Means of formal description of syntax and denotational semantics of various languages in next-generation intelligent computer systems

Artem Goylo
Minsk State Linguistic University
Minsk, Belarus
Email: artemgoylo@gmail.com

Sergei Nikiforov
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: nikiforov.sergei.al@gmail.com

***Abstract*—The article is devoted to linguistic means of formal description of syntax and denotational semantics of different languages in next-generation intelligent computer systems. A formal ontology of different languages is proposed. The treatment of such notions as language, sign, information construction, sign construction, syntax, semantics, meaning, value, etc. is specified. The subject domains of syntax and denotational semantics of natural languages are formalized. As a result, an upper-level ontology is obtained, using which it becomes possible to formalize more specific subject domains of specific languages, both natural and artificial, for their further use in natural-language interfaces of next-generation intelligent computer systems.**

***Index Terms*—language, sign, information construction, natural language processing, natural language understanding, ontology, semantic network, Open Semantic Technology for Intelligent Systems (OSTIS), SC-code (Semantic Computer Code), constituency grammar**

I. Introduction

The incredible diversity of natural languages and the existence of a large number of artificial languages undermines interoperability between different people(s), different computer systems, as well as between people and computer systems.

This leads to a great number of problems connected with:

- a rapid increase in the amount of information to be processed by people for performing various kinds of tasks;
- difficulties and low efficiency of human interaction with computer systems.

These problems can be solved by designing intelligent computer systems, which are already used in automating all kinds of human activities.

Such computer systems help to solve problems that exist in the general domain of human-computer interaction, including but not limited to:

- automating the processing of such multilingual documents that cannot be processed manually in sufficient time;
- developing natural language interfaces, which are highly dependent on the quality of natural language processing;
- developing machine translation systems;

Thus, a very urgent issue is the problem of developing natural language understanding subsystems for next-generation intelligent computer systems.

In order to ensure interoperability between systems and between components of those systems, it is necessary to introduce a kind of shared foundation, upon which systems can be built. An ontology can be utilized as said foundation. Therefore, the implementation of an NLP-component for next-generation intelligent computer systems is dependent on designing a set of ontologies necessary for the component to function. The most vital ontology for natural language interfaces