Semantic Space Integration of Logical Knowledge Representation and Knowledge Processing Models

Valerian Ivashenko
Department of ntelligent Information Technologies
Belarusian State University of Informatics and Radiolectronics
Minsk, Republic of Belarus
Email: ivashenko@bsuir.by

***Abstract*—This article deals with an approach to represent both logical structures and schemes, as well as logical knowledge processing models within semantic space in the form of semantic networks. Some properties of the semantic space and logical models such as topological, metric, and valuation (norm) properties are investigated. Concepts for ontological representation of classical and non-classical logical formulas, classes, and relations are proposed.**

***Keywords*—semantic space, integration, problem-solving model integration, logical knowledge representation model, logical knowledge processing model, semantic topological space, semantic metric space, distensible set, unified semantic representation model, language key element, right proposition, uncorrupted proposition, finite set deduction**

I. Introduction

The main quality of such intelligent systems as intelligent computer systems of a new generation is the ability to solve problems. Let us consider the indicators of quality which are necessary to ensure, maintain, and develop this quality. Intelligent computer systems of a new generation are classified as cybernetic systems (adaptive systems [1]). For such systems, some of the important quality indicators are [2]:

- self-learning, implemented (or automated) through self-improvement, provided by the universality of the intelligent system operating on a variety of knowledge types and (hybrid) problem-solving models, as well as their deep integration (degree of interosculation);
- interoperability that requires mutual understanding based on semantic compatibility;
- ability to coordinate their plans and intentions and coordinate them in a decentralized manner that requires the integration of plans or their parts;
- semantic compatibility (for types of knowledge and problem-solving models) that is matching between systems and concepts and requires formalization of semantic representation of information through unification, where the latter is the main indicator of degree of convergence between intelligent computer systems and their components.

Thus, the requirement of deep integration (knowledge types and problem-solving models) is represented in the intelligent computer systems of a new generation. At the same time, intelligent computer systems of a new generation are characterized by the degree of convergence, unification, and standardization of intelligent computer systems and their components and the corresponding degree of integration (depth of integration) of intelligent computer systems and their components. Currently, there is a strong need for focusing on potentially universal (that is, capable of quickly acquiring any knowledge and skills), synergistic intelligent computer systems with "strong" intelligence, since the maximum level of cybernetic system processor quality in terms of the variety of problem-solving models interpreted by the processor of a cybernetic system is its universality, that is, its "principal ability to interpret any model for solving both intelligent and non-intelligent problems". In order to ensure quality, the importance of moving to hybrid individual intelligent computer systems is highlited, where convergence and integration of various problem-solving models and various knowledge types is carried out. Due to the fact that different intelligent computer systems may require different combinations of problem-solving models (models for representing and processing knowledge of various types), which have been developed by a large number at the present time, and in the development and implementation of various intelligent computer systems, the appropriate methods and tools must guarantee logical semantic compatibility of the developed components and, in particular, their ability to use common information resources, then for this, the need to unify these models is indicated. The creation of intelligent computer systems of a new generation involves the creation of an appropriate technology for integrated design and integrated support for the stages of the life cycle of these systems. It is noted that in order to create a technology for integrated design and comprehensive support for the subsequent stages of the life cycle of intelligent computer systems of a new generation, in particular, it is necessary:

to unify the formalization of various models for representing various types of used information stored in the memory of intelligent computer systems and various models for solving intelligent problems to ensure semantic compatibility and simple automated integrability of various knowledge types and problem-solving models in intelligent computer systems. To do this, it is proposed to develop a basic universal abstract model for the representation and processing of knowledge, which ensures the implementation of various problem-solving models.

Let us consider the stages of solving these problems in the aspect of integrating logical models of representation and knowledge processing. The need for this consideration is caused by the need of applying logical problem-solving models in intelligent computer systems of a new generation (including knowledge-driven systems), while ensuring the quality of knowledge in accordance with the problems of knowledge management.

II. Knowledge integration and semantic space models

In order to solve the problem of unifying the formalization of various models for representing various types of information used, a model of a unified semantic representation of knowledge [3] has been developed, as well as models for representing data in the form of texts of generalized formal languages [4] and processing generalized strings (and lists) for knowledge-driven systems [5]. Based on and in accordance with the model of unified knowledge representation, a family of sc-languages [1], [3], [6] has been developed, to clarify the semantics of which a model of event (distensible) sets [4] has been developed and an ontological model of spatio-temporal relations of events and phenomena for knowledge processing operations has been proposed.

In order to ensure the integration of knowledge and quality assurance in the process of knowledge integration, models for the specification and integration of knowledge are proposed. Solving the problem of knowledge integration allows considering and studying by formal means the semantic neighborhoods of sc-language text elements, the key elements of sc-languages, and studying the similarity of structures that are formed as a result of integration. Based on the proposed models, a meta-model of the semantic space was developed [4], within which it is possible to study the semantic space [7]–[14] and consider semantic subspaces of various types.

The system of transitions from texts of sc-languages to topological space is studied. Below is a fragment of the ontology that describes the types of topological spaces and inclusion relations of topological spaces of various types [**?**], [15].

generalized sc-tuple
:= [non-empty sc-set]

generalized sc-relation
:= [sc-set of non-empty sc-sets]
⇒ *explanation**:
[A generalized sc-relation is a sc-set of generalized sc-tuples.]

binary sc-relation
⇒ *explanation**:
[A binary sc-relation is an sc-set of sc-pairs (or generalized sc-tuples to which there are two different memberships of sc-elements or the same sc-element).]

nodal sc-pair
⇒ *explanation**:
[A nodal sc-pair is an sc-pair that cannot be denoted by a membership sc-arc (positive, negative, or fuzzy).]

slot sc-relation
⇒ *explanation**:
[A slot sc-relation is a binary sc-relation (an sc-set of (oriented) sc-pairs) whose elements are not nodal sc-pairs.]

membership phenomenon
⇒ *explanation**:
[A membership phenomenon is a set of phenomena each of which is a slot sc-relation, while any sc-arc of permanent non-membership does not belong permanently to each of them.]

becoming*
⇒ *explanation**:
[becoming* is a binary sc-relation between events (states) or phenomena.]

immediately before'
⇒ *first domain**:
*becoming**
⇒ *second domain**:
established event or phenomenon

immediately after'
⇒ *first domain**:
*becoming**
⇒ *second domain**:
constitutive event or phenomenon

continuance*
⇒ *explanation**:
[The continuance* is the transitive closure of the

sc-relation of becoming.]

earlier'
⇒ *first domain**:
*continuance**
⇒ *second domain**:
early event or phenomenon

later'
⇒ *first domain**:
*continuance**
⇒ *second domain**:
later event or phenomenon

sc-structure*
⇒ *explanation**:
[A sc-structure* is an sc-set that contains a non-empty support sc-subset (the set of primary elements of the sc-structure*).]

sc-structure'
⇒ *first domain**:
*sc-structure**
⇒ *second domain**:
non-empty sc-set

support of sc-structure'
⇒ *first domain**:
*sc-structure**
⇒ *second domain**:
non-empty sc-set

elementarily represented sc-set'
:= [elementarily represented element']
⇒ *explanation**:
[An elementarily represented element' is an element of an sc-structure* and an sc-set all of whose elements are elements of an sc-structure*.]

full-connectivily represented sc-set sc-set'
:= [full-connectivily represented element']
⇒ *explanation**:
[A full-connectivily represented sc-set element' is an element of an sc-structure* an sc-set all of whose elements and all its memberships are elements of an sc-structure* or an sc-arc that is an elementarily-represented element' of this sc-structure*.]

fully represented sc-set'
:= [fully represented element']
⇒ *explanation**:
[A fully represented element' is a full-connectively represented element' of an sc-structure* with any its element that is not an sc-arc outgoing from it connected by a membership sc-arc or a non-membership sc-arc belonging to this sc-structure*.]

sc-tuple'
⇒ *explanation**:
[A sc-tuple' is a sc-tuple' is a full-connectively represented element' of sc-structure* that is an sc-tuple and belongs to the sc-relation' of this sc-structure*.]

sc-relation'
⇒ *explanation**:
[A sc-relation' is a full-connectively represented element' of sc-structure* being a sc-relation whose elements are all sc-tuples' of this sc-structure*.]

sc-class'
⇒ *explanation**:
[A sc-class' is a full-connectively represented element' of sc-structure* all of whose elements are members of an sc-structure* that is neither an sc-relation' nor an sc-tuple' of that sc-struct*.]

entitive closure*
⇒ *explanation**:
[An entitive closure* is the smallest superset* (structure*) in which each element is elementarily represented'.]

entitive closure'
⇒ *first domain**:
*entitive closure**
⇒ *second domain**:
entitive closure

support of entitive closure '
⇒ *first domain**:
*entitive closure**
⇒ *second domain**:
non-empty sc-set

substantial closure*
⇒ *explanation**:
[A substantial closure* is the smallest superset* (structure*) in which each element is a full-connectively represented element']

substantial closure'
⇒ *first domain**:
*substantial closure**
⇒ *second domain**:
substantial closure

support of substantial closure '

⇒ *first domain**:
*substantial closure**

⇒ *second domain**:
non-empty sc-set

sc-relation of similarity by slot relations*

⇒ *explanation**:
[A similarity sc-relation by slot sc-relations* is a sc-relation that is reflexive by these slot relations, i.e. for any element included in the tuple of this sc-relation under one of the slot sc-relations, there is a tuple of this sc-relation in which it enters under each of these slot sc-relations.]

sc-relation of similarity by slot relations '

⇒ *first domain**:
*sc-relation of similarity by slot relations**

⇒ *second domain**:
sc-relation of similarity by slot relations

slot relations of similarity sc-relation '

⇒ *first domain**:
*sc-relation of similarity by slot relations**

⇒ *second domain**:
slot relations of similarity sc-relation

sc-relation of semantic similarity by slot relations*

⇒ *explanation**:
[A semantic similarity sc-relation by slot relations* is a similarity sc-relation by slot relations* si and sj, in which each element under the slot sc-relation si can be converted to an element of the syntactic type of the element under the slot sc-relation sj; two incident sc-elements under the slot sc-relation si, within this sc-relation of semantic similarity correspond to the incident elements, respectively, under the slot sc-relation sj.]

