

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра интеллектуальных информационных технологий

**Формальные основы семантического
представления знаний в интеллектуальных
системах**

*Рекомендовано УМО по образованию в области
информатики и радиоэлектроники
в качестве учебно-методического пособия
для специальности 1-40 03 01 «Искусственный интеллект»*

Минск БГУИР 2014

УДК 004.82(075.8)
ББК 32.813я73
Ф79

Авторы:

В. В. Голенков, Н. А. Гулякина,
М. Д. Степанова, С. А. Самодумкин

Рецензенты:

кафедра интеллектуальных систем Белорусского государственного университета
(протокол №5 от 21.11.2013);

ведущий научный сотрудник государственного научного учреждения
«Объединенный институт проблем
информатики Национальной академии наук Беларуси»,
кандидат физико-математических наук, доцент Ю. В. Поттосин

Ф79 **Формальные** основы семантического представления знаний в
интеллектуальных системах : учеб.-метод. пособие / В. В. Голенков [и др.] . –
Минск : БГУИР, 2014. – 68 с. : ил.
ISBN 978-985-488-961-0.

Предназначено для студентов вузов, обучающихся по специальности «Искусственный интеллект»,
а также родственным специальностям. Может быть полезно специалистам в области разработки
интеллектуальных систем различного назначения.

УДК 004.82(075.8)
ББК 32.813я73

ISBN 978-985-488-961-0

© УО «Белорусский государственный
университет информатики
и радиоэлектроники», 2014

ВВЕДЕНИЕ

В настоящее время основным результатом исследований в области искусственного интеллекта являются не столько разработка конкретных интеллектуальных систем, обладающих высокими интеллектуальными возможностями, сколько **разработка технологий**, позволяющих быстро и в большом количестве порождать самые разнообразные интеллектуальные системы, имеющие большую практическую ценность. В состав таких технологий входит:

- формальная теория интеллектуальных систем (формальное описание того, как они устроены);
- методы проектирования интеллектуальных систем;
- инструментальные средства (средства автоматизации проектирования интеллектуальных систем);
- средства информационной поддержки (информационного обслуживания) разработчиков интеллектуальных систем;
- средства компьютерной поддержки управления коллективной разработкой интеллектуальных систем.

Современные технологии проектирования интеллектуальных систем имеют ряд недостатков:

- технологии искусственного интеллекта не ориентированы на широкий круг разработчиков интеллектуальных систем и, следовательно, не получили массового распространения;
- велики сроки разработки интеллектуальных систем и велика трудоемкость их сопровождения;
- высока степень зависимости технологий искусственного интеллекта от платформ, на которых они реализованы, что приводит к существенным изменениям технологий при переходе на новые платформы;
- для эффективной реализации даже существующих моделей представления знаний и моделей решения трудно формализуемых задач современные компьютеры оказываются плохо приспособленными, что требует разработки принципиально новых компьютеров;
- современное состояние в области проектирования интеллектуальных компьютерных систем представляет собой огромное разнообразие самых различных моделей, методов, средств, платформ;
- отсутствуют подходы, позволяющие на некоторой универсальной основе интегрировать научные и практические результаты в области искусственного интеллекта, что порождает высокую степень дублирования результатов. В частности, высока трудоемкость интеграции различных моделей представления и обработки информации, моделей решения задач и, следовательно, различных интеллектуальных компьютерных систем.

Искусственный интеллект является междисциплинарной научной дисциплиной. Этим обусловлен большой ее потенциал, т. к. на стыках научных направлений рождаются сильные результаты. Но этим же обусловлены и большие трудности, т. к. развитие искусственного интеллекта требует глубокого взаимопонимания и сотрудничества исследователей, имеющих различный стиль мышления, подход к объекту и предмету исследования, различный менталитет, целевые установки и традиции. Современный этап развития искусственного интеллекта остро нуждается в преодолении указанных трудностей.

Важнейшей задачей искусственного интеллекта в настоящее время является построение общей комплексной теории интеллектуальных систем, в рамках которой сочетались бы самые разные направления искусственного интеллекта – и теория представления знаний, и теория решения задач (в том числе различные исчисления, эвристики, стратегии), и теория программ (процедурных, декларативных, параллельных, последовательных), и архитектуры интеллектуальных систем (в том числе детализированные до уровня аппаратной поддержки), теория интеллектуальных пользовательских интерфейсов, и компьютерная лингвистика.

Сейчас эпицентром развития искусственного интеллекта является не столько разработка отдельных его направлений, сколько их глубокая **семантическая интеграция**, целью которой должна быть не только общая теория интеллектуальных систем, но и общая, доступная технология их комплексного проектирования.

В основе рассматриваемой семантической компонентной технологии проектирования интеллектуальных систем лежат следующие принципы:

1. Использование опыта наиболее продвинутых технологий.
2. Графодинамические модели.
3. Параллельные асинхронные графодинамические модели.
4. Семантические модели представления и обработки знаний.
5. Унификация семантического представления знаний.
6. Унификация структуризации баз знаний.
7. Графовые языки программирования.
8. Унификация формального описания агентов, работающих над семантической памятью.
9. Унификация семантических моделей обработки знаний.
10. Унификация семантических моделей информационного поиска.
11. Унификация семантических моделей интеграции знаний и семантических моделей интеграции целых интеллектуальных систем.
12. Унификация и интеграция различных семантических моделей решения задач.
13. Унификация визуализации семантических сетей.
14. Унификация семантических моделей различных пользовательских интерфейсов.
15. Библиотека типовых семантически совместимых компонентов интеллектуальных систем и методика модульного проектирования интеллектуальных систем.
16. Платформенно-независимый характер проектирования интеллектуальных систем.
17. Семантический ассоциативный параллельный компьютер.
18. Встроенные подсистемы интеллектуальных систем, обеспечивающие их эффективную эксплуатацию и эволюцию.
19. Доступность и открытость технологии.
20. Эволюционная методика проектирования.
21. Реализация предлагаемой технологии в виде интеллектуальной метасистемы.

1. Опыт наиболее продвинутых технологий

В первую очередь имеется в виду технология проектирования микросхем, которая за последнее время обеспечила существенное сокращение времени и повышение качества разработок благодаря:

- 1) созданию языковых средств формального описания проектируемых микросхем на разных уровнях детализации;
- 2) четкому разделению процесса разработки формальных описаний микросхем и процесса их реализации по заданным формальным описаниям;
- 3) созданию мощных и доступных библиотек формальных описаний типовых (многократно используемых) компонентов микросхем.

Для того чтобы аналогичным образом построить технологию проектирования интеллектуальных систем, необходимо:

- 1) создать языковые средства полного унифицированного формального описания интеллектуальных систем;
- 2) четко отделить разработку полного унифицированного формального описания проектируемой интеллектуальной системы от разработки различных вариантов интерпретации таких формальных описаний интеллектуальных систем;
- 3) создать библиотеку формальных описаний типовых (многократно используемых) компонентов интеллектуальных систем, для этого необходимо обеспечить **интегрируемость** (семантическую совместимость) указанных компонентов.

2. Графодинамические модели

В качестве формальной основы проектируемых интеллектуальных систем будем использовать **графодинамические модели обработки информации**. Такие модели трактуют процесс обработки информации как процесс преобразования графовой структуры, в ходе которого меняется не только состояние элементов этой структуры, но и ее конфигурация (появляются или удаляются ее вершины, а также связи между ними).

Заметим, что для создания графодинамических моделей обработки информации недостаточно тех видов графовых структур, которые в настоящее время исследуются в теории графов. Нам потребуется не только увеличение числа компонентов, инцидентных ребру (т. е. переход от ребра к **гиперребру**), но и увеличение числа компонентов, инцидентных дуге (т. е. переход от дуги к **гипердуге**, которая по сути является графовой трактовкой многоместного кортежа). Нам потребуются не только те ребра, гиперребра, дуги, гипердуги, компонентами которых являются вершины графовой структуры, но и те, компонентами которых являются другие ребра, гиперребра, дуги и гипердуги. Нам потребуются такие связующие элементы графовых структур, которые задают целые фрагменты (подструктуры) заданной графовой структуры, в состав которых входят соответствующие вершины, ребра, гиперребра, дуги, гипердуги [46].

Приведем общее определение **графовой структуры**, на основе которого можно строить практически полезные графодинамические модели обработки информации.

Графовая структура G задается пятеркой $\langle V, C, K, M, I \rangle$,

где V – множество **вершин** (первичных элементов, терминальных элементов);

C – множество **связующих элементов** графовой структуры, каждый из которых задает некоторый фрагмент графовой структуры;

K – множество **ключевых вершин** графовой структуры, каждая из которых задает некоторый класс эквивалентных (однотипных) в определенном смысле элементов графовой структуры ($K \subset V$);

M – множество **меток** элементов (алфавит элементов) графовой структуры, каждая из которых задает некоторый базовый класс эквивалентных в определенном смысле элементов графовой структуры. К ним, в частности, относятся следующие классы:

- всех вершин графовой структуры;
- всех связующих элементов графовой структуры;

- всех ключевых вершин графовой структуры;
- всех меток графовой структуры;
- всех используемых в графовой структуре отношений инцидентности элементов графовой структуры.

I – множество используемых в графовой структуре **отношений инцидентности** элементов, все они являются бинарными ориентированными отношениями. Среди них выделим следующие:

- отношения инцидентности вершин, например, последовательность символов в строке символов;
- отношения инцидентности, каждая пара которых соединяет связующий элемент графовой структуры с элементом (компонентом) того фрагмента, который задается этим связующим элементом. Подчеркнем, что компонентами связующего элемента могут быть элементы графовой структуры любого вида (вершины, связующие элементы, метки, отношения инцидентности). Отметим также, что компоненты одного и того же связующего элемента могут выполнять разные роли в рамках соответствующего фрагмента графовой структуры, который задается связующим элементом. Указанные роли соответствуют разным отношениям инцидентности, входящим во множество I ;
- отношения инцидентности, каждая пара которых связывает ключевую вершину графовой структуры с тем ее элементом, который входит в класс элементов, задаваемый этой ключевой вершиной. Подчеркнем, что элементами, инцидентными ключевой вершине, могут быть элементы графовой структуры любого вида;
- отношения инцидентности, каждая пара которых связывает метку графовой структуры с тем ее элементом, который имеет указанную метку. Подчеркнем, что элементами, инцидентными метке, могут быть элементы графовой структуры любого вида.

Заметим, что связь каждого отношения инцидентности (каждого элемента множества I) с соответствующими парами инцидентности и связь каждой такой пары с элементами графовой структуры, соединяемыми этой парой инцидентности, можно условно считать неявно задаваемыми связями инцидентности более низкого уровня.

Заметим также, что каждую графовую структуру G мы будем трактовать как множество всех элементов, входящих в ее состав:

$$G = (V \cup C \cup K \cup M \cup I).$$

Таким образом, в число элементов графовой структуры входят все ее вершины (в том числе ключевые), связующие элементы, метки и отношения инцидентности.

Заметим также, что множество связующих элементов графовой структуры можно разбить на:

- множество **связок** (простых связующих элементов);
- множество **подструктур**, каждая из которых задает фрагмент графовой структуры, в состав которого входят такие связующие элементы, которые связывают элементы графовой структуры, входящие в указанный фрагмент.

В свою очередь, по признаку ориентированности множество связок можно разбить на следующие:

- множество **ориентированных связок**, некоторые компоненты которых выполняют в рамках этих связок разные роли;
- множество **неориентированных связок**, все компоненты которых выполняют в рамках этих отношений связок одинаковые роли.

Важным частным случаем ориентированной связки является **кортеж**. Кортеж задает такое подмножество элементов графовой структуры, в котором роли всех элементов пронумерованы. То есть в рамках указанного подмножества имеется элемент, который выполняет роль первого элемента (первого компонента), а также элемент, который выполняет роль второго элемента (второго компонента) данного подмножества и т. д.

Если в графовой структуре имеются кортежи, то в число ее отношений инцидентности должны входить следующие отношения:

- быть первым компонентом;
- быть вторым компонентом;
- быть третьим компонентом и т. д.

По количеству компонентов множество связей можно разбить на:

- множество **унарных связей** (одноместных, однокомпонентных);
- множество **бинарных связей** (двухместных, двухкомпонентных);
- множество **многокомпонентных связей** (многоместных), имеющих более двух компонентов.

Неориентированные бинарные связи будем называть **ребрами**, ориентированные – **дугами**. Неориентированные многокомпонентные связи будем называть **гиперребрами**, а ориентированные – **гипердугами**.

Множество связующих элементов графовой структуры можно трактовать как подмножество **шкалы множеств**, заданной над множеством $(V \cup M \cup I)$.

Шкала множеств (H) над указанным множеством определяется рекурсивно:

1) $H \supset (V \cup M \cup I)$

2) если $hj_1, hj_2, \dots, hj_n \in H$,

то $\{hj_1, hj_2, \dots, hj_n\} \in H$,

т. е. любое множество, состоящее из любых элементов шкалы множеств, само также является одним из элементов шкалы множеств;

3) если $hj_1, hj_2, \dots, hj_n \in H$; $ij_1, ij_2, \dots, ije \in I$,

то множество $hj = \{hj_1, hj_2, \dots, hj_n\}$, между которым и множеством $\{ij_1, ij_2, \dots, ije\}$ задано произвольное соответствие, определяющее то или иное распределение ролей между элементами множества hj , также является элементом шкалы (т. е. $hj \in H$).

Основные вопросы, которые ассоциируются с рассмотрением графодинамической модели (графодинамической парадигмы) обработки информации, следующие:

- Можно ли на основе теории графов построить **универсальную абстрактную модель обработки информации**, которая могла бы конкурировать с абстрактной машиной фон Неймана, лежащей в основе традиционных компьютерных систем?
- Нужно ли это делать? Какими преимуществами эта модель обладает по сравнению с абстрактной машиной фон Неймана? Какими преимуществами обладают системы, создаваемые на основе этой модели? Что принципиально нового дает графодинамическая парадигма обработки информации?
- Есть ли к этому предпосылки?
- Можно ли различные модели решения задач (в том числе различные логические исчисления) формально описать в виде графодинамических моделей обработки информации и можно ли обеспечить совместимость (интегрируемость) таких графодинамических моделей?

Интерес к графодинамическим моделям обработки информации имеет достаточно длительную историю. Для подтверждения этого достаточно отметить:

- предложенное А. Н. Колмогоровым уточнение понятия алгоритм [26];
- работы школы М. А. Айзермана по графодинамике [1];
- исследования по графовым грамматикам [42];
- исследования по теории программирования и CASE-технологиям [23];
- разработка параллельных моделей обработки информации [28];
- предложенные В. Б. Борщевым и М. В. Хомяковым клубные системы и вегетативная машина [3].

Для разработки графодинамических моделей обработки информации необходимо рассматривать графовую структуру с позиций семиотики и трактовать ее как **знаковую структуру** (текст), представляющую собой систему взаимосвязанных символов. Такая трактовка графовых структур позволяет привнести семантику в теорию графов.

Почему тексты обязательно должны быть линейными (т. е. цепочками символов)? Но, как только мы введем понятие **графового языка** (т. е. языка, текстами которого являются в общем случае графовые структуры различной конфигурации), возникают следующие вопросы:

- В чем преимущество графовых языков по сравнению с традиционными линейными языками, текстами которых являются цепочки (строки) символов?

- Можно ли построить **универсальный графовый язык**, обеспечивающий представление информации (знаний) любого семантического вида?
- Можно ли в универсальном графовом языке сделать так, чтобы множество всех меток, используемых во всех графовых структурах, являющихся текстами универсального графового языка, было конечным?

Говоря о графовых языках, следует подчеркнуть то, что графовые структуры, являющиеся текстами таких языков, представляют собой абстрактные математические структуры, не уточняющие (не детализирующие) способ их материального представления (например, способ кодирования в компьютерной памяти, способ графического изображения, ориентированного на человеческое восприятие). То есть графовая структура как абстрактный математический объект и ее, например, графическое представление – это принципиально разные вещи. Из этого, в частности, следует, что каждому графовому языку может соответствовать несколько языков, использующих разные способы представления (кодирования, изображения) текстов этого графового языка.

Накопленный опыт развития и применения теории графов и все полученные в ней результаты становятся хорошим математическим фундаментом для разработки различных графовых языков и различных графодинамических моделей обработки информации, а также для создания теории таких языков и моделей. На стыке теории графов и семиотики может появиться очень интересный раздел семиотики – **графовая семиотика**.

3. Параллельные асинхронные графодинамические модели

В качестве формальной основы проектируемых интеллектуальных систем будем использовать графодинамические модели специального вида, ориентированные на **параллельную и асинхронную** обработку информации.

Акцентирование внимания на **параллельных** графодинамических моделях связано со следующими обстоятельствами. Во-первых, потому что без организации параллельной обработки информации невозможно рассчитывать на необходимую производительность подавляющего числа практически полезных интеллектуальных систем. Во-вторых, потому что целый ряд исследований [28] показал перспективность создания параллельных моделей обработки информации именно на основе графодинамического подхода.

Предпочтение **асинхронному** варианту управления обработкой информации отдается потому, что асинхронные модели обработки информации являются более гибкими, их легче интегрировать и наращивать новыми функциональными возможностями.

Графодинамическая модель параллельной асинхронной обработки информации, которую будем также называть **графодинамической параллельной асинхронной машиной**, трактуется нами как абстрактная **многоагентная система**, состоящая из:

- абстрактной **графодинамической памяти**, в которой хранятся обрабатываемые графовые структуры;
- **коллектива агентов**, работающих над общей для них графодинамической памятью и обменивающихся информацией только через эту память (в т. ч. и для координации своих действий).

Графодинамическая память носит реконфигурируемый, структурно перестраиваемый характер, поскольку процесс обработки графовых структур в конечном счете сводится к генерации и удалению различных элементов графовых структур, а также к генерации и удалению пар инцидентности между этими элементами. Другими словами, процесс обработки информации в графодинамической памяти сводится не только к изменению состояния элементов памяти, но и к изменению конфигурации связей между ними.

Агенты, работающие над общей графодинамической памятью, делятся на три вида:

- **внутренние агенты**, каждый из которых реагирует на определенного вида ситуации или события в графодинамической памяти и осуществляет изменение состояния графодинамической памяти, соответствующее своему функциональному назначению;
- **рецепторные агенты**, каждый из которых реагирует на определенные события во внешней среде и осуществляет первичное отражение этих событий в графодинамической памяти;

- эффекторные агенты, каждый из которых реагирует на определенного вида команды, формируемые внутренними агентами в графодинамической памяти, и осуществляет соответствующее изменение материального (физического) состояния интеллектуальной системы, которое определенным образом влияет на изменение ее внешней среды.

Агенты могут работать параллельно, если одновременно возникают условия инициирования агентов.

Асинхронность деятельности внутренних агентов заключается в том, что наличие условия инициирования агента еще не означает начала его работы. То есть время реакции каждого внутреннего агента в известной мере субъективно и достаточно произвольно. В этом смысле указанные агенты обладают:

- свободой выбора момента начала реакции на условие инициирования;
- свободой выбора последовательности обработки условий инициирования, если в текущий момент таких условий возникло несколько.

Для обеспечения эффективного взаимодействия агентов, работающих над общей графодинамической памятью, наряду с предоставляемой им свободой, необходима разработка таких правил их поведения, которые гарантируют безопасность и производительность каждого из них. В конечном счете, эти правила сводятся к двум положениям:

- позаботиться о своей безопасности, точнее, об обеспечении безопасного выполнения своей задачи;
- не навреди другим агентам (помни о том, что ты не один, не создавай для других «аварийных» ситуаций).

Для обеспечения безопасного выполнения своей задачи агент блокирует некоторые элементы графовой структуры, которая хранится в общей графодинамической памяти. Блокировка – это запрет, установленный заданным агентом и адресованный другим агентам, на выполнение тех или иных действий над заданным элементом хранимой графовой структуры. Таким образом, существует несколько видов таких блокировок. Приведем некоторые из них:

- запрет на удаление заданного элемента графовой структуры;
- запрет на удаление всех элементов хранимой графовой структуры, инцидентных заданному (блокируемому) элементу;
- запрет на удаление заданного вида элементов хранимой графовой структуры, которые связаны с блокируемым элементом выходящей из него (или входящей в него) парой инцидентности, принадлежащей заданному отношению инцидентности;
- запрет на генерацию заданного вида элементов в хранимой графовой структуре, которые связаны с блокируемым элементом выходящей из него (или входящей в него) парой инцидентности, принадлежащей заданному отношению инцидентности.

Приведем некоторые правила поведения агента, работающего в коллективе агентов над общей графодинамической памятью:

- не нарушать блокировочные запреты, сформированные другими агентами;
- самому заблокировать тот фрагмент обрабатываемой графовой структуры, целостность которого необходимо сохранить до завершения своей работы;
- не блокировать больше, чем надо;
- снимать свои блокировки как можно быстрее, как только в них отпадает необходимость (т. е. желательно это делать до завершения своей работы);
- удалять сгенерированные для своей работы вспомогательные структуры (информационные «леса») как можно быстрее, как только в них отпадает необходимость (т. е. желательно убирать информационный «мусор» по мере возможности до завершения своей работы);
- поиск фрагментов хранимой графовой структуры, являющихся условиями инициирования агента, осуществлять поэтапно, начиная с поиска тех частей этих условий, которые реже появляются в памяти (это необходимо для того, чтобы скорее установить факт отсутствия условий инициирования).

В случае возникновения **конфликтов** между агентами используются внутренние агенты специального вида, реагирующие на возникновение таких конфликтов и обеспечивающие их разрешение (такие агенты будем называть метаагентами-судьями).

Заметим, что для организации своей деятельности над графодинамической памятью каждый агент «копируется» на соответствующее ему семейство постоянно присутствующих в памяти (резидентных) элементов хранимой в памяти графовой структуры. Указанные элементны будем называть ключевыми элементами агентов. Очевидно, что такие элементы соответствуют константам программ, описывающих поведение агентов.

Сложность комплексного перехода на графодинамическую парадигму параллельной асинхронной обработки информации определяется исключительно психологическими обстоятельствами – это непривычно. Но накопленный человечеством опыт по созданию компьютерных систем и, в частности, интеллектуальных систем позволяет этот переход сделать достаточно быстро, т. к. многие проблемы, возникающие при реализации и применении графодинамических моделей, имеют достаточно близкие аналоги в традиционных компьютерных системах, но многие из них могут быть решены значительно проще. Кроме того, в результате перехода к графодинамическим моделям выявляется целый ряд проблем, которые ранее просто были не видны. Эпицентром такого перехода является формализация семантики и разработка семантически совместимых языковых средств представления различных видов знаний.

4. Семантические модели представления и обработки знаний

В качестве формальной основы проектируемых интеллектуальных систем и основы абстрактных логико-семантических моделей интеллектуальных систем будем использовать графодинамические модели специального вида – семантические модели представления и обработки знаний, в основе которых лежат **семантические сети** [30, 31, 44, 56, 68, 73].

Фактически речь идет о создании формальных средств описания семантики различных видов знаний и формальных средств описания обработки знаний на семантическом уровне.

Семантическая сеть – это графовая структура G , задаваемая пятеркой $\langle V, C, K, M, I \rangle$, где V – множество **вершин** (первичных элементов);

C – множество **связок** (вторичных элементов). Каждая такая связка связывает между собой некоторое число элементов семантической сети. Такими связываемыми элементами (компонентами связки) могут быть как вершины, так и другие связки семантической сети. В зависимости от числа связываемых элементов (компонентов) связки бывают бинарными, тернарными, 4-арными и т. д. В рамках связки ее компоненты могут иметь либо разные, либо одинаковые роли. В зависимости от этого связки бывают ориентированными и неориентированными;

I – семейство бинарных ориентированных **отношений инцидентности**, которые соединяют связки семантической сети с их компонентами. При этом разные отношения инцидентности задают разные роли компонентов связок;

M – **алфавит элементов семантической сети**, который представляет собой «синтаксически» (формально) реализуемое выделение различных множеств элементов путем «приписывания» этим элементам различных меток из множества M ;

K – множество **ключевых узлов**. Множество M от множества K отличается только синтаксическим способом выделения различных подмножеств из множества $(V \cup C)$. При этом использование ключевых узлов (множества K) требует явного введения связок принадлежности во множество связок, соединяющих ключевые узлы с теми элементами семантической сети, которые принадлежат множествам, обозначаемым этими ключевыми узлами.

При этом рассматриваемая математическая структура (G) должна удовлетворять следующим семантическим требованиям:

- каждая вершина $v_j \in V$ является знаком одного из объектов, описываемых семантической сетью, т. е. вершины этой структуры должны быть знаками (обозначениями) различных описываемых объектов;
- каждая ключевая вершина $k_j \in K$ является знаком соответствующего класса элементов графовой структуры G ;

- каждая метка $m_j \in M$ также является знаком соответствующего класса элементов графовой структуры G ;
- каждая пара инцидентности, принадлежащая любому отношению инцидентности $ij \in I$, является парой принадлежности, связывающей знак некоторого множества элементов семантической сети с одним из этих элементов;
- связки этой математической структуры должны быть знаками (обозначениями) различных связей, соединяющих между собой описываемые объекты либо объекты с другими связями, либо различные связи;
- алфавит элементов этой математической структуры должен трактоваться как семейство знаков, каждый из которых обозначает соответствующий тип (класс) описываемых объектов и/или связей;
- в семантической сети вершины $v_j, v_k \in V$ могут быть инцидентны друг другу, но только в том случае, если по крайней мере одна из них (например v_k) является ключевой ($v_k \in K$), а вторая является вершиной, принадлежащей множеству, обозначаемому ключевой вершиной v_k (т. е. $v_j \in v_k$);
- каждый элемент множества I ($ij \in I$) является знаком некоторого подмножества отношения принадлежности, задающего определенную роль, выполняемую соответствующими элементами семантической сети в рамках соответствующих множеств таких элементов. Иначе говоря, отношения инцидентности этой математической структуры должны быть знаками (обозначениями) различных ролей, которые выполняют различные описываемые объекты или связи в рамках тех связей, компонентами которых они являются. Указанные подмножества отношения принадлежности будем называть **ролевыми отношениями**;
- каждый связующий элемент $c_j \in C$ является знаком некоторого фрагмента графовой структуры G , а точнее знаком некоторого подмножества множества всех элементов графовой структуры G ;
- среди элементов графовой структуры G нет пар, синонимичных друг другу знаков, т. е. знаков, обозначающих один и тот же объект (одну и ту же сущность) либо один и тот же внешний описываемый объект, либо одно и то же множество элементов графовой структуры;
- среди элементов графовой структуры G нет омонимичных знаков, которые в разных контекстах, в разных обстоятельствах могут обозначать разные сущности.

Множеством элементов семантической сети $G = \langle V, C, I, M, K \rangle$ будем называть множество $(V \cup C \cup I \cup M \cup K)$.

