

# Semantic Coding for Semantic Knowledge Inference

Igor Boyko

Belarus State University

Minsk, Belarus

Email: igor\_m\_boyko@hotmail.com

**Abstract**—A pressing task of developing the effective knowledge inference tools attracts researchers of different interests in Computer Science. Semantic coding is a main component of computational semantics for knowledge inference on a semantically formalized basis. Universal semantic code (USC) of professor V.V.Martynov is a tool of semantic coding for meaning representation and inference. USC converts knowledge in the semantic code, implement axiomatic inference, and convert the semantic code into natural language. Independence of the semantic code from any kind of natural language is a tool of real-time multilingual knowledge generating and knowledge exchange through computer.

**Keywords**—universal semantic code, knowledge representation, knowledge inference, formal semantics, natural language processing, semantic coding, verb classification.

## I. INTRODUCTION

Can we explain how our long-time memory works? For instance, you just have listened the scientific report and memorized it. Later, on your department meeting, you retold the report to your coworkers. However, try to compare the initial text of the report with what you retold and you notice the essential difference between two verbal representations of the same report. If every listener of the report retells it after you, you will get many different verbal representations of the same text or the same knowledge.

Human memory determines the verbal perception. However, it is a question, in what form does our mind record the initial text? This is impossible to remember the content of the text without some expression tool. A psychologist Hoffman [8] writes, "... the information in the memory can be kept only in the form of the specific code." Repeating this idea several times, he assumes - some internal semantic code works between human information input and output.

Modeling the internal semantic code for the meaning representation and inference in the informational computer systems is a crucial problem of artificial intelligence.

USC defines semantic criteria in the following way [12], [13]:

- 1) Every semantic string of symbols corresponds to only one meaning and every string conversion corresponds to only one meaning conversion.
- 2) Declarative knowledge represented as procedural. Each object in the system considered not from the point what it is but what function it performs.
- 3) A USC verb classifier defines names of the classes and gives their symbolic representation paired with natural language interpretation

It seems expedient to determine the place USC occupies within the scope of formal knowledge representation models designed for artificial intelligence (AI) systems. Formal grammars generated based on a Chomsky [4] conception could be used for creating programming languages, but absence of semantic interpretation does not allow them to become knowledge representation languages with one to one correlation between syntactic and semantic elements.

As per Amarel [1] and Schank [17] priority in intellectual problem solving is given to knowledge representation, thus the necessity of developing language system has been predetermined. Natural language in its non-canonized form cannot meet this requirement [19]. A formal system with full semantic interpretation is embodied in Montague [16] grammar. However, Montague semantics is defined on the natural language word level without exposing its inner semantic structure. We could say by analogy that in this case the research covers only a molecular level (an atomic structure is missing), and easy to imagine what the level would have reached our technology if physics had shared the same fate.

The interpretation of Rene Tom [18] has the partial similarity of signs with their spatial analogues. For example,  $\rightarrow$  is the conjunction,  $\leftarrow$  is the separation,  $\vdash$  is the start,  $\dashv$  is the end. Unfortunately, it is impossible to convert one interpretation into another without formal representation of the notions.

Frames of Minsky [15], associative nets and semantic networks, first order predicates logic or modal logic are the most traditional models [9]. Several conventions for knowledge modeling are determined including production rules [5], structured objects [7] and logic programming [10]. The main disadvantage of them is the absence of tools of the word meaning representation. The understanding of the problem had initiated the development of the fuzzy and pseudo-physical logics [6].

There are three projects to create a language with formalized semantics, besides USC: the model of conceptual dependence proposed by Schank [17] the "meaning-text" model of Melchuk [14], and the theory of automatic generation of knowledge architecture (TAGKA) of Hardzei [21]. The basis of the first two are some primitives (semantic elements): primitive actions in Schank's model and lexical functions in Melchuk's, which form semantic notation of utterances. The primitives of the given models do not claim to be complete, independent and consistent in the strict sense of the word because of their empirical elaboration. USC is the first project of the language deductive theory of knowledge representation.

Hardzei proposes the theory having in the foundation semantic formalisms for knowledge representation and knowledge inference as well. The theory is based on USC of prof.