sc-relation of semantic similarity by slot relations '

⇒ *first domain**:
*sc-relation of semantic similarity by slot relations**

⇒ *second domain**:
sc-relation of semantic similarity by slot relations

slot relations of semantic similarity sc-relation '

⇒ *first domain**:
*sc-relation of semantic similarity by slot relations**

⇒ *second domain**:
slot relations of semantic similarity sc-relation

connected sc-structure*

⇒ *explanation**:
[A connected sc-structure* is a sc-structure* that is connected.]

connected sc-structure '

⇒ *first domain**:
connected sc-structure

⇒ *second domain**:
connected non-empty sc-set

support of connected sc-structure '

⇒ *first domain**:
*connected sc-structure**

⇒ *second domain**:
non-empty sc-set

semantic similarity of sc-structures*

⇒ *explanation**:
[A semantic similarity of sc-structures* connects the sc-set of sc-structures* with the sc-structure* sc-relation of semantic similarity by slot sc-relations si, sj so that for each sc-structure* from the sc-set there is its an element and a tuple of this sc-relation of similarity, in which it is included under the slot sc-relation si, and under the slot sc-relation sj there is an element of the sc-structure *, also for each element of the sc-structure there is a tuple of this sc-relation of similarity, in which it enters under the slot sc-relation sj, and under the slot sc-relation si enters an element of the sc-structure* from the sc-set.]

sc-relation of semantic similarity of sc-structures '

⇒ *first domain**:
*semantic similarity of sc-structures**

⇒ *second domain**:
*sc-relation of semantic similarity by slot relations**

semantic similarity of sc-structures '

⇒ *first domain**:
*semantic similarity of sc-structures**

⇒ *second domain**:
*sc-structure of semantic similarity of sc-structures**

sc-structure of semantic similarity of sc-structures '

⇒ *first domain**:
*sc-structure of semantic similarity of sc-structures**

⇒ *second domain**:
sc-structure of semantic similarity of sc-structures

set of semantically similar sc-structures′
⇒ *first domain**:
*sc-structure of semantic similarity of sc-structures**
⇒ *second domain**:
set of semantically similar sc-structures

semantic continuous similarity of sc-structures*
⇒ *explanation**:
[A semantic continuous similarity of sc-structures* connects an sc-set of sc-structures* with a connected sc-structure* sc-relation of semantic similarity by slot sc-relations si, sj so that for each sc-structure* from sc-set find its element and the tuple of this similarity sc-relation, in which it enters under the slot sc-relation si, and under the slot sc-relation sj there is an element of the connected sc-structure*, also for each element of the connected sc-structure there is a tuple of this sc- a similarity relation in which it enters under the slot sc-relation sj, and under the slot sc-relation si the element of the sc-structure* from the sc-set enters.]

sc-relation of semantic continuous similarity of sc-structures′
⇒ *first domain**:
*semantic continuous similarity of sc-structures**
⇒ *second domain**:
*sc-relation of semantic continuous similarity by slot relations**

semantic continuous similarity of sc-structures*′
⇒ *first domain**:
*semantic continuous similarity of sc-structures**
⇒ *second domain**:
*sc-structure of semantic continuous similarity of sc-structures**

sc-structure of semantic continuous similarity of sc-structures′
⇒ *first domain**:
*sc-structure of semantic continuous similarity of sc-structures**
⇒ *second domain**:
sc-structure of semantic continuous similarity of sc-structures

set of semantically continuously similar sc-structures′
⇒ *first domain**:
*sc-structure of semantic continuous similarity of sc-structures**
⇒ *second domain**:
set of semantically continuously similar sc-structures

key query′
⇒ *first domain**:
*key query**
⇒ *second domain**:
key query
⇒ *explanation**:
[A key query′is a search-verify query (from one known element) that is executed from at least one element and is not executed from at least one element.]

element of key query′
⇒ *first domain**:
*key query**
⇒ *second domain**:
element of key query

minimal key query′
⊂ *key query′*
⇒ *explanation**:
[A minimal key query is a key query that finds sc-subsets of the element sets found by all other key queries that have the same domains of known satisfiability and non-satisfiability elements.]

element of minimal key query′
⇒ *first domain**:
*minimal key query**
⇒ *second domain**:
element of minimal key query

full semantic neighborhood of an element*
⇒ *explanation**:
[A full semantic neighborhood of an element* is all elements found by minimal key queries from this element (taking into account disjunctive search and negation of search).]

full semantic neighborhood of an element′
⇒ *first domain**:
*full semantic neighborhood of an element**
⇒ *second domain**:
full semantic neighborhood of an element

element of full semantic neighborhood′
⇒ *first domain**:
*full semantic neighborhood of an element**
⇒ *second domain**:
element of full semantic neighborhood

introspective key element′
⇒ *explanation**:
[An introspective (basic) key element is an element of a set (from the class of the smallest such sets) of elements such that any full semantic

neighborhood of any element is an sc-subset of the union of their full semantic neighborhoods.]

topological space
⇒ *explanation**:
[A topological space is a set with a set (family) of (open) subsets defined over it, including the set itself and the empty set. For any subset of the family, the result of the union belongs to the family, and for any finite subset of the family, the result of the intersection also belongs to the family. The complements of the sets of a family to the largest of the sets are called closed sets.]

topological space of connector incidence closure
⇒ *explanation**:
[A topological space of connector incidence closure on a set of sc-elements is a topological space, all closed sets of which contain all sc-elements of this set, to which there is a route along oriented connectives of the incidence relation of connectors.]
⇒ *note**:
[In the general case, it does not satisfy the Tikhonov separability axiom. The pragmatics of considering such spaces is determined by the operations of removing sc-elements and connectors with which they are incident. Deleting an sc element requires deleting all connectors whose closure of any open neighborhood it belongs to.]

topological subspace of connector incidence closure'
⇒ *first domain**:
*inclusion of topological spaces of connector incidence closure**
⇒ *second domain**:
topological space of connector incidence closure

topological superspace of connector incidence closure'
⇒ *first domain**:
*inclusion of topological spaces of connector incidence closure**
⇒ *second domain**:
topological space of connector incidence closure

topological space of syntactic closure
⇒ *explanation**:
[A topological space of syntactic closure on a set of sc-elements is a topological space, all closed sets of which contain all sc-elements of this set, to which there is a route along oriented connectives of the incidence relation.]
⇒ *note**:
[In the general case, it does not satisfy the Kolmogorov separability axiom. Syntactic closure can be singled out as the basis of closed sets of a topological space, however, due to the possibility of drawing arcs from any sc-node to any, in the final case (as a result of the process of eliminating non-factors), such a space is a trivial (antidiscrete) space. The union relation of topological spaces of syntactic closure is not algebraically closed on the set of topological spaces of syntactic closure. For the same reason, for any incomplete topological space of syntactic closure, one can consider a topological space of syntactic closure whose support is a superset of the support of the former and which does not preserve closed sets. In this sense, the topology based on syntactic closure is not stable with respect to the processes of knowledge formation and its consideration is not pragmatically justified. The topology of the complete topological space of syntactic closure is antidiscrete (trivial). Thus, for a complete topological space of syntactic closure, all topological subspaces of syntactic closure have antidiscrete (trivial) topology.]

topological space of entitive closure
⇒ *explanation**:
[The topological space of an entitive closure on the set of sc-elements is a topological space all of whose closed sets are entitive closures.]
⇒ *note**:
[In the general case, it does not satisfy the Tikhonov separability axiom. An entitive closure can be considered as a support of a topological (sub)space of entitive closure. The topological space of entitive closure preserves the closed sets of any topological spaces of entitive closure whose support is a subset of its support and an entitive closure. Such spaces form a structure of topological subspaces – topological superspaces of entitive closure. The topology of spaces in this structure is diverse.]

topological subspace of entitive closure'
⇒ *first domain**:
*inclusion of topological spaces of entitive closure**
⇒ *second domain**:
topological space of entitive closure

topological superspace of entitive closure'
⇒ *first domain**:
*inclusion of topological spaces of entitive closure**
⇒ *second domain**:
topological space of entitive closure

topological space of substantial closure

⇒ *explanation**:

[The topological space of a substantial closure on the set of sc-elements is a topological space all of whose closed sets are substantial closures.]

⇒ *note**:

[In the general case, it does not fit the axiom of separability according to Tikhonov. An substantial closure can be considered as a support of a topological (sub)space of substantial closure. The topological space of a substantial closure preserves closed any topological spaces of a substantial closure whose support is a subset of its support and a substantial closure. Such spatial formations of topological subspaces are topological superspaces of substantial closure. Topology of spaces in this vast area.]

topological subspace of substantial closure'

⇒ *first domain**:

*inclusion of topological spaces of substantial closure**

⇒ *second domain**:

topological space of substantial closure

topological superspace of substantial closure'

⇒ *first domain**:

*inclusion of topological spaces of substantial closure**

⇒ *second domain**:

topological space of substantial closure

There is a possible transition from sc-structures to manifolds and topological spaces by reducing sc-structures to graph structures. The issues of reducing sc-structures to graph structures and further towards to manifolds and topological spaces are considered in detail in [4] (see Fig. 1).

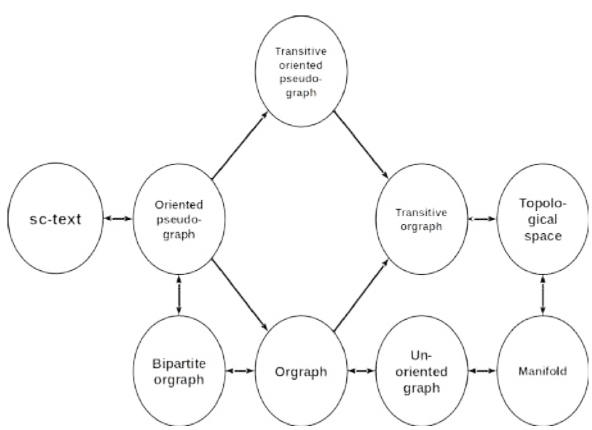

Figure 1. Transformations for sc-texts, graphs, manifolds and topological spaces.