Следовательно, все элементы (атомарные фрагменты) семантической сети являются знаками различных сущностей (объектов). Такими сущностями могут быть всевозможные внешние описываемые объекты, а также различные множества, состоящие их элементов (атомарных фрагментов) этой же семантической сети.

Таким образом, семантическая сеть – это абстрактная знаковая конструкция «рафинированного вида», в которой нет ничего кроме знаков и инцидентности этих знаков. В частности, в семантической сети отсутствуют элементарные незначащие фрагменты (символы), имена описываемых объектов, слова, из которых эти имена состоят, всевозможные разделители и ограничители, обеспечивающие структуризацию текста. В отличие от текстов традиционного вида семантическая сеть имеет в общем случае нелинейный характер, поскольку каждый элемент семантической сети может быть инцидентен более чем двум другим элементам.

Семантическую сеть можно трактовать как абстрактный текст, который является семантическим инвариантом соответствующего максимального множества семантически эквивалентных текстов, принадлежащих всевозможным языкам.

На основе понятия семантической сети вводится понятие языка семантических сетей в заданном алфавите и с заданным набором ключевых узлов.

Язык семантических сетей в заданном алфавите (M) – это бесконечное множество семантических сетей, которому ставится в соответствие фиксированный конечный алфавит, за пределы которого не выходит ни одна из семантических сетей указанного множества.

Язык семантических сетей в заданном алфавите (M) и с заданным семейством ключевых узлов (K) – это множество семантических сетей, для каждой из которых алфавитом является множество A , а семейством ключевых узлов – множество K .

Язык семантических сетей L в алфавите M с ключевыми узлами K является результатом **интеграции** языка L_1 (который имеет алфавит M_1 и набор ключевых узлов K_1) и языка L_2 (который имеет алфавит M_2 и набор ключевых узлов K_2) в том и только в том случае, если $M = M_1 \cup M_2$; $K = K_1 \cup K_2$.

Из приведенного выше уточнения понятия семантической сети следует, что в общем случае, в отличие от традиционных текстов, семантические сети являются нелинейными знаковыми конструкциями (нелинейными текстами). Это обусловлено тем, что:

- 1) знак каждого объекта описываемой предметной области входит в семантическую сеть однократно, т. к. синонимия в семантической сети запрещена;
- 2) каждый объект описываемой предметной области имеет неограниченное число связей с другими объектами этой области и все эти связи обозначаются связками семантической сети, инцидентными тому элементу семантической сети, который обозначает рассматриваемый объект.

Тексты (знаковые конструкции) с которыми мы привыкли иметь дело устроены совсем по-другому.

Во-первых, они линейны, т. к. являются цепочками (строками), состоящими из элементов (атомарных фрагментов, символов) текста, связанных отношением инцидентности. Каждая связка этого отношения связывает предшествующий (левый) символ с последующим (правым) символом. При этом у каждого символа существует не более одного предшествующего и не более одного последующего символа.

Во-вторых, в отличие от семантических сетей элементы (атомарные фрагменты, символы) традиционных текстов в общем случае не являются знаками объектов описываемой предметной области. Такие знаки представляются в виде неатомарных фрагментов текстов, т. е. в виде строк символов (слов, словосочетаний), входящих в состав таких текстов и, соответственно, являющихся именами описываемых объектов. В семантических сетях знак каждого описываемого объекта, используемого понятия, описываемой связи является синтаксически элементарным фрагментом (элементом) семантической сети.

В-третьих, в отличие от семантических сетей в текстах традиционного вида необходимы специальные средства выделения различных фрагментов (различных строк символов, входящих в состав текста). К таким средствам относятся различные разделители: пробелы (разделитель слов), запятые, точки, скобки различного вида. При этом разделитель и ограничитель может быть представлен как одним символом, так и строкой (цепочкой) символов.

В-четвертых, в текстах традиционного вида возможна тривиальная синонимия. Будем говорить, что строки символов u_i , u_j тривиально синонимичны, в том и только в том случае, если:

- 1) строки u_i и u_j не имеют общих элементов (символов);
- 2) строки u_i и u_j имеют одинаковое число элементов и одинаковую структуру (т. е. являются изоморфными для заданного алфавита, являются разными вхождениями одной и той же обобщенной строки);
- 3) строки u_i и u_j являются знаками одного и того же элемента описываемой предметной области.

Таким образом, в обычных (линейных) текстах имя описываемого объекта в общем случае может входить в состав текста неоднократно. В семантических сетях ничего подобного нет, т. к. там синонимия знаков запрещена.

В-пятых, в текстах традиционного вида возможна нетривиальная синонимия. Будем говорить, что строки символов u_i , u_j нетривиально синонимичны в том и только в том случае, если:

- строки u_i , u_j не имеют общих элементов;
- строки u_i , u_j имеют разную структуру;

- строки u_i, u_j являются знаками одного и того же элемента описываемой предметной области.

В-шестых, в текстах традиционного вида возможна омонимия. Будем говорить что строки символов u_i, u_j омонимичны в том и только в том случае, если:

- строки u_i, u_j не имеют общих элементов;
- строки u_i, u_j имеют одинаковую структуру (т. е. являются изоморфами для заданного алфавита)
- строки u_i, u_j являются знаками разных элементов описываемой предметной области.

В обычных (линейный) языках семантические связи между описываемыми объектами выражаются через различные синтаксические зависимости между именами описываемых объектов (такие зависимости оформляются с помощью падежей, предлогов, последовательности слов, ограничителей, разделителей и т. д.).

В семантической сети каждой описываемой связи между описываемыми объектами соответствует один связующий элемент семантической сети. Объектам, связанным этой связью, соответствуют обозначающие их элементы семантической сети, инцидентные указанному выше связующему элементу (связке элементов семантической сети). Такие связки могут быть бинарными (т. е. имеющими два инцидентных элемента) и небинарными, а также могут быть ориентированными, компоненты которых имеют разные роли, и неориентированными.

Таким образом, структура линейного текста требует больших усилий синтаксического анализа, поскольку связи между объектами описываемой предметной области носят нелинейный характер, но эта нелинейность загоняется в жесткие рамки синтаксической линейности текста.

Отсюда проблемы выделения словосочетаний, установление синтаксических и семантических связей между ними (части речи, падежи, склонения, спряжения и т. д.).

Есть еще проблемы, обусловленные стремлением к компактности (минимизации числа символов) линейного текста: омонимичные сокращения, осуществляемые путем удаления некоторых слов из словосочетаний, местоимения и многое другое.

Семантическая сеть принципиально отличается от традиционного (линейного) текста тем, что ее понимание (т. е. установление ее соответствия с описываемой предметной областью, ее интерпретация на описываемой предметной области) в отличие от понимания традиционного текста не требует сложных процедур синтаксического и семантического анализа, целью которых является переход от связей между элементами и фрагментами традиционного текста к связям между элементами описываемой предметной области.

Следовательно, понимание традиционного текста в интеллектуальной системе, база знаний которой представлена в виде семантической сети, можно формально трактовать как перевод указанного традиционного текста с заданного линейного языка на заданный язык семантических сетей, на котором оформлена база знаний.

Семантические сети как модели представления знаний известны давно. Так, например, обобщенное понятие графовой структуры можно считать эквивалентом понятия семантической сети, т. к. вершины графовой структуры вряд ли могут быть синонимичными или омонимичными, хотя это явно нигде не оговаривается, поскольку теория графов семантикой графовых структур не занимается. Фактически теория графов рассматривает синтаксический аспект семантических сетей, исследуя различные виды их конфигураций с точностью до изоморфизма. Общепринятые способы представления принципиальных электрических схем, логических схем, блок-схем тоже не что иное, как семантические сети, представленные в различных алфавитах. Очевидно также, что к числу семантических сетей также относятся такие способы представления информации, как когнитивные карты, карты знаний и многое другое [5]. Но в отличие от фреймовых, продукционных и логических моделей для семантических сетей не были разработаны достаточно удобные и практически используемые языки представления знаний и языки программирования, специально ориентированные на обработку семантических сетей. И, как следствие этого, не были созданы широко используемые комплексные технологии проектирования интеллектуальных систем, в основе которых лежат семантические сети. Причин тому много. Одна из них – это не совсем привычный, нетрадиционный характер таких моделей и возникший на этой основе миф о сложности их реализации. Если такие графодинамические семантические модели реализовывать «в лоб», без ориентации на последующую аппаратную поддержку,

то, конечно, это будет неэффективно. Но жизнь берет свое. И на фоне бурного развития микроэлектронных технологий подобного рода мифы выглядят все менее и менее убедительными. Более того, развитие Internet-технологий привело к необходимости формализации семантики информации, обрабатываемой в сети Internet, что вызывало бурное развитие целого направления – Semantic Web.

Особо подчеркнем то, что семантическую сеть как абстрактную математическую структуру следует четко отличать от различных вариантов ее материального (физического) представления в компьютерной памяти, графической визуализации. Абстрактное понятие семантической сети от этого как раз и абстрагируется. Различные семантические сети могут иметь различные алфавиты элементов (различные наборы меток на элементах семантических сетей). Следовательно, можно говорить о семантических сетях, представленных в разных алфавитах, а также о различных **языках семантических сетей**, каждому из которых ставится в соответствие свой фиксированный алфавит, в котором представляются все семантические сети, принадлежащие этому языку.

Чем же хороши семантические сети и в чем достоинство семантических моделей обработки информации?

- Представление знаний в виде семантических сетей позволяет существенно упростить процедуру интеграции знаний и свести эту процедуру к выявлению и склеиванию синонимичных элементов интегрируемых семантических сетей.
- Специфика обработки баз знаний заключается в том, что порождаемые (генерируемые) новые фрагменты знаний необходимо не просто построить, но и погрузить, интегрировать в текущее состояние базы знаний, т. к. в этих порождаемых фрагментах знаний могут появиться знаки, синонимичные тем, которые уже присутствуют в текущем состоянии базы знаний. Таким образом, процедура интеграции порождаемых фрагментов обрабатываемой базы знаний является процедурой, постоянно используемой в ходе обработки знаний. Следовательно, представление знаний в виде семантических сетей, благодаря упрощению процедуры интеграции знаний, позволяет упростить не только ввод новых знаний из вне, но и интеграцию в состав текущего состояния базы знаний новых знаний, порождаемых в ходе решения задач.
- База знаний интеллектуальной системы, представленная в виде корректно построенной семантической сети, полностью исключает дублирование информации в рамках такой базы знаний – каждая информация, представленная соответствующим фрагментом семантической сети, должна находиться в рамках этой семантической сети там и только там, где она должна находиться, и нигде больше.
- Представление знаний в виде семантических сетей позволяет существенно упростить процедуру ассоциативного доступа к различным видам фрагментов хранимой базы знаний, а также существенно расширить типологию запросов (вопросов) к базе знаний.
- Семантические модели обработки знаний не просто хорошо приспособлены к поддержке параллельной асинхронной обработки информации, но и обеспечивают обмен информацией через общую графодинамическую память между различными выполняемыми одновременно процессами, что может существенно ускорить каждый из этих процессов. Примером такого взаимодействия параллельно протекающих процессов является одновременная реализация разных стратегий и тактик, направленных на поиск пути решения заданной нетривиальной задачи.
- С помощью семантических моделей представления и обработки знаний можно проинтерпретировать все известные виды моделей представления и обработки знаний (фреймовые, продукционные, логические), а также все известные модели решения задач различного вида и модели рассуждений. Это дает возможность рассматривать перечисленные модели не как альтернативные, а как дополняющие друг друга модели, которые могут сосуществовать в разных сочетаниях в разных интеллектуальных системах.

Семантическая модель обработки знаний представляет собой абстрактную многоагентную систему, состоящую из **абстрактной семантической памяти**, в которой хранятся семантические сети, и из множества агентов, ориентированных на обработку семантических сетей, хранимых в указанной семантической памяти.

Семантическую память можно трактовать как абстрактную семантическую модель памяти интеллектуальной системы.

Семейство абстрактных агентов, работающих над семантической памятью вместе с этой памятью можно трактовать как **семантическую модель решения задач**, используемую в соответствующей интеллектуальной системе, или как операционную семантику этой интеллектуальной системы. Подчеркнем, что семантическую модель обработки информации можно построить для любой компьютерной системы (как для интеллектуальной, так и для системы традиционного вида), обеспечивая тем самым семантическую совместимость на абстрактном уровне не только интеллектуальных систем, но и компьютерных любого уровня интеллектуальности.

Всю максимальную семантическую сеть, хранимую в семантической памяти абстрактной логико-семантической модели интеллектуальной системы, будем называть **абстрактной семантической моделью базы знаний** этой интеллектуальной системы.

База знаний должна содержать в себе всю информацию, необходимую агентам, работающим над семантической памятью, для организации коллективной деятельности по решению задач, с которыми должна справляться интеллектуальная система (сюда в том числе входит и описание блокировок, задаваемых разными процессами в семантической памяти).

Семантическая модель базы знаний интеллектуальной системы – это, образно говоря, формальная трактовка «семантического пространства», в котором находится эта интеллектуальная система, а точнее такого фрагмента указанного «семантического пространства», который в текущий момент данной интеллектуальной системе известен.

В целом логико-семантическая модель интеллектуальной системы включает в себя:

- семантическую модель базы знаний этой интеллектуальной системы;
- семантическую машину обработки знаний этой интеллектуальной системы, которая, в свою очередь, состоит из:
 - семантической памяти;
 - коллектива агентов над семантической памятью.

5. Унификация семантического представления знаний

Данный принцип дает возможность определить структуру унифицированных семантических сетей, обеспечивающих представление самых различных видов знаний.

Это предполагает разработку соответствующего стандарта, выделяющего из всего многообразия абстрактных языков семантических сетей определенный базовый универсальный язык семантических сетей – SC-код (Semantic Computer code).

Для того чтобы перейти от семантических сетей произвольного вида к текстам SC-кода, уточним направления такого перехода, т.е. задаваемые критерии качества разрабатываемого языка семантических сетей (SC-кода). К таким критериям мы отнесем:

- 1) переход от семантических сетей, имеющих унарные и многокомпонентные (многоместные) связи, к семантическим сетям, имеющим только бинарные связи. Такие сети будем называть бинарными семантическими сетями [22, 45]. При прочих равных условиях лучше тот язык семантических сетей, алфавит которого состоит из меньшего числа элементов (меток). Очевидно, что предельно минимальный алфавит языка семантических сетей должен как минимум обеспечивать расшифровку связи каждого ключевого узла семантической сети с теми, которые принадлежат множеству, обозначаемому этим ключевым узлом. То есть в число элементов алфавита семантической сети как минимум должен быть включен знак отношения принадлежности (re);
- 2) универсальность языка, которая означает, что алфавит, соответствующий языку семантических сетей, и набор его ключевых узлов должен обеспечивать представление (запись) любых видов знаний;
- 3) бинарность семантических сетей. Этот критерий означает, что при прочих равных условиях предпочтение отдается языкам семантических сетей, текстами которых являются бинарные семантические сети;
- 4) алфавит языка семантических сетей должен быть семейством попарно непересекающихся множеств, чтобы каждый элемент семантической сети имел не более одной метки, т.е. принадлежал не более, чем одному множеству, входящему в состав алфавита семантической сети.

Семантическую сеть G_b , задаваемую пятеркой $\langle V_b, C_b, K_b, M_b, I_b \rangle$, будем называть **бинарной семантической сетью** в том и только в том случае, если:

- все связующие элементы, входящие во множество C_b , являются бинарными связками, каждая из которых имеет только два компонента (ими могут быть элементы множеств V_b , C_b и K_b);
- $I_b = \{i_1, i_2\}$, где i_1 – бинарное ориентированное отношение инцидентности, связывающее бинарную связку с ее компонентом; i_2 – бинарное ориентированное отношение инцидентности, связывающее бинарную ориентированную связку с ее вторым компонентом. Очевидно, что $i_1 \subset i_2$.

Для любой семантической сети G , задаваемой пятеркой $\langle V, C, K, M, I \rangle$, можно построить семантически эквивалентную ей бинарную семантическую сеть G_b , задаваемую пятеркой

$\langle V_b, C_b, K_b, M_b, I_b \rangle$ следующим образом:

- во множество V_b включаются все вершины (V) семантической сети G и все небинарные связующие элементы (C_n) семантической сети G ($C_n \subset G$);
- $I_b = \{i_1, i_2\}$ – это отношения инцидентности, связывающие бинарные связки (в том числе из множества C) с их компонентами;
- $M_b = (M \cup \{re\})$. Если в алфавит M не входила метка отношения принадлежности (re), то в алфавит M_b она дополнительно вводится;
- $K_b = K \cup (I \setminus (I_b \cup \{re\}))$. То есть знаки отношений инцидентности, связывающих в рамках G небинарные связующие элементы с их компонентами, становятся в рамках G_b ключевыми узлами;
- во множество C_b включаются: 1) все бинарные связки семантической сети G ; 2) все ориентированные пары отношения принадлежности (re), связывающие ключевые вершины семантической сети G с инцидентными им элементами семантической сети; 3) все ориентированные пары всех отношений инцидентности (I), связывающих небинарные связующие элементы (из множества C) с их компонентами; 4) все пары принадлежности, явно связывающие ключевые узлы семантической сети G_b , соответствующие различным отношениям инцидентности (I) семантической сети G , с парами принадлежности которые явно связывают небинарные связующие элементы с их компонентами, выполняющими в рамках этих элементов роли, обозначаемые указанными отношениями инцидентности.

Нетрудно заметить, что приведение семантической сети к бинарному виду приводит также к минимизации числа отношений инцидентности.

Нетрудно также заметить, что все пары инцидентности, связывающие небинарные связующие элементы семантической сети G с их компонентами, в бинарной семантической сети C_b превращаются в бинарные ориентированные связки, которые принадлежат отношению принадлежности и которые, следовательно, должны быть помечены меткой re ($re \in M_b$), являющейся знаком отношения принадлежности.

Таким образом, каждый небинарный связующий элемент семантической сети G в семантической сети G_b трактуется как множество связываемых им элементов семантической сети, связь которого с его элементами представляется явно – не в виде пар инцидентности, а в виде дополнительно вводимых связок принадлежности. Эти связки принадлежности, в свою очередь, могут быть вторыми компонентами других связок принадлежности, связывающих указанные связки с ключевыми узлами, обозначающими различные роли компонентов небинарных связующих элементов исходной семантической сети.

В семантических сетях важно избавляться от небинарных связок. Во-первых, бинарные связки легко изображать графически: в виде линий, каждая из которых соединяет графические изображения двух связываемых элементов семантической сети. Во-вторых, использование только бинарных связок существенно упрощает машинное кодирование семантических сетей и, в частности, разработку специальной памяти для хранения семантических сетей.

В качестве примера приведем бинарное представление небинарной связки отношения **возведение в степень***, а также представление нескольких алгебраических операций, соответствующих одному небинарному отношению. В приведенном примере это операции логарифмирования взятия корня.

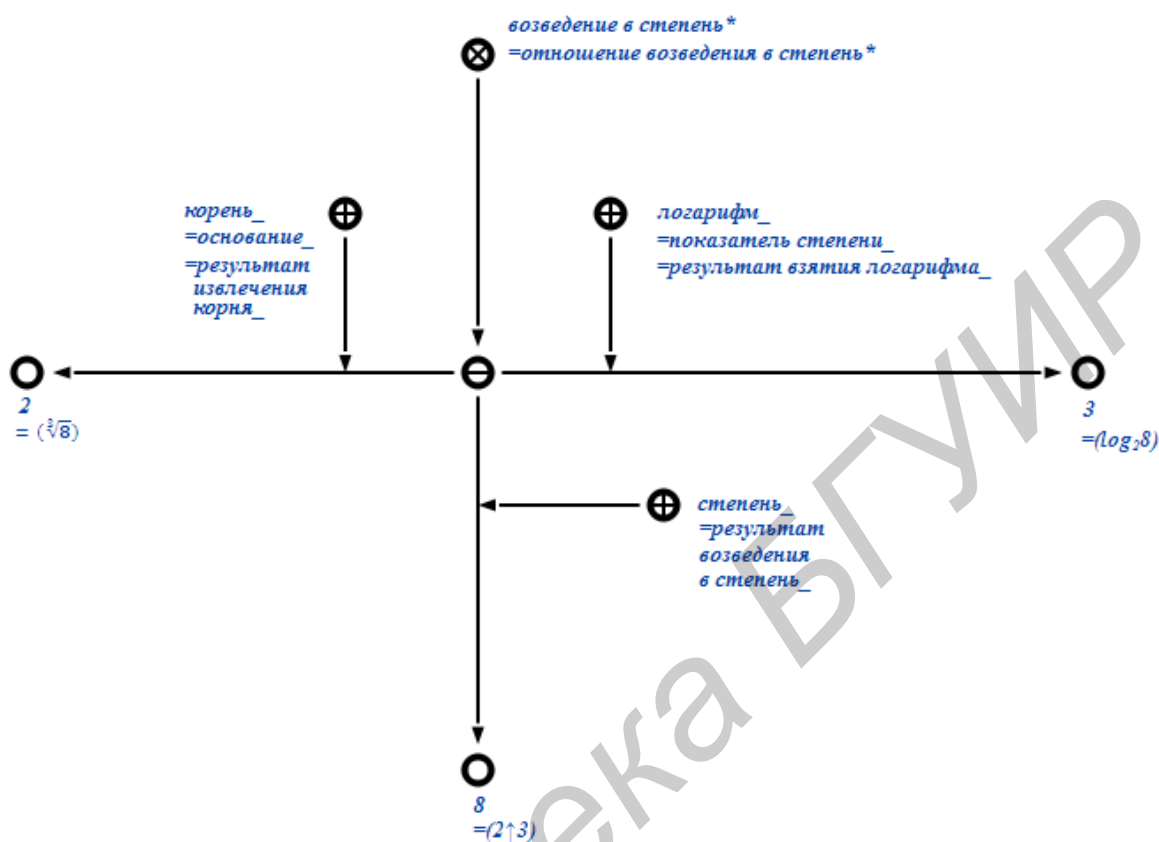


Рис. 1. Бинарное представление связки отношения **возведение в степень***

Алгебраическая операция o_i , соответствующая отношению r_i , трактуется как бинарное ориентированное отношение, являющееся подмножеством отношения принадлежности и связывающее связку отношения r_i с тем компонентом этой связки, который однозначно задается остальными компонентами указанной связки.

Любой текст традиционного линейного вида можно свести к алфавиту, состоящему из двух классов элементов текста (из двух классов символов). То есть любую строку символов можно представить (закодировать) в двоичном алфавите в виде битовой строки.

Можно ли аналогичным образом минимизировать алфавит семантических сетей?

В основе **минимизации алфавита** элементов лежит следующее свойство семантических сетей. Метки, входящие в состав алфавита элементов семантической сети, и ее ключевые узлы семантически эквивалентны в том смысле, что просто являются синтаксически различными способами выделения (задания) разных классов элементов семантической сети. При этом заметим, что в отличие от алфавита символов линейного текста все элементы алфавита (все метки) семантической сети, как и все ее ключевые узлы, имеют семантическую интерпретацию на описываемой предметной области. Таким образом, метки элементов семантической сети без какого-либо изменения семантики этой сети можно преобразовывать в ее ключевые узлы. При этом семантическая интерпретация каждого такого ключевого узла будет совпадать с семантикой соответствующей преобразованной метки.

Если мы имеем дело с корректно построенной семантической сетью и записанной в некотором языке семантических сетей (L_i), то перевод этой семантической сети на любой другой язык семантических сетей (L_j) не требует больших усилий, т. к. имеет место большое сходство синтаксического и семантического устройства всех языков семантических сетей.

Фактически эти языки отличаются своими алфавитами, элементы которых, как было показано выше, легко преобразуются в ключевые узлы семантических сетей.

Итак, число меток семантической сети можно уменьшать ценой расширения множества ее ключевых узлов. Вопрос в том, до какого предела это можно делать и как выглядит минимальный алфавит универсального языка семантических сетей. Универсальным языком семантических сетей будем называть такое множество семантических сетей, элементами которых являются семантические сети, представляющие любую информацию о любой описываемой предметной области.

Из сказанного ранее следует, это в состав минимального алфавита универсального языка семантических сетей, по крайней мере, должна входить метка *re*, обозначающая **отношение принадлежности**. Без этой метки невозможно описать связи ключевых узлов семантической сети с элементами обозначаемых ими классов, а также невозможно осуществить переход к бинарным семантическим сетям.

Текст, принадлежащий SC-коду (**sc-текст**, sc-структура, sc-конструкция), является семантической сетью частного вида, имеющей следующие особенности:

- все связи sc-текстов являются бинарными связками, которые будем называть **sc-коннекторами**. Неориентированные sc-коннекторы будем называть **sc-ребрами**, ориентированные – **sc-дугами**.
- множество меток элементов sc-текстов (алфавит sc-элементов, алфавит SC-кода) включает в себя:
 - метку **sc-узлов** (вершин sc-текстов);
 - метку **sc-ребер**;
 - метку **sc-дуг общего вида**;
 - метку **sc-дуг принадлежности**;
 - метку **sc-дуг основного вида**.
- множество отношений инцидентности элементов sc-текстов состоит из двух следующих отношений:
 - быть **компонентом sc-коннектора** (sc-ребра или sc-дуги);
 - быть **вторым компонентом sc-дуги**;
- множество ключевых узлов SC-кода (ключевых узлов sc-текстов) вместе с метками sc-элементов задает базовую семантическую типологию sc-элементов, т. е. базовую онтологию SC-кода.

К числу ключевых узлов SC-кода, определяющих разбиение множества sc-элементов по признаку **константности**, относятся:

- ключевой sc-узел, обозначающий множество всевозможных константных sc-элементов (**sc-констант**), являющихся обозначением некоторого конкретного фиксированного объекта;
- ключевой sc-узел, обозначающий множество всевозможных переменных sc-элементов (**sc-переменных**), являющихся обозначением некоторого произвольного, нефиксированного объекта из некоторого дополнительно уточняемого множества объектов. Используются sc-переменные в логических формулах (в т. ч. высказываниях), программах (обобщенных описаниях способов решения различных классов задач), формулировках вопросов.