V.V. Martynov where were proposed semantic primitives, i.e. semantically irreducible kernel words, and the rules of their combinatorics were defined. In general, semantic coding is a process of converting natural language phrases to a chain of semantic primitives or semantic formulas and back. The author sees the essential difference of TAGKA vs USC in the method of defining a structure of the semantic formula and operations of conversion of semantic formulas to each other. Semantic coding has a crucial difference with Semantic Web where is no semantic formulas but semantic tags represented not formally but by means of natural language.

## II. LINGUISTIC INTERPRETATION AND INFERENCE

There is in USC a formulated set of semantic axioms within the scope of some algebra. Each axiom represents a regular conversion of sense in explicit form. No kind of artificial intelligence systems can exist without semantic explication of sense.

For example, it is natural enough for a human to come to the following conclusion: ‘The engineer has seen the device before that is why he would recognize it’ or in a more general form,  $X$  has seen  $Y \rightarrow X$  would recognize  $Y$ . If our system would be intellectual, it would know how to draw this immediate inference. In other words, the developer of AI system has to know the way of teaching a computer to draw such a kind of inference. Regretfully he does not know how to do it. Moreover, he cannot perceive how the human does it.

Here is another case of deduction: ‘He has already played Rossini’s “Tarantella” that is why he would play it’ or in a more general form,  $X$  has played  $Y \rightarrow X$  can possibly play  $Y$ . A human identifies the verb ‘play’ in spite of grammatical differences. Though the first deduction is reducible to the same postulate, we do not know how to. The human pure intuitively easily uses such deduction [13].

Formalization of lexical semantics cannot solve AI problems because of the natural language vagueness that follows from the discrepancy between the complexity of the syntactic and semantic structures. Such discrepancy arises due to the ellipticity of natural language phrases. Thus the following phrase “Your child eats with his hands” is reconstructed in full as “Your child eats with his mouth, holding food with his hands” [19].

Making comparative analysis of the following phrases “John beats Jim” and “John expects Jim” we figure out that in spite of their full syntactic coincidence they have important semantic distinction. Asking the question “what does John do to Jim?” we get a regular answer “He beats him” and meaningless, in this case, “He expects him”. Actually, the phrase “John beats Jim” has an “atomic” semantic structure while the phrase “John expects Jim” has a “molecular” one. Semantic reconstruction of the second phrase is: “John is staying where he expects Jim to come soon” [13].

## III. USC VERB CLASSIFIER

In the semantic verb classifier, each verb stays in some class. USC operates with 54 verb classes of physical and 54 verb classes of informational strings [20].

Each verb is coded by the USC string defining a number of arguments of the string and their roles:  $X$  – subject,  $Y$  – instrument,  $Z$  – object,  $W$  – result.

How to use the classifier? Suppose, in the user’s text there is a sentence “The master restores a painting”. The verb “restore” is a name of the class and according to its USC string  $((XY)Z)((ZW)W)$  we can construct a phrase:  $X$  by means of  $Y$  affects  $Z$  so that  $W$  is restored. The user can specify the role of the variables like that: “An artist by means of a brush affects on the paint so that the painting is restored”.

Having assigned names of performers of the action, represented by the verb, and tools, that are being used, to the variables the user determines the initial situation (he does not realize that now he is developing a knowledge base) and the target situation. The user just has to fill out a form of the following type: who “ $X$ ” by means of what “ $Y$ ” acting on what “ $Z$ ” gets what “ $W$ ” and computer provides possible ways of changing from the initial to the final state.

## IV. USC AXIOMATICS

Using axioms of conversion, we demonstrate how to generate and convert USC strings and show how this knowledge representation and inference language can become a solver of intellectual problems.

The KB operates with the axioms of the USC algebra and formed as an oriented graph. The nodes of the graph are the USC strings, the arcs are the USC axioms. The solution of the intellectual problem is implemented as a route of the arcs. The inference algorithm is based on the successive drawing of the route from the target situation to the initial one.

### A. Axiom of diffusion

The axiom defines transferring the variable from one position to another in the right part of the string.