Types of metric and pseudometric semantic subspaces and normalized semantic subspaces can also be represented. Further, we will consider such metric spaces as finite subspaces with full-connectively represented sc-elements, in particular.

metric

⇒ *explanation**:

[A metric is a function of two arguments that takes values on a (linearly) ordered support of the group, is non-negative, is equal to the neutral element (zero) only when the arguments are equal, is symmetric, satisfies the triangle inequality.]

metric space

⇒ *explanation**:

[A metric space is a set with a function of two arguments defined on it, which is a metric that takes values on the ordered support of the group.]

metric finite syntactic space

⇒ *explanation**:

[A metric finite syntactic space of the SC-code is a metric space with a finite support, whose elements are designations (sc-elements), and the value of the metric can be determined through the incidence relations of the elements without taking into account their semantic type.]

metric finite syntactic subspace'

⇒ *first domain**:

*inclusion of metric finite syntactic spaces**

⇒ *second domain**:

metric finite syntactic space

metric finite syntactic superspace'

⇒ *first domain**:

*inclusion of metric finite syntactic spaces**

⇒ *second domain**:

metric finite syntactic space

metric finite semantic space

⇒ *explanation**:

[A metric finite semantic space of the SC-code is a metric space with a finite support, whose elements are designations (sc-elements), and the value of the metric cannot be determined through the incidence relations of elements without taking into account their semantic type.]

metric finite semantic subspace'

⇒ *first domain**:

*inclusion of metric finite semantic spaces***

⇒ *second domain**:
metric finite semantic space

metric finite semantic superspace′
⇒ *first domain**:
*inclusion of metric finite semantic spaces**
⇒ *second domain**:
metric finite semantic space

A metric finite syntactic space can be constructed by [4] according to the string processing model and metric definitions given in [5].

pseudometric
⇒ *explanation**:
[A pseudometric is a function of two arguments that takes values on a (linearly) ordered group support, is non-negative, symmetric, and satisfies the triangle inequality.]

psudometric space
⇒ *explanation**:
[A pseudometric space is a set with a function of two arguments defined on it, which is a pseudometric [16] taking values on the ordered support of the group.]

pseudometric finite semantic space
⇒ *explanation**:
[A pseudometric finite semantic space of the SC-code is a pseudometric space with a finite support whose elements are designations (sc-elements), and the value of the pseudometric cannot be determined through the incidence relations of elements without taking into account their semantic type.]

Some models of more complex structures that take into account non-factors [17] associated with space-time have been successfully proposed in [4]. The proposed models rely on a representation capable of expressing the semantics of variable notation and operational semantics by extended means of the alphabet. To build such models, in addition to the extended alphabet tools, it is proposed to rely on models that describe the processes of integration and formation of knowledge [18], on knowledge specification tools [3], [4], focused on consideration of finite structures, which allow proceeding with consideration of complex metric relationships within the semantic space meta-model (see Fig. 2).

The possibility of considering the metric in the semantic space allows speaking about the semantic metric, which, along with activity, scaling, interpretability, and the presence of a complex structure and coherence, is a hallmark of knowledge.

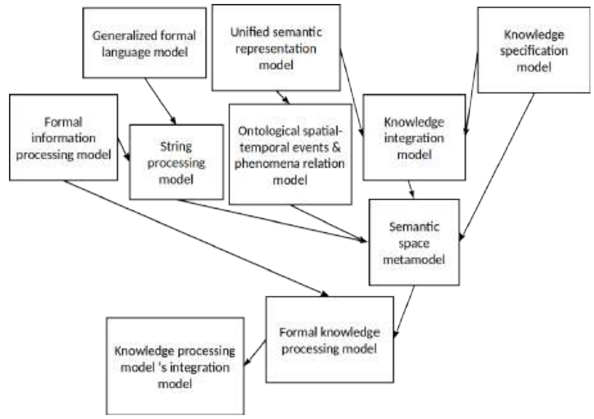

Figure 2. Models providing integration.

semantic metric
:= [semantic similarity]
⇒ *explanation**:
[Semantic metric is a metric defined on signs and quantitatively expressing the proximity of their meanings.]

In addition to factual knowledge (facts), rules are used in knowledge bases. Within logical models of knowledge processing, rules are represented as logical formulas. Thus, the transition to the integration of such types of knowledge as (logical) rules allows talking about the integration of knowledge processing models (problem-solving models).

III. Integration of logical processing models as problem solving models

In order to solve the problem of integrating problem-solving models, the concept of a formal model for knowledge processing is proposed, which is a development for the concept of a formal model of information processing. The approach is used in the works of V. Kuzmitsky [19] and A. Kalinichenko [20]. A meta-model for the integration of formal models of knowledge processing is proposed.

The integration of knowledge processing models boils down to the following steps:

- For each state of the integrating model, its one-to-one (i) representation is constructed in the model of the unified semantic representation of knowledge.
- Next, a mapping π of this representation to a set of sc-texts immersed in a metric semantic subspace is constructed, and a one-to-one mapping i of operations $i(\rho)$ of this model to operations ρ on sc-texts from this set is constructed, so that:

$$i \circ i^{-1} \subseteq I = \left\{ \langle x, x\rangle \,|\exists y \,\langle x, y\rangle \in i^{-1} \cup i \right\}$$

$$i^{-1} \circ i \subseteq I$$

$$\forall\rho\left(i^{-1}\circ i\circ\rho\circ i\circ i^{-1}=\rho\right)$$

$$\forall\rho\exists i\left(\rho\right)\left(\pi^{-1}\circ i^{-1}\circ\rho\subseteq i\circ\pi\circ i\left(\rho\right)\right)$$

$$\forall\rho\exists i\left(\rho\right)\left(i\circ\pi\circ i\left(\rho\right)\subseteq\rho\circ i\circ\pi\right)$$

$$\forall\rho\exists i\left(\rho\right)\left(\rho=i\circ\pi\circ i\left(\rho\right)\circ\pi^{-1}\circ i^{-1}\right)$$

$$\forall\rho\exists i\left(\rho\right)\left(i\left(\rho\right)=\pi^{-1}\circ i^{-1}\circ\rho\circ i\circ\pi\right)$$

- Syntactic relations are distinguished on the elements of sc-texts.
- Interpretation functions are built on the states of the original model (in projective semantics) or on their representation in sc-texts (in reflexive semantics).
- The metric is set in accordance with the metric of the metric semantic subspace.
- In addition to the specified requirements, additional requirements τ and σ can be specified in accordance with a given scale on the set of states of the integrating information processing model: bijection (trivial order), out-degree, in-degree, etc.

$$\forall\rho\exists i\left(\rho\right)\left(\rho\circ\tau=i\circ\pi\circ i\left(\rho\right)\circ\pi^{-1}\circ i^{-1}\right)$$

$$\forall\rho\exists i\left(\rho\right)\left(i\left(\rho\right)\circ\sigma=\pi^{-1}\circ i^{-1}\circ\rho\circ i\circ\pi\right)$$

It should be noted that in the previous article [21], the mapping requirements were considered to be quite strong ($\tau=I$ and $\sigma=I$):

$$\forall\rho\exists i\left(\rho\right)\left(\rho\subseteq i\circ\pi\circ i\left(\rho\right)\circ\pi^{-1}\circ i^{-1}\right)$$

$$\forall\rho\exists i\left(\rho\right)\left(i\left(\rho\right)\subseteq\pi^{-1}\circ i^{-1}\circ\rho\circ i\circ\pi\right)$$

The current text contains proposal for weakening these requirements. Other additional requirements (including the quantitative properties of the information) may also be taken into account.

Let us consider some examples (Fig. 3–25).

From the point of view of topological properties, for each state of the model, there is its topological closure with respect to the set of operations. Moreover, these topological properties are preserved during integration. Thus, integration is a continuous mapping. However, for classical logical models of information processing, it is known that the closure with respect to deducibility is not topological closure (not additive):

$$[S]\cup[T]\neq[S\cup T]\,.$$

The seeming contradiction can be resolved if we notice that in the first case, the elements of the closure are

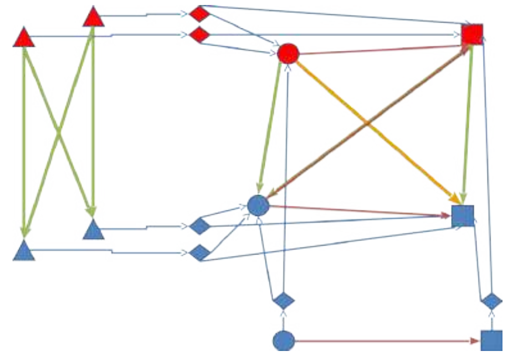

Figure 3. The reconvergent integration of non-deterministic knowledge processing operation as non-deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as non-deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

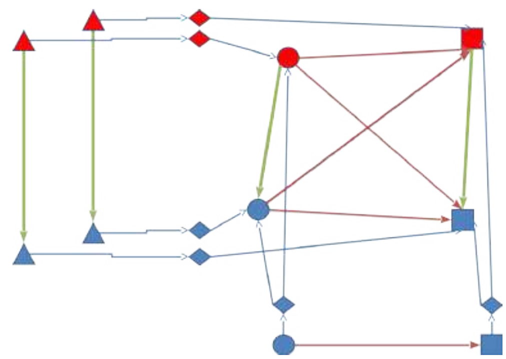

Figure 4. The convergent integration of deterministic knowledge processing operation as deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as non-deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