К числу ключевых узлов SC-кода, определяющих разбиение множества sc-элементов по **структурному признаку**, относятся:

- ключевой sc-узел, обозначающий множество всевозможных **sc-коннекторов**, т. е. атомарных связок sc-элементов. Более детальное разбиение множества sc-коннекторов по структурному признаку осуществляется с помощью меток sc-элементов на **sc-ребра**, **sc-дуги общего вида**, **sc-дуги принадлежности**, **sc-дуги основного вида**;
- ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторое связующее множество sc-элементов. Указанные sc-элементы будем называть **связующими sc-узлами**. К числу ключевых узлов SC-кода, определяющих более детальную структурную типологию связующих sc-узлов, относятся:
 - ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторую неатомарную связь между sc-элементами, т. е. связь, не являющуюся sc-коннектором. Такие sc-узлы будем называть **неатомарными sc-связками**. Более детальная структурная типология неатомарных sc-связок задается

такими ключевыми узлами SC-кода, как быть **унарной sc-связкой**, быть **бинарной неатомарной sc-связкой**, быть **многокомпонентной sc-связкой**, быть **ориентированной неатомарной sc-связкой**, быть **неориентированной неатомарной sc-связкой**;

- ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторую структуру из sc-элементов. Такие структуры будем называть обозначениями sc-структур, или просто **sc-структурами**;
- ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторый класс sc-элементов. Такие sc-узлы будем называть обозначениями sc-понятий, или просто **sc-понятиями**. Более детальная структурная типология sc-понятий задается следующими ключевыми узлами SC-кода: быть **отношением** (классом однотипных связей), быть **бинарным отношением**, быть **унарным отношением**, быть **многоместным отношением**, быть **ориентированным отношением**, быть **неориентированным отношением**, быть **ролевым отношением** (т. е. отношением, которое является подмножеством отношения принадлежности), быть **классом структур**, быть **классом терминальных sc-узлов** (первичных sc-узлов, которые не являются обозначениями множеств sc-элементов);
- ключевой sc-узел, обозначающий множество всевозможных sc-узлов, каждый из которых обозначает некоторый объект, не являющийся множеством sc-элементов. Такие sc-узлы будем называть **терминальными sc-узлами** (первичными sc-узлами). Более детальная структурная типология терминальных sc-узлов задается следующими ключевыми узлами sc-кода:
 - быть **предметным sc-узлом**, каждый из которых обозначает некоторый реальный (материальный, физический) или вымышленный внешний объект некоторой предметной области;
 - быть **sc-ссылкой**, каждая из которых обозначает либо определенный файл, который можно просматривать или в котором закодирована в определенном формате некоторая внешняя, инородная для SC-кода информационная конструкция, либо некоторую компьютерную систему, с которой можно взаимодействовать;
 - быть **терминальным элементом шкалы или шаблона** (это sc-элементы, для которых трудно установить обозначаемые ими объекты, поскольку эти sc-узлы просто являются терминальными элементами каких-либо шкал, шаблонов, типовых структур, с которыми устанавливаются соответствия, а также сравниваются, сопоставляются различные объекты и структуры).

К числу ключевых узлов SC-кода, уточняющих семантику sc-дуг принадлежности, относятся:

- ключевые узлы SC-кода, определяющие разбиение множества sc-дуг принадлежности по признаку **позитивности**:
 - быть **sc-дугой позитивной принадлежности**;
 - быть **sc-дугой негативной принадлежности**;
 - быть **sc-дугой нечеткой принадлежности** (т. е. sc-дугой, позитивность или негативность которой в текущий момент не установлена);
- ключевые узлы SC-кода, определяющие разбиение множества sc-дуг принадлежности по признаку **стационарности**:
 - быть **sc-дугой стационарной принадлежности**, семантический тип которой является постоянным, не изменяющимся во времени;
 - быть **sc-дугой нестационарной принадлежности**, семантический тип которой изменяется во времени.

Заметим, что **sc-дуги основного вида**, которые выделяются с помощью соответствующей метки, семантически трактуются как **sc-дуги позитивной стационарной принадлежности**.

Перечислим основные особенности и достоинства SC-кода.

Унифицированные семантические сети (sc-тексты) – это абстрактная семантическая модель знаний, являющаяся инвариантом различных способов представления и кодирования этих же знаний (в том числе и самих семантических сетей). Наличие такого инварианта необходимо для решения проблемы интеграции самых различных видов

знаний. На основе унифицированных семантических сетей можно строить семантические модели различных компьютерных систем и решить проблему интеграции таких систем.

SC-код является абстрактным языком в том смысле, что способ изображения (материализации) его текстов не уточняется. Следовательно, можно разрабатывать различные графические уточнения SC-кода (например SCg-код), различные варианты изображения sc-текстов в виде строк символов (например SCs-код), различные варианты машинного представления sc-конструкций в адресной памяти традиционных компьютеров, а также в специальной структурно перестраиваемой ассоциативной памяти будущих компьютеров, ориентированных на обработку баз знаний.

Все sc-элементы, кроме терминальных sc-узлов, являются обозначениями множеств, состоящих из sc-элементов (множеств sc-элементов). Такие sc-элементы будем называть **вторичными sc-элементами**. Из этого следует то, что SC-код имеет базовую теоретико-множественную семантическую интерпретацию.

SC-код представляет собой достаточно простой компьютерный код семантических сетей, который является не «инородным» представлением семантических сетей, а их представлением тоже в виде семантических сетей, но максимально простого вида – с минимальным алфавитом и с бинарными связками.

SC-код ориентирован на представление информации в компьютерной памяти и может рассматриваться как основа модели структурно перестраиваемой ассоциативной памяти будущих компьютеров, ориентированных на обработку семантических сетей. То есть SC-код можно рассматривать как универсальную основу машинного кодирования знаний в памяти будущих компьютеров, ориентированных на обработку семантических сетей. В такой памяти биты и байты уступят место sc-дугам, sc-ребрам и sc-узлам.

На базе SC-кода можно создавать целое семейство совместимых специализированных языков, ориентированных на представление самых разных видов знаний (логических формул и высказываний, программ, вопросов, поведенческих целей, различных видов моделей динамических систем и т. п.), таким образом, чтобы тексты всех этих специализированных языков полностью соответствовали SC-коду (т. е. были sc-текстами). Такие специализированные языки, общим носителем которых является SC-код, будем называть **sc-языками**. Каждый sc-язык определяется своим расширением множества ключевых узлов SC-кода. При этом алфавит всех sc-языков совпадает с алфавитом SC-кода.

SC-код представляет собой ядро **универсального открытого языка семантических сетей**, являющегося результатом интеграции всевозможных языков семантических сетей, построенных на основе SC-кода, и задаваемого:

- фиксированным алфавитом (алфавитом SC-кода);
- постоянно расширяемым (открытым) семейством ключевых узлов, в состав которого входят все ключевые узлы всех интегрируемых языков.

SC-код представляет собой **единство языка и метаязыка**. Так, например, в виде sc-конструкций можно описать синтаксис, семантику и онтологию SC-кода. С формальной точки зрения SC-код можно трактовать как метаязык базовой семантической спецификации sc-элементов с помощью специального набора ключевых узлов SC-кода.

Единство языка и метаязыка в SC-коде проявляется в том числе и на самом низком уровне – на уровне sc-дуг принадлежности $\langle si, ei \rangle$, в которых сам sc-элемент ei , а не обозначаемый им объект, является элементом множества, обозначаемого sc-узлом si .

SC-код позволяет описать структуру любой информационной конструкции, не принадлежащей SC-коду, на любом этапе синтаксического и семантического анализа. В частности, первичную синтаксическую структуру любой информационной конструкции можно представить в виде изоморфной sc-конструкции. Следовательно, SC-код может быть использован в качестве метаязыка для описания любого внешнего языка, т. е. языка, тексты которого не являются sc-конструкциями.

SC-код хорошо приспособлен к использованию в условиях так называемых **не-факторов**: нестационарности, неточности, противоречивости, неактуальности знаний, а также неполноты знаний (нечеткости, несформированности множеств, несформированности внешних информационных конструкций) [40].

Информационные конструкции SC-кода (sc-конструкции) легко **визуализируются**.

Основные положения SC-кода. В качестве языка семантических сетей, удовлетворяющего указанным выше требованиям, подходит SC-код (Semantic Computer Code). Основной особенностью SC-кода является то, что в нем минимизирован не только алфавит (набор) меток, но и набор ключевых узлов, используемых конструкциями (текстами) SC-кода. В этом смысле SC-код следует рассматривать не как универсальный язык представления знаний, а как универсальный носитель (базис, основу) такого языка. Очевидно, что указанный универсальный язык представления знаний, построенный на основе SC-кода, должен быть открытым языком, поскольку могут появляться новые виды знаний, и должен рассматриваться как результат интеграции открытого семейства совместимых языков, каждый из которых ориентируется на представление соответствующего вида знаний, имеет алфавит, совпадающий с алфавитом SC-кода, и набор ключевых узлов, являющийся расширением (надмножеством) набора ключевых узлов SC-кода. Указанные языки, построенные на основе SC-кода, будем называть *sc-языками*.

Семантические сети, представленные в SC-коде, будем называть *sc-конструкциями*. Вершины, связки, метки и ключевые узлы *sc-конструкций* будем называть *sc-элементами*. Вершины *sc-конструкций* будем называть *sc-узлами*. Связки *sc-конструкций* будем называть *sc-коннекторами*.

Алфавит SC-кода

Алфавит SC-кода представляет собой набор из следующих пяти меток:

- *Метка sc-узла*
= *Метка вершины sc-конструкции*
- *Метка sc-ребра*
= *Метка бинарной неориентированной связки либо бинарной ориентированной связки неизвестной направленности*
- *Метка sc-дуги общего вида*
- *Метка sc-дуги основного вида*
= *Метка ориентированной пары sc-элементов, принадлежащей отношению принадлежности (re)*
- *Метка sc-ссылки*
= *Метка sc-элемента, обозначающего либо файл интеллектуальной системы (т. е. информационную конструкцию, закодированную иным от SC-кода способом), либо компьютерную систему (которую, например, можно инициализировать).*

Общие принципы семантической интерпретации элементов семантических сетей, принадлежащих SC-коду

От отношения принадлежности, каждая связка которого связывает знак множества с обозначением одного из элементов этого множества, мы переходим к семантически нормализованному отношению принадлежности, каждая связка которого связывает знак множества с *sc-элементом*, который сам (а не обозначаемый им объект) является элементом указанного множества.

Множество, элементами которого являются *sc-элементы*, будем называть семантически нормализованным множеством. Любое множество можно привести к семантически нормализованному виду.

SC-код оперирует только семантически нормализованными множествами. Это дает возможность четко отличать первичные (терминальные) *sc-элементы*, являющиеся обозначениями внешних объектов (не являющихся множествами) от вторичных *sc-элементов*, служащих обозначениями множеств, все элементы которых могут непосредственно входить в состав *sc-конструкций*. К таким вторичным *sc-элементам* относятся и связки семантически нормализованной принадлежности. Указанная семантическая нормализация множеств предоставляет неограниченные возможности для перехода от *sc-конструкций* к *sc-метаконструкциям*, это является принципиально важным для семантического представления различных метазнаний, предметных метаобластей, а точнее, для описания связи между знаниями и соответствующими описывающими их метазнаниями.

Ключевые узлы SC-кода

Ключевые узлы SC-кода отражают универсальную базовую семантическую типологию sc-элементов и, соответственно, обозначаемых ими объектов самых различных предметных областей, в том числе и различных метаобластей, определяющих семантику различных видов знаний.

Ключевые узлы SC-кода, соответствующие его алфавиту, т. е. узлы, синонимичные всем используемым меткам (всем синтаксически выделяемым классам sc-элементов).

В число ключевых узлов SC-кода, прежде всего, входят узлы, обозначающие все элементы алфавита SC-кода. Это необходимо для того, чтобы иметь возможность описывать свойства синтаксически выделяемых классов sc-элементов. Для того, чтобы явно выделить группу таких ключевых узлов, введем ключевой узел *Алфавит SC-кода*.

Алфавит SC-кода

Э *Метка sc-узла*

Э *Метка sc-ребра*

Э *Метка sc-дуги общего вида*

Э *Метка sc-дуги основного вида*

Э *Метка sc-ссылки*

Алфавит SC-кода является семейством попарно непересекающихся множеств всех sc-элементов. Множество sc-элементов разбивается на подклассы не только в соответствии с алфавитом sc-элементов, но и по целому ряду других признаков, которые рассмотрены ниже.

Ключевые узлы SC-кода, определяющие разбиение множества элементов, принадлежащих текстам SC-кода, на константные и переменные

Каждый sc-элемент является обозначением некоторого объекта. При этом речь может идти либо о вполне конкретном объекте, либо о произвольном объекте, взятом из некоторого множества объектов. В первом случае мы имеем дело с константными sc-элементами (*sc-константами*), а во втором – с переменными sc-элементами (*sc-переменными*). Каждой sc-переменной однозначно ставится в соответствие *область значений** (множество sc-элементов, которое «пробегает» эта sc-переменная). Область значений sc-переменной может быть не известна, но всегда должна быть известна *область возможных значений** этой переменной, которая является некоторым надмножеством области ее значений.

Итак, по признаку константности множество sc-элементов разбивается на класс *sc-констант* и класс *sc-переменных*.

sc-элемент

– Р а з б и е н и е по признаку константности:

- *sc-константа*
- *sc-переменная*

sc-константа

= *Класс всевозможных sc-констант*

= *Понятие sc-константы*

= *константный sc-элемент*

= *sc-элемент, являющийся обозначением конкретного объекта /** в частности конкретного множества sc-элементов */

= *обозначение конкретного объекта*

= *знак конкретного множества sc-элементов или внешнего объекта*

= *знак множества sc-элементов или внешнего объекта /** слово «конкретного» можно опустить, т. к. слово «знак» предполагает обозначение именно конкретного объекта */

= *множество sc-элементов или внешний объект /** слово «знак» можно опустить, т. к. в случае отсутствия слова «обозначение» по умолчанию предполагается слово «знак» */

sc-переменная

- = Класс всевозможных *sc*-переменных
- = Понятие *sc*-переменной
- = переменный *sc*-элемент
- = обозначение произвольного объекта
- = обозначение произвольного множества *sc*-элементов или внешнего объекта
- = *sc*-переменная общего вида
- = *sc*-элемент, представляющий собой произвольный *sc*-элемент из некоторого множества *sc*-элементов
 - / * такое множество *sc*-элементов будем называть областью возможных значений* соответствующей *sc*-переменной */
- = *sc*-элемент, являющийся представителем произвольного *sc*-элемента из некоторого множества *sc*-элементов
 - / * следует отличать друг от друга такие понятия, как быть представителем произвольного *sc*-элемента, обозначать множество *sc*-элементов, обозначать терминальное (первичное) множество-синглетон, быть элементом (принадлежать, входить в состав в качестве элемента) */
- = Множество всех тех и только тех *sc*-элементов, каждый из которых является переменным *sc*-элементом

При идентификации (именовании) *sc*-элементов будем отличать термин «обозначение» от термина «знак». Это отличие заключается в следующем:

- *sc*-элемент, названный *обозначением* некоторого объекта, может быть как константным, так и переменным, т. е. может обозначать как конкретный, так и произвольный объект;
- *sc*-элемент, названный *знаком* некоторого объекта, может быть только константным.

Для более детального описания семантики *sc*-переменных, как уже было отмечено, вводятся отношения, связывающие *sc*-переменные с их значениями, областями значений, областями возможных значений.

значение*

- = значение *sc*-элемента*
- = быть значением *sc*-элемента*
- = одно из значений заданного *sc*-элемента
- = Отношение, связывающее *sc*-элемент с одним из его значений
- П р и м е ч а н и е .
sc-переменная в общем случае может иметь несколько значений, но каждая *sc*-константа всегда имеет единственное значение, каковым является она сама.

область значений*

- = Отношение, связывающее *sc*-переменную с областью ее значений
- = быть областью значений заданной *sc*-переменной / * в частности, это может быть множеством корней некоторого уравнения */
- = множество *sc*-элементов, которое пробегает заданная *sc*-переменная

область возможных значений*

- = Отношение, связывающее *sc*-переменную с областью ее возможных значений
- = быть областью возможных значений заданной *sc*-переменной
- П р и м е ч а н и е . Для каждой *sc*-переменной область ее возможных значений является нестрогим надмножеством области ее значений.

Тип *sc*-переменной определяется характером области ее возможных значений. В связи с этим можно выделить следующие классы *sc*-переменных.

***sc*-переменная**

- Р а з б и е н и е по логико-семантическому признаку:
 - *первичная sc-переменная*
 - = *sc*-переменная первого логико-семантического уровня
 - = *sc*-переменная первого логического уровня
 - = *sc*-переменная первого уровня
 - = *sc*-переменная, значениями которой являются только *sc*-константы

- = *sc-переменная, область возможных значений которой состоит только из sc-констант*
- = *Класс первичных sc-переменных*
- *sc-метапеременная*
 - = *непервичная sc-переменная*
 - = *sc-переменная, не являющаяся первичной*
 - = *sc-переменная, в область возможных значений которой входит по крайней мере одна sc-переменная*
- **Р а з б и е н и е** по признаку фиксированности логико-семантического уровня sc-метапеременной:
 - *sc-метапеременная фиксированного уровня*
 - = *sc-метапеременная, значениями которой являются sc-переменные одного логико-семантического уровня*
 - = *sc-метапеременная, область возможных значений которой состоит из sc-переменных одного логического уровня*
 - **Р а з б и е н и е** по логико-семантическому уровню:
 - *sc-метапеременная второго уровня*
 - = *sc-метапеременная второго логико-семантического уровня*
 - = *вторичная sc-метапеременная*
 - = *sc-метапеременная, область возможных значений которой состоит только из первичных sc-переменных*
 - = *Класс sc-переменных второго уровня*
 - *sc-метапеременная третьего уровня*
 - = *sc-метапеременная, область возможных значений которой состоит только из sc-переменных второго уровня*
 - *sc-метапеременная четвертого уровня*
 - */* и так далее */*
 - *sc-метапеременная нефиксированного уровня*
 - = *sc-метапеременная смешанного уровня*
 - = *sc-метапеременная, не относящаяся к классу sc-метапеременных фиксированного уровня*
 - = *sc-метапеременная, область возможных значений которой может включать в себя и sc-константы, и первичные sc-переменные, и sc-переменные второго уровня, и sc-метапеременные других логических уровней, и sc-метапеременные нефиксированного уровня*
 - **Р а з б и е н и е** по признаку универсальности:
 - *универсальная sc-метапеременная :*
 - = *sc-метапеременная, область возможных значений которой является множество всевозможных sc-элементов /*в том числе и сама эта метапеременная*/*
 - *неуниверсальная sc-метапеременная нефиксированного уровня.*

Подчеркнем то, что связь между sc-переменной и областью ее возможных значений, связь между sc-переменной и некоторым ее конкретным значением может быть описана явно средствами SC-кода.

Ключевые узлы SC-кода, определяющие структурную типологию элементов, которые принадлежат текстам SC-кода:

sc-элемент

- **Р а з б и е н и е** по структурному признаку:
 - *обозначение внешнего объекта*
 - *обозначение материального объекта*
 - *обозначение внешнего абстрактного объекта*
/* т. е. абстрактного объекта, не являющегося множеством */
 - *обозначение файла sc-системы*
/* т. е. компьютерной системы, основанной на SC-коде */
 - *обозначение компьютерной системы, которую можно инициализировать*
/* такой системой может быть как sc-система, так и система, построенная по

- другим технологиям */
- **обозначение множества sc-элементов**
 - **обозначение структуры sc-элементов**
 - **обозначение понятия**
 - = **обозначение класса sc-элементов, семантически однотипных в заданном смысле**
 - **обозначение отношения**
 - = **обозначение класса связок sc-элементов, семантически однотипных в заданном смысле**
 - = **обозначение относительного понятия**
 - **обозначение класса структур, не являющихся связками**
 - **обозначение класса объектов, не являющихся однотипными структурами**
 - = **обозначение абсолютного понятия**

обозначение структуры sc-элементов

- = **обозначение множества sc-элементов, обладающего определенной целостностью**
- **Р а з б и е н и е** по уровню сложности структур:
 - **обозначение связки sc-элементов**
 - **sc-коннектор**
 - = **обозначение атомарной связки sc-элементов**
 - = **обозначение элементарной связки sc-элементов**
 - **обозначение неатомарной связки sc-элементов**
 - **обозначение тривиальной структуры sc-элементов**
 - = **обозначение одноуровневой структуры, не являющейся связкой**
 - = **обозначение множества sc-элементов, которое не является связкой и для которого инцидентность между его элементами не является существенной и, следовательно, не учитывается**
 - **П р и м е р а м и** тривиальных структур являются геометрические фигуры, каждая из которых трактуется в SC-коде как множество знаков геометрических точек
 - **обозначение двухуровневой структуры sc-элементов**
 - = **обозначение реляционной структуры sc-элементов**
 - = **обозначение графовой структуры sc-элементов**
 - **П о я с н е н и е** . Это множество sc-элементов, в состав которого входят связки, связывающие элементы этого множества. При этом инцидентность указанных связок является существенной и, следовательно, учитывается
 - **обозначение иерархической структуры sc-элементов**
 - = **обозначение многоуровневой структуры sc-элементов**
 - **П р и м е ч а н и е** . Это рекурсивное обобщение понятия двухуровневой структуры sc-элементов.

обозначение отношения

- **Р а з б и е н и е** по признаку ориентированности:
 - **обозначение ориентированного отношения**
 - **обозначение неориентированного отношения**
- **Р а з б и е н и е** по признаку равенства числа компонентов связок:
 - **обозначение отношения, связки которого могут иметь разную арность**
 - **обозначение отношения, все связки которого имеют одинаковую арность**
- **Р а з б и е н и е** по признаку арности:
 - **обозначение бинарного отношения**
 - **обозначение тернарного отношения**
 - **обозначение четырехарного отношения**
 - **/*и т. д.*/**
- **Р а з б и е н и е** по признаку существования связок, компонентами которых являются другие связки этого же отношения:
 - **обозначение отношения, в область определения которого не входят связки этого же отношения**
 - **обозначение отношения, в область определения которого входят связки этого же отношения**
- **Р а з б и е н и е** по признаку существования связок, одним из компонентов которых

является знак этого отношения:

- обозначение отношения, в область определения которого не входит знак этого отношения
 - обозначение отношения, в область определения которого входит знак этого отношения
- Р а з б и е н и е , основанное на теоретико-множественной связи с отношением принадлежности:
- обозначение отношения, являющееся подмножеством отношения принадлежности
 \subset обозначение бинарного ориентированного отношения
/* это результат наследования свойств надмножества */
 - обозначение отношения, строго пересекающегося с отношением принадлежности
 - обозначение отношения, непересекающегося с отношением принадлежности.

Ключевые узлы SC-кода, определяющие семантическую типологию связок, описывающих принадлежность или непринадлежность:

обозначение принадлежности-непринадлежности

= обозначение принадлежности или непринадлежности

= обозначение связки нечеткой принадлежности

- Р а з б и е н и е по признаку позитивности:
- обозначение принадлежности
= обозначение связки принадлежности
 - обозначение непринадлежности
= обозначение связки непринадлежности
- Р а з б и е н и е по признаку стационарности:
- обозначение стационарной принадлежности-непринадлежности
= обозначение связки стационарной нечеткой принадлежности
 - обозначение нестационарной принадлежности-непринадлежности
= обозначение нестационарной принадлежности или непринадлежности
= обозначение связки нестационарной нечеткой принадлежности
- Р а з б и е н и е по сочетанию признака позитивности и признака стационарности:
- обозначение стационарной принадлежности
= обозначение связки стационарной принадлежности
 - обозначение стационарной непринадлежности
= обозначение связки стационарной непринадлежности
 - обозначение нестационарной принадлежности
= обозначение связки нестационарной принадлежности
 - обозначение нестационарной непринадлежности
= обозначение связки нестационарной непринадлежности.

6. Унификация структуризации баз знаний

Будем трактовать семантическую структуру **базы знаний** интеллектуальной системы как отражение иерархической системы взаимосвязанных друг с другом **предметных областей**, представляемых в базе знаний. Это предполагает:

- 1) уточнение понятия предметной области;
- 2) разработку языковых средств описания структуры предметных областей с помощью унифицированных семантических сетей;
- 3) разработку языковых средств описания типологии предметных областей и различных видов связей между ними.

Уточнение понятия предметной области

Семантика базы знаний интеллектуальной системы – это соотношение между базой знаний и описываемой ею предметной областью. Поэтому для формального рассмотрения семантики баз знаний необходимо иметь формальное уточнение понятия предметной области.

Предметная область (S) – это математическая структура, которая задается следующим набором ее компонентов $\langle A, H, Q, R, O, B, P \rangle$,

где A – непустое множество, которое будем называть **носителем** (основным множеством, множеством первичных элементов, множеством рассматриваемых сущностей) предметной области (S);

H – **шкала множеств**, построенная над основным множеством (A) предметной области (S).

Шкала множеств, построенная над произвольным множеством x , обозначается $H(x)$ и определяется рекурсивно:

1) $H(x) \supset x$

2) для любого набора элементов из множества $H(x)$

$$h_1 \in H(x), \dots, h_n \in H(x)$$

имеет место

$$\{h_1, \dots, h_n\} \in H(x)$$

$$\langle h_1, \dots, h_n \rangle \in H(x)$$

Из определения шкалы множеств следует, что $A \subset H$.

Q – **семейство классов**, каждый из которых имеет непустое пересечение со множеством (H). При этом большинство из этих классов имеют непустое пересечение со множеством (A).

R – **семейство отношений** разной арности, заданных в общем случае на некотором надмножестве множества (H), а чаще всего – множества (A).

Из определения шкалы множеств следует, что каждое из указанных отношений имеет непустое пересечение со множеством (H), т. е. выделяет из множества (H) некоторое подмножество. В этом смысле множество (R) аналогично множеству (Q).

O – **семейство алгебраических операций** разной арности, заданных на некотором надмножестве множества (H), а чаще всего – множества (A).

B – специально выделенный элемент семейства классов (Q) ($B \in Q$), который является подмножеством в общем случае шкалы множеств, построенной над основным множеством предметной области ($B \subset H$) и который будем называть **классом исследуемых объектов** предметной области (S) (классом объектов исследования).

Заметим, что чаще всего множество B является подмножеством основного множества A предметной области S .

P – специально выделенное семейство отношений, алгебраических операций и классов первичных объектов предметной области (S) ($P \subseteq (Q \cup R \cup O)$). Такое специально выделенное семейство будем называть **предметом исследования** предметной области (S).

Введем ряд дополнительных понятий, связанных с понятием предметной области.

Множеством элементов предметной области будем называть множество ($H \cup Q \cup R \cup O$). Очевидно, что к элементам предметной области (S) относятся также:

- все ее первичные элементы, т. к. $A \subset H$;
- класс ее объектов исследования, т. к. $B \in Q$;
- все ее объекты исследования, т. к. $B \subset H$;
- все элементы ее предмета исследования, т. к. $P \subseteq (Q \cup R \cup O)$.

Сигнатурой предметной области (S) (множеством ее сигнатурных элементов) будем называть множество ($Q \cup R \cup O$). С семантической точки зрения сигнатура предметной области – это множество всех понятий, используемых в этой предметной области. Очевидно, что к числу сигнатурных элементов предметной области S относятся также:

- класс ее объектов исследования, т. к. $B \in Q$;
- все элементы ее предмета исследования, т. к. $P \subseteq (Q \cup R \cup O)$.