- 1) Transferring the variable from the first position to the second:  
 $((XY)Z)((ZY)W) \rightarrow ((XY)Z)((ZZ)W)$
- 2) Transferring the variable from the second position to the third:  
 $((XY)Z)((ZY)W) \rightarrow ((XY)Z)((ZY)Y)$
- 3) Transferring the variable from the first position into the third if and only if variables are not repeated inside the initial string:  $((XY)Z)((ZY)W) \rightarrow ((XY)Z)((ZY)Z)$

On the Fig. 2 see the graph for physical strings conversion with the axiom.

### B. Axiom of transposition

The axiom defines shifting of internal parenthesis in the right part of the string and transition from ‘active’ to ‘passive’ verb:

$$((XY)Z)((ZZ)W) \rightarrow ((XY)Z)(Z(ZW)).$$

On the Fig. 3 see the combined graph for physical strings conversion with the axiom of diffusion and transposition.

1.1	(ZW)Y	<b>Connect</b> - make joined or united  X bmo Y connects Z and W	2.1	(ZY)W	<b>Disconnect</b> - make disconnected, disjoined  X bmo Y disconnects Z and W
3.1	(ZW)Z	<b>Insert</b> - put or introduce into something  X bmo Y inserts Z in W	4.1	(ZZ)W	<b>Extract</b> - force to leave or move out  X bmo Y extracts Z from W
5.1	(ZZ)Y	<b>Squeeze</b> - press firmly  <i>X bmo Y squeezes Z</i>	6.1	(ZY)Z	<b>Expand</b> - make bigger or wider in size, volume, or quantity  X bmo Y expands Z
7.1	(ZY)Y	<b>Destroy</b> - damage irreparably  X bmo Y destroys Z	8.1	(ZW)W	<b>Produce</b> - create or manufacture  X bmo Y produces W from Z
9.1	(ZZ)Z	<b>Change</b> - cause a physical transformation  X bmo Y changes Z			