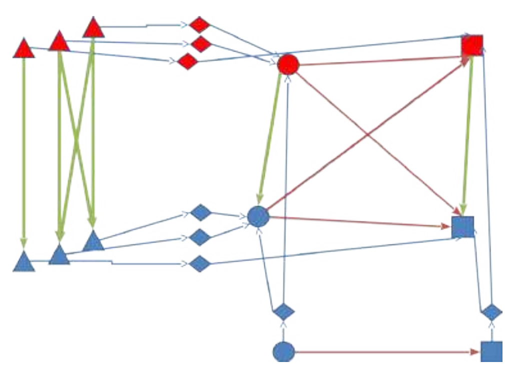

Figure 5. The reconvergent integration of non-deterministic knowledge processing operation as deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

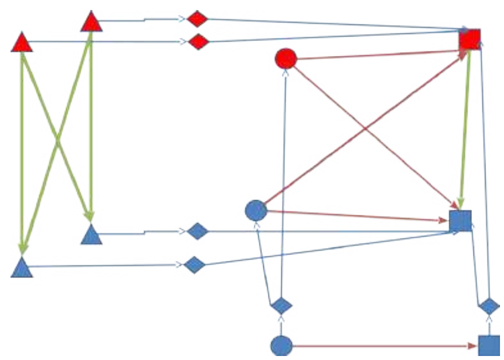

Figure 6. The convergent integration of non-deterministic knowledge processing operation as deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

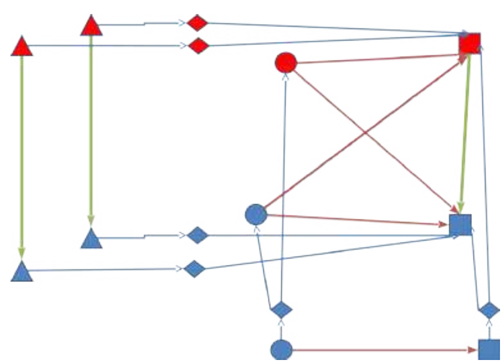

Figure 7. The convergent integration of ndeterministic knowledge processing operation as deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

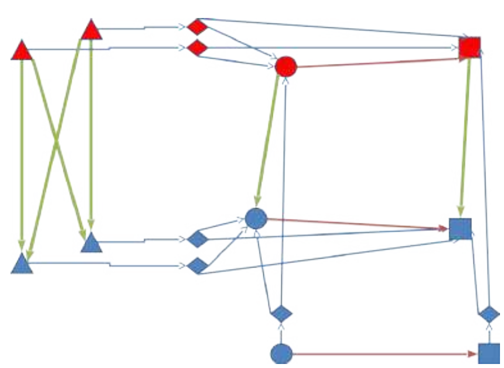

Figure 8. The reconvergent integration of non-deterministic knowledge processing operation as deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

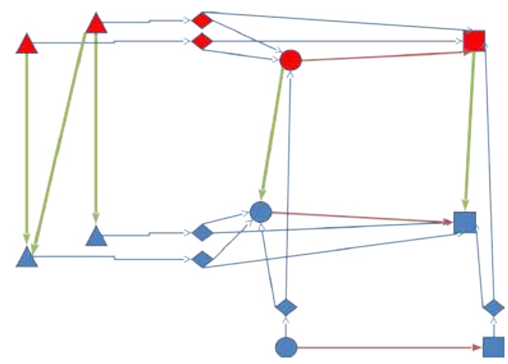

Figure 9. The asymmetrical reconvergent integration of non-deterministic knowledge processing operation as deterministic one (green lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one (red lines).

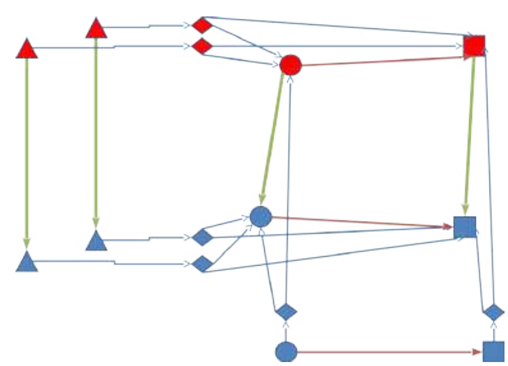

Figure 10. The reconvergent integration of deterministic knowledge processing operation as deterministic one (green lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one (red lines).

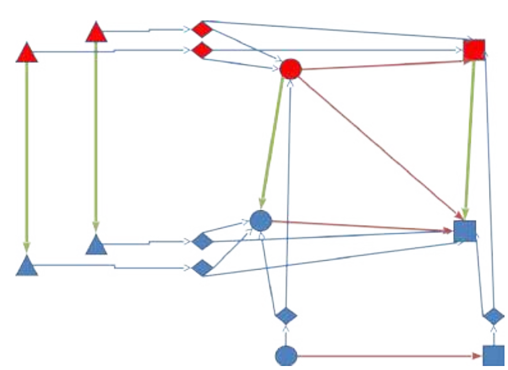

Figure 11. The reconvergent integration of deterministic knowledge processing operation as deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

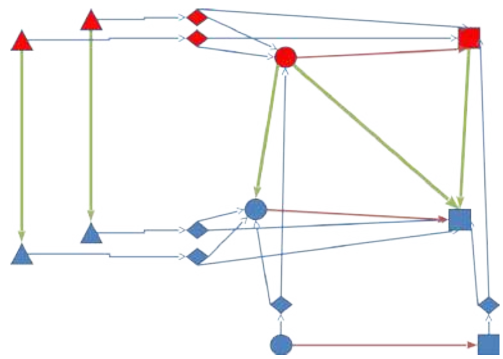

Figure 12. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one (red lines).

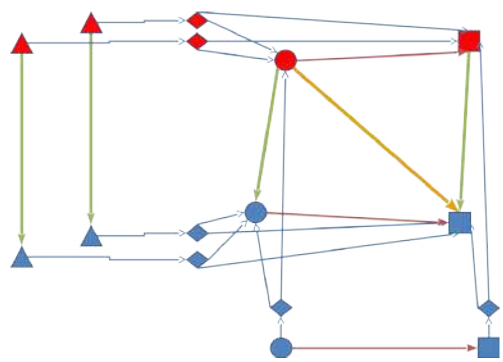

Figure 13. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (yellow (diagonal) and green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (yellow (diagonal) and red lines).

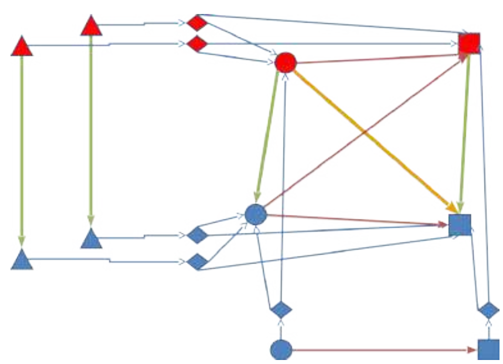

Figure 14. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as non-deterministic one (yellow and green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (yellow and red lines).

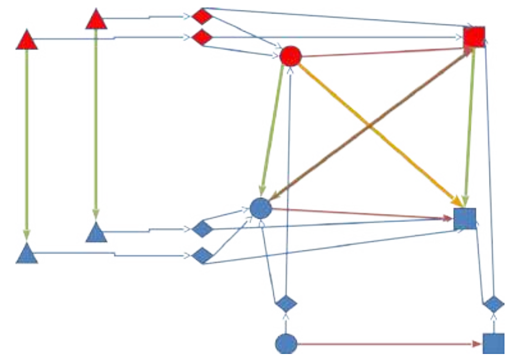

Figure 15. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as non-deterministic one (yellow and green lines) with the (symmetrical) divergent integration of deterministic knowledge processing operation as non-deterministic operation one (yellow and red lines).

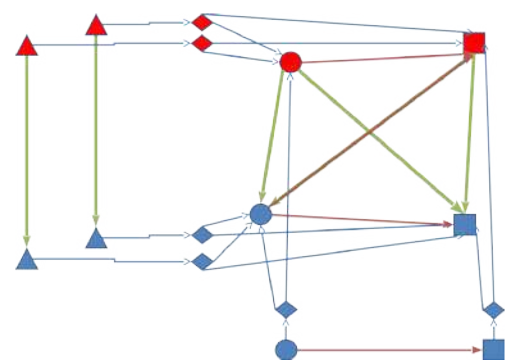

Figure 16. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

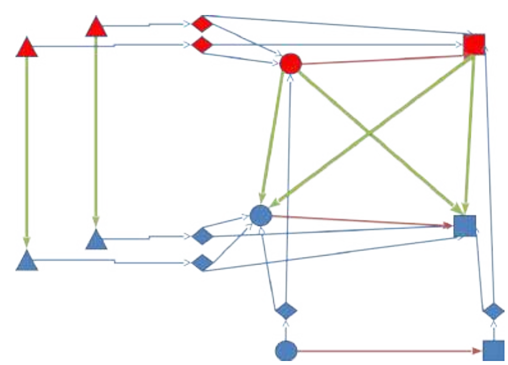

Figure 17. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the (symmetrical) divergent integration of deterministic knowledge processing operation as deterministic operation one (red lines).