Соответственно этому шкалу множеств (H) над основным множеством (A) заданной предметной области будем называть **множеством несигнатурных элементов** этой предметной области.

Множеством вторичных элементов (связующих) предметной области (S) будем называть множество $H \setminus A$. Речь идет о несигнатурных элементах предметной области,

которые не являются ее основными (первичными) элементами. Вторичные элементы предметной области связывают всевозможные элементы заданной предметной области с помощью отношений и алгебраических операций, входящих в сигнатуру этой предметной области.

Семейством вспомогательных сигнатурных элементов предметной области S будем называть множество $((Q \cup R \cup O) \setminus P)$. То есть речь идет о сигнатурных элементах, не входящих в состав предмета исследования (P) предметной области (S). Соответственно этому сигнатурные элементы, входящие в состав предмета исследования, будем также называть основными сигнатурными элементами предметной области.

Множеством вспомогательных первичных элементов предметной области (S) будем называть множество $(A \setminus B)$. Речь идет о первичных элементах предметной области, не являющихся ее объектами исследования, но связанных с этими исследуемыми объектами различными отношениями и алгебраическими операциями в рамках заданной предметной области.

Введенное понятие предметной области представляет собой модификацию понятия алгебраической системы, ориентированную на рассмотрение семантики баз знаний.

Перечислим некоторые формальные отличия понятия предметной области от классического понятия алгебраической системы.

- Явно вводится класс объектов исследования (B) и предмет исследования (P) как специально введенное подмножество сигнатуры предметной области. Фактически, указанные множества являются семантической спецификацией предметной области. Множество (B) выделяет класс объектов, на которых концентрируется внимание, а множество (P) задает ракурс исследования этих объектов (какие рассматриваются связи этих объектов как между собой, так и с другими второстепенными объектами).
- Отношения (R) и алгебраические операции (O) предметной области задаются не на основном множестве (носителе) (A) предметной области и даже не на множестве несигнатурных элементов (H) этой предметной области, а на некотором надмножестве указанного множества (H). Это означает, что отношение $r_i (r_i \in R)$ или алгебраическая операция $o_i (o_i \in O)$ могут быть использованы не только в заданной предметной области (S), но и в другой предметной области (S'), множество несигнатурных элементов (H') которой может существенно отличаться от множества (H).
- Выделяемые классы элементов (Q) предметной области являются подмножествами не основного множества (A) предметной области и даже не множества несигнатурных элементов (H) этой предметной области, а некоторого надмножества указанного множества H . Это означает, что класс $q_i (q_i \in Q)$ может быть использован не только в заданной предметной области (S), но и в другой предметной области (S'), множество несигнатурных элементов (H') которой может отличаться от множества (H).
- Понятие алгебраической операции и понятие отношения предметной области являются обобщениями соответствующих классических понятий, поскольку они задаются не на основном множестве предметной области, а на *шкале множеств*, построенной на базе этого основного множества. Это означает, что связки алгебраических операций и отношений могут связывать между собой не только элементы основного множества, но и любые множества, состоящие из этих элементов, элементов этих элементов и т. д. В частности, компонентами одних связок могут быть другие связки, принадлежащие тому же или любому другому отношению или алгебраической операции.

Предметная область позволяет рассматривать исследуемые объекты на разных уровнях детализации:

- детализировать классификацию класса исследуемых объектов (B). Эту классификацию можно проводить по самым различным признакам и иметь сколь угодно большое число уровней ее детализации;
- осуществлять декомпозицию самих исследуемых объектов, т. е. переходить к рассмотрению структур взаимосвязанных частей этих объектов. При этом число иерархических уровней такой декомпозиции может быть сколь угодно большим;
- рассматривать связи исследуемых объектов с новыми и новыми вспомогательными объектами, не входящими в класс исследуемых объектов;

- расширять сигнатуру предметной области и вводить новые и новые отношения и алгебраические операции, рассматривающие новые связи исследуемых объектов как между собой, так и со вспомогательными объектами.

Указанную детализацию рассмотрения исследуемых объектов можно осуществлять как в рамках исходной (заданной) предметной области, расширяя ее в соответствующих направлениях, а можно переходить к системе самостоятельных, но связанных между собой предметных областей.

Первым и важнейшим этапом проектирования базы знаний является уточнение структуры описываемой предметной области или нескольких взаимосвязанных предметных областей. Уточнение такой структуры – это прежде всего уточнение класса исследуемых объектов, предмета исследования, всего семейства сигнатурных элементов. При этом для заданного класса исследуемых объектов и заданного предмета исследования можно построить более качественную и менее качественную предметную область.

Критерий качества предметных областей – это отдельная большая тема. Отметим один из таких критериев – критерий минимизации числа вводимых понятий (сигнатурных элементов). При выборе минимального числа вводимых классов исследуемых объектов и вводимых отношений необходимо помнить следующее:

- Любые два объекта имеют сходство. То есть для любой пары объектов существует понятие, которому принадлежат оба этих объекта. Но вопрос в том, нужно ли явно вводить все такие понятия.
- Все в мире взаимосвязано. То есть для любой n -ки объектов существует отношение, которому эта n -ка принадлежит. Но вопрос в том, нужно ли явно вводить все такие отношения. Лучше минимизировать число таких отношений, а n -ку у неявно вводимого отношения трактовать как композицию (цепочку) n -ок, принадлежащую явно введенным отношениям.

Нетрудно заметить, что решение рассматриваемых проблем имеет весьма сложный, творческий и во многом субъективный характер.

Отношения, заданные на множестве предметных областей

Переходя к рассмотрению отношений, заданных на множестве предметных областей, мы фактически переходим к некоторой предметной метаобласти, объектами исследования которой являются всевозможные предметные области (в том числе и сама эта предметная метаобласть).

Обобщая понятия гомоморфизма и изоморфизма алгебраических систем, можно говорить о гомоморфизме и изоморфизме предметных областей, что дает хорошую основу для выявления глубоких нетривиальных аналогий предметных областей.

Различные предметные области могут пересекаться. То есть элементы одной предметной области могут быть также и элементами другой предметной области. При этом возможны самые различные варианты такого пересечения: строгое пересечение, строгое включение. Общие элементы пересекающихся предметных областей могут в рамках этих областей выполнять как одинаковые, так и разные роли. Например, первичные элементы одной предметной области могут входить в состав другой в качестве вторичных или сигнатурных элементов. Объекты исследования одной предметной области могут входить в состав другой в качестве вспомогательных элементов. С семантической точки зрения предметные области могут быть связаны между собой следующим образом:

- Множество всех сигнатурных элементов одной предметной области (S_1) может быть множеством главных (исследуемых) объектов другой предметной области (S_2). Это означает, что предметная область S_2 является онтологией по отношению к предметной области S_1 . В зависимости от сигнатуры предметной области S_2 она может быть, например, либо теоретико-множественной либо логической онтологией предметной области S_1 .
- Некоторый класс вспомогательных объектов области S_1 может быть классом главных исследуемых объектов в области S_2 .
- Множество главных (исследуемых) объектов области S_1 может совпадать со множеством главных объектов области S_2 . Но при совпадении объектов исследования в этих областях их предметы исследования, определяемые наборами сигнатурных элементов, могут быть разными.

- Предметные области могут иметь совпадающие предметы исследования, но разные классы объектов исследования.
- Множество главных (исследуемых) объектов предметной области S_1 может быть надмножеством множества главных объектов области S_2 . Это может означать, что область S_2 ориентирована на более детальное, чем в области S_1 , рассмотрение соответствующего класса объектов исследования.

Будем говорить, что предметная область $S_j = \langle A_j, H_j, Q_j, R_j, O_j, B_j, P_j \rangle$ **включается** (является подобластью, является частной) по отношению к предметной области $S_i = \langle A_i, H_i, Q_i, R_i, O_i, B_i, P_i \rangle$ в том и только в том случае, если $A_j \subseteq A_i; H_j \subseteq H_i; Q_j \subseteq Q_i; R_j \subseteq R_i; O_j \subseteq O_i; B_j \subseteq B_i; P_j \subseteq P_i$.

Будем говорить, что предметная область $S_i = \langle A_i, H_i, Q_i, R_i, O_i, B_i, P_i \rangle$ является результатом **объединения** семейства предметных областей $\{S_1, \dots, S_n\}$, где

$S_{i1} = \langle A_{i1}, H_{i1}, Q_{i1}, R_{i1}, O_{i1}, B_{i1}, P_{i1} \rangle, \dots, S_{in} = \langle A_{in}, H_{in}, Q_{in}, R_{in}, O_{in}, B_{in}, P_{in} \rangle$ в том и только в том случае, если $A_i = \bigcup_{j=1}^n A_{ij}; H_i = \bigcup_{j=1}^n H_{ij}; Q_i = \bigcup_{j=1}^n Q_{ij}; R_i = \bigcup_{j=1}^n R_{ij}; O_i = \bigcup_{j=1}^n O_{ij}; B_i = \bigcup_{j=1}^n B_{ij}; P_i = \bigcup_{j=1}^n P_{ij}$.

Будем говорить, что семейство предметных областей $\{S_1, \dots, S_n\}$, где $S_{i1} = \langle A_{i1}, H_{i1}, Q_{i1}, R_{i1}, O_{i1}, B_{i1}, P_{i1} \rangle, \dots, S_{in} = \langle A_{in}, H_{in}, Q_{in}, R_{in}, O_{in}, B_{in}, P_{in} \rangle$ является **декомпозицией** (разбиением) предметной области $S_i = \langle A_i, H_i, Q_i, R_i, O_i, B_i, P_i \rangle$ в том и только в том случае, если:

- предметная область S_i есть объединение семейства предметных областей $\{S_1, \dots, S_n\}$;
- множество $(H_i \setminus A_i)$ разбивается на семейство подмножеств $\{H_{i1} \setminus A_{i1}, \dots, H_{in} \setminus A_{in}\}$. Это означает, что каждая связь рассматриваемых объектов представлена только в одной из семейства областей, на которые осуществляется декомпозиция.

Будем говорить, что предметная область S_j является **теоретико-множественной онтологией** по отношению к предметной области S_i , т. е. $\langle S_i, S_j \rangle \in$ *теоретико-множественная онтология** в том и только в том случае, если:

- множеством исследуемых объектов предметной области S_j является семейство сигнатурных элементов предметной области S_i ;
- в состав предмета исследования предметной области S_j входят различные теоретико-множественные понятия: *включение**, *строгое включение**, *семейство пересекающихся множеств**, *семейство непересекающихся множеств**, *семейство попарно непересекающихся множеств**, *пересечение**, *объединение**, *разбиение**, *разность множеств**, *булеан**, *бинарное отношение*, *небинарное отношение*, *ориентированное отношение*, *неориентированное отношение*, *отношение из связок одинаковой арности*, *отношение из связок разной арности*, *область определения** (отношения), *домен** (заданного отношения для заданного атрибута), *схема отношения** и т. д.

Будем говорить, что предметная область S_j является **логическим описанием** предметной области S_i , т. е. $\langle S_i, S_j \rangle \in$ *логическое описание** в том и только в том случае, если:

- множеством основных исследуемых объектов предметной области S_j является множество логических формул и, в частности, высказываний, интерпретируемых на предметной области S_i ;
- носитель (основное множество) предметной области S_j является надмножеством носителя предметной области S_i . В рамках предметной области S_j элементы носителя предметной области S_i трактуются как предметные константы, а остальные элементы носителя области S_j – как обозначения произвольных объектов, т. е. переменные, значениями которых являются элементы носителя области S_i ;
- аналогично этому шкала множеств, построенная на носителе области S_j , включает в себя шкалу множеств, построенную на носителе области S_i , которая в рамках области S_j трактуется как множество констант, а также включает в себя переменные, значениями которых являются указанные константы;

- в состав предмета исследования области S_j входят различные логические понятия, обозначающие различные классы логических формул, а также различные отношения, заданные на множестве логических формул. К таким понятиям относятся:
 - *атомарная логическая формула* (некоторое множество связанных между собой констант и переменных);
 - *неатомарная логическая формула*, трактуемая как связка, компонентами которой являются логические формулы, входящие в состав этой неатомарной логической формулы;
 - *конъюнктивная логическая формула* (конъюнкция заданного множества логических формул);
 - *дизъюнктивная логическая формула* (дизъюнкция заданного множества логических формул);
 - *имплицативная логическая формула* (ориентированная пара логических формул);
 - *логическая формула эквиваленции*;
 - *отрицательная логическая формула*, трактуемая как унарная связка, единственным компонентом которой является отрицаемая логическая формула;
 - *кванторная формула*, трактуемая как бинарная ориентированная связка, компонентами которой являются:
 - множество переменных, связываемых квантором;
 - логическая формула, на которую действует квантор;
 - *формула существования*, трактуемая как связка, одним из компонентов которой является множество переменных, связываемых квантором существования, а другим компонентом – логическая формула, на которую действует квантор существования (в случае, если подкванторная формула является конъюнктивной, она может явно не вводиться, а ее компоненты могут рассматриваться как компоненты формулы существования);
 - *логическая теория*, трактуемая как множество истинных высказываний, описывающих свойства и закономерности описываемой предметной области, на которой указанные высказывания интерпретируются;
 - *логическое следствие* (связка, связывающая некоторое высказывание с теми высказываниями, на основании которых оно может быть доказано);
 - *декомпозиция логического следствия* (связь логического следствия с соответствующим доказательством);
 - *доказательство* (множество логических следствий, обеспечивающих логический вывод некоторого высказывания);
 - *главное доказательство* (выделяемое из множества доказательств одного и того же высказывания);
 - *аксиома* (априори истинное высказывание);
 - *теорема* (доказываемое высказывание).

Здесь мы говорим не о формальном описании свойств и закономерностей заданной предметной области на логическом языке традиционного вида, а о таком описании этой предметной области, которое оформляется также в виде предметной области, надстраиваемой над исходной (заданной) предметной областью. Просто объектами исследования в указанной надстраиваемой предметной области являются логические формулы, интерпретируемые на исходной предметной области.

Типология предметных областей, определяемых их стационарностью, а также их объектами и предметами исследования

Множество предметных областей разбивается на два основных класса:

- **стационарные** (статические) предметные области, структура которых от времени не зависит;
- **нестационарные** (динамические) предметные области, структура которых зависит от времени. Это значит, что связи между описываемыми объектами могут появляться и исчезать.

Важнейшим этапом описания **нестационарных предметных областей** является переход от них к стационарным предметным областям, которые являются их стационарными эквивалентами (своего рода «летописями», протоколами). Такие области будем называть предметными областями ситуаций и событий. Предмет исследования каждой такой предметной области включает в себя следующие понятия:

- *ситуация* (структура, сохраняющая стационарность в течение некоторого отрезка времени);
- *событие* (структура, отражающая переход от одной ситуации к другой);
- *быть отметкой начала события или ситуации* (это отметка на шкале времени);
- *быть отметкой завершения события или ситуации*;
- семейство понятий, описывающих самые различные соотношения между ситуациями и событиями во времени (с учетом различных вариантов соотношения между отметками начала и завершения ситуаций и событий).

Рассмотрим типологию **стационарных предметных областей**. Перечислим основные классы стационарных предметных областей, каждому из которых соответствует свой определенный предмет исследования (но разным предметным областям, входящим в каждый такой класс, соответствуют разные классы объектов исследования):

- *предметная область ситуаций и событий* – это класс предметных областей, для каждой из которых классом объектов исследования является множество всевозможных ситуаций и событий в некоторой конкретной описываемой нестационарной предметной области;
- *предметная область действий* – это класс предметных областей, для каждой из которых классом объектов исследования является множество всевозможных действий, целенаправленных субъектов, инструментов и объектов действий (объектов, на которые действия направлены);
- *логическое описание стационарной предметной области* общего вида – это класс предметных областей, для каждой из которых классом объектов исследования является множество всевозможных логических формул (в частности высказываний), интерпретируемых на некоторой описываемой стационарной предметной области;
- *логическое описание предметной области ситуаций и событий* – это класс предметных областей, для каждой из которых объектами исследования являются всевозможные логические формулы, описывающие причинно-следственные закономерности, интерпретируемые на некоторой описываемой предметной области ситуаций и событий;
- *логическое описание предметной области действий* – это класс предметных областей, для каждой из которых объектами исследования являются всевозможные логические формулы, интерпретируемые на некоторой предметной области действий;
- *теоретико-множественная онтология* – это класс предметных областей, для каждой из которых объектами исследования являются понятия, составляющие сигнатуру некоторой описываемой предметной области, а элементами предмета исследования являются отношения и алгебраические операции, описывающие теоретико-множественные связи;
- *логическая онтология* – это класс предметных областей, для каждой из которых объектами исследования являются понятия, составляющие сигнатуру некоторой описываемой предметной области, и определения этих понятий, а элементами предмета исследования, в частности, являются:
 - понятие «*быть неопределяемым понятием*» (в рамках соответствующего логического описания, описываемой предметной области);
 - понятие «*быть определяемым понятием*»;
 - отношение, связывающее каждое определяемое понятие с теми понятиями, на основании которых оно определяется;
- *терминологическая онтология* – это класс предметных областей, для каждой из которых объектами исследования являются термины (словосочетания), соответствующие различным элементам описываемой предметной области;
- *предметная область вопросов и информационных задач* – это класс предметных областей, для каждой из которых объектами исследования являются вопросы, информационные задачи, обобщенные вопросы и обобщенные информационные задачи, задаваемые по отношению к некоторой описываемой предметной области, а также соответствующие им способы решения информационных задач (т. е. различные программы);
- *предметная область поведенческих целей и задач* – это класс предметных областей, для каждой из которых объектами исследования являются поведенческие цели и задачи, обобщенные поведенческие цели и задачи, задаваемые по отношению к некоторой описываемой нестационарной предметной области, а также соответствующие им способы решения поведенческих задач.

Содержательная структура базы знаний интеллектуальной системы, определяемая набором предметных областей, описываемых этой базой знаний

Что должна «знать» интеллектуальная система кроме той предметной области, на которой она специализируется? Оказывается, много чего. И это прежде всего обусловлено тем, что интеллектуальная система должна не только уметь решать задачи в заданной предметной области, но и эффективно взаимодействовать как с конечными пользователями, так и с разработчиками.

Поэтому база знаний любой интеллектуальной системы представляет собой результат интеграции фактически нескольких баз знаний, каждая из которых описывает свою предметную область. К их числу относятся:

- база знаний, описывающая основную предметную область, в которой специализируется данная интеллектуальная система;
- система баз знаний, описывающих внешние языки и/или воспринимаемые образы внешней среды;
- база знаний, описывающая пользовательский интерфейс;
- база знаний, описывающая особенности и возможности собственного Я как субъекта, взаимодействующего с внешней средой (в частности с пользователями);
- семейство баз знаний, описывающих пользователей как субъектов, взаимодействующих с системой (как партнеров диалога);
- база знаний, описывающая процесс (в том числе историю) взаимодействия системы с внешней средой (в частности с пользователями).

Семантическая структура базы знаний как иерархическая система описываемых ею предметных областей

Семантическую структуру базы знаний, описывающей некоторую заданную (исходную) предметную область, можно рассматривать как иерархическую систему предметных областей различного специального вида, надстраиваемых над заданной предметной областью. Пусть нам дан некоторый атомарный (недекомпозируемый нами) раздел некоторой предметной области. Такой предметной областью может быть, например, геометрическая предметная область, т. е. та, объектами исследования которой являются геометрические фигуры. А указанным атомарным разделом заданной предметной области может быть, например, предметная область прямоугольных треугольников. Это предметная область, которая включается в состав (является подобластью) геометрической предметной области и объекты исследования которой – прямоугольные треугольники. Проектирование базы знаний, которая является описанием предметной области прямоугольных треугольников, условно можно разбить на следующие этапы:

- уточнение структуры описываемой предметной области;
 - уточнение предмета исследования;
 - уточнение всей сигнатуры (для рассматриваемой предметной области имеются в виду такие понятия, как *гипотенуза*, *катет*, *площадь*, *вершина*, *сторона*, *биссектриса треугольника*, *медиана треугольника*, *высота треугольника*, *мера угла*, *длина*, *граница* и т. д.);
 - уточнение набора вспомогательных объектов, связь с которыми имеет существенное значение для рассмотрения исследуемых объектов (в данной предметной области – это *геометрические точки*, *отрезки*, *прямые*, *углы*, *окружности*);
- построение предметной области, которая является теоретико-множественной онтологией рассматриваемой предметной области;
- построение предметной области, которая является логической онтологией рассматриваемой предметной области;
- построение предметной области, которая является терминологической онтологией рассматриваемой предметной области;
- построение предметной области, которая является логическим описанием заданной предметной области. В рассматриваемом нами примере в число объектов исследования рассматриваемой на данном этапе предметной области войдут *Теорема Пифагора*, *Доказательство теоремы Пифагора*, *Теорема о площади прямоугольного треугольника* и т. д.;

- построение предметной области вопросов и информационных задач для заданной предметной области;
- построение предметной области когнитивных мультимедийных иллюстраций и библиографических источников для заданной предметной области.

Таким образом, проектирование базы знаний можно рассматривать как процесс построения некоторой исходной предметной области и процесс наращивания указанной предметной области целым рядом надобластей, у каждой из которых есть свой класс исследуемых объектов.

Завершая рассмотрение отношений, связывающих между собой различные предметные области, особо подчеркнем то, что одна предметная область может быть метаобластью (описанием) по отношению к другой предметной области и то, что любым видам знаний соответствует своя предметная область со своими объектами исследования и своим предметом исследования. Сложность заключается в том, чтобы эту предметную область увидеть и явно задать.

Структуризация базы знаний, выделение в ней различных связанных между собой подструктур необходимы по целому ряду причин. В частности, для дидактических целей (человеку, усваивающему некоторые знания, желательно иметь своего рода оглавление или карту этих знаний, что позволяет планировать их усвоение и рассматривать их с различной степенью детализации), а также для организации распределения работ по проектированию баз знаний (когда разным исполнителям поручается разработка разных фрагментов базы знаний, имеющих достаточно четкие границы).

Таким образом, база знаний рассматривается как система взаимосвязанных между собой интегрируемых структур, которые будем называть **фрагментами базы знаний**. Связи между фрагментами базы знаний могут быть самыми различными. Каждый фрагмент и вся база знаний в целом может иметь несколько вариантов декомпозиции на подфрагменты (частные фрагменты).

По структурно-семантическому принципу можно выделить следующие типы фрагментов баз знаний:

- база фактов некоторой предметной области, которую сокращенно будем называть просто **предметной областью** и которая представляет собой результат интеграции всех известных в текущий момент фактографических высказываний, являющихся истинными для указанной предметной области;
- **иерархическая система нескольких предметных областей**, которые нецелесообразно объединять (интегрировать) в одну предметную область;
- **семантическая окрестность заданного объекта**;
- **связная семантическая окрестность конечного множества заданных объектов**, представляющая сравнительный анализ и связи объектов из заданного конечного множества, т. е. описывающая сходства (аналогии), отличия заданных объектов, а также близкие связи между ними.

При структуризации базы знаний некоторым ее фрагментам приписывается статус **разделов базы знаний**, которые именуются, нумеруются и входят в состав (оглавление) базы знаний.

Рассмотрим формальное уточнение понятия **предметной области** с помощью SC-кода. Если в рамках **sc-модели базы знаний** явно вводится некоторая предметная область, то она трактуется как sc-структура, для которой в базе знаний явно вводится обозначающий ее sc-узел, который, в свою очередь, связывается входящей в него sc-дугой основного вида с sc-узлом, обозначающим класс sc-структур, являющихся предметными областями. После этого в указанной sc-структуре необходимо явно задать роли некоторых узлов, входящих в состав этой sc-структуры. К числу таких ролей относятся:

- быть **максимальным классом исследуемых объектов**, т. е. множеством всех исследуемых объектов и только их. В каждой предметной области существует только один ключевой узел, выполняющий такую роль;
- быть **классом исследуемых объектов**. Каждая предметная область может иметь любое число таких классов;
- быть **классом вторичных объектов**, построенных на основе исследуемых;

- быть **классом вспомогательных объектов**, через связи с которыми описываются некоторые характеристики исследуемых объектов;
- быть **отношением, каждая связка которого связывает только исследуемые объекты** или вторичные объекты, построенные на основе исследуемых;
- быть **отношением, каждая связка которого связывает исследуемые объекты со вспомогательными**.

Такое явное указание ролей ключевых элементов предметных областей есть не что иное, как их семантическая спецификация, уточняющая то, какие объекты описываются (исследуются) в данной предметной области, и о каких характеристиках, связях исследуемых объектов в данной предметной области идет речь.

Можно говорить о достаточно богатой типологии предметных областей. А именно, выделяют следующие классы предметных областей:

- предметная область, описывающая теоретико-множественные характеристики и связи заданного семейства объектов. Такие предметные области могут быть онтологиями других предметных областей;
- терминологическая сеть заданного фрагмента базы знаний;
- текст формальной теории, описывающей свойства и закономерности заданной предметной области. Классами объектов исследования такой предметной метаобласти являются: 1) класс логических формул и, в частности, высказываний интерпретируемых на заданной предметной области; 2) класс элементов заданной предметной области, используемых в качестве констант в указанных логических формулах; 3) класс переменных, используемых в указанных логических формулах и возможными значениями которых являются соответствующие элементы заданной предметной области;
- логическая система понятий, описываемых в заданной формальной теории. Эта предметная метаобласть выделяет класс понятий, не определяемых в заданной формальной теории, и связывает каждое определяемое понятие с теми, на основе которых оно определяется;
- логическая система утверждений заданной формальной теории. Эта предметная метаобласть выделяет класс аксиом для заданной формальной теории, каждой теореме ставит в соответствие одно из ее доказательств (основное доказательство) и связывает каждую теорему со всеми теми утверждениями и определениями, которые используются в основном доказательстве этой теоремы;
- логическая система фрагментов баз знаний, связывающая каждый фрагмент с теми, в которых даются определения понятий, используемых в заданном фрагменте, и вводятся и доказываются используемые утверждения. На основании такой логической системы строятся различные варианты последовательности изучения (прочтения) разделов баз знаний.

Заметим, что некоторым из перечисленных классов предметных областей может соответствовать одинаковый (унифицированный, фиксируемый) набор используемых в них ключевых понятий. Унификация (стандартизация) таких наборов понятий является важнейшим средством более глубокой семантической совместимости (интегрируемости) различных фрагментов базы знаний. Результатом такой унификации фактически является разработка средств SC-кода, ориентированных на представление предметных областей соответствующего класса. Такие языковые средства будем называть специализированным **SC-языком**.

К числу таких специализированных sc-языков можно отнести:

- **теоретико-множественный sc-язык**, обеспечивающий описание теоретико-множественных характеристик и связей заданного семейства объектов. С помощью такого языка, в частности, могут быть представлены предметные области, являющиеся теоретико-множественными онтологиями других предметных областей;
- **терминологический sc-язык**, обеспечивающий построение терминологических сетей;
- **логический sc-язык**, обеспечивающий построение sc-текстов формальных теорий.