Figure 1. The example of physical classes

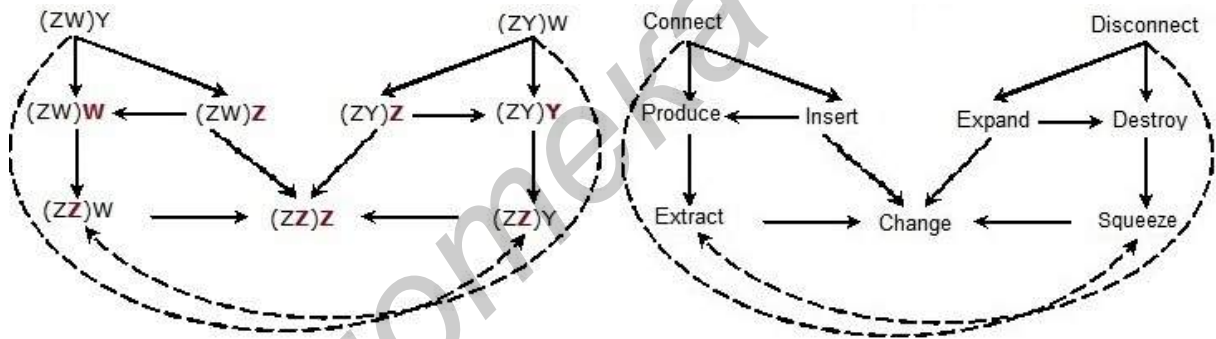


Figure 2. Axiom of diffusion

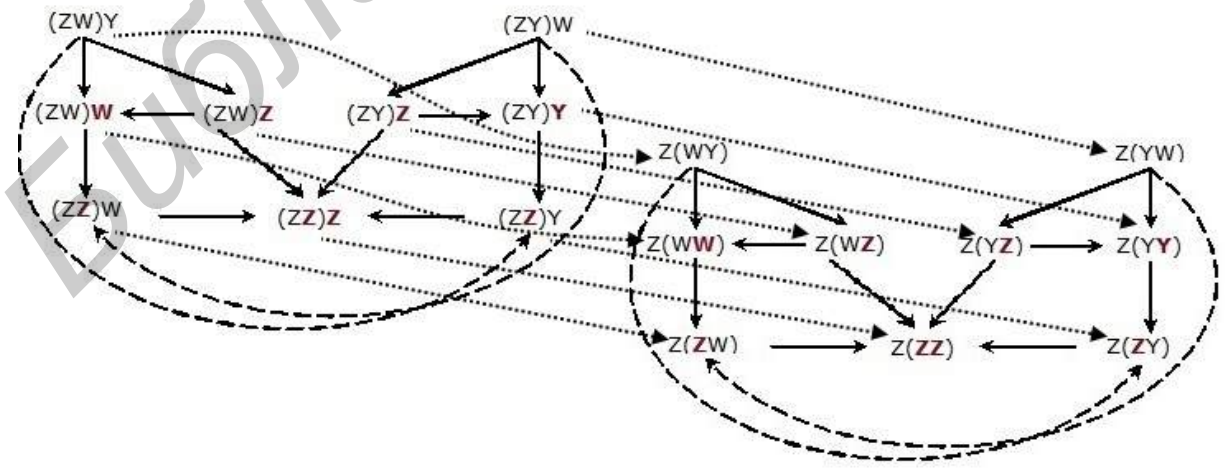


Figure 3. Axioms of transposition and diffusion

### C. Axiom of complement

The axiom defines converting one string into another, in the right part of the string, according to the spatial relation:

$$\begin{aligned} ((XY)Z)((ZZ)W) &\rightarrow ((XY)Z)((ZZ)W') &\rightarrow \\ ((XY)Z)((ZZ)W_j) & & \end{aligned}$$

On the Fig. 4 see the graph for physical strings conversion with the axiom.

The complete graph of USC conversion is on the Fig. 5.

## V. CONCLUSION

Inference is a main procedure of intellectual problems solving including inventive problems. Inventive solution comprises consequence of technical operations. Any technological process is a consequence of the technical operations, where every operation is represented by physical verb (charge, magnetize, bond, split, etc.).

Conversion of the USC strings, representing physical verbs, by means of the USC axioms generates string consequences where each of them can be considered as a technological process. Substitution the real objects in the positions of variables in the USC strings forms a technical solution [2].

Another possible application is a machine translation by converting phrases of one natural language into the USC strings and converting the strings into another natural language. During the translation, USC strings may represent verbs, nouns, and adjectives independently of the particular natural language [3].

Axiomatic conversions of strings represent inference of knowledge from knowledge on the abstract level of verb classes and then with specifying the knowledge by substitution of synonyms in the positions of the classes.

USC displays the situation when the set of semantic primitives is not postulated but recursively calculated, and primitives are displaced in the semantic field in the range of semantic nearness of meanings. Eventually, KB is semantically ordered.

## REFERENCES

- [1] Amarel S., On the Representation of Problems of Reasoning about Actions, *Machine Intelligence*, 3, (1968).
- [2] Boyko I., Computer Semantic Search of Inventive Solutions, *TRIZ Journal*, PA USA, March. (2001).
- [3] Boyko I., Terminological Abstractions for Terminology Classification, 6th International Conference Terminology and Knowledge Engineering, Nancy, France, (2002).
- [4] Chomsky N., *Lecture on Government and Binding*, Dordrecht, (1981).
- [5] Davis R. and King J., *An Overview of Production Systems*, *Machine Intelligence*, 8, New York: Wiley, (1977).
- [6] Dubois D., Prade H., and Yager R., *Fuzzy Sets for Intelligent Systems*, Morgan Kaufmann, San Mateo, CA USA, (1993).
- [7] Findler N., *Associative Networks*. New York: Academic Press, (1979).
- [8] Hoffmann J., *Aktive Gedächtnis: Psychologische Experimente und Theorien zur menschlichen Gedächtnistätigkeit*. VEB Deutscher Verlag der Wissenschaften, Berlin, (1982).
- [9] Jackson P., *Introduction to Expert Systems*, England, (1999).
- [10] Kowalski R., *Logic for Problem Solving*. Amsterdam: North-Holland, (1979).