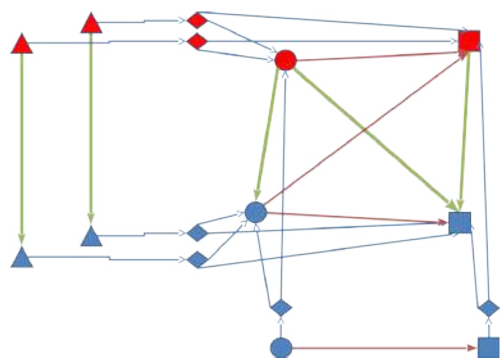

Figure 18. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

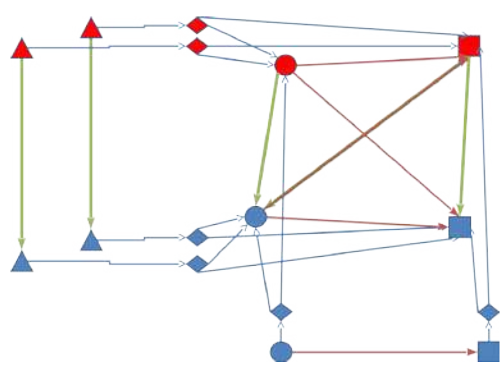

Figure 19. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the (symmetrical) divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

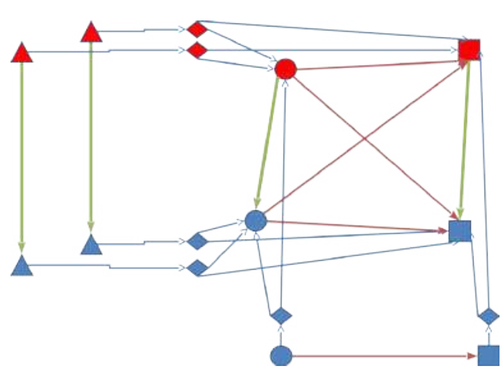

Figure 20. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as deterministic one (green lines) with the (symmetrical) divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

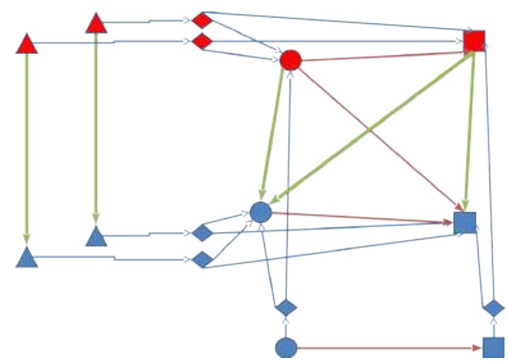

Figure 21. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

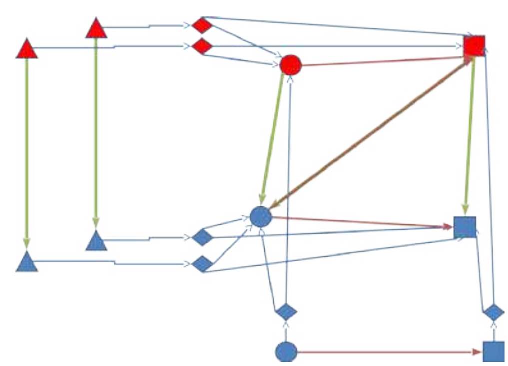

Figure 22. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

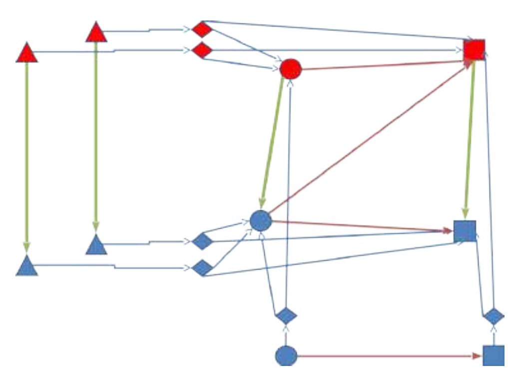

Figure 23. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

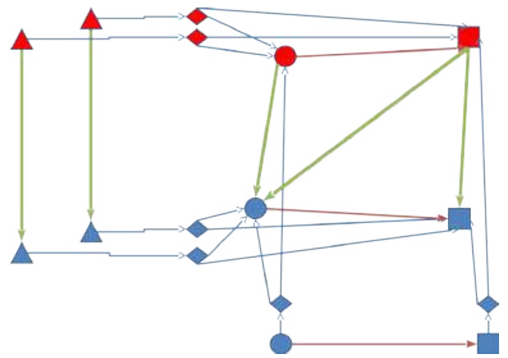

Figure 24. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the (symmetrical) divergent integration of deterministic knowledge processing operation as deterministic operation one (red lines).

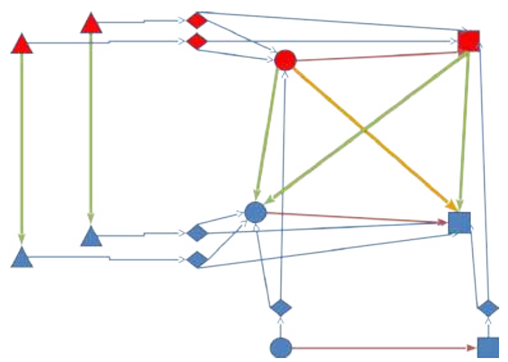

Figure 25. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

states (sets of formulas), and in the second case, they are formulas.

$$\begin{gathered}\{(A\to B),(B\to C)\}\vdash\{(A\to B),(B\to C),(A\to C)\}\\ \{(D\to B),(B\to E)\}\vdash\{(D\to B),(B\to E),(D\to E)\}\\ \{(A\to B),(B\to C)\}\cup\{(D\to B),(B\to E)\}\vdash\\ \{(A\to B),(B\to C),(A\to C),(D\to C),\\ (D\to B),(B\to E),(D\to E),(A\to E)\}\end{gathered}$$

For classical logics and logical models of knowledge processing, it is possible to naturally introduce a metric on sets of literal conjuncts (of a given length), if we take a finite subject domain and accept the assumption (hypothesis) of a closed world, then the metric is introduced as the sum of exclusive-OR from each pairs of matching literals.

$$\mu\left(\langle x,y\rangle\right)=\sum_{i=1}^{n}x_i\veebar y_i$$

Moreover, if the conjuncts define a set-ring algebra, then we can speak of a normed space and a norm (valuation) over a vector space, where the field is GF(2). Any (meaningful) proposition over this domain, except for the identically false one, can be represented as (PDNF). Then for PDNF we get a metric vector. The proposition is true if and only if the metric vector contains 0. For PCNF, there is no 0.

$$\mu\left(\langle x,y\rangle\right)=min_j\sum_{i=1}^{n}x_{ij}\veebar y_{ij}$$

In addition to (null-local predicates) of literals (constants), unary (semantic) predicates for the type of an element of an sc-text and unary and binary (syntactic) predicates for element incidence tuples (variables) can be considered (by analogy of relation algebra [22], [23]). In the case of search by pattern (homomorphism), the metric is calculated on multisets. Possible task is to minimize (maximize) the metric from the pragmatic view of logics. Associated with the search for a relevant structure, this task is of practical importance in reference and testing (checking) (dialog) systems [21], [24]. Also, metrics with other quantor elimination approaches ([23], [25]) can be used for logical inference and theorem proving purposes. The system of natural inference and sequent calculus consider finite sets of formulas. One of the algorithms for solving inference problems is a conflict-driven (contradiction-driven) clause learning (CDCL) [26]. These techniques seemed to be promising. To express complex patterns and regularities, it is possible to construct a metatheory using metastructures. For this, meta-relations and modal operators are introduced. Such a formalism allows describing the complex introspective reasoning characteristic of modal logics (see Fig. 26–28 for the sages hat puzzle solution [27]).

Applied logics [27]–[32] consider applications of classical logic to abstract and subject domains describing reality: logical theories about equality and order relations [28], [29], logical theories of arithmetic [28], logical theories of time [27], [31], logical proof theories [28], [30], graph and geometric theories [32], theories of natural and social systems [29].

Classification of logical theories corresponds to the classification of subject domains. Let us consider some concepts and examples that are considered within applied logics.

slot binary relation

⇒ *note**:
[slot binary relation is a slot sc-relation that is a set.]

non-slot binary relation

⇒ *note**:
[non-slot binary relation is a binary sc-relation that is a set but is not a slot sc-relation.]

irreflexive slot binary relation

⊂ *irreflexive binary relation*

⇒ *note**:

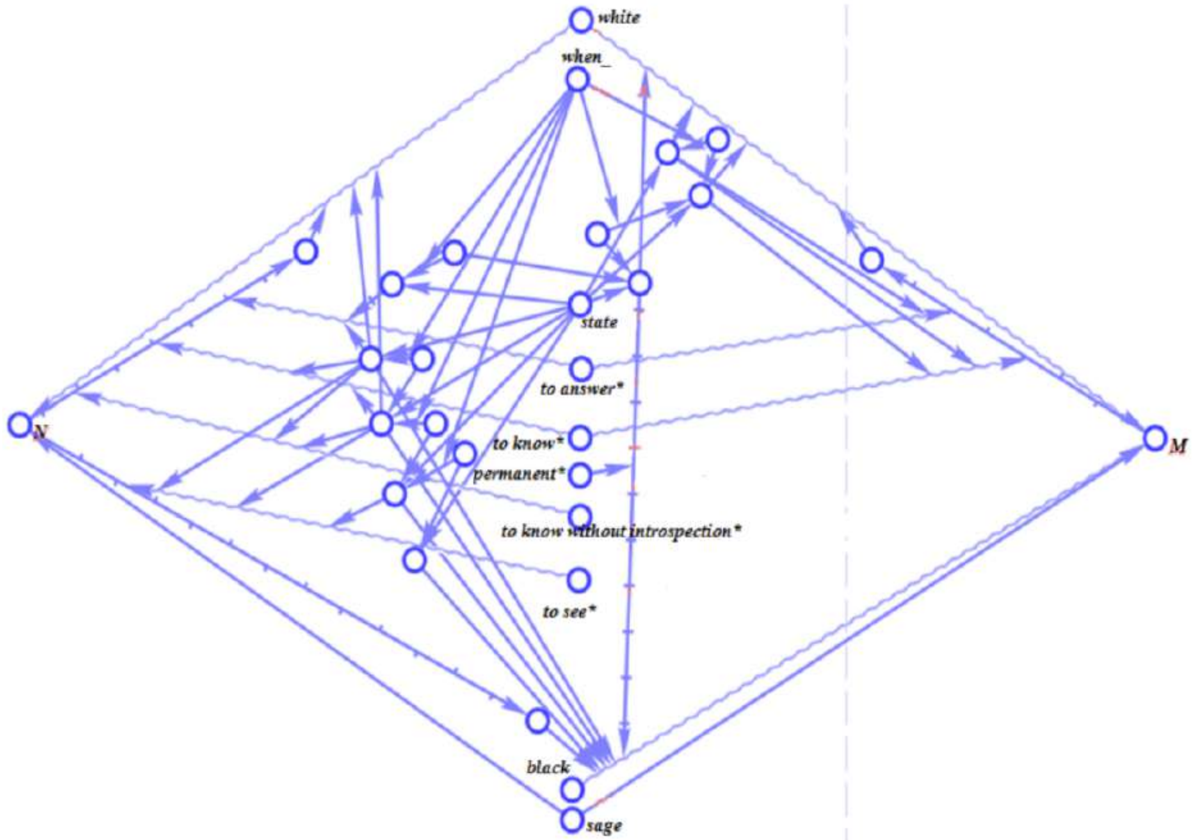

Figure 26. Example of solving the puzzle with sages hats for one of two sages.