Таким образом, **SC-код** является ядром целого семейства самых различных **SC-языков**, ориентированных на описание различных классов предметных областей, в каждый из которых входят предметные области с разными множествами исследуемых объектов, но с одинаковыми предметами исследования.

Каждому такому специализированному sc-языку ставится в соответствие **множество ключевых узлов**, обозначающих различные классы исследуемых объектов, различные отношения и алгебраические операции, заданные на множестве исследуемых объектов.

SC-язык, являющийся объединением всевозможных специализированных sc-языков будем называть **языком SCK** (Semantic Code Knowledge). Этот язык рассматривается нами как интегрированный язык представления знаний. Язык SCK является открытым (расширяемым) языком, поскольку его всегда можно пополнить новым sc-языком, описывающим структуры нового вида предметных областей.

Построение **семантической структуры базы знаний** интеллектуальной системы требует не только явного представления спецификаций каждой описываемой предметной области в виде sc-текста, но и явного описания всевозможных связей между этими предметными областями.

Переходя к рассмотрению отношений, заданных на множестве предметных областей, мы фактически переходим к некоторой предметной **метаобласти**, объектами исследования которой являются всевозможные предметные области, в том числе и сама эта предметная метаобласть.

Обобщая понятия гомоморфизма и изоморфизма алгебраических систем, можно говорить о гомоморфизме и изоморфизме предметных областей, что дает хорошую основу для выявления глубоких нетривиальных аналогий между предметными областями.

Различные предметные области могут пересекаться. То есть элементы одной предметной области могут быть также и элементами другой предметной области. При этом возможны самые различные варианты такого пересечения. Это может быть строгое пересечение или строгое включение. Общие элементы пересекающихся предметных областей могут в рамках этих областей выполнять как одинаковые, так и разные роли. Например, первичные элементы одной предметной области могут входить в состав другой в качестве вторичных или ключевых элементов. Объекты исследования одной предметной области могут входить в состав другой и в качестве вспомогательных элементов.

В качестве примеров отношений, заданных на множестве предметных областей, рассмотрим несколько вариантов выделения частных предметных областей:

- выделение частной предметной области, на основе выделения подмножества из максимального класса исследуемых объектов. Таким способом из предметной области геометрии, объектами исследования которой являются геометрические точки, фигуры и семейства фигур, можно выделить: 1) предметную область планиметрии, изучающую планарные фигуры и планарные семейства фигур (т. е. семейства фигур, лежащих в одной плоскости); 2) предметную область стереометрии, которая изучает непланарные фигуры и непланарные семейства фигур, которые могут состоять как из непланарных, так и из планарных фигур;
- выделение частной предметной области, на основе выделения подмножества из семейства классов исследуемых объектов. Таким способом из предметной области, изучающей треугольники, можно выделить: 1) предметную область, изучающую остроугольные, тупоугольные и прямоугольные треугольники; 2) предметную область, изучающую равносторонние, разносторонние и равнобедренные треугольники;
- выделение частной предметной области, на основе выделения подмножества из семейства отношений, заданных на исследуемых объектах. Таким способом из геометрической предметной области можно выделить: 1) предметную область, объектами исследования которой являются геометрические фигуры, а предметом исследования – их числовые характеристики; 2) предметную область, объектами исследования которой являются геометрические фигуры, а предметом исследования – различные виды их конгруэнтности (движений).

Предметная область позволяет рассматривать исследуемые объекты на разных уровнях детализации. Детализацию рассмотрения исследуемых объектов можно осуществлять как в рамках исходной (заданной) предметной области, расширяя эту предметную область в соответствующих направлениях, а можно переходить к системе самостоятельных, но связанных между собой предметных областей.

Первым и важнейшим этапом проектирования базы знаний является уточнение структуры описываемой предметной области или нескольких взаимосвязанных предметных областей. Уточнение такой структуры – это, прежде всего, уточнение класса исследуемых объектов, предмета исследования, всего семейства ключевых узлов семантической сети, представляющей предметную область. При этом для заданного класса исследуемых объектов и заданного предмета исследования можно построить более качественную и менее качественную предметную область.

Рассмотрим еще один тип фрагментов баз знаний – **семантические окрестности**. В общем случае семантическая окрестность заданного объекта – это описание некоторых числовых характеристик, свойств и связей заданного объекта. Частными видами семантических окрестностей являются:

- описание характеристик, свойств или связей заданного объекта, однозначно определяющих (устанавливающих) заданный объект (для понятий – это определение или любое другое высказывание определяющего типа);
- полная (интегрированная) семантическая окрестность заданного объекта, содержащая все известные в текущий момент сведения об этом объекте в рамках заданного раздела базы знаний или в рамках всего текущего состояния базы знаний;
- описание числовых характеристик (параметров, признаков) заданного объекта;
- описание свойств заданного объекта (это те истинные нефактографические высказывания, в которых знак этого объекта используется в качестве константы);
- описание теоретико-множественных связей заданного объекта с другими объектами;
- описание разноязычных терминов, иероглифов, пиктограмм, используемых для внешней идентификации заданного объекта (в т. ч. и описание происхождения этих идентификаторов);
- перечень отношений, соответствующих заданному объекту (отношений, в области определения которых заданный объект входит в качестве элемента; отношений, в области определения которых заданный объект входит в качестве подмножества; отношений, для областей определения которых заданный объект является надмножеством; отношений, область определения каждого из которых строго пересекается с заданным объектом).

В **sc-модели базы знаний** каждая семантическая окрестность представляется в виде соответствующего sc-текста. Для явного введения (задания) этой семантической окрестности в рамках sc-модели базы знаний необходимо:

- 1) явно ввести sc-узел, обозначающий эту семантическую окрестность;
- 2) явно связать введенный sc-узел sc-дугами основного вида со всеми элементами обозначаемого им sc-текста;
- 3) с помощью ролевого отношения **быть центром семантической окрестности** явно указать центральный элемент семантической окрестности;
- 4) явно связать введенный sc-узел sc-дугой основного вида со специальным sc-узлом, обозначающим **класс sc-структур, являющихся семантическими окрестностями**.

После этого можно описывать различные характеристики, в частности, типологию явно введенной семантической окрестности, а также различные связи этой семантической окрестности с другими фрагментами базы знаний.

Технология проектирования унифицированных семантических моделей баз знаний рассмотрена в работе [19].

Семейство совместимых языков семантических сетей, построенных на основе SC-кода и имеющих разные наборы ключевых узлов

Выше, рассматривая иерархию и типологию предметных областей, мы выделили классы предметных областей, задаваемые общими сигнатурами и обеспечивающими формальное уточнение (на основе понятия предметной области) различных видов знаний.

Поскольку от каждой предметной области легко перейти к семантической сети, которая является структурной моделью этой предметной области, каждому классу предметных областей с заданной сигнатурой можно поставить в соответствие язык семантических сетей с заданным (общим для всех семантических сетей, принадлежащих этому языку) алфавитом и набором ключевых узлов.

Так как каждую семантическую сеть можно представить в виде семантической сети, принадлежащей SC-коду, то каждому из указанных языков семантических сетей можно поставить в соответствие множество конструкций SC-кода, в которых кроме ключевых узлов самого SC-кода используются ключевые узлы, определяемые сигнатурными элементами соответствующего класса предметных областей. Каждое такое множество конструкций SC-кода будем называть *sc-языком*, ориентированным на представление структурных моделей предметных областей соответствующего класса.

Таким образом, *sc-язык* – это язык семантических сетей, ориентированный на представление структурных моделей определенного класса предметных областей, которым соответствует определенная сигнатура.

Каждый *sc-язык* задается алфавитом, который совпадает с алфавитом SC-кода, и семейством ключевых узлов, которое включает в себя семейство ключевых узлов SC-кода.

Перечислим некоторые *sc-языки*:

sc-язык ситуаций и событий

= *sc-язык, ориентированный на представление структурных моделей предметных областей, в которых объектами исследования являются ситуации и события некоторой нестационарной (динамической) предметной области, а предметом исследования – различные классы ситуаций и событий и различные отношения, заданные на объектах исследования*

– К л ю ч е в ы м и у з л а м и данного *sc-языка* являются:

- *ситуация*
= *Понятие ситуации*
- *событие*
= *Понятие события*
- *текущая ситуация*
= *ситуация, имеющая место в текущий момент времени*
= *ситуативный синглетон, элементом которого является знак текущей ситуации*

sc-язык ситуаций и событий в sc-памяти

= *sc-язык описания процесса обработки sc-конструкций*

= *sc-язык описания sc-памяти как некоторой динамической системы*

– К л ю ч е в ы м и у з л а м и данного *sc-языка* являются:

- *сформированное множество*
= *множество sc-элементов, каждый из которых присутствует в рамках соответствующего состояния sc-памяти*
*/*в рамках соответствующей ситуации в sc-памяти*/*
- *неактуальный sc-элемент*
= *sc-элемент, отсутствующий в соответствующем состоянии sc-памяти*
- П р и м е ч а н и е . В описании динамики *sc-памяти* следует отличать:
 - понятие *sc-переменной* (значение которой от времени не зависит)
 - понятие *sc-переменной*, значение которой может меняться во времени (например программная переменная).

sc-язык действий

логический sc-язык

логическая формула

- ⊃ *атомарная логическая формула*
= *атомарная формула существования*
- ⊃ *формула существования*
 - ⊃ *единственность существования*

- ▷ всеобщность над импликацией
= импликативная формула с неявно навешенным квантором всеобщности
- ▷ логическая теория

описываемая предметная область_

- = быть предметной областью, которая описывается заданной логической теорией
- = быть семантической сетью, которая является структурной моделью описываемой предметной области

Рассмотрим фрагмент предметной области, объектами исследования которой являются логические формулы и, в том числе, высказывания, интерпретируемые на рассмотренной выше геометрической предметной области.

В качестве примера такого фрагмента рассмотрим одну из аксиом геометрии Евклида, которая на естественном языке может иметь следующие эквивалентные формулировки:

- через каждую пару точек можно провести единственную прямую;
- для любой пары точек существует единственная инцидентная им прямая;
- для любых $_ti, _tj$ имеет место импликация:
 - если $_ti, _tj \ni$ точка,
 - то существует единственная $_pij$ такая, что $_pij \ni$ прямая; $_ti, _tj \ni$ $_pij$;
- На логическом языке традиционного вида данное высказывание может быть представлено следующим образом:
 - $\forall _ti, _tj ([_ti, _tj \in \text{точка}] \Rightarrow \exists !_pij [_pij \in \text{прямая}; _ti, _tj \in _pij])$
- Уточним структуру изучаемого фрагмента рассматриваемой предметной области. Теперь приведем представление рассматриваемой логической формулы в SC-коде.

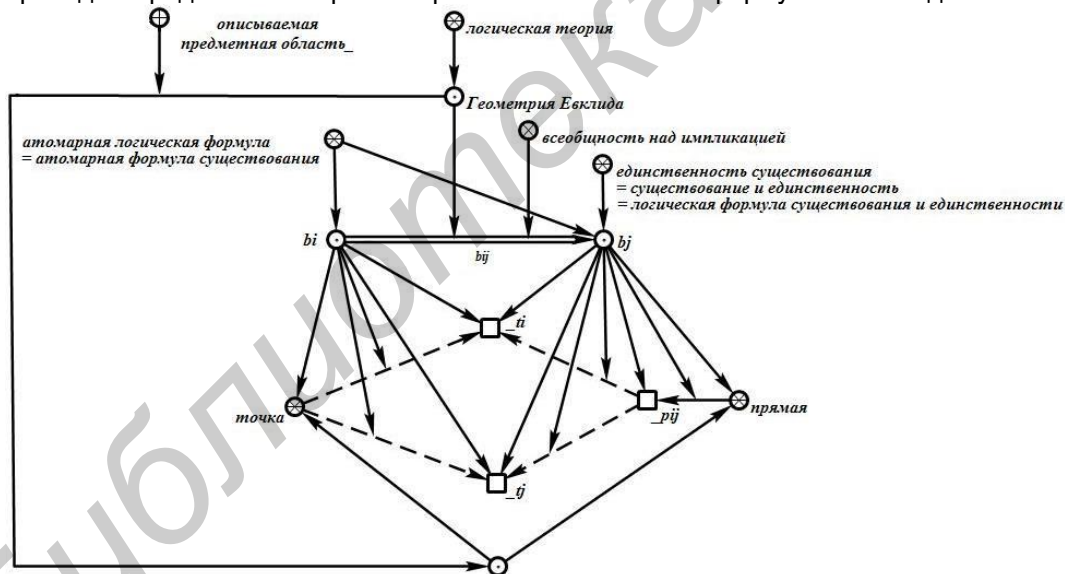




Рис. 2. Представление рассматриваемой логической формулы в SC-коде

- Здесь переменные, связываемые кванторами, указываются неявно. Все переменные, входящие в состав атомарной логической формулы и не связанные в вышестоящих логических формулах, в состав которых эта атомарная формула входит, неявно связываются квантором существования в рамках этой атомарной логической формулы.
- Если переменная (например $_ti$) входит в состав логической формулы (например в bij), которая является неатомарной логической формулой, относящейся к классу формул всеобщности над импликацией, при этом, если указанная переменная ($_ti$) не связана в вышестоящих логических формулах и входит в состав обеих логических формул, являющихся компонентами формулы в bij , то указанная переменная ($_ti$) неявно связывается квантором всеобщности в рамках логической формулы в bij .
- В рассмотренной scg-конструкции используются следующие графические примитивы в дополнение к тем, которые были указаны выше:

	<ul style="list-style-type: none"> • графический примитив, изображающий переменную, значениями которой являются пары принадлежности
	<ul style="list-style-type: none"> • графический примитив, изображающий переменную, значениями которой являются константные sc-узлы

sc-язык логических формул, описывающих стационарные предметные области общего вида

sc-язык логических формул, описывающих предметные области ситуаций и событий

sc-язык логических формул, описывающих предметные области действий

sc-язык множеств

= sc-язык, ориентированный на представление структурных моделей предметных областей, в которых объектами исследования являются множества, а предметом исследования – различные классы множеств и различные отношения, заданные на множествах

sc-язык отношений

= sc-язык, ориентированный на представление структурных моделей предметных областей, в которых объектами исследования являются отношения, а предметом исследования – различные классы отношений и различные отношения, заданные на отношениях

sc-язык логических онтологий

sc-язык терминологических онтологий

sc-язык вопросов и информационных задач

sc-язык поведенческих целей и задач

sc-язык способов решения информационных задач

– Пояснение.

- Данный sc-язык представляет собой результат интеграции целого семейства sc-языков, каждому из которых соответствует своя модель решения информационных задач: процедурная или непроцедурная, последовательная или параллельная, высокоуровневая или низкоуровневая

sc-язык способов решения поведенческих задач

sc-язык графовых структур

sc-язык числовых моделей

= sc-язык, ориентированный на представление структурных моделей предметных областей, в которых объектами исследования являются числа и числовые отношения, а предметом исследования – различные классы чисел и различные отношения, заданные на числах и числовых отношениях

– К л ю ч е в ы м и у з л а м и данного sc-языка являются:

- величина
= инвариант измерения какой-либо измеряемой характеристики в различных шкалах
- измеряемая характеристика
= измеряемый параметр
= бинарное ориентированное отношение, связывающее измеряемый объект с соответствующей величиной (результатом измерения)
- шкала
= бинарное ориентированное отношение, связывающее величину с ее числовым значением
- число /* натуральное число, целое число, действительное число*/
- сложение*
- умножение*
- возведение в степень*
- /*и т. д.*

sc-язык гипермедиальных структур

= sc-язык, ориентированный на представление структурных моделей предметных

областей, в которых объектами исследования являются отображаемые пользователям файлы интеллектуальной системы, а предметом исследования – семейство понятий, обеспечивающих спецификацию этих файлов и описание различных синтаксических и семантических связей между ними

В качестве примера гипермедийной структуры приведем sc-конструкцию, которая описывает связь знака некоторого действительного числа (в данном случае числа 25) с различными вариантами представления (записи) этого числа в разных системах счисления.

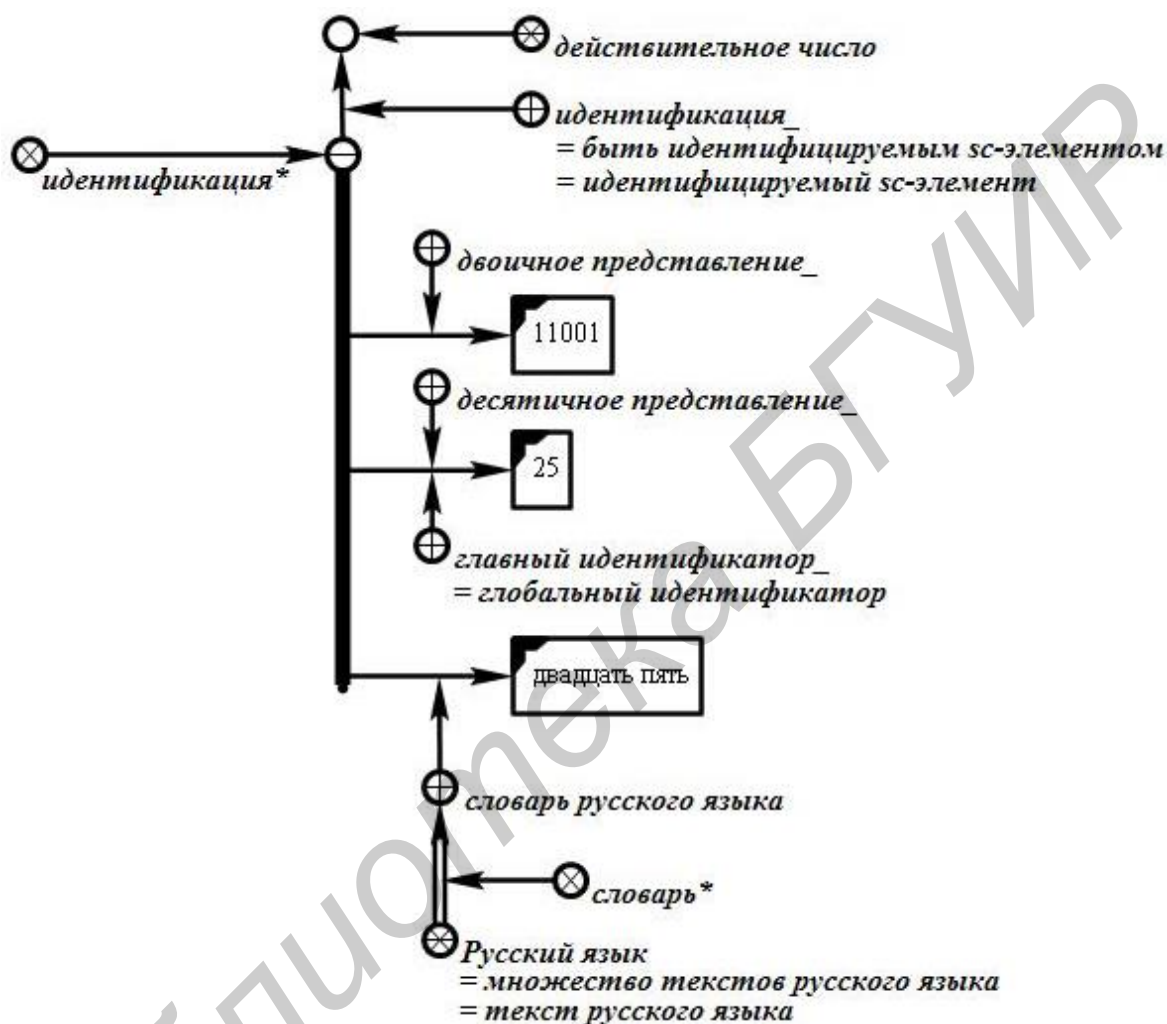


Рис. 3. SC-конструкция, которая описывает связь числа 25 с различными вариантами записи этого числа в разных системах счисления

7. Графовые языки программирования

Для описания способов решения задач и поведения агентов над общей графодинамической памятью используют **графовые языки программирования**, которые ориентированы на обработку унифицированных семантических сетей и программы которых сами являются унифицированными семантическими сетями.

Если все используемые в интеллектуальной системе графовые языки программирования привести к общему унифицированному стандарту – SC-коду (это требует представления в виде sc-текстов не только самих программ, но и обрабатываемых ими данных), то можно достаточно эффективно решать проблему формализации семантической совместимости программ, написанных не только на одном, но и на разных языках программирования.

В традиционных языках программирования синтаксическая структура и семантика хранящихся в памяти обрабатываемых данных остаются на усмотрение программиста, в результате чего наблюдаем разнородность. Поэтому нельзя говорить о семантической совместимости программ. В этом смысле традиционные языки программирования несовершенны.

Наряду с применением в интеллектуальных системах множества sc-языков самого различного назначения востребованным является использование целого семейства совместимых sc-языков программирования, которые могут иметь разный уровень, а также быть последовательными, процедурными и декларативными.

Важнейшей особенностью всех этих языков является использование ассоциативного доступа к обрабатываемым фрагментам хранимого в графодинамической памяти sc-текста.

Операционная семантика каждого такого графового языка программирования (точнее, sc-языка программирования) задается коллективом агентов над общей графодинамической памятью, которые обеспечивают интерпретацию любой программы указанного языка программирования, хранящейся вместе с обрабатываемой информацией в указанной графодинамической памяти.

Программы, представленные в виде семантической сети и описывающие обработку семантических сетей, а также соответствующие им языки программирования, фактически открывают новую страницу в теории программирования, которую можно назвать семантической теорией программ и языков программирования.

Основным лейтмотивом такой теории должно быть обеспечение семантической совместимости программ и языков программирования.

8. Унификация формального описания агентов, работающих над семантической памятью

Из всех используемых в интеллектуальной системе графовых языков программирования (sc-языков) выделим **базовый графовый язык программирования**, ориентированный на описание агентов, работающих над общей графодинамической памятью, в которой хранятся и обрабатываются унифицированные семантические сети.

Выделение базового sc-языка программирования предназначено для обеспечения унифицированного, платформенно-независимого формального описания поведения агентов, работающих над общей графодинамической памятью.

Такой базовый sc-язык программирования будем называть **языком SCP** (Semantic Code Programming), а написанные на нем программы – **scp-программы**.

Перечислим основные особенности языка SCP:

- относится к классу **графовых языков программирования**;
- ориентирован на обработку **унифицированных семантических сетей** (sc-текстов), хранимых в семантической памяти;
- принадлежит классу sc-языков: программы языка SCP представляются также в виде унифицированных семантических сетей (sc-текстов);
- ориентирован на описание **параллельной асинхронной** обработки sc-текстов, хранимых в семантической памяти;
- использует **ассоциативный доступ** к фрагментам обрабатываемых sc-текстов;
- является процедурным языком программирования низкого уровня, предназначенным для описания поведения агентов, работающих над семантической памятью;
- уникальной особенностью языка SCP является то, что на нем можно писать **реконфигурируемые программы**, т. е. программы, которые в процессе своего выполнения могут изменять сами себя (удалять или порождать операторы, корректировать порядок их выполнения и т. п.). Такая особенность языка SCP обусловлена не только тем, что scp-программы и обрабатываемые ими данные хранятся в общей памяти, но и тем, что они принадлежат одному и тому же базовому языку (SC-коду), имеющему четко заданную семантическую интерпретацию;

- sc-моделям интеллектуальной системы соответствует достаточно простая процедура их интеграции;
- sc-модель интеллектуальной системы является параллельной и асинхронной моделью обработки знаний, причем параллелизм имеет место не только между процессами, выполняемыми разными sc-агентами, но и при функционировании каждого из них. Это дает существенный резерв для повышения производительности интеллектуальных систем при соответствующей поддержке этого параллелизма на уровне реализации интеллектуальных систем;
- в виде семантической сети представляется как сама программа, так и обрабатываемая ею информация, хранимая в ассоциативной реконфигурируемой памяти.

Более подробно графовый язык программирования SCP и технология проектирования scr-программ рассмотрены в работах [8, 11] и в документации проекта OSTIS [71].

9. Унификация семантических моделей обработки знаний

На основе унифицированных семантических сетей (sc-текстов) уточним понятие **унифицированной модели обработки информации**, а также понятие унифицированной модели решения задач.

Все указанные абстрактные модели будем называть **sc-моделями обработки знаний** или **sc-машинами**, поскольку в основе их лежит использование SC-кода. Каждая такая модель (sc-машина) представляет собой многоагентную систему, состоящую из:

- 1) графодинамической памяти, в которой хранятся и обрабатываются тексты SC-кода и переработка информации в которой сводится к изменению конфигурации связей между sc-элементами – такую память будем называть **sc-памятью**;
- 2) коллектива агентов, работающих над общей для них sc-памятью и взаимодействующих между собой только через эту память – такие агенты будем называть **sc-агентами**.

Очевидно, что sc-модели обработки знаний являются частным, унифицированным видом графодинамических моделей параллельной асинхронной обработки информации.

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. Типология sc-агентов достаточно богата. Можно выделить следующие классы sc-агентов:

- sc-агенты, обеспечивающие интерпретацию программ различных sc-языков программирования высокого уровня;
- sc-агенты информационного поиска;
- sc-агенты, обеспечивающие реализацию правил логического вывода, соответствующих самым различным логическим исчислениям;
- sc-агенты сведения задач к подзадам;
- sc-агенты анализа качества хранимой базы знаний, в частности, ее корректности, полноты;
- sc-агенты обнаружения и автоматического склеивания синонимичных sc-элементов;
- sc-агенты автоматического устранения некоторых ошибок в базе знаний;
- sc-агенты удаления информационного мусора (например, фрагментов базы знаний, которые редко востребованы и могут быть достаточно легко восстановлены в случае их отсутствия);
- sc-агенты, обеспечивающие трансляцию вводимой информации с различных внешних языков в SC-код;
- sc-агенты, обеспечивающие трансляцию sc-текстов, вводимых пользователю на различные внешние языки;
- рецепторные sc-агенты;
- эффекторные sc-агенты.

В понятии sc-машины набор агентов не задается, т. е. могут существовать разные sc-машины с разным набором sc-агентов. Несколько разных sc-машин можно **интегрировать**. С формальной точки зрения это сделать не очень сложно:

- 1) интегрировать sc-текст, описывающий текущее состояние взаимодействия sc-агентов одной sc-машины с аналогичным sc-текстом другой sc-машины;

- 2) полученный интегрированный sc-текст поместить в sc-память интегрированной sc-машины;
- 3) в интегрированную sc-машину включить все sc-агенты первой интегрируемой sc-машины и все sc-агенты второй интегрируемой sc-машины.