- [11] Lukasiewicz J., *Elementy Logiki Matematycznej*, Warszawa, (1958).
- [12] Martynov V., *Foundations of semantic coding. Summary.* / European Humanity University. Minsk. 2001.
- [13] Martynov V. *In the middle of conscious of the human.* / Belarus State University. Minsk. 2009. 272 pages. (in Russian)
- [14] Melchuk I., *Experience of Theory of Linguistic Models "Sense  $\Leftrightarrow$  Text"*, Moscow, (1974).
- [15] Minsky M., *Framework for Representation Knowledge, The Psychology of Computer Vision*. New York, (1975).
- [16] Montague R., *Formal Philosophy*. New Haven, London, (1974).
- [17] Schank R., *Conceptual Information Processing*. Amsterdam-Oxford-New-York, (1975).
- [18] Thom, R., *Structural Stability and Morphogenesis*. Benjamin Addison Wesley, New York, (1975).
- [19] Boyko I. *Canonization of Natural Language by Means of Universal Semantic Code.* / International Congress on Computer Science: Informational Systems and Technologies (CSIST 2016), Minsk, 2016.
- [20] Boyko I. *Semantic classification of actions for knowledge inference.* / OSTIS-2016 (Open Semantic Technologies for Intelligent Systems) conference proceeding. Minsk. 2016.
- [21] Hardzei A. *Theory of automatic generation of knowledge architecture.* / OSTIS-2014 (Open Semantic Technologies for Intelligent Systems) conference proceeding. Minsk. 2014. (in Russian)

## СЕМАНТИЧЕСКОЕ КОДИРОВАНИЕ ДЛЯ СЕМАНТИЧЕСКОГО ВЫВОДА ЗНАНИЙ

Бойко И.М.

Статья посвящена семантическому кодированию знаний на основе Универсального Семантического Кода (УСК), который был разработан профессором Виктором Владимировичем Мартыновым.

Семантический вывод знаний осуществляется с помощью набора УСК аксиом, которые задают правила последовательного преобразования знаний.

Семантический код скоординирован с языковым представлением через классификатор глаголов, представляющих действия.

Аксиоматический граф вывода знаний показывает конечность возможных преобразований одних глаголов в другие, тем самым определяя конечность количества возможных способов решения интеллектуальных задач.