[An irreflexive slot binary relation is a slot (binary) relation. Any tuple of which is not denoted by a loop arc (an arc with the same beginning and end).]

irreflexive non-slot binary relation
⊂ *irreflexive binary relation*
⇒ *note**:
[An irreflexive non-slot binary relation is a non-slot binary sc-relation. Different memberships of any tuple of which are the memberships of different elements.]

reflexive slot binary relation
⊂ *reflexive binary relation*
⇒ *note**:
[A reflexive slot binary relation is a slot binary sc-relation. For any tuple's element of which, there is a its tuple denoted by a loop arc (an arc with the same beginning and end).]

reflexive non-slot binary relation
⊂ *reflexive binary relation*
⇒ *note**:
[A reflexive non-slot binary relation is a non-slot binary sc-relation. For any tuple's element of which, there is a tuple with two different memberships of this element.]

transitive slot binary relation
⊂ *transitive binary relation*
⇒ *note**:
[A transitive slot binary relation is a slot binary relation. For any two tuples of which, the end of one of them is the beginning of the second, there is a link whose beginning is the beginning of the first link, and the end is the end of the second link.]

transitive non-slot binary relation
⊂ *transitive binary relation*
⇒ *note**:
[A transitive non-slot binary relation is a non-slot binary relation. For which, there is a role relation, the first domain of which is this binary relation such that for any two tuples of this binary relation, if any element's membership to one of them does not belong to this role relation and the

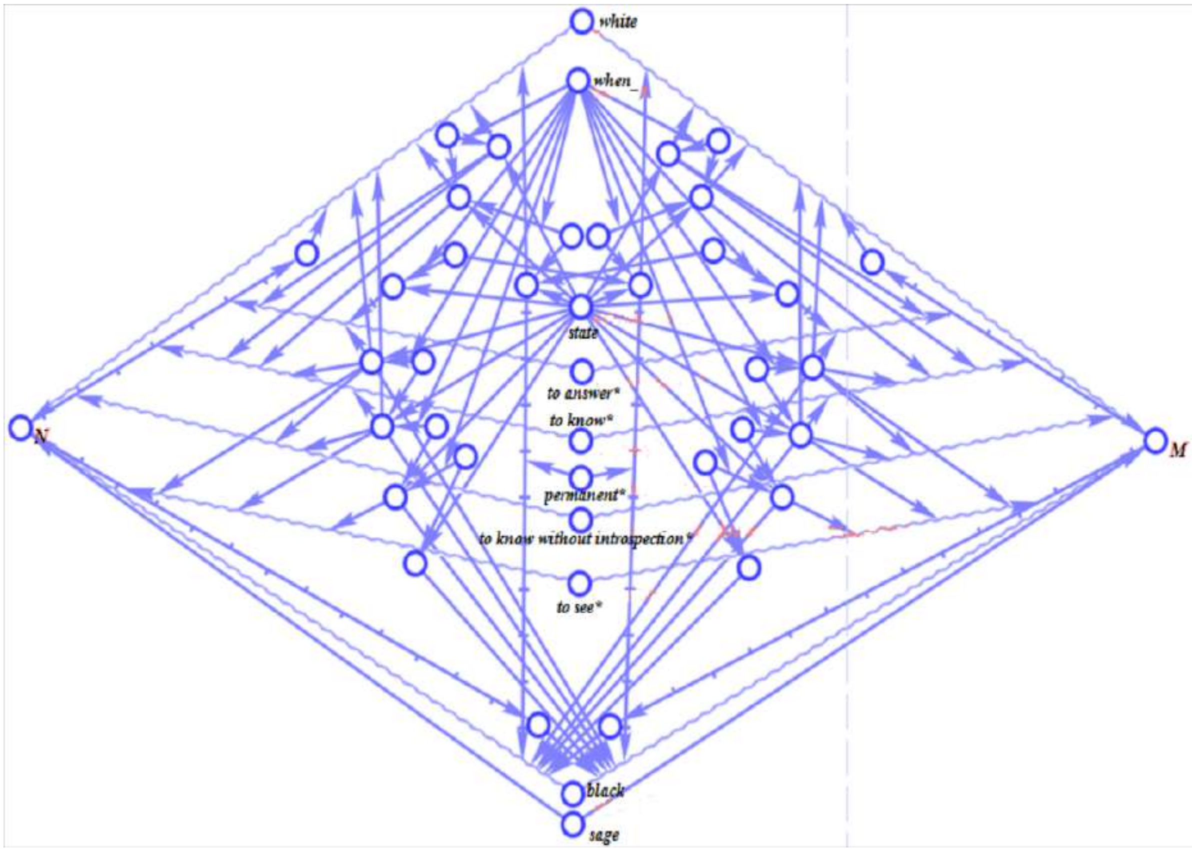

Figure 27. Example of solving the puzzle with sages hats for both of two sages.

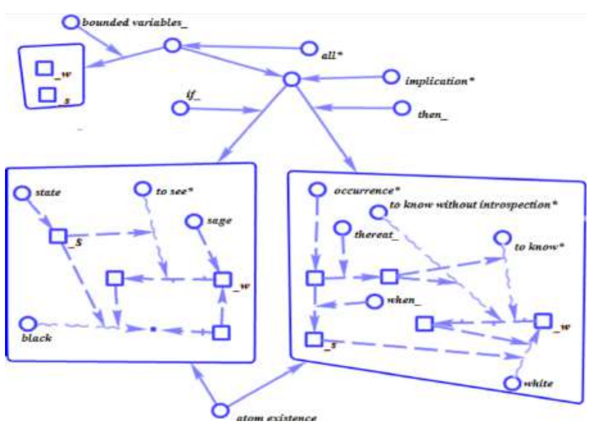

Figure 28. Rule for solving the sages hat puzzle.

membership of the same element to the second tuple belongs to this role relation then there is a tuple with element's membership belonging to the role relation, whose membership to the first tuple belongs to the role relation, and with element's membership that does not belong to the role relation, whose membership to the second tuple does not belong to this role relation.]

symmetric slot binary relation

⊂ *symmetric binary relation*

⇒ *note**:

[A symmetric slot binary relation is a slot binary relation. For any tuple of which, there is a tuple the end of the last is the beginning of the first tuple and the beginning of the last is end of the first (i.e. these tuples are denoted by opposite arcs).]

symmetric non-slot binary relation

⊂ *symmetric binary relation*

⇒ *note**:

[A symmetric non-slot binary relation is a non-slot binary relation. For which, there is a role relation the first domain of which is this binary relation. Also, for any tuple of which, there is a tuple with an element whose membership in the first tuple belongs to the role relation while the membership of that element in the last tuple does not belong to this role relation and with an element whose first tuple membership does not belong to the role relation while its membership in the last tuple belongs to this role relation.]

antisymmetric slot binary relation

⊂ *antisymmetric binary relation*

⇒ *note**:

[An antisymmetric slot binary relation is a slot binary relation. For any tuple of which having a different beginning and end, there is no tuple the end of which is the beginning of the first tuple and the beginning of the last is the end of the first one (i.e. these tuples are denoted by opposite arcs) .]

antisymmetric non-slot binary relation

⊂ *antisymmetric binary relation*

⇒ *note**:

[An antisymmetric non-slot binary relation is a non-slot binary relation. For which, there is a role relation the first domain of which is this binary relation. For any tuple of which, there is no tuple with an element whose membership in the first link belongs to the role relation while its membership to the second link does not belong to this role relation and with an another element whose first tuple membership does not belong to the role relation but its membership to the second tuple belongs to this role relation.]

monotonic slot binary relation*

⊂ *monotonic binary relation**

⇒ *note**:

[Monotone slot binary relation * is slot binary relation with respect to an order relation. Thus, if there is a tuple of this binary relation then for any of its two tuples, the beginning of the second is connected by the tuple of this order relation with the beginning of the first one, there is a third tuple of the binary relation the beginning of which coincides with the beginning of the second tuple and the end coincides with the end of the first.]

monotonic relation order relation'

⇒ *first domain**:
*monotonic binary relation**

⇒ *second domain**:
order relation

monotonic binary relation'

⇒ *first domain**:
*monotonic binary relation**

⇒ *second domain**:
monotonic binary relation

monotonic non-slot binary relation*

⊂ *monotonic binary relation**

⇒ *note**:

[Monotonic non-slot binary relation* is non-slot binary relation with respect to an order relation. For which, there is a role relation, the first domain of which is this binary relation, such that if there is a tuple of this binary relation, then for any of its two tuple, if the element belongs to the last of them under this role relation unlike the other one and also it is connected by the tuple of this order relation with the element belonging to the first tuple under the same role relation in contrast to another element of the first tuple then there is a third tuple of this binary relation such that the element belonging to it under the role relation belongs to the second tuple under the same role relation and the element belonging not under this role relation to the third tuple belongs without this role relationship to the first tuple.]

slot equivalence relation

⊂ *equivalence sc-relation*

⇒ *note**:

[A slot equivalence relation is a slot transitive binary relation which is a slot reflexive and symmetric relation.]

non-slot equivalence relation

⊂ *equivalence sc-relation*

⇒ *note**:

[A non-slot equivalence relation is a non-slot transitive binary relation that is a non-slot reflexive and symmetric relation (over the respective domains).]

slot non-strict order relation

⊂ *sc-relation of non-strict order*

⇒ *note**:

[A slot relation of nonstrict order is a transitive binary relation that is reflexive and antisymmetric.]

non-slot non-strict order relation

⊂ *sc-relation of non-strict order*

⇒ *note**:

[A non-slot non-strict order relation is a transitive binary relation that is reflexive and antisymmetric (with respect to the respective domains).]