Более того, одна sc-машина может **интерпретировать** другую. Это значит, что при интерпретации sc-машин можно не выходить за пределы класса sc-машин. Для этого необходима разработка целого семейства sc-языков программирования различного уровня. Тексты (программы) всех этих языков должны храниться в sc-памяти, т. е. должны быть семантическими сетями, представленными в SC-коде. Операционная семантика (интерпретация) каждого из этих языков задается определенным набором sc-агентов, процедура выполнения (поведения) каждой из которых описывается программой, написанной на языке более низкого уровня.

В абстрактных sc-машинах можно выделить следующие языки программирования:

- Семейство sc-языков программирования высокого и сверхвысокого уровня (как процедурных, так и непроцедурных). Тексты программ этих языков хранятся в базе знаний интеллектуальной системы и описывают способы решения различных классов задач в соответствующих предметных областях.
- Базовый sc-язык программирования (язык SCP), на котором описываются sc-агенты и интерпретации sc-языков программирования высокого и сверхвысокого уровня, а также sc-операции, обеспечивающие интерпретацию различных логических исчислений, различных моделей интеллектуального решения задач.
- SC-язык программирования, на котором описываются sc-агенты интерпретации базового sc-языка программирования. Фактически это 1-й язык микропрограмм для **sc-компьютера**, обеспечивающего аппаратную интерпретацию базового sc-языка программирования (языка SCP).
- При необходимости можно ввести 2-й язык микропрограммирования, описывающий интерпретацию 1-го, и т. д.

На основе понятия абстрактной sc-машины можно уточнить понятие унифицированной логико-семантической модели интеллектуальной системы. Интеллектуальная система в целом рассматривается нами как многоагентная система, ориентированная на обработку семантического пространства, представленного семантической сетью, принадлежащей SC-коду. Назовем это sc-моделью интеллектуальных систем.

Абстрактная sc-модель интеллектуальной системы – это динамическая система, состоящая из:

- 1) абстрактной структурно перестраиваемой ассоциативной sc-памяти, в которой хранятся и обрабатываются тексты SC-кода;
- 2) коллектива внутренних sc-агентов – самостоятельных субъектов, каждый из которых способен выполнять соответствующий ему класс действий, направленных на изменение состояния sc-памяти. Каждое такое действие инициируется возникновением в sc-памяти ситуации или события определенного вида. Результатом каждого такого действия является генерация (создание) новых ситуаций или событий в sc-памяти. Частью такого результата является регистрация факта успешно или неуспешно выполненного действия (это часть языковых средств синхронизации действий sc-агентов);
- 3) коллектива рецепторных sc-агентов (интеллектуальных датчиков), каждый из которых реагирует на определенные ситуации или события во внешней среде и отображает (описывает) их в sc-памяти;
- 4) коллектива эффекторных sc-агентов, каждый из которых выполняет действие соответствующего класса, направленные на изменения состояния внешней среды. Каждое действие инициируется соответствующей командой, появившейся в sc-памяти.

Обратим внимание, что общая память данной многоагентной системы является не только средой, над которой работают агенты, но и средой, через которую они обмениваются информацией. То есть не существует никаких других каналов обмена информацией между такими агентами, кроме общей памяти.

Нетрудно заметить, что sc-текст базы знаний интеллектуальной системы является формальным и унифицированным уточнением того, что должна знать проектируемая интеллектуальная система, а sc-машина интеллектуальной системы и, в первую очередь, набор входящих в ее состав sc-агентов является формальным и унифицированным

уточнением того, что должна проектируемая интеллектуальная система уметь делать со своими знаниями.

Подчеркнем также, что четкое выделение абстрактного семантического уровня интеллектуальной системы позволяет не только обеспечивать семантическую совместимость интеллектуальных систем, но и сформировать критерии сравнения интеллектуальных систем по уровню их возможностей. Очевидно, что уровень этих возможностей определяется качеством (корректностью, полнотой, многообразием) ее знаний и эффективностью ее умений (эффективностью используемых ею моделей решения задач).

Заметим также, что абстрактную логико-семантическую модель можно построить для любой компьютерной системы (как для интеллектуальной, так и для системы традиционного вида), обеспечивая их семантическую совместимость на абстрактном логико-семантическом уровне.

Среди достоинств предложенной модели интеллектуальных систем можно назвать следующие:

- sc-моделям интеллектуальной системы соответствует достаточно простая процедура их интеграции;
- sc-модель интеллектуальной системы является параллельной и асинхронной моделью обработки знаний, причем, параллелизм имеет место не только между процессами, выполняемыми разными sc-агентами, но и при функционировании каждого из них. Это дает существенный резерв для повышения производительности интеллектуальных систем при соответствующей поддержке этого параллелизма на уровне реализации интеллектуальных систем.

10. Унификация семантических моделей информационного поиска

На основе унифицированных семантических сетей (SC-кода) обеспечивается построение **унифицированных семантических моделей информационного поиска** (унифицированных семантических моделей ассоциативного доступа к информации, хранимой в семантической памяти).

Ассоциативный доступ – это доступ, основанный не на знании того, где находится искомая (требуемая) информация (адрес или имя соответствующей области памяти), а на знании того, как искомая информация связана с известной информацией, хранимой в памяти, т. е. на знании некоторой спецификации искомой информации.

Эффективность организации информационного поиска в базе знаний интеллектуальной системы во многом определяет эффективность самой интеллектуальной системы. Это обусловлено тем, что время, затрачиваемое интеллектуальной системой на поиск нужных в текущий момент знаний и навыков, занимает не меньше половины времени, затрачиваемого на решение задачи в целом.

Унифицированная семантическая модель информационного поиска, которую будем называть **sc-моделью информационного поиска**, включает в себя:

- 1) **SC-язык вопросов**, с помощью которого в виде sc-текстов осуществляется описание (спецификация) запрашиваемых (искомых) фрагментов всего интегрированного sc-текста, хранимого в текущий момент в sc-памяти (т. е. sc-текста, который является sc-моделью базы знаний). Тексты, принадлежащие SC-языку вопросов, будем называть **sc-вопросами**.
- 2) **SC-язык оформления ответов**, с помощью которого осуществляется явное выделение sc-текстов, являющихся ответами, и явное описание их связи с явно выделенными sc-текстами, которые представляют вопросы, соответствующие указанным ответам.
- 3) Семейство информационно-поисковых sc-агентов, каждый из которых реагирует на соответствующий ему тип sc-вопроса (при этом он должен быть иницирован) и выполняет соответствующую поисковую процедуру в sc-памяти.

Семантическая типология вопросов является предметом отдельного рассмотрения. Приведем фрагмент такой типологии, чтобы проиллюстрировать семантическую мощь sc-языка вопросов.

Прежде всего по аналогии с логическими формулами множество вопросов разбивается на **атомарные** и **неатомарные вопросы**, каждый из последних представляет собой конечное множество вопросов.

Компонентами неатомарного вопроса могут быть как атомарные, так и неатомарные вопросы. При этом, если построить орграф, вершинами которого будут знаки всех вопросов, входящих в состав заданного неатомарного вопроса, а дуги которого будут связывать знаки неатомарных вопросов, входящих в состав заданного неатомарного вопроса, с их компонентами, то этот орграф будет деревом, все конечные вершины которого являются знаками атомарных вопросов. Частным видом неатомарного вопроса является конъюнктивный вопрос, ответом на который является конъюнкция (интеграция) ответов на все вопросы, являющиеся компонентами этого конъюнктивного вопроса.

Поскольку в общем случае вопросу может соответствовать несколько правильных ответов, удовлетворяющих, релевантных, соответствующих заданному вопросу, множество вопросов разбивается на вопросы:

- запрашивающие все правильные ответы;
- запрашивающие один (или по крайней мере один) правильный ответ;
- запрашивающие несколько разнообразных правильных ответов;
- запрашивающие точно указанное число (больше единицы) правильных ответов.

Специальным видом неатомарных вопросов являются **сколько-вопросы**, запрашивающие не сами правильные ответы на некоторый вопрос, который может быть как атомарным, так и неатомарным, и который является единственным компонентом сколько-вопроса, а количество таких правильных ответов.

Приведем некоторые типы атомарных вопросов:

- **какой-вопрос атомарного вида**. Каждый такой вопрос запрашивает фрагменты базы знаний, изоморфные заданному **образцу**, который может иметь произвольный размер и конфигурацию и может быть представлен не только логической формулой существования, где квантор существования действует на конъюнкцию атомарных логических формул, но также и логической формулой существования, где квантор существования действует на логическую формулу произвольного вида. Суть атомарного *какой-вопроса* заключается в поиске знаков таких объектов, которые заданным образом связаны с другими известными и неизвестными (искомыми) объектами, удовлетворяющими заданным требованиям. На основе *какой-вопросов* атомарного вида строится важный класс неатомарных вопросов, ключевыми компонентами которых являются атомарные *какой-вопросы*, а остальными компонентами – вопросы любого вида, в формулировках которых используются переменные, входящие в состав соответствующих ключевых *какой-вопросов*;
- **запрос всех элементов** заданного конечного множества (чаще всего – это множество из элементов некоторой структуры);
- **запрос внешней информационной конструкции**, представленной некоторым файлом в том или ином формате;
- запрос полного текста заданного высказывания;
- **ли-вопрос**, запрашивающий факт истинности или ложности заданного высказывания в рамках заданной формальной теории;
- **вопрос выбора альтернатив**, запрашивающий одно или несколько истинных высказываний из заданного множества высказываний;
- **почему-вопрос**, запрашивающий обоснование (доказательство) истинности заданного высказывания;
- **что-это-вопрос**, запрашивающий основные сведения об указываемом объекте. Фактически, речь идет о выделении из базы знаний семантической окрестности, центром которой является знак указываемого объекта. Таким объектом может быть понятие, предметная область, формальная теория, высказывание, любая структура, материальный объект;
- **запрос общих свойств** объектов, принадлежащих заданному классу;

- **запрос идентифицирующих признаков** заданного объекта. Здесь запрашиваются фрагменты базы знаний, каждый из которых однозначно определяет (устанавливает, идентифицирует) заданный объект. Если таким объектом является понятие, то идентифицирующим признаком будет либо определение этого понятия, либо соответствующая теорема о необходимости и достаточности;
- **запрос связей между заданными объектами**;
- **запрос сравнительного анализа заданных объектов**;
- **запрос сходств** заданных объектов (сходства, аналогии – это частный вид связей между объектами);
- **запрос отличий** заданных объектов (отличия объектов – это тоже частный вид связей между ними);
- **запрос плана решения** заданной конкретной задачи, т. е. плана достижения определенной цели в заданных конкретных условиях;
- **запрос обобщенного способа решения** любой задачи из заданного класса задач. Таким обобщенным способом может быть алгоритм, декларативная (непроцедурная) программа, нестрогое предписание (рекомендация);
- **зачем-вопрос**, запрашивающий то, какой надцели соответствует заданная цель, которая может быть сформулирована как в декларативной, так и в процедурной форме.

Список типов атомарных вопросов можно продолжить, но почти все они будут подтипами (подмножествами) перечисленных типов вопросов. В основе **сц-языка** вопросов лежит построение онтологии вопросов, в рамках которой четко прописываются все теоретико-множественные (в первую очередь, родо-видовые) связи между всеми выделенными типами и подтипами вопросов. При этом в формулировке каждого конкретного **сц-вопроса** явным образом отражается иерархия всех типов вопросов, которым принадлежит данный конкретный **сц-вопрос**. Для этого каждому типу вопросов ставится в соответствие ключевой **сц-узел**, обозначающий этот тип вопросов.

В заключение заметим, что в **сц-языке оформления ответов** кроме отношения релевантности, связывающего вопросы с правильными на него ответами, используются языковые средства, описывающие качество, полноту ответов. Это вызвано тем, что некоторые типы вопросов предполагают наличие целого множества правильных ответов, но разного качества, с разной степенью полноты.

11. Унификация семантических моделей интеграции знаний и семантических моделей интеграции целых интеллектуальных систем

На основе унифицированных семантических сетей обеспечивается построение **унифицированных семантических моделей интеграции знаний** (понимания знаний) и используют эти модели как основу:

- 1) процесса приобретения интеллектуальной системой новых знаний как со стороны конечных пользователей, так и со стороны разработчиков;
- 2) интеграции программ и различных семантических моделей расширения задач;
- 3) интеграции абстрактных логико-семантических моделей интеллектуальных систем.

Главное свойство интеллектуальной системы – не те интеллектуальные знания и навыки, а также интеллектуальные способности, которые она имеет в текущий момент, а метаспособность приобретать любые необходимые ей новые знания и навыки. А для этого интеллектуальная система как минимум должна уметь интегрировать эти приобретаемые знания и навыки. Следовательно, проблема формализации интеграции знаний и навыков является центральной для деятельности интеллектуальных систем.

Принципиальное свойство интеграции двух интеллектуальных систем заключается в следующем. Пусть имеются две интеллектуальные системы s_1 , s_2 , первая из которых способна решать задачи из множества q_1 , в вторая – q_2 . В результате простого соединения этих систем мы получаем систему, которая способна решать задачи из множества $(q_1 \cup q_2)$. Тогда как в результате интеграции мы получаем систему, которая способна решать значительно большее число задач, чем $(q_1 \cup q_2)$. Такое расширение числа решаемых задач

происходит за счет тех задач, для решения которых некоторые знания и навыки находятся в системе s_1 , а другие – в системе s_2 .

Таким образом, при интеграции интеллектуальных систем происходит приобретение нового качества на стыке интегрируемых систем, когда для решения некоторых задач одна часть необходимых знаний и/или умений находится в одной интегрируемой системе, а другая часть – в другой.

Очевидно, что теоретико-множественное объединение множества элементов одной семантической сети с другой приведет к знаковой конструкции, которая удовлетворяет всем требованиям, предъявляемым к семантическим сетям, кроме одного – в этой объединенной конструкции могут появиться пары синонимичных элементов, в которых один элемент принадлежит одной объединяемой семантической сети, а другой элемент – другой.

Процесс интеграции двух семантических сетей рассмотрим как систему следующих взаимодействующих подпроцессов, некоторые из них могут выполняться параллельно:

- приведение интегрируемых семантических сетей к унифицированному виду, т. е. представление (запись) их в SC-коде;
- приведение интегрируемых семантических сетей к бинарному виду и общему алфавиту (например, это можно сделать путем представления этих семантических сетей в виде конструкций рассмотренного выше SC-кода);
- согласование ключевых узлов и онтологий, используемых в интегрируемых sc-текстах. Очевидно, что полностью автоматизировать такое согласование невозможно, поэтому разработчикам интегрируемых фрагментов баз знаний и целых баз знаний необходимо уметь договариваться друг с другом;
- выделение в интегрируемых sc-текстах таких sc-элементов, которые имеют глобальные (уникальные) идентификаторы (внешние имена);
- выделение в интегрируемых sc-текстах sc-элементов, имеющих локальные идентификаторы вместе с областью действий каждого такого идентификатора. Область действия локального идентификатора – это такой фрагмент базы знаний, в рамках которого разные sc-элементы, имеющие этот локальный идентификатор, считаются синонимичными;
- склеивание sc-элементов, имеющих одинаковые глобальные (главные, уникальные, согласованные) идентификаторы (внешние имена). При этом sc-элементы, которые имеют совпадающие локальные (дополнительные, неосновные) идентификаторы рассматриваются как предположительно синонимичные sc-элементы;
- склеивание sc-элементов, имеющих одинаковые локальные идентификаторы, если каждый из этих sc-элементов принадлежит области действия своего локального идентификатора или действия локального идентификатора другого sc-элемента;
- склеивание sc-элементов на основании однозначности используемых алгебраических операций (в каждой алгебраической операции одним и тем же аргументам не может соответствовать два разных результата);
- склеивание sc-элементов на основании логических высказываний о существовании единственности (в процессе выявления пар синонимичных sc-элементов указанные высказывания могут порождаться путем логического вывода);
- склеивание кратных связей, принадлежащих отношениям:
 - не имеющим кратных связей;
 - имеющим кратные связи, но не для заданных типов компонентов (например, кратные связи принадлежности не могут выходить из знаков канторовских множеств).

Таким образом, интеграция семантических сетей, т. е. процесс погружения (понимания) одной семантической сети в другую, – это нетривиальный процесс рассуждений, направленный на выявление пар синонимичных элементов семантической сети на основе определенных знаний, имеющихся в базе знаний интеллектуальной системы.

От унифицированной семантической модели интеграции знаний (точнее sc-текстов) можно достаточно легко перейти к интеграции sc-моделей интеллектуальных систем, поскольку после интеграции sc-моделей баз знаний интегрируемых интеллектуальных систем интеграция соответствующих им наборов sc-агентов сводится к простому теоретико-множественному объединению указанных множеств sc-агентов.

12. Унификация и интеграция различных семантических моделей решения задач

Данный принцип позволяет обеспечить в рамках проектируемой интеллектуальной системы использование не только самых различных видов знаний, но и **моделей и стратегий решения задач**.

Для этого необходимо акцентировать внимание не столько на разработке новых моделей решения задач, сколько на унификации и интеграции в рамках проектируемых интеллектуальных систем уже разработанных и хорошо зарекомендовавших себя моделей (дедуктивных, индуктивных, абдуктивных, четких, нечетких, универсальных, специализированных и т. д.). Подчеркнем то, что в разных проектируемых интеллектуальных системах могут быть востребованы самые разные сочетания известных моделей и стратегий решения задач. Подавляющее число моделей представления знаний и решения задач не является альтернативным, они дополняют друг друга. Не составляют исключение и такие классы моделей, как фреймовые, логические, продукционные.

Рассмотренное выше понятие **вопроса** и его формализация является основой не только для информационно-поисковых моделей, но и для самых различных моделей решения задач. С точки зрения решателя задач вопрос – это **непроцедурная формулировка информационной цели**, т. е. декларативная формулировка определенной информационной цели, которая описывает спецификацию (свойства) той информации, которую требуется либо найти, если она уже присутствует в текущем состоянии базы знаний, либо построить (сгенерировать, вывести), если она отсутствует в текущем состоянии памяти. Таким образом, вопрос можно считать описанием целевого (требуемого) состояния обрабатываемой базы знаний, а точнее определенного фрагмента этой базы знаний. Вопрос также можно считать одним из видов метазнаний, описывающих (специфицирующих) наше незнание, т. е. наше знание о том, что мы не знаем, но хотели бы знать.

Вопросы могут инициироваться (задаваться) как пользователями, так и самой системой. Это означает, что в процессе обработки информации интеллектуальная система сама себе может задавать (генерировать, порождать) вопросы. Если инициирован некоторый *sc*-вопрос, то сначала активизируются соответствующие агенты информационного поиска с расчетом на то, что запрашиваемый ответ (или ответы) на указанный *sc*-вопрос уже присутствует в текущем состоянии базы знаний. И только после того, как информационно-поисковые *sc*-агенты обнаружат отсутствие ответа в текущем состоянии базы знаний, начинается работа решателя задач, направленная на генерацию (построение, порождение, вывод) требуемого ответа.

Кроме вопроса используется также и **процедурная формулировка информационной цели** – это описание (спецификация) определенного действия, которое требуется выполнить и которое направлено на преобразование (изменение состояния) базы знаний, хранимой в некоторой памяти. Указанное действие выполняется либо одним *sc*-агентом в случае, если это элементарное действие над *sc*-памятью, либо несколькими *sc*-агентами и порождает определенное событие (изменение состояния *sc*-памяти).

Для унификации различных моделей решения задач необходимо уточнить не только понятие **информационной цели**, но и понятие **информационной задачи**. Информационная задача включает формулировку информационной цели (описание того, что требуется) и ту хранимую в памяти информацию, которая семантически связана с заданной информационной целью, является ее контекстом (тем, что дано). В пределе контекстом информационной цели можно считать текущее состояние всей хранимой базы знаний.

Формальное рассмотрение контекстов различных информационных задач требует разработки специальных языковых средств, предназначенных для описания текущего состояния хранимой базы знаний, а точнее для описания грани между тем, что в текущем состоянии базы знаний известно и тем, что неизвестно. К числу таких языковых средств, в частности, относятся следующие ключевые узлы, являющиеся знаками нестационарных множеств (тех, которые в разные моменты времени могут иметь разные элементы):

- быть *sc*-дугой нечеткой принадлежности (такая *sc*-дуга связывает *sc*-узел, обозначающий некоторое множество, с *sc*-элементом, о котором нам в текущий момент времени не известно, принадлежит он указанному множеству или нет);

- быть построенным конечным множеством (у каждого такого множества в текущем состоянии базы знаний известны и явно указаны все его элементы);
- быть построенным высказыванием (для каждого такого высказывания в текущем состоянии базы знаний представлен не только его знак, но и полный текст);
- быть построенной внешней информационной конструкцией (файлом);
- быть аксиоматизированной формальной теорией;
- быть построенным рассуждением (обоснованием, доказательством, решением);
- быть построенной программой.

Более подробно унифицированные семантические модели решения задач и технология их проектирования рассмотрены в работе [18].

13. Унификация визуализации семантических сетей

В качестве основы организации графического пользовательского интерфейса используется язык унифицированного визуального представления абстрактных унифицированных семантических сетей в виде, близком к изоморфному.

Указанный язык графического изображения sc-текстов назван SCg-кодом (Semantic Code graphical). Подчеркнем, что следует четко отличать язык абстрактных унифицированных семантических сетей (SC-код), который абстрагируется от того, как должны быть физически представлены узлы и коннекторы текстов этого языка (sc-текстов), и язык графического изображения таких семантических сетей. То есть абстрактная семантическая сеть и ее рисунок – принципиально разные вещи.

С помощью SCg-кода осуществляется отображение на экране не только пользовательских сообщений, адресуемых системе или пользователю, но и всей остальной информации, необходимой для организации работы пользователя (прежде всего – это элементы управления интерфейсом). Такая унификация отображаемой пользователю информации дает возможность организовать взаимодействие пользователя с help-системой точно так же, как и его взаимодействие с основной (предметной) системой.

Трактовка элементов управления пользовательским интерфейсом как элементов отображаемого на экране SCg-текста позволяет унифицировать:


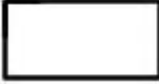
- 1) представление любой информации, отображаемой на экране;
- 2) способы инициирования различных вопросов, касающихся любой отображаемой на экране информации (в том числе и элементов управления);

Для того чтобы четко отделить те средства SCg-кода, которые обусловлены самим SC-кодом, от тех средств, которые обусловлены стремлением повысить наглядность SCg-текстов, введем ядро SCg-кода (или просто SCg-ядро), алфавит которого взаимно однозначно соответствует алфавиту SC-кода и соответственно тексты которого изоморфны семантически эквивалентным текстам SC-кода. Переход от SCg-ядра к SCg-коду заключается в ослаблении требований, предъявляемых к изображениям семантических сетей, в целях обеспечения удобства для человеческого восприятия. Такое ослабление осуществляется в следующих направлениях: вводится приписывание идентификаторов изображаемых sc-элементов, расширяется алфавит графических примитивов, допускается уникальное изображение некоторых sc-узлов, допускается синонимия scg-элементов, но при этом синонимичным элементам должны быть приписаны одинаковые идентификаторы, вводятся специальные графические средства, направленные на повышение наглядности (шинные линии, контуры).

Ниже приведем алфавит графических примитивов текстов SCg-ядра, который полностью соответствует алфавиту SC-кода.

Таблица 2

•	<i>scg-узел общего вида</i> является изображением sc-узла общего вида
— · — · — · —	<i>scg-ребро общего вида</i> является изображением sc-ребра общего вида
— · — · — · — →	<i>scg-дуга общего вида</i> является изображением sc-дуги 1-го вида

	scg-дуга базового вида является изображением sc-дуги 2-го вида (т. е. конкретной, постоянной существующей пары принадлежности)
	scg-рамка является ограничителем внешней информационной конструкции, визуально изображаемой в этой рамке, и является изображением sc-ссылки, обозначающей эту внешнюю информационную конструкцию

Переход от SCg-ядра к SCg-коду подразумевает:

- 1) расширение алфавита, позволяющее выделить различные семантические типы элементов семантических сетей;
- 2) введение идентификаторов и копирования sc-узлов;
- 3) введение контуров в целях минимизации числа явно изображаемых элементов;
- 4) введение шин в целях обеспечения более наглядного размещения текстов.

SCg-конструкции могут изображаться в двухмерном, трехмерном и многослойном виде.

Далее приведем пример перехода от предметной области к семантической сети, которая является структурной моделью фрагмента указанной предметной области.

Рассмотрим геометрическую предметную область, объектами исследования которой являются геометрические точки и геометрические фигуры самого различного вида, а предметом исследования – различные классы геометрических фигур (прямые плоскости, планарные геометрические фигуры, треугольники, многоугольники, окружности, геометрические тела и т. д.) и различные геометрические отношения, заданные на множестве геометрических фигур и точек (лежать между, конгруэнтность, подобие, равновеликость, длина, площадь, объем, быть вершиной ломаной линии, линейного угла, многоугольника, быть центром окружности, круга, сферы, шара, быть вписанной фигурой, граничной точкой, границей).

Рассмотрим небольшой фрагмент указанной предметной области, в состав которого входят четыре точки, один треугольник и одна окружность. Естественно-языковое описание структуры этого фрагмента выглядит следующим образом.

Точки $T_{чкА}$, $T_{чкВ}$, $T_{чкС}$ являются вершинами треугольника $Зугк(T_{чкА}, T_{чкВ}, T_{чкС})$. Через указанные точки также проходит окружность $Окр(T_{чкО}, T_{чкА})$, центром которой является точка $T_{чкО}$.

На языке геометрических чертежей данная информация представляется следующим образом.

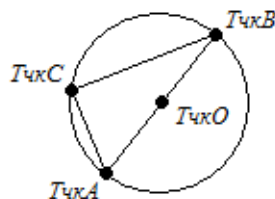


Рис. 4. Геометрический чертеж

На данном чертеже изображены следующие геометрические объекты:

- геометрические точки $T_{чкА}$, $T_{чкВ}$, $T_{чкС}$, $T_{чкО}$
- треугольник $Зугк(T_{чкА}, T_{чкВ}, T_{чкС})$
- окружность $Окр(T_{чкО}, T_{чкА})$

При этом на приведенном чертеже подписаны только точки, а изображенные фигуры выделяются на экране монитора яркостью или цветом при наведении указателя на их идентификаторы.

Уточним структуру рассматриваемого фрагмента геометрической предметной области. Элементами основного множества такого фрагмента являются следующие объекты: $T_{чкА}$, $T_{чкВ}$, $T_{чкС}$, $T_{чкО}$, $Зугк(T_{чкА}, T_{чкВ}, T_{чкС})$, $Окр(T_{чкО}, T_{чкА})$.

Выделенными классами объектов для рассматриваемого фрагмента являются: *треугольник* /*Понятие треугольника*/

точка /*Понятие геометрической точки*/
окружность /*Понятие окружности*/

Отношениями, используемыми в рассматриваемом фрагменте являются:
вершина_ /*быть вершиной*/
*центр** /*быть центром*/

Приведем описание структуры рассматриваемого фрагмента геометрической предметной области на теоретико-множественном языке.

