2004кн

⇒ *библиографическая ссылка**:
 [В. Стефанюк, *Локальная организация интеллектуальных систем. Модели и приложения*. М.: Физматлит]

Степанов.Ю.С.Семиотика-1971ст

⇒ *библиографическая ссылка**:
 [Ю. Степанов, “Семиотика”, сер. вып. 3, М., 1971]

Таранчук.В.Б.WolframMСТРИОС-2015ст

⇒ *библиографическая ссылка**:

[В. Б. Таранчук, “Wolfram Mathematica – средства и технологии разработки интеллектуальных обучающих систем”, в *Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015) : материалы V Междунар. науч.-техн. конф., Минск, 19–21 февр. 2015 г.*, Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: В. В. Голенков (отв. ред.) [и др.], Минск, 2015, с. 339—346]

Тарасов.В.Б.ЭволюСНСНвИИ-1997кн

⇒ *библиографическая ссылка**:

[В. Тарасов, “Эволюционная семиотика – новое синергетическое направление в искусственном интеллекте”, 1997]

Тарасов.В.Б.ЭволюСиНМСОТПкПНО-1998ст

⇒ *библиографическая ссылка**:

[В. Тарасов, “Эволюционная семиотика и нечеткие многоагентные системы – основные теоретические подходы к построению интеллектуальных организаций”, 1998]

Trudeau.R.J.IntroGT-1993кн

⇒ *библиографическая ссылка**:

[R. Trudeau, *Introduction to Graph Theory*, англ., сер. Dover Books on Mathematics. Dover Pub., 1993, isbn: 9780486678702]

Углев.В.А.АктуализацияССПСТОП-2012ст

⇒ *библиографическая ссылка**:

[В. А. Углев, “Актуализация содержания стандартов проектирования сложных технических объектов: онтологический подход”, 2012, с. 80—86]

Limbourg.Q..UsiXML-AUIDLSMLOI-2004ст

⇒ *библиографическая ссылка**:

[Q. Limbourg и J. Vanderdonckt, “UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence”, янв. 2004, с. 325—338]

Винер.Н.Кибернетика-1952кн

⇒ *библиографическая ссылка**:

[Н. Винер, *Кибернетика и общество*. АСТ]

Virtuoso-эл

⇒ *библиографическая ссылка**:

[(сент. 2021). OpenLink Software: Virtuoso Homepage. англ., url: <https://virtuoso.openlinksw.com/>]

Wooldridge.M.IntroMAS-2009кн

⇒ *библиографическая ссылка**:

[M. Wooldridge, *An introduction to multiagent systems*, англ., 2nd. Chichester: J. Wiley, 2009]

XForms-эл

⇒ *библиографическая ссылка**:

[*XForms 1.1 [Electronic resource]*, англ., Режим доступа: <https://www.w3.org/TR/xforms>. – Дата доступа: 25.05.2021]

Ярушкина.Н.Г.НечетГС-2007кн

⇒ *библиографическая ссылка**:

[Н. Ярушкина, *Нечёткие гибридные системы*. М.: ООО Издательская фирма "Физико-математическая литература", 2007]

Golovko.V..IntegrationANNKB-2018

⇒ *библиографическая ссылка**:

[V. A. Golovko, V. V. Golenkov, V. P. Ivashenko и др., “Integration of artificial neural networks and knowledge bases”, в *Open semantic technologies for intelligent systems*, V. Golenkov, ред., сер. 2, BSUIR, Minsk, 2018, с. 133—145]

Д.фон.Нейман.ТеорСамовосАвтомат-1971кн

⇒ *библиографическая ссылка**:

[Д. фон Нейман, *Теория самовоспроизводящихся автоматов*. М.: Мир]

FXML-эл

⇒ *библиографическая ссылка**:

[*Introduction to FXML [Electronic resource]*, англ., Режим доступа: https://openjfx.io/javadoc/12/javafx.fxml/javafx/fxml/doc-files/introduction_to_fxml.html. – Дата доступа: 23.05.2021]

Castelvecchi.B.AIBlackBox-2016

⇒ *библиографическая ссылка**:

[D. Castelvecchi, “Can we open the black box of AI?”, *Nature News*, т. 538, № 7623, Oct 2016.]

Gaulke.W.Using-POTLMDUIG-2015cm

⇒ *библиографическая ссылка**:

[W. Gaulke и J. Ziegler, “Using profiled ontologies to leverage model driven user interface generation”, июнь 2015, с. 254—259. doi: 10.1145/2774225.2775070]

Hitz.M.UsingAOFTAGOUIFDBA-2016cm

⇒ *библиографическая ссылка**:

[M. Hitz и T. Kessel, “Using Application Ontologies for the Automatic Generation of User Interfaces for Dialog-Based Applications”, т. 268, дек. 2016, isbn: 978-3-319-49943-7. doi: 10.1007/978-3-319-49944-4_2]

Tarek.R..NeuralSymbolic-2017

⇒ *библиографическая ссылка**:

[T. R. Besold, A. d’Avila Garcez, S. Bader и др., “Neural-Symbolic Learning and Reasoning: A Survey and Interpretation”, нояб. 2017, (accessed 2020, Jun). url: <https://arxiv.org/pdf/1711.03902.pdf>]

ScWeb-эл

⇒ *библиографическая ссылка**:

[(сент. 2021). *ostis-apps/sc-web*: The sc-web enchancement of <https://github.com/deniskoronchik/sc-web/tree/master>. англ., url: <https://github.com/ostis-apps/sc-web>]

Liu.B..Deriving-UIFOAMBA-2005cm

⇒ *библиографическая ссылка**:

[B. Liu, H. Chen и W. He, “Deriving user interface from ontologies: A model-based approach”, т. 2005, дек. 2005, с. 6, isbn: 0-7695-2488-5. doi: 10.1109/ICTAI.2005.55]

Грибова.В.В..АвтоматизацияРПИСДД-2011cm

⇒ *библиографическая ссылка**:

[В. В. Грибова и А. В. Тарасов, “Автоматизация разработки пользовательских интерфейсов с динамическими данными”, в *Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2011) : материалы Междунар. научн.-техн. конф., Минск, 10-12 февраля 2011г.*, Белорус. гос. ун-т информатики и радиоэлектроники ; редкол.: В. В. Голенков (отв. ред.) [и др.], Минск, 2011, с. 287—292]

Грибова.В.В..ГенераторКПИУО-2005cm

⇒ *библиографическая ссылка**:

[В. В. Грибова и А. В. Тарасов, “Генератор кода пользовательского интерфейса, управляемый онтологией”, в *Искусственный интеллект, Т.4*, 2005, с. 457—464]

} /* Завершили Раздел “Библиографический раздел Стандарта OSTIS” */

Научное издание

Голенков Владимир Васильевич
Гулякина Наталья Анатольевна
Шункевич Даниил Вячеславович

**Открытая технология
онтологического проектирования,
производства и эксплуатации
семантически совместимых гибридных
интеллектуальных компьютерных систем**

Ответственный за выпуск *С. Л. Бочкарева*

Редактор *Д. В. Шункевич*

В авторской редакции

Подписано в печать 08.09.2021. Формат 60x84 1/8.

Бумага офсетная. Печать цифровая.

Усл. печ. л. 80,21. Уч.-изд. л. 40,73.

Тираж 130 экз. Заказ 181.

Издатель и полиграфическое исполнение
УП «Бестпринт». Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий № 1/160 от 27.01.14.
Ул. Филатова, д. 9, к. 1, 220026, г. Минск.