Inference relation

⊃ *Inference relation on finite sets*

  ⊃ *Inference relation on finite sets of full-connectively represented sets*

∈ *reflexive binary relation*

∈ *transitive binary relation*

∈ *monotonic binary relation*

⇒ *note**:

[The inference relation is a reflexive, transitive, monotone binary relation on sets of premises (judgments (propositions), logical formulas). The properties of the inference relation are the rules of inference by Gentzen.]

sequent
⇒ *note**:
[A sequent is a tuple (of an implicative form) between a conjunctive set of logical formulas (conjunction) and a disjunctive set of logical formulas (disjunction). An example of a sequent is an expression (judgement) like: $A_1 \wedge A_2 \wedge ... \wedge A_n \Rightarrow\ C_1 \vee C_2 \vee ... \vee C_m$.]

antecedent'
⇒ *first domain**:
sequent
⇒ *second domain**:
conjunction

consequent'
⇒ *first domain**:
sequent
⇒ *second domain**:
disjunction

Inference relation on sequents
⇒ *note**:
[The inference relation on sequents satisfies the rules of inference of the sequent calculus.]

metastructure
⇒ *note**:
[A metastructure is a structure whose full-connectively represented element is another structure.]

modal operator
⇒ *note**:
[A modal operator is a logical connectives that links a logical formula to a structure (and sometimes other elements) in a metastructure. An example of a modal operator is the knowledge operator: Δ.]

modal inference rule
⇒ *note**:
[A modal inference rule the modal operator binding of a formula is true (has a true interpretation) in a structure if and only if the formula is true (has a true interpretation) in the structure that precedes it. An example of an inference rule is the knowledge operator: $\Gamma \cup \{\alpha\} \vdash \Gamma \cup \{\Delta\alpha\}$.]

relation of becoming of structures
⇒ *note**:
[The relation of becoming of structures is a binary relation on the set of structures having a non-empty common support. The roles in the tuples of the relation of becoming are the role relations of the previous structure (preceding structure) and the subsequent structure.]

thinking sequence
:= [fate of thinking]
:= [thought]
⇒ *note**:
[A sequence of thinking is a sequence of sc-sets of propositions (logical formulas).]
⇒ *subdividing**:
{• *irrational thinking sequence*
• *rational thinking sequence*
}

sequence of rational thinking of classical logic
:= [fate of rational thinking of classical logic]
⊂ *rational thinking sequence*
⇒ *note**:
[A sequence of rational thinking is a sequence (given by the relation of becoming of structures) of (classically) satisfible sc-sets (sc-subsets or sc-supersets) of propositions.]

sequence of classical rational deductive thinking
⊂ *sequence of rational thinking of classical logic*
⊃ *sequence of rational deductive thinking of classical logic on finite sc-sets*
⇒ *note**:
[A sequence of classical rational deductive thinking is a sequence of rational thinking, the sequence (of becoming) of satisfiable sc-sets of propositions deductively logically following (according to classical rules) one after another.]

sequence of classical rational deductive cognition
:= [will]
⇒ *note**:
[A sequence of classical rational deductive cognition is a sequence (given by the relation of becoming of structures) of non-contradictory sc-supersets of propositions logically following from one to another.]

Non-classical logics [27], [29]–[31], [33], [34] consider (1) non-classical inference whose inference relation having unusual properties [27], [29]–[31], following which, it is possible or impossible to deduce results which is deduced in classical logic, as well as (2) other scales of truth constants for logical formulas, their interpretations, and [33], [34] values that are different from false and (ground) truth.

The event-like nature of event (distensible) sets [3], [4], used by the languages of the unified knowledge representation model, allows naturally not only representing transitions between deducible sets of facts under the conditions of the open world assumption (hypothesis) but also the non-monotonic modifiable reasoning corresponding to them under the conditions of the closed world assumption (hypothesis). To implement non-monotone inference, special relations of non-monotone inference are introduced. Modifiable reasoning considers four kinds of predicates:

$$P1_i = P1_j$$
$$P2_i \subseteq P2_j$$
$$P3_i \supseteq P3_j$$
$$P4_i ? P4_j$$

An efficient solution of the problem is possible by adding additional restrictions on the class of considered formulas and functions (Horn clauses, monotone predicates). The transition to fuzzy logic allows reducing a discrete problem to a continuous problem (embed it in a continuous model which is close to quantor elimination methods). The used formalism of event (distensible) sets [3], [4] allows representing fuzzy logic expressions in a natural way. Let us consider some definitions of fuzzy connectives that link and express causality and exclusivity in Lukasiewicz's fuzzy logic [34].

$$\varphi \tilde{\rightarrow} \psi$$
$$\tilde{1} \overset{\text{def}}{=} \tilde{0} \tilde{\rightarrow} \tilde{0}$$
$$\sim \varphi \overset{\text{def}}{=} \varphi \tilde{\rightarrow} \tilde{0}$$
$$\varphi \tilde{\wedge} \psi \overset{\text{def}}{=} (\varphi \tilde{\rightarrow} \psi) \tilde{\rightarrow} \psi$$
$$\varphi \tilde{\vee} \psi \overset{\text{def}}{=} \sim (\sim \varphi \tilde{\wedge} \sim \psi)$$
$$\varphi \tilde{\oplus} \psi \overset{\text{def}}{=} (\sim \varphi) \tilde{\rightarrow} \psi$$
$$\varphi \tilde{\otimes} \psi \overset{\text{def}}{=} \sim \left(\sim \varphi \tilde{\oplus} \sim \psi\right)$$
$$\varphi \tilde{\ominus} \psi \overset{\text{def}}{=} \sim (\varphi \tilde{\rightarrow} \psi)$$
$$\varphi \tilde{\leftrightarrow} \psi \overset{\text{def}}{=} (\varphi \tilde{\rightarrow} \psi) \vee (\psi \tilde{\rightarrow} \varphi)$$

To implement the inference in other non-classical logics, including those dealing with subject domains in which non-deterministic sets and structures are considered, additional concepts are also considered.

nonmonotonic inference on finite sc-set of premises
⇒ *note**:
[Nonmonotonic inference on a finite sc-set of premises is a relation between (finite) sc-sets of true logical statements (premises). If there is no embedding of the structure of an atomic logical formula in the relational structure (sc-subset of the subject area) of the sc-set of true (consistent) premises and the negation of this atomic formula is true with respect to them, then there is an sc-set with a relational structure belonging to it, including all elements of the previously mentioned relational structure and the constants of this atomic formula, to which all premises of the previously mentioned sc-set of true (consistent) premises and the mentioned atomic logical formula belong.**]**

inferencing set
⇒ *note**:
[An inferencing set is an event sc-set, the (temporary) belonging of logical formulas to which is established in the order of formation of the process of deriving these logical formulas.**]**

fuzzy truth*
⇒ *note**:
[Fuzzy truth connects a finite sc-set with temporal belongings on a finite set of finite phenomena of belonging to a statement. On the membership phenomena, a finite sc-subset of the sc-relation of becoming (immediately before, immediately after) is given, which defines the structure of the corresponding sc-subsets. This structure is a directed tree. Fuzzy truth is a binary relation between the (fuzzy) membership of a link of a formal theory statement and a finite sc-set and a real number from 0.0 to 1.0. The fuzzy truth of the negation of the statement is equal to the difference 1.0 and the fuzzy truth of the statement belonging to the negation. The fuzzy truth of the conjunction of propositions does not exceed the minimum of the fuzzy truth of the elements of this conjunction and is not lower than the (boundary or drastic) product of the fuzzy truth of the same elements of the conjunction. The fuzzy truth of a disjunction does not exceed the (boundary or drastically) sum of the fuzzy truth of the elements of this conjunction and not less than the maximum fuzzy truth of the same elements of the conjunction. The fuzzy truth of atomic propositions is equal to the arithmetic mean of the isomorphic embedding of the proposition structure in each of the sc-subsets of the finite sc-set that are included in the structure given by the becoming sc-relation.**]**

constructively true proposition*
⇒ *note**:
[A constructively true proposition* is a subset of a true proposition*. True atomic logical formulas or their true interpretations are constructively true if and only if they have an isomorphic embedding in the domain where all elements of the embedding

are full-connectively represented. A conjunction of constructively true logical formulas (or having corresponding constructively true interpretations) is constructively true (or has a constructively true corresponding interpretation). The disjunction of at least one constructively true logical formula (or having a corresponding complete constructively true interpretation) is constructively true (or has a constructively true corresponding interpretation). The negation of a false logical formula (or having a false corresponding interpretation) is constructively true (or has a constructively true interpretation). If all the logical formulas in the disjunction are false (have corresponding false interpretations), then the disjunction is also false (has a corresponding false interpretation). The negation of a false logical formula (or having a false corresponding interpretation) is constructively true (or has a constructively true interpretation). An implication with a false premise (or having a corresponding false interpretation) is constructively true (or has a constructively true interpretation). An implication with a constructively true consequence (or having a corresponding constructively true interpretation) is constructively true (or has a constructively true interpretation). A constructively true implication (or having a constructively true interpretation) with a constructively true premise (or having a corresponding constructively true interpretation) has a constructively true consequence (or having a corresponding constructively true interpretation). A constructively true implication (or having a constructively true interpretation) with a false consequence (or having a corresponding false interpretation) has a false premise (or having a corresponding false interpretation). The existence of variable values for a logical formula is constructively true (or has a corresponding constructive true interpretation) if the universality of variable values for that logical formula is constructively true (or has a corresponding constructive true interpretation). If a logical formula has only constructively true corresponding interpretations, then the universality of the values of the variable for this logical formula is constructively true (or has a corresponding constructively true interpretation).]

right proposition*
⇒ *note**:
[A right proposition is a proposition that is true or uncourrupted.]

uncorrupted proposition*
⇒ *note**:
[An uncorrupted proposition is a proposition whose truth or falsity (untruth) does not lead to a contradiction.]