точка \exists *ТчкаА*, *ТчкаВ*, *ТчкаС*, *Тчко*;

треугольник \exists *Зугк*(*ТчкаА*, *ТчкаВ*, *ТчкаС*);

окружность \exists *Окр*(*Тчко*, *ТчкаА*);

вершина_ \exists \langle *Зугк*(*ТчкаА*, *ТчкаВ*, *ТчкаС*) ,, *ТчкаА* \rangle ,,
 \langle *Зугк*(*ТчкаА*, *ТчкаВ*, *ТчкаС*) ,, *ТчкаВ* \rangle ,,
 \langle *Зугк*(*ТчкаА*, *ТчкаВ*, *ТчкаС*) ,, *ТчкаС* \rangle ;

*центр** \exists \langle *Окр*(*Тчко*, *ТчкаА*) ,, *Тчко* \rangle ;

Окр(*Тчко*, *ТчкаА*) \exists *ТчкаА*, *ТчкаВ*, *ТчкаС*;

Теперь приведем описание структуры рассматриваемого фрагмента геометрической предметной области в виде унифицированной семантической сети, принадлежащей SC-коду и графически изображенной в виде конструкции SCg-кода.

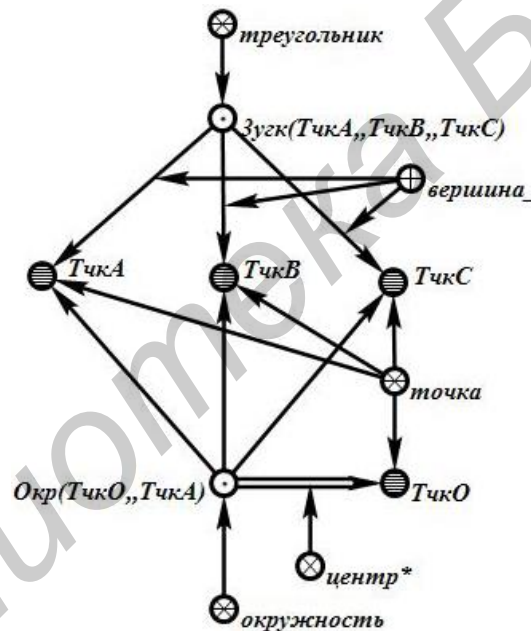


Рис. 5. Конструкция scg-кода

Уточним использованный здесь алфавит графических примитивов.

Таблица 3

	Метка знака выделенного класса объектов
	Метка знака отношения
	Метка знака отношения, являющегося подмножеством отношения принадлежности
	Метка знака объекта, принадлежащего основному множеству
	Метка знака бинарной ориентированной связи
	Метка знака связи стационарной принадлежности

Следует четко отличать:

- 1) ключевые узлы языка семантических сетей (в данном случае – ключевые узлы SC-кода);
- 2) дополнительные ключевые узлы, соответствующие сигнатурным элементам той предметной области, структурной моделью которой является семантическая сеть (в данном примере это ключевые узлы, обозначающие такие понятия, как *точка*, *окружность*, *треугольник*, *быть вершиной*, *быть центром*).

Заметим также, что кроме SCg-кода для внешнего представления абстрактных унифицированных семантических сетей используются также и другие языки:

- SCs-код, обеспечивающий представление унифицированных абстрактных семантических сетей (sc-текстов) в виде, близком к традиционным текстам;
- SCn-код, обеспечивающий гипертекстовое представление абстрактных sc-текстов, предназначенное для оформления исходных текстов баз знаний.

Более подробно различные языки внешнего представления абстрактных sc-текстов вместе с большим количеством примеров рассмотрены в работах [8] и в документации проекта OSTIS [71].

14. Унификация семантических моделей различных пользовательских интерфейсов

Пользовательский интерфейс интеллектуальной системы, построенной на основе предлагаемой технологии, рассматривается как **специализированная интеллектуальная система**, построенная по той же технологии и предназначенная для **трансляции адресуемых пользователю сообщений** с внутреннего абстрактного семантического языка представления знаний (SC-кода) на тот или иной внешний язык, тексты которого отображаются пользователю в удобном для него виде, а также для **трансляции пользовательских сообщений** с внешнего. В основе графических интерфейсов лежит SCg-код (Semantic Code graphical – один из возможных способов визуального представления SC-кода). Вся информация представляется в виде scg-текста (в том числе и элементы управления).

Трактовка пользовательских интерфейсов как интеллектуальных систем и унификация семантических моделей таких систем дает возможность:

- 1) унифицировать проектирование пользовательских интерфейсов;
- 2) легко наращивать возможности пользовательских интерфейсов;
- 3) легко интегрировать пользовательские интерфейсы с предметными (основными) интеллектуальными системами;
- 4) неограниченно использовать базу знаний предметных интеллектуальных систем для семантического анализа и понимания вводимой пользователем информации (в частности естественно-языковых текстов).

База знаний пользовательского интерфейса состоит из:

- 1) описания синтаксиса и семантики всех используемых внешних языков;
- 2) описания пользовательских команд;
- 3) описания принципов работы пользовательского интерфейса;
- 4) портрета пользователя.

Машина обработки знаний пользовательского интерфейса включает в себя следующие агенты:

- 1) обработки устройств ввода (мышь и клавиатура);
- 2) вывода конструкций пользователю на экран, в наушники и т. д.;
- 3) трансляции конструкций с sc-кода и обратно.

Структура графического пользовательского интерфейса интеллектуальной системы:

- представляется в виде главного окна, которое можно сравнить с рабочим столом в операционной системе;

- в рамках главного окна могут присутствовать дочерние окна;
- вся информация в рамках главного окна отображается с помощью SCg-кода;
- в главном окне присутствуют элементы управления, которые представлены с помощью SCg-кода. Выделены следующие классы элементов управления:
 - неатарные классы пользовательских команд;
 - атомарные классы пользовательских команд. Они делятся на подклассы:
 - команда-вопрос;
 - команда редактирования;
 - команда просмотра;
 - элементы управления изображающие объект действий;
 - конкретные пользовательские команды.

Диалог с системой осуществляется в рамках главного окна, путем обмена сообщениями. Сообщения формируются с помощью SCg-кода, путем инициирования пользовательских команд с помощью элементов управления. Сообщения системы пользователю выводятся также в главное окно с помощью SCg-кода. Частным видом такого сообщения является scg-рамка с содержимым, представленным на каком-то внешнем языке.

Структура естественно-языкового пользовательского интерфейса:

- Естественно-языковой пользовательский интерфейс
 - База знаний
 - База знаний естественно-языкового пользовательского интерфейса
 - Лингвистическая база знаний
 - Привязка к предметной базе знаний
 - Машина обработки знаний
 - Трансляторы
 - SC -> текст
 - Запрос-> SC
 - Операции
 - Поисковые операции
 - Операции вывода
 - Пользовательский интерфейс
 - Синтезаторы и анализаторы речи
 - Редакторы и просмотрщики текстов.

Более подробно унифицированные семантические модели пользовательских интерфейсов и технология их проектирования рассмотрены в работе [27].

15. Библиотека типовых семантически совместимых компонентов интеллектуальных систем и методика модульного проектирования интеллектуальных систем

В целях ускорения процесса проектирования интеллектуальных систем необходимо создать общую библиотеку многократно используемых семантически совместимых компонентов интеллектуальных систем, на основе которой разработать методику модульного (компонентного, сборочного) проектирования интеллектуальных систем.

В указанной библиотеке можно выделить следующие разделы (частные библиотеки):

- библиотека многократно используемых компонентов баз знаний. Прежде всего в эту библиотеку входят самые различные по содержанию, но семантически совместимые онтологии. Кроме того, сюда относятся различные стандартные наборы знаний, которыми должны владеть образованные интеллектуальные системы. К ним относятся знания по арифметике, теории множеств (каждая интеллектуальная система должна, по крайней мере, отличать элемент заданного множества от его подмножества), базовые знания по теории отношений (каждая интеллектуальная система должна уметь отличать бинарное отношение от многоместного отношения, понимать, что такое соответствие), логике (каждая интеллектуальная система должна понимать, что такое теория, высказывание, определение, переменная, должна отличать фактографическое высказывание от высказывания, не являющегося фактографическим, а также высказывание от логической

формулы, не являющейся высказыванием) и многие другие знания, востребованность которых может быть самой разной;

- библиотека компонентов семантических моделей информационного поиска. Сюда прежде всего входят различные информационно-поисковые агенты;
- библиотека компонентов семантических моделей интеграции знаний и машин обработки знаний;
- библиотека интерпретаторов программ, соответствующих различным языкам программирования;
- библиотека различных стратегий решения задач, моделей решения задач и агентов, входящих в состав таких моделей;
- библиотека компонентов пользовательских интерфейсов.

Все компоненты, включаемые в состав общей библиотеки компонентов интеллектуальных систем, оформляются как компоненты интеллектуальной собственности (intellectual property), поэтому будем их также называть ip-компонентами.

Особо подчеркнем то, что модульное проектирование интеллектуальных систем возможно только в том случае, если отбор компонентов, включаемых в состав рассмотренной библиотеки, будет осуществляться на основе тщательного анализа качества этих компонентов. Одним из важнейших критериев такого анализа является семантическая совместимость анализируемых компонентов со всеми компонентами, имеющимися в текущей версии библиотеки.

Для обеспечения семантической совместимости таких компонентов интеллектуальных систем, которые являются унифицированными семантическими моделями (sc-моделями знаний, sc-моделями машин обработки знаний, sc-агентов, sc-моделями интеллектуальных подсистем), необходимо согласовать семантику (смысл) всех используемых ключевых узлов и глобальные идентификаторы ключевых узлов, используемых в разных компонентах. После этого интеграция всех компонентов, входящих в состав библиотеки, и в любых комбинациях осуществляется автоматически без вмешательства разработчика.

16. Платформенно-независимый характер проектирования интеллектуальных систем

Для максимальной платформенной независимости технологии необходимо обеспечить четкое разделение процесса проектирования формального описания логико-семантической модели разрабатываемой интеллектуальной системы от процесса реализации (интерпретации) этой модели на той или иной платформе.

Подчеркнем при этом следующее. Если каждой интеллектуальной системе соответствует своя уникальная логико-семантическая модель, то каждый интерпретатор абстрактных логико-семантических моделей интеллектуальных систем должен обеспечивать интерпретацию целого класса таких моделей, а в идеале – интерпретацию любой такой модели. Следовательно, разработка указанных интерпретаторов может осуществляться абсолютно независимо от разработки логико-семантических моделей конкретных интеллектуальных систем.

Таким образом, SC-код, обеспечивающий унификацию семантического представления любых знаний, вместе с языком SCP, обеспечивающим унификацию формального описания агентов, работающих над семантической памятью, являясь средством унификации логико-семантических моделей интеллектуальных систем, выполняет в рамках предлагаемой технологии роль, аналогичную той, которую выполняет язык VHDL в современных микроэлектронных технологиях. В лице SC-кода и языка SCP мы имеем стандарт полного формального описания логико-семантических моделей интеллектуальных систем, обеспечивающий независимость проектирования абстрактных логико-семантических моделей конкретных интеллектуальных систем от разработки различных вариантов реализации (т. е. их интерпретации на различных платформах). Такой стандарт является своего рода разделом между полным платформенно-независимым описанием интеллектуальной системы (абстрактной логико-семантической моделью) и платформенно-зависимой реализацией (интерпретацией) этой абстрактной модели.

Полностью построенная абстрактная логико-семантическая модель проектируемой интеллектуальной системы:

- 1) является открытой, поскольку ее можно легко пополнять новыми знаниями и навыками, интегрируя их в текущую версию модели;
- 2) концентрирует внимание на семантических аспектах функционирования интеллектуальной системы и не содержит никаких лишних деталей, обусловленных тем или иным способом ее технической реализации (интерпретации);
- 3) является абстрактным инвариантом целого множества самых различных способов ее технологической реализации (в том числе и с помощью принципиально новых компьютеров).

Разработка прототипа интеллектуальной системы завершается разработкой полной sc-модели этой системы, которая записывается в виде исходного текста с использованием таких языковых средств, как SCg-код, SCs-код, SCn-код. После этого разработчик выбирает один из универсальных вариантов интерпретации (реализации) sc-моделей, загружает разработанные им исходные тексты в выбранный интерпретатор и получает прототип, пригодный для опытной эксплуатации и последующего совершенствования.

Если же после этого разработчиков интеллектуальной системы что-то не устраивает в выбранном варианте интерпретации sc-моделей (в частности производительность), должна существовать достаточно продуманная методика совершенствования выбранного варианта интерпретатора sc-моделей интеллектуальных систем. Очевидно, что для каждого варианта интерпретации sc-моделей интеллектуальных систем указанная методика будет иметь свои особенности.

Следовательно, нижние уровни детализации проектируемых интеллектуальных систем в отличие от верхнего (логико-семантического) являются платформенно-зависимыми. Можно говорить о различных модификациях технологии проектирования интеллектуальных систем, соответствующих разным платформам. Напомним при этом, что основная трудоемкость проектирования интеллектуальных систем, полностью определяющая уровень ее возможностей (уровень знаний и навыков) концентрируется именно на 1-м этапе проектирования – на разработке ее абстрактной логико-семантической модели.

Таким образом, проектирование интеллектуальной системы можно организовать как два следующих самостоятельных процесса, выполняемых одновременно и независимо друг от друга:

- 1) процесс разработки абстрактной унифицированной логико-семантической модели проектируемой интеллектуальной системы;
- 2) процесс совершенствования выбранного интерпретатора абстрактных унифицированных логико-семантических моделей интеллектуальных систем.

Заметим, что сама идея обеспечения кросс-платформенной разработки компьютерных систем путем внедрения формального языка, обеспечивающего описание абстрактных (логических) моделей этих систем, не нова. Существует целый ряд кросс-платформенных технологий. Различают их по:

- 1) классу разрабатываемых компьютерных систем;
- 2) свойствам используемых абстрактных моделей компьютерных систем;
- 3) достоинствам технологии разработки самих этих абстрактных моделей.

Вопросы программной реализации и, в частности, web-ориентированной реализации унифицированных логико-семантических моделей интеллектуальных систем рассмотрены в работе [25].

17. Семантический ассоциативный параллельный компьютер

Обеспечить возможность реализации унифицированных логико-семантических моделей интеллектуальных систем на **семантических ассоциативных параллельных компьютерах**, специально ориентированных на аппаратную реализацию таких моделей.

Очевидно, что для указанных компьютеров базовый графовый язык программирования (язык SCP) является их ассемблером, т. е. аппаратно интерпретируемым языком программирования.

В связи с проблемой создания компьютеров, ориентированных на обработку знаний, необходимо отметить следующее:

- 1) в таких компьютерах принципиально важна поддержка именно параллельной обработки знаний;
- 2) опыт использования параллельных компьютеров показывает, что эффективное их использование, предполагающее разработку качественных параллельных программ, требует особой профессиональной подготовки и высокой квалификации. Мир параллельного программирования требует особой культуры и стиля мышления. Еще более серьезная профессиональная подготовка необходима для разработки параллельных программ, ориентированных на обработку знаний и использующих ассоциативный доступ к обрабатываемой информации;
- 3) уровень развития микроэлектронных технологий сейчас позволяет достаточно быстро реализовывать самые смелые компьютерные архитектуры и модели обработки информации. Необходима только четкая постановка задачи;
- 4) созданию параллельных компьютеров для обработки знаний должно предшествовать создание **технологии** проектирования интеллектуальных систем, в основе которой лежат те модели параллельной обработки знаний, которые будут аппаратно поддерживаться в указанных компьютерах. Иначе мы получим груду талантливо сделанного железа, эффективность использования которого будет весьма низкой. Это главная причина неудач такого рода проектов;
- 5) предлагаемая технология проектирования интеллектуальных систем как раз и предполагает последовательное выполнение следующих этапов:
 - разработка технологии проектирования абстрактных унифицированных логико-семантических моделей интеллектуальных систем;
 - разработка нескольких вариантов программной реализации абстрактных унифицированных логико-семантических моделей интеллектуальных систем, выполненных на современных компьютерах;
 - разработка, эксплуатация достаточно большого количества прикладных интеллектуальных систем и совершенствование технологии проектирования интеллектуальных систем на основе приобретенного опыта;
 - и только после этого разработка семантического ассоциативного компьютера, появление которого не отменит абсолютно ничего, сделанного ранее. Просто появится еще один, но уже аппаратный вариант реализации абстрактных унифицированных логико-семантических моделей интеллектуальных систем, применение которого для уже разработанных абстрактных унифицированных логико-семантических моделей самых различных прикладных систем для конечных пользователей этих интеллектуальных систем абсолютно ничего не изменит, кроме существенного повышения быстродействия.

Рассматривая абстрактную sc-машину обработки знаний на самом верхнем уровне, мы не детализируем внутреннее устройство sc-агентов обработки знаний. Разработав язык SCP, мы получили возможность формально описывать поведение sc-агентов обработки знаний. Если трактовать язык SCP как ассемблер семантического ассоциативного компьютера, то проектирование этого компьютера можно рассматривать как формальный переход к sc-машинам более низкого уровня, обеспечивающим интерпретацию sc-машин более высокого уровня. Существенным здесь является то, что при этом мы не выходим за пределы класса абстрактных sc-машин. Просто вводится последовательность sc-языков программирования все более и более низкого уровня, каждый из которых обеспечивает формальное описание sc-агентов, входящих в состав sc-машины, интерпретирующей программы непосредственно предшествующего ему sc-языка программирования более высокого уровня (см. гл. 9). При этом число таких уровней, т. е. специальных sc-языков программирования (их можно назвать sc-языками микропрограммирования) должно быть таким, которое необходимо для доведения формального описания sc-машин до уровня детализации, позволяющего перейти от соответствующего абстрактного языка микропрограммирования к формальному описанию цифровой аппаратуры на языке VHDL.

Архитектуру аппаратной реализации семантических моделей обработки знаний можно рассматривать как иерархию абстрактных машин, описывающих переход от агентов, имеющих доступ ко всей семантической памяти, к агентам, имеющим доступ только к своей

семантической окрестности и, в конечном счете, взаимодействующим только со своими семантическими соседями.

Аппаратная интерпретация абстрактных sc-машин предполагает создание реконфигурируемой памяти с распределенными в ней процессорными элементами. Такую интеграцию памяти и процессора будем называть **процессоро-памятью**. Реконфигурируемость (структурная перестраиваемость) памяти может быть обеспечена коммутационной средой для процессорных элементов. Можно рассматривать целый ряд подходов к реализации реконфигурируемой семантической ассоциативной процессоро-памяти. В частности, процессорным элементам можно ставить в соответствие узлы обрабатываемых унифицированных семантических сетей, а коммутируемым каналам связи между процессорными элементами – коннекторы этой семантической сети. В этом случае текущее состояние конфигурации коммутируемых каналов связи будет полностью соответствовать текущему состоянию конфигурации обрабатываемой семантической сети. Следовательно, память превращается из пассивного хранилища байтов в коммутационную среду между процессорными элементами.

18. Встроенные подсистемы интеллектуальных систем, обеспечивающие их эффективную эксплуатацию и эволюцию

Каждая проектируемая интеллектуальная система трактуется как **результат интеграции следующих интеллектуальных подсистем**:

- предметной (основной) интеллектуальной системы;
- интеллектуального пользовательского интерфейса;
- интеллектуальной подсистемы адаптивного управления диалогом с конечным пользователем;
- интеллектуальной help-системы для информационного обслуживания и обучения конечных пользователей предметной интеллектуальной системы, которые, начиная работать с системой, не обязаны иметь сразу высокую квалификацию;
- интеллектуальных систем управления проектированием интеллектуальной системы, которая координирует деятельность разработчиков предметной интеллектуальной системы [9];
- интеллектуальных систем управления информационной безопасностью предметной интеллектуальной системы.

Подчеркнем, что для обеспечения интегрируемости (семантической совместимости) перечисленных интеллектуальных систем они должны проектироваться на основе одной и той же технологии.

Таким образом, проектируя каждую интеллектуальную систему, необходимо одновременно проектировать следующие подсистемы:

- осуществляющую информационное обслуживание и обучение конечных пользователей данной интеллектуальной системы, т. е. фактически являющуюся оформлением документации по эксплуатации системы в виде интеллектуальной справочной и обучающей системы. Это существенно расширит контингент конечных пользователей, повысит эффективность эксплуатации системы и существенно упростит эту эксплуатацию;
- обеспечивающую координацию разработчиков проектируемой интеллектуальной системы, поскольку разработка (совершенствование) системы продолжается в ходе ее эксплуатации и требует создания специальных методов и компьютерных средств постоянного совершенствования предметной интеллектуальной системы непосредственно в ходе ее эксплуатации. Это существенно отодвинет срок ее морального старения;
- обеспечивающую управление информационной безопасностью проектируемой интеллектуальной системы.

Если подсистема управления проектированием интеллектуальной системы будет создаваться действительно как интеллектуальная система, интегрируемая с основной (предметной) интеллектуальной системой, то в перспективе она может стать не только координатором деятельности разработчиков, но и самостоятельным субъектом проектирования, способным тестировать, диагностировать, анализировать как основную проектируемую интеллектуальную систему, так и саму себя.

19. Доступность и открытость технологии

Обеспечить максимально возможное **расширение контингента разработчиков** интеллектуальных систем, использующих предлагаемую технологию, за счет максимальной доступности этой технологии и открытого характера ее развития.

Если технология проектирования интеллектуальных систем ориентируется на широкое, массовое распространение и на интенсивное собственное развитие, опирающееся на накапливаемый опыт ее использования, она должна быть доступной и открытой. Это означает:

- свободный доступ ко всей документации и основанным средствам автоматизации (компьютерной поддержки) проектирования интеллектуальных систем;
- открытость исходных текстов всех основных средств компьютерной поддержки проектирования интеллектуальных систем, а также основных многократно используемых (типовых) компонентов интеллектуальных систем;
- открытость исходных текстов всех предварительных проектов прикладных интеллектуальных систем, выполняющих роль образцовых, показательных проектов;
- открытый характер организации (project-менеджмента) процесса развития технологии, имеющий форму открытого (open source) проекта, участником которого может быть любой желающий, в том числе, и пользователь этой технологии, указывающий на различные ошибки и высказывающий различные пожелания.

В завершение отметим следующие характеристики открытого характера предлагаемой технологии:

- не является препятствием для реализации коммерческих интересов, связанных с этой технологией. Так, например, на коммерческой основе могут создаваться и предоставляться:
 - 1) самые различные прикладные интеллектуальные системы;
 - 2) некоторые варианты реализации различных многократно используемых (типовых) компонентов интеллектуальных систем;
 - 3) некоторые варианты реализации интерпретатора абстрактных семантических логико-семантических моделей интеллектуальных систем, в частности, различные варианты построения семантических ассоциативных параллельных компьютеров;
- при грамотном использовании фактора открытости способствует обеспечению информационной безопасности как самой технологии, так и прикладных интеллектуальных систем, созданных на ее основе;
- может быть эффективно реализован только на базе **технологии облачных вычислений**, в рамках которой вся предлагаемая технология проектирования интеллектуальных систем рассматривается как некий Internet-сервис [9].

20. Эволюционная методика проектирования

Технология OSTIS представляет собой комплекс совместимых частных технологий, ориентированных на проектирование различных компонентов интеллектуальных систем:

- на проектирование баз знаний и, в частности,
 - на проектирование онтологий
 - на проектирование программ (в широком смысле трактовки этого вида знаний)
- на проектирование машин обработки знаний (решателей задач) и, в частности,
 - на проектирование машин дедуктивного вывода
 - на проектирование машин, обеспечивающих интерпретацию самых различных неклассических логик
- на проектирование пользовательских интерфейсов и, в частности,
 - на проектирование интеллектуальных графических интерфейсов
 - на проектирование естественно-языковых пользовательских интерфейсов (в текстовой и речевой форме).

Кроме того, она представляет собой комплекс совместимых частных технологий, ориентированных на проектирование различных классов интеллектуальных систем:

- справочных (интеллектуальных систем информационного обслуживания пользователей в различных предметных областях);
- обучающих (систем управления обучением);
- help-систем;
- автоматизированного проектирования;
- управления проектами;
- поддержки принятия решений;
- геоинформационных и т. д.

Как вся комплексная технология OSTIS в целом, так и каждая, входящая в нее частная технология включают в себя следующие компоненты:

- достаточно детальную теорию проектируемого класса систем (это формальное уточнение того, как устроены проектируемые системы);
- систематизированную библиотеку типовых многократно используемых компонентов проектируемых систем;
- инструментальные средства (средства автоматизации) проектирования;
- методику проектирования;
- методику обучения проектированию;
- интеллектуальную help-подсистему для информационного обслуживания проектировщиков;
- интеллектуальную подсистему управления проектом (подсистему автоматизации взаимодействия разработчиков).

Наконец, технология OSTIS представляет собой интеллектуальную метасистему (IMS.OSTIS), которая предназначена для поддержки проектирования интеллектуальных систем (в том числе и самой себя) и которая полностью отражает текущее состояние технологии OSTIS и поддерживает процесс постоянного совершенствования этой технологии (см. разд. 21).

Предлагается использовать **методику поэтапного эволюционного проектирования** интеллектуальных систем.

Указанная методика предполагает:

- быстрое проектирование;
- скорейшее введение в эксплуатацию первых версий проектируемой системы с минимальными, но практически полезными возможностями;
- эволюционное поэтапное совершенствование проектируемой интеллектуальной системы путем ее расширения новыми знаниями и навыками непосредственно в ходе эксплуатации интеллектуальной системы и активным привлечением ее конечных пользователей.

С формальной точки зрения проектирование унифицированной логико-семантической модели (sc-модели) интеллектуальной системы в конечном счете сводится к проектированию sc-модели **базы знаний** этой интеллектуальной системы, поскольку scr-программы, описывающие поведение sc-агентов, можно рассматривать как часть базы знаний. Таким образом, проектируемая база знаний включает в себя:

- базу знаний предметной (основной) интеллектуальной системы;
- тексты всех scr-программ, описывающих поведение sc-агентов;
- текст документации, представленный в виде базы знаний интеллектуальной help-системы, обеспечивающей всестороннее информационное обслуживание пользователей проектируемой интеллектуальной системы.

Начальный этап проектирования базы знаний интеллектуальной системы – это уточнение иерархической системы предметных областей, которые должны быть описаны в проектируемой базе знаний. Каждой такой предметной области ставится в соответствие определенный раздел проектируемой базы знаний. Среди выделенных разделов проектируемой базы знаний имеются разделы, которые делятся (декомпозируются) на подразделы, а также атомарные (недекопируемые) разделы. Далее процесс

проектирования всей базы знаний сводится к проектированию каждого ее атомарного раздела с последующей их интеграцией в единую базу знаний.