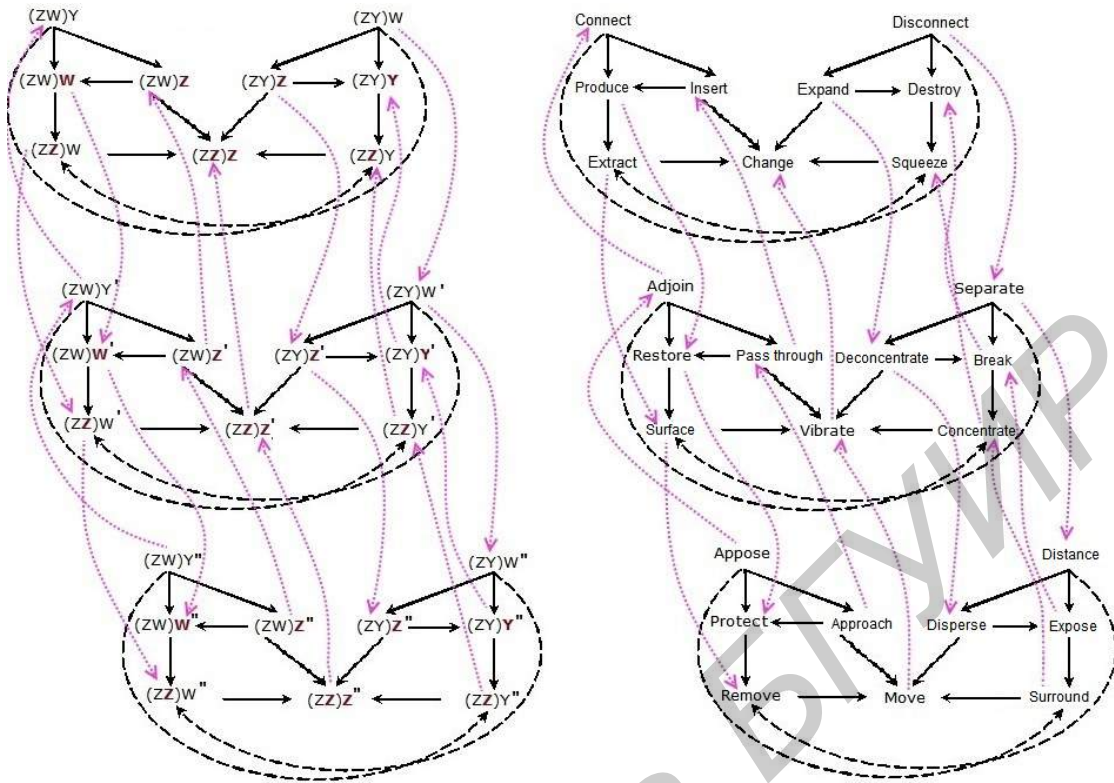


Figure 4. Axiom of complement

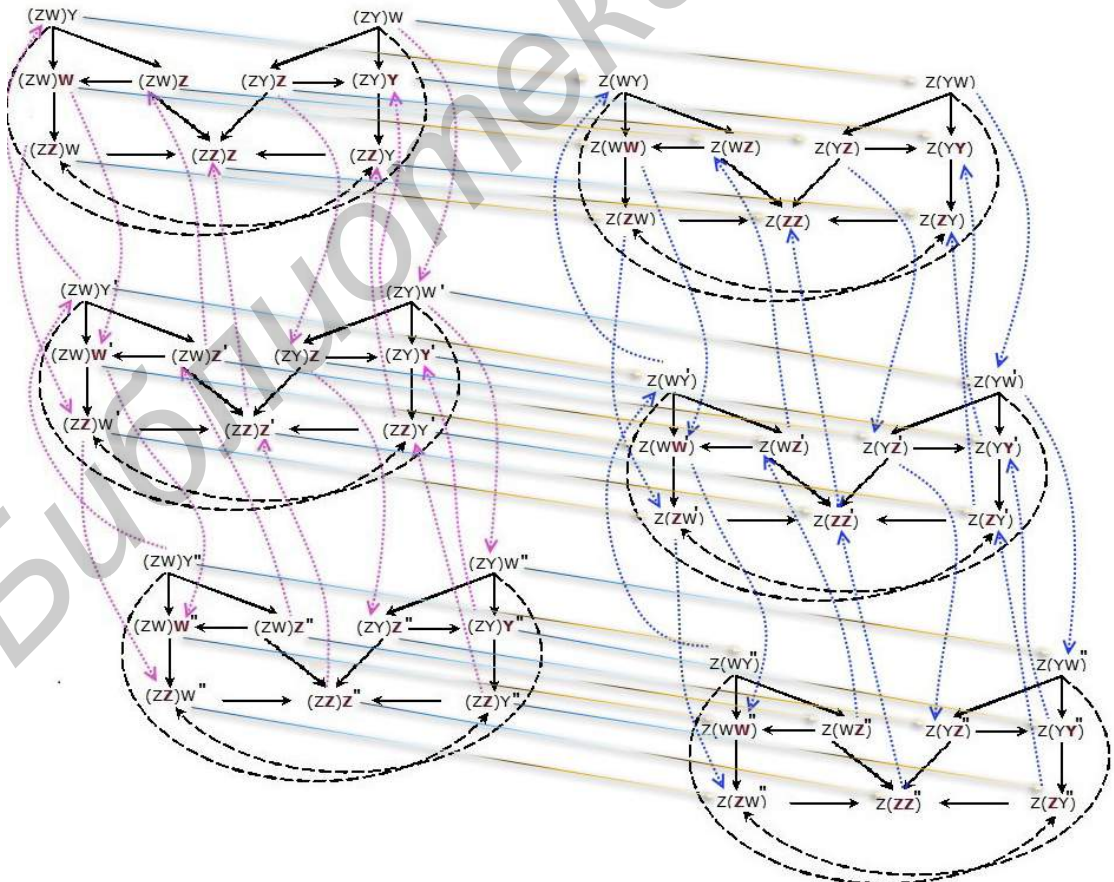


Figure 5. Complete graph of axioms: diffusion, transposition, and complement