The approach based on the semantic space [4], [8], in addition to considering useful metric properties, allows imposing additional requirements on the topological properties of the corresponding logics, when all substructures of the relational structure of the subject domain, on which the logical formulas are interpreted, are meaningfully closed.

IV. Conclusion

In reference and testing (checking) systems, there is a problem of analyzing complex answers characteristic of modified reasoning. The representation of such answers as logical constructions in the semantic space makes it possible to quantify the correctness of such answers, as well as the quality of the knowledge that the system is able to acquire in the process of dialog within the OSTIS Ecosystem.

References

[1] V. Golenkov, N. Gulyakina, and D. Shunkevich, *Standart otkrytoi tekhnologii ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheski sovmestimykh gibridnykh intellektual'nykh komp'yuternykh sistem [Open technology standard for ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint, 2021, (In Russ.).

[2] V. V. Golenkov and N. A. Gulyakina, "Next-generation intelligent computer systems and technology of complex support of their life cycle," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 27–40, 2022.

[3] V. P. Ivashenko, "Modeli i algoritmy integratsii znanii na osnove odnorodnykh semanticheskikh setei (diss. na soiskanie stepeni kand. tekhn. nauk: 05.13.17) [models and algorithms for knowledge integration based on homogeneous semantic networks (diss. for the degree of candidate of technical sciences: 05.13.17)]," dis. ... kand. tekhn. nauk : 05.13.17, Belorus. gos. un-t informatiki i radioelektroniki [Belarusian State University for Informatics and Radioelectronics], Minsk, 2014, (In Russ.).

[4] V. Ivashenko, "General-purpose semantic representation language and semantic space," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 41–64, 2022.

[5] ——, "String processing model for knowledge-driven systems," *Doklady BGUIR*, vol. 18, pp. 33–40, 10 2020.

[6] V. Golenkov, V. Golovko, N. Guliakina, and V. Krasnoproshin, "The standardization of intelligent computer systems as a key challenge of the current stage of development of artificial intelligence technologies," pp. 73–88, 2020.

[7] W. Lowe, "Towards a theory of semantic space," 01 2001, pp. 576–581.

[8] Y. I. Manin and M. Marcolli, "Semantic spaces," *Mathematics in Computer Science*, vol. 10, pp. 459–477, 2016.

[9] I. Mel'cuk, *Language: from Meaning to Text*, D. Beck, Ed. Moscow & Boston: Academic Studies Press, 04 2016.

[10] Y. M. A. Kostrikin, *Linear Algebra and Geometry*. Gordon and Breach Science Publishers, 1997.

[11] D. Bohm and B. Hiley, *The Undivided Universe: An Ontological Interpretation of Quantum Theory*. London: Routledge, 1993.

[12] D. Bohm, *Wholeness and the Implicate Order*. London: Routledge, 2002.

[13] A. N. Gordei. Theory for automatic generation of knowledge architecture: Tapaz-2. Available at: http://tapaz.by (accessed 18.04.2021).

[14] A. Hardzei, M. Sviatoshchik, L. Bobyor, and S. Nikiforov, "Universal semantic markup and top-level ontology representation," in *Open Semantic Technologies for Intelligent Systems*. Springer International Publishing, 2022, pp. 192–221. [Online]. Available: https://doi.org/10.1007/978-3-031-15882-7_11

[15] M. A. Armstrong, *Basic Topology*. M.: Springer, 1983.

[16] L. Collatz, *Functional Analysis and Numerical Mathematics*. New York, San Francisco, London: Academic Press, 1966.

[17] A. Narin'yani, "Ne-faktory: kratkoe vvedenie [non-factors: a brief introduction]," *Novosti iskusstvennogo intellekta [AI news]*, no. 2, pp. 52–63, 2004.

[18] V. P. Ivashenko, "Ontologicheskaya model' prostranstvenno-vremennykh otnoshenii sobytii i yavlenii v protsessakh obrabotki znanii [ontological model of the space-time relations of the event and the phenomenon in the processes of knowledge processing," *Vestnik Brestskogo gosudarstvennogo tekhnicheskogo universiteta [Bulletin of Brest State Technical University]*, vol. 107, no. 5, pp. 13–17, 2017, (In Russ.).

[19] V. M. Kuz'mitskii, "Printsipy postroeniya grafodinamicheskogo parallel'nogo komp'yutera, orientirovannogo na reshenie zadach iskusstvennogo intellekta [principles of constructing a graph-dynamic parallel computer focused on solving artificial intelligence problems]," dis. ... kand. tekhn. nauk : 05.13.11 [dis. ... Cand. of tech. sciences: 05.13.11], Minsk, 2000, 236 p.

[20] L. Kalinichenko, *Metody i sredstva integratsii neodnorodnykh baz dannykh [Methods and tools for integrating heterogeneous databases]*. M.: Nauka, 1983, (In Russ.).

[21] V. Ivashenko, "Application of an integration platform for ontological model-based problem solving using an unified semantic knowledge representation," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 179–186, 2022.

[22] V. P. Ivashenko, *Modeli resheniya zadach v intellektual'nykh sistemakh. V 2 ch. Ch. 1 : Formal'nye modeli obrabotki informatsii i parallel'nye modeli resheniya zadach : ucheb.-metod. posobie [Models for solving problems in intelligent systems. In 2 parts, Part 1: Formal models of information processing and parallel models for solving problems: a tutorial]*. Minsk : BSUIR, 2020, (In Russ.).

[23] A. Tarski, "On the calculus of relations," *J. Symbolic Logic*, vol. 6, 09 2014.

[24] W. Li and L. Qian, "Semantic approach to user answer verification in intelligent tutoring systems," in *Open Semantic Technologies for Intelligent Systems*, V. Golenkov, V. Krasnoproshin, V. Golovko, and D. Shunkevich, Eds. Springer International Publishing, 2022, pp. 242–266. [Online]. Available: https://doi.org/10.1007/978-3-031-15882-7_13

[25] T. Imielinski and W. Lipski, "The relational model of data and cylindric algebras," *J. Comput. Syst. Sci.*, vol. 28, pp. 80–102, 1984.

[26] Z. Lintao, C. Madigan, M. Moskewicz, and S. Malik, "Efficient conflict driven learning in a boolean satisfiability solver," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers*, 02 2001, pp. 279–285.

[27] A. Teiz, P. Gribomon, G. Yulen, A. Pirot, D. Rolan, D. Snaiers, M. Vokler, P. Goshe, P. Vol'per, E. Greguar, and F. Del'sart, *Logicheskii podkhod k iskusstvennomu intellektu: ot modal'noi logiki k logike baz dannykh [Logic approach to Artificial Intelligence: from modal logic to database logic]*, G. Gavrilov, Ed. M.: Mir, 1998, (In Russ.).

[28] S. K. Klini, *Matematicheskaya logika [Mathematical logic]*. Moskva : Mir, 1973, (In Russ.).

[29] A. Teiz, P. Gribomon, and Z. L. i dr., *Logicheskii podkhod k iskusstvennomu intellektu: ot klassicheskoi logiki k logicheskomu programmirovaniyu [Logic Approach to Artificial Intelligence: From Classical Logic to Logic Programming]*, G. Gavrilov, Ed. M.: Mir, 1990, (In Russ.).

[30] A.G.Dragalin, *Konstruktivnaya teoriya dokazatel'stv i nestandartnyi analiz [Constructive evidence theory and non-standard analysis]*, G. E. Mints, M. K. Valiev, E. G. Dragalina-Chernaya, N. M. Nagornyi, N. N. Nepeivoda, V. N. Sadovskii, and E. D. Smirnova, Eds. Editorial URSS, 2003, (In Russ.).

[31] V. N. Vagin, E. Y. Golovina, A. A. Zagoryanskaya, M. V. Fomina, and D. A. Pospelov, *Dostovernyi i pravdopodobnyi vyvod v intellektual'nykh sistemakh [Reliable and plausible inference in intelligent systems]*, 2nd ed., pod red. V. N. Vagina and D. A. Pospelova, Eds. M. : Fizmatlit, 2008, (In Russ.).

[32] V. Golenkov, V. Tarasov, O. Eliseeva, N. Gulyakina, T. Lemesheva, N. Bezzubenok, V. Ivashenko, T. Gavrilova, E. Vasil'eva, V. Emel'yanov, R. Serdyukov, S. Akulich, O. Lubnevskii, and Y. Tel'nov, *Intellektual'nye obuchayushchie sistemy i virtual'nye uchebnye organizatsii: Monografiya [Intelligent Learning Systems and Virtual Learning Organizations: Monograph]*, V. Golenkov and V. Tarasov, Eds. Mn.: BGUIR, 2001, (In Russ.).

[33] I. Z. Batyrshin, A. O. Nedosekin, A. A. Stetsko, V. B. Tarasov, A. V. Yazenin, and N. G. Yarushkina, *Nechetkie gibridnye sistemy. Teoriya i praktika [Fuzzy hybrid systems. Theory and practice]*, pod red. N. G. Yarushinoi, Ed. M. : Fizmatlit, 2007, (In Russ.).

[34] L. Behounek, P. Cintula, and P. Hájek, *Introduction to mathematical fuzzy logic*, 01 2011, vol. 37, pp. 1–101.

Интеграция логических моделей представления и обработки знаний в смысловом пространстве

Ивашенко В. П.

В данной статье рассматривается подход к представлению логических структур и схем, как моделей логической обработки знаний, в семантическом пространстве в виде семантических сетей. Исследуются некоторые свойства семантического пространства и логических моделей, такие как топологические, метрические и валюационные (нормовые) свойства. Предложены понятия для онтологического представления классических и неклассических логических формул, классов и отношений.

Received 13.03.2023