В целом начальную стадию проектирования всей интеллектуальной системы на основе предлагаемой технологии условно разбивают на следующие четыре этапа:

1. Разработка 1-й версии интеллектуальной системы, которая включает в себя:
 - 1-ю версию ее базы знаний;
 - типовое ядро интеллектуальной информационно-поисковой машины, которое входит в состав библиотеки многократно используемых компонентов интеллектуальных систем;
 - типовое ядро интеллектуального решателя, которое входит в состав библиотеки многократно используемых компонентов интеллектуальных систем;
 - типовое ядро пользовательского интерфейса, которое входит в состав библиотеки многократно используемых компонентов интеллектуальных систем.Разработанная 1-я версия интеллектуальной системы уже обладает определенной целостностью, ее можно тестировать и запускать в предварительную опытную эксплуатацию.
2. Разработка 2-й версии интеллектуальной системы, которая включает в себя:
 - 2-ю версию ее базы знаний;
 - 1-ю версию ее информационно-поисковой машины;
 - типовое ядро ее интеллектуального решателя;
 - типовое ядро ее пользовательского интерфейса.
3. Разработка 3-й версии интеллектуальной системы, включающей в себя:
 - 3-ю версию ее базы знаний;
 - 2-ю версию ее информационно-поисковой машины;
 - 1-ю версию ее интеллектуального решателя;
 - типовое ядро ее пользовательского интерфейса.
4. Разработка 4-й версии интеллектуальной систем, включающей в себя:
 - 4-ю версию ее базы знаний;
 - 3-ю версию ее информационно-поисковой машины;
 - 2-ю версию ее интеллектуального решателя;
 - 1-ю версию ее пользовательского интерфейса.

Дальнейшее развитие проектируемой интеллектуальной системы может акцентировать внимание на самых разных направлениях, приоритетность которых определяется самим приложением.

Более подробно методика эволюционного коллективного проектирования унифицированных семантических моделей интеллектуальных систем на основе содержательной структуризации знаний (см. разд. 6) описана в работе [13].

21. Реализация предлагаемой технологии в виде интеллектуальной метасистемы

Данный принцип позволяет реализовать предлагаемую технологию как **интеллектуальную метасистему, ориентированную на поддержку проектирования интеллектуальных систем**, построенную по тем же самым принципам (т. е. по той же технологии), что и интеллектуальные системы, разрабатываемые на ее основе.

Указанная интеллектуальная система должна включать в себя:

- теорию (принципы построения) проектируемых интеллектуальных систем, которая входит в состав базы знаний метасистемы;
- структурированную библиотеку типовых многократно используемых компонентов (ip-компонентов) интеллектуальных систем, которая входит в состав базы знаний рассматриваемой метасистемы;
- инструментальные средства автоматизации синтеза, анализа и имитационного моделирования проектируемых интеллектуальных систем и их компонентов (это подсистема интеллектуальной метасистемы, ориентированная на решение задач проектирования интеллектуальных систем);

- интеллектуальную help-систему, являющуюся подсистемой рассматриваемой интеллектуальной метасистемы, ориентированной на информационное обслуживание и обучение разработчиков интеллектуальных систем;
- методику проектирования интеллектуальных систем, которая оформляется как часть базы знаний метасистемы;
- методику обучения проектированию интеллектуальной системы, которая также является частью базы знаний метасистемы;
- интеллектуальную подсистему управления проектированием самой метасистемы;
- интеллектуальную подсистему управления информационной безопасностью метасистемы;
- семейство различных вариантов реализации интерпретаторов унифицированных абстрактных логико-семантических моделей интеллектуальных систем.

Учитывая рассматриваемые выше принципы построения предлагаемой нами технологии, ее назвали Открытой Семантической Технологией проектирования Интеллектуальных Систем (Open Semantic Technology for Intelligent Systems – OSTIS). Также ее можно было бы назвать **SC-технологией**, поскольку основой этой технологии является SC-код. Соответственно этому интеллектуальную метасистему, ориентированную на поддержку проектирования интеллектуальных систем, будем называть **метасистемой OSTIS**.

В интеллектуальной метасистеме OSTIS можно выделить целый ряд подсистем, ориентированных на поддержку проектирования различных компонентов интеллектуальных систем, таких, как:

- базы знаний и различные фрагменты баз знаний (онтологии, формальные теории, программы);
- информационно-поисковые машины, машины интеграции знаний, решатели задач;
- пользовательские интерфейсы (графические, естественно-языковые, мультимодальные).

В интеллектуальной метасистеме OSTIS можно также выделить семейство интеллектуальных подсистем, ориентированных на поддержку проектирования различных классов интеллектуальных систем, таких, как:

- справочные системы (системы информационного обслуживания);
- обучающие системы (имеющие подсистемы интеллектуального управления обучением);
- help-системы для пользователей различных компьютерных систем;
- системы автоматизированного проектирования;
- системы управления проектами.

Кроме того, нами предлагается семейство совместимых технологий проектирования интеллектуальных систем, а именно:

- Семейство совместимых семантических технологий проектирования различных компонентов интеллектуальных систем
 - Семантическая технология проектирования баз знаний
 - Семантическая технология проектирования программ, входящих в состав баз знаний и ориентированных на обработку этих знаний
 - Семантическая технология проектирования нейронных сетей, входящих в состав баз знаний и ориентированных на обработку этих знаний
 - Семантическая технология проектирования машин обработки знаний (интеллектуальных информационно-поисковых машин и интеллектуальных решателей задач)
 - Семантическая технология проектирования пользовательских интерфейсов интеллектуальных систем
 - Семантическая технология проектирования естественно-языковых пользовательских интерфейсов интеллектуальных систем
- Семейство совместимых семантических технологий проектирования различных классов интеллектуальных систем
 - Комплексная семантическая технология проектирования интеллектуальных справочных систем (интеллектуальных систем информационного обслуживания пользователей)
 - Комплексная семантическая технология проектирования интеллектуальных обучающих систем

- Комплексная семантическая технология проектирования интеллектуальных help-систем
- Комплексная семантическая технология проектирования интеллектуальных систем поддержки принятия решений
- Комплексная семантическая технология проектирования интеллектуальных геоинформационных систем.

Приведенный список частных технологий может быть существенно расширен. Подробнее об этом см. в документации открытого проекта OSTIS [71].

ЗАКЛЮЧЕНИЕ

В рассмотренной в учебно-методическом пособии технологии OSTIS существенным являются не сами рассмотренные выше принципы, некоторые из них выглядят очевидными и бесспорными, а весь целостный комплекс этих принципов и их максимально возможная согласованность.

Ключевыми проблемами, решение которых лежит в основе предлагаемой технологии, являются:

- обеспечение семантической совместимости (интегрируемости) различных моделей представления и обработки знаний;
- создание общей теории абстрактных семантических моделей интеллектуальных систем, не противопоставляя, а интегрируя самые различные подходы;
- обеспечение максимально возможной независимости интеллектуальных систем от многообразия вариантов и платформ их технической реализации (в т. ч. и от будущих компьютеров, специально ориентированных на аппаратную поддержку обработки знаний).

Таблица 4

sc-модели интеллектуальных систем:			
<ul style="list-style-type: none"> • пользовательских интерфейсов; • help-систем; • подсистем управления проектами; • систем поддержки проектирования; • подсистем управления информационной безопасностью 			
sc-модели решателей задач			
sc-модели информационного поиска		sc-модели интеграции знаний	
sc-модели баз знаний			
язык SCK		унифицированные идентификаторы sc-элементов	
sc-язык онтологий	sc-языки целей, вопросов и задач	логический sc-язык	sc-языки программирования
язык SCP			
scp-машина			
SC-код	SCg-код	SCs-код	SCn-код
программные интерпретаторы scp-машины		scp-компьютеры	
Internet	Локальные платформы		

Библиографический список

1. [Айзерман и др., 1988] Айзерман, М. А. Динамический подход к анализу структур, описываемых графами (основы графодинамики) / М. А. Айзерман [и др.] // Исследования по теории структур. – М. : Наука, 1988. – С. 5–76.
2. [Бениаминов, 1988] Бениаминов, Е. М. Основания категорного подхода к представлению знаний. Категорные средства / Е. М. Бениаминов // Изв. АН СССР. Техническая кибернетика. – 1988. – №2. – С. 21–33.
3. [Борщев, 1983] Борщев, В. Б. Схемы на клубных системах и вегетативная машина / В. Б. Борщев // Семиотика и информатика. – 1983. – Вып. 22. – С. 3–44.
4. [Вагин и др., 2008] Вагин, В. Н. Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин [и др.]. – М. : ФИЗМАТЛИТ, 2008.
5. [Гаврилова и др., 2000] Гаврилова, Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб. : Питер, 2000.
6. [Гастев, 1975] Гастев, Ю. А. Гомоморфизмы и модели. Логико-алгебраические аспекты моделирования / Ю. А. Гастев. – М. : Наука, 1975.
7. [Гладун, 1994] Гладун, В. П. Процессы формирования новых знаний / В. П. Гладун. – София : Педагог, 1994.
8. [Голенков и др., 2001] Голенков, В. В. Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков [и др.]. – Минск : БГУИР, 2001.
9. [Грибова и др., 2011] Грибова, В. В. Автоматизация разработки пользовательских интерфейсов с динамическими данными / В. В. Грибова, Н. Н. Черкезишвили // Открытые семантические технологии проектирования интеллектуальных систем: материалы междунар. II науч.-техн. конф., Минск, 16–18 февраля 2012 / БГУИР; редкол. : В. В. Голенков [и др.]. – Минск, 2012. – С. 287–292.
10. [Гуляева, 1989] Гуляева, Д. М. Решение прикладных задач на расширенных семантических сетях / Д. М. Гуляева // Математическое обеспечение ЭВМ и систем программирования: учеб. пособие / Д. М. Гуляева. – М., 1989.
11. [Гулякина, 2012] Гулякина, Н. А. Языки и технологии программирования, ориентированные на обработку семантических сетей / Н. А. Гулякина, О. В. Пивоварчик, Д. А. Лазуркин // Открытые семантические технологии проектирования интеллектуальных систем: материалы II междунар. науч.-техн. конф., Минск, 16–18 февраля 2012 / БГУИР; редкол. : В. В. Голенков [и др.]. – Минск, 2012. – С. 221–228.
12. [Гусаков и др., 1981] Гусаков, В. Я. Динамические алгебраические системы как математическая модель банка данных / В. Я. Гусаков, С. М. Гусакова // Семиотика и информатика. – 1981. – Вып. 17. – С. 43–52.
13. [Давыденко, 2012] Давыденко, И. Т. Комплексная методика проектирования семантических моделей интеллектуальных справочных систем / И. Т. Давыденко // Открытые семантические технологии проектирования интеллектуальных систем: материалы II междунар. науч.-техн. конф., Минск, 16–18 февраля 2012 / БГУИР; редкол. : В. В. Голенков [и др.]. – Минск, 2012. – С. 457–466.
14. [Евгенов, 2008] Евгенов, Г. Б. Технология создания многоагентных прикладных систем / Г. Б. Евгенов // Одиннадцатая нац. конф. по искусственному интеллекту : тр. конф., Дубна, 29 сентября – 3 октября 2008 г. – М., 2008. – С. 306 – 312.
15. [Епифанов, 1984] Епифанов, М. Е. Индуктивное обобщение в ассоциативных сетях / М. Е. Епифанов // Изв. АН СССР. Техническая кибернетика. – 1984. – №5 – С. 132–146.
16. [Ефимова и др., 1988] Ефимова, С. М. Поиск в базах знаний, опирающихся на модель П-графов, и его аппаратная реализация на основе метода M^3 / С. М. Ефимова, Е. В. Суворова. – М. : Вычислительный центр АН СССР, 1988.
17. [Загорулько, 1988] Загорулько, Ю. А. Технология конструирования средств обработки знаний на основе семантических сетей. Средства спецификации и настройки / Ю. А. Загорулько. – Новосибирск, 1988.
18. [Заливако и др., 2012] Заливако, С. С. Семантическая технология компонентного проектирования интеллектуальных решателей задач / С. С. Заливако, Д. В. Шункевич // Открытые семантические технологии проектирования интеллектуальных систем:

- материалы II междунар. науч.-техн. конф., Минск, 16–18 февраля 2012 / БГУИР; редкол. : В. В. Голенков [и др.]. – Минск, 2012. – С. 297–314.
19. **[Ивашенко, 2012]** Ивашенко, В. П. Семантические модели и средства интеграции и отладки баз знаний / В. П. Ивашенко // Открытые семантические технологии проектирования интеллектуальных систем: материалы II междунар. науч.-техн. конф., Минск, 16–18 февраля 2012 / БГУИР; редкол. : В. В. Голенков [и др.]. – Минск, 2012. – С. 193–204.
 20. **[Калиниченко, 1983]** Калиниченко, Л. А. Методы и средства интеграции неоднородных баз данных / Л. А. Калиниченко. – М.: Наука, 1983.
 21. **[Кандрашина и др., 1989]** Кандрашина, Е. Ю. Представление знаний о времени и пространстве в интеллектуальных системах / Е. Ю. Кандрашина, Л. В. Литвинцева, Д. А. Поспелов. – М.: Наука, 1989.
 22. **[Карабеков и др., 2008]** Карабеков, Б. А. Система «Бинарная Модель Знаний» как инструмент для концептуального моделирования бизнес-процессов // Одиннадцатая нац. конф. по искусственному интеллекту : тр. конф., Дубна, 29 сентября–3 октября 2008 г. – М., 2008. – С. 282–291.
 23. **[Касьянов, 2003]** Касьянов, В. Н. Графы в программировании: обработка, визуализация и применение / В. Н. Касьянов, В. А. Евстигнеев. – СПб.: BHV – Санкт-Петербург, 2003.
 24. **[Клещев, 1986]** Клещев, А. С. Семантические порождающие модели. Общая точка зрения на фреймы и продукции в экспертных системах / А. С. Клещев. – Владивосток, 1986.
 25. **[Колб, 2012]** Колб, Д. Г. Web-ориентированная реализация семантических моделей интеллектуальных систем / Д. Г. Колб // Открытые семантические технологии проектирования интеллектуальных систем: материалы II междунар. науч.-техн. конф., Минск, 16–18 февраля 2012 / БГУИР; редкол. : В. В. Голенков [и др.]. – Минск, 2012. – С. 111–122.
 26. **[Колмогоров, 1958]** Колмогоров, А. Н. К определению алгоритма / А. Н. Колмогоров, В. А. Успенский // Успехи математических наук. – 1958. – Т.13. – №4(82). – С. 3–28.
 27. **[Корончик, 2012]** Корончик, Д. Н. Семантические модели мультимодальных пользовательских интерфейсов и семантическая технология их проектирования / Д. Н. Корончик // Открытые семантические технологии проектирования интеллектуальных систем: материалы II междунар. науч.-техн. конф., Минск, 16–18 февраля 2012 / БГУИР; редкол. : В. В. Голенков [и др.]. – Минск, 2012. – С. 339–346.
 28. **[Котов и др., 1966]** Котов, В. Е. Асинхронные вычислительные процессы над общей памятью / В. Е. Котов, А. С. Нариньяни // Кибернетика. – 1966. – № 3. – С. 64–71.
 29. **[Кузнецов В.Е., 1989]** Кузнецов, В. Е. Представление в ЭВМ неформальных процедур / В. Е. Кузнецов. – М.: Наука, 1989.
 30. **[Кузнецов И.П., 1986]** Кузнецов, И. П. Семантические представления / И. П. Кузнецов. – М.: Наука, 1986.
 31. **[Лозовский, 1984]** Лозовский, В. С. Семантические сети / В. С. Лозовский // Представление знаний в человеко-машинных и робототехнических системах: учеб. пособие / В. С. Лозовский. – М.: ВИНТИ, 1984. – С. 84–121.
 32. **[Любарский, 1980]** Любарский, Ю. Я. Интеллектуальные информационные системы / Ю. Я. Любарский. – М.: Наука, 1980.
 33. **[Люгер, 2003]** Люгер, Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж. Ф. Люгер. – М.: Вильямс, 2003.
 34. **[Мальцев, 1970]** Мальцев, А. И. Алгебраические системы / А. И. Мальцев. – М.: Наука, 1970.
 35. **[Марковский, 1997]** Марковский, А. В. Анализ структуры знаковых ориентированных графов / А. В. Марковский. // Изв. РАН : Теория и системы управления. – 1997. – №5.
 36. **[Мартынов, 1977]** Мартынов, В. В. Универсальный семантический код / В. В. Мартынов. – Минск : Наука и техника, 1977.
 37. **[Мельчук, 1974]** Мельчук, И. А. Опыт теории лингвистических моделей «Смысл-Текст». Семантика, синтаксис / И. А. Мельчук. – М.: Наука, 1974.
 38. **[Месарович и др., 1978]** Месарович, М. Общая теория систем: математические основы / М. Месарович, Я. Тахакара. – М.: Мир, 1978.

39. **[Молокова, 1992]** Молокова, О. С. Методология анализа предметных знаний / О. С. Молокова. // Новости искусственного интеллекта. – 1992. – №3. – С.11–60.
40. **[Нариньяни, 1994]** Нариньяни, А. С. НЕ-факторы и инженерия знаний: от наивной формализации к естественной программатике / А. С. Нариньяни // КИИ-94: сб. тр. Нац. конф. с международным участием по ИИ. «Искусственный интеллект-94»: в 2 т.– Тверь : АИИ, 1994. – Т. 1. – С. 9–18.
41. **[Осипов, 1990]** Осипов, Г. С. Построение моделей предметных областей. Неординарные семантические сети / Г. С. Осипов. // Изв. АН СССР. Техническая кибернетика. – 1990. – №5. – С. 32–45.
42. **[Петров, 1978]** Петров, С. В. Графовые грамматики и автоматы (обзор) / С. В. Петров. // Автоматика и телемеханика. – 1978. – №7. – С. 116–136.
43. **[Петрушкин, 1992]** Петрушкин, В. А. Экспертно-обучающие системы / В. А. Петрушкин. – Киев : Наукова думка. – 1992.
44. **[Плесневич, 1982]** Плесневич, Г. С. Представление знаний в ассоциативных сетях / Г. С. Плесневич // Изв. АН СССР. Техническая кибернетика. – 1982. – №5. – С. 6–22.
45. **[Плесневич, 2008]** Плесневич, Г. С. Бинарные модели знаний / Г. С. Плесневич // Интеллектуальные системы (AIS'08) и Интеллектуальные САПР (CAD-2008): тр. междунар. науч.-техн. конф. : в 4 т. – М : Физматлит, 2008. – Т.2. – С. 135–146.
46. **[Попков, 1986]** Попков, В. К. Гиперсети и их характеристики связности / В. К. Попков. // Исследования по прикладной теории графов: учеб. пособие / В. К. Попков. – Новосибирск : Наука, 1986. – С. 25–58.
47. **[Поспелов, 1986a]** Поспелов, Д. А. Представление знаний. Опыт системного анализа / Д. А. Поспелов. // Системные исследования. Методологические проблемы. Ежегодник. – М. : Наука, 1986. – С. 83–102.
48. **[Поспелов, 1986b]** Поспелов, Д. А. Ситуационное управление. Теория и практика / Д. А. Поспелов. – М : Наука, 1986.
49. **[Рабинович, 1995]** Рабинович, З. Л. О концепции машинного интеллекта и ее развитии / З. Л. Рабинович // Кибернетика и системный анализ. – 1995. – №2. – С.163–173.
50. **[Рассел, 2006]** Рассел, С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – М. : Вильямс, 2006.
51. **[Резанов, 1989]** Резанов, С. Н. Об одном методе обобщения на семантических сетях в системе управления энергообъединением / С. Н. Резанов // Изв. АН СССР. Техническая кибернетика – 1989. – №5. – С. 55–62.
52. **[Рубашкин, 1989]** Рубашкин, В. Ш. Представление и анализ смысла в интеллектуальных информационных системах / В. Ш. Рубашкин. – М. : Наука, 1989.
53. **[Рыбина, 2010]** Рыбина, Г. В. Основы построения интеллектуальных систем: учеб. пособие / Г. В. Рыбина. – М. : Финансы и статистика, 2010.
54. **[Сапатый, 1983]** Сапатый, П. С. Об эффективности структурной реализации операций над семантическими сетями / П. С. Сапатый // Изв. АН СССР. Техническая кибернетика. – 1983. – №5. – С. 128–134.
55. **[Семенов, 1980]** Семенов, В. В. Семантические фреймовые сети как модели предметной области для САПР САУ / В. В. Семенов // Представление знаний в системах искусственного интеллекта: учеб. пособие / В. В. Сапатый. – М. : МДНТП, 1980. – С. 117–122.
56. **[Скороходько, 1989]** Скороходько, Э. Ф. Семантические сети и автоматическая обработка текста. / Э. Ф. Скороходько. – Киев : Наукова думка, 1983.
57. **[Скрэгг, 1983]** Скрэгг, Г. Семантические сети как модели памяти / Г. Скрэгг // Новое в зарубежной лингвистике. – М. : Радуга, 1983. – Вып. 12: Прикладная лингвистика. – С. 228–271.
58. **[Соловьев, 1990]** Соловьев, В. А. Формирование на семантической сети понятий и суждений с помощью рассуждений по аналогии / В. А. Соловьев // Искусственный интеллект-90: II Всесоюз. конф. Секционные и стендовые доклады. – Минск, 1990. – Т. 1. – С. 166–169.
59. **[Тузов, 1984]** Тузов, В. А. Математическая модель языка / В. А. Тузов. – Л. : Изд-во Ленингр. ун-та, 1984.

60. **[Тузов, 1986]** Тузов, В. А. О формализации понятия задачи / В. А. Тузов. – М. : Наука, 1986. – С. 73–83.
61. **[Тыугу, 1989]** Тыугу, Э. Х. Интеграция знаний / Э. Х. Тыугу // Изв. АН СССР. Техническая кибернетика. – 1989. – №5. – С. 3–13.
62. **[Финн, 2008]** Финн, В. К. Многозначные логики и их применения: в 2 т. / под ред. В. К. Финн. – М. : ЛКИ, 2008. – Т.1, Т.2.
63. **[Уварова, 1987]** Уварова, Т. Г. Формальное описание операционного языка для семантических сетей. / Т. Г. Уварова, Л. Л. Лифшиц – М. : ВЦ АН СССР, 1987.
64. **[Хельбиг, 1980]** Хельбиг, Г. Семантическое представление знаний в вопросно-ответной системе FAS-80 / Г. Хельбиг. // Представление знаний и моделирование процессов понимания: учеб. пособие / Г. Хельбиг. – Новосибирск, 1980. – С. 97–123.
65. **[Хендрикс, 1975]** Хендрикс, Г. О расширении применимости семантических сетей введением разбиений / Г. О. Хендрикс // Тр. IV Междунар. объединенной конф. по искусственному интеллекту: в 10 т. – М., 1975. – Т. 1. – С. 190–206.
66. **[Хорошевский, 2008]** Хорошевский, В. Ф. Пространства знаний в сети Интернет и Semantic Web (Ч. 1) / В. Ф. Хорошевский. // Искусственный интеллект и принятие решений. – 2008. – №1. – С. 80–97.
67. **[Цаленко, 1989]** Цаленко, М. Ш. Моделирование семантики в базах данных / М. Ш. Цаленко. – М. : Наука, 1989.
68. **[Шенк, 1980]** Шенк, Р. Обработка концептуальной информации / Р. Шенк. – М. : Энергия, 1980.
69. **[Шрейдер, 1971]** Шрейдер, Ю. А. Системы и модели / Ю. А. Шрейдер, А. А. Шаров. – М. : Радио и связь, 1982.
70. **[Шуберт, 1979]** Шуберт, Л. Усиление выразительной мощности семантических сетей / Л. Шуберт // Кибернетический сборник. Новая серия. – 1979. – Вып. 16. – С. 171–212.
71. **[OSTIS, 2011]** Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2011. – Режим доступа: <http://ostis.net>.
72. **[Russell et al., 1995]** Russell, S. Artificial Intelligence. A Modern Approach / S. Russell [et al.]. – New Jersey : Prentice Hall, 1995.
73. **[Sowa, 2008]** Sowa, J. Conceptual Graphs / J. Sowa F. [et al.] // Handbook of Knowledge Representation, Elsevier, 2008. – P. 213–237.
74. **[Wooldridge et al., 1994]** Wooldridge, M. Agent Theories, Architectures and Languages: A Survey / M. Wooldridge [et al.] // Intelligent Agents. Languages. Amsterdam : Springer Verlag, August. – 1994. – P. 3–39.

Оглавление

Введение	3
1. Опыт наиболее продвинутых технологий	5
2. Графодинамические модели	5
3. Параллельные асинхронные графодинамические модели.....	8
4. Семантические модели представления и обработки знаний	10
5. Унификация семантического представления знаний.....	15
6. Унификация структуризации баз знаний	26
7. Графовые языки программирования	41
8. Унификация формального описания агентов, работающих над семантической памятью	42
9. Унификация семантических моделей обработки знаний.....	43
10. Унификация семантических моделей информационного поиска	45
11. Унификация семантических моделей интеграции знаний и семантических моделей интеграции целых интеллектуальных систем.....	47
12. Унификация и интеграция различных семантических моделей решения задач.....	49
13. Унификация визуализации семантических сетей.....	50
14. Унификация семантических моделей различных пользовательских интерфейсов.....	53
15. Библиотека типовых семантически совместимых компонентов интеллектуальных систем и методика модульного проектирования интеллектуальных систем.....	54
16. Платформенно-независимый характер проектирования интеллектуальных систем	55
17. Семантический ассоциативный параллельный компьютер	56
18. Встроенные подсистемы интеллектуальных систем, обеспечивающие их эффективную эксплуатацию и эволюцию	58
19. Доступность и открытость технологии.....	59
20. Эволюционная методика проектирования	59
21. Реализация предлагаемой технологии в виде интеллектуальной метасистемы.....	61
Заключение	63
Библиографический список	64

Учебное издание

Голенков Владимир Васильевич
Гулякина Наталья Анатольевна
Степанова Маргарита Дмитриевна
Самодумкин Сергей Александрович

**ФОРМАЛЬНЫЕ ОСНОВЫ СЕМАНТИЧЕСКОГО ПРЕДСТАВЛЕНИЯ
ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ**

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Редактор М. А. Зайцева
Корректор Е. Н. Батурчик
Компьютерная правка, оригинал-макет В. М. Задоя

Подписано в печать 15.05.2014. Формат 60x84 1/8. Бумага офсетная. Гарнитура «Ариал».
Отпечатано на ризографе. Усл. печ. л. 8,37. Уч.-изд. л. 10,7. Тираж 100 экз. Заказ 793.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014, №2/113 от 07.04.2014.
ЛП №02330/264 от 14.04.2014.
2200013, Минск, П. Бровки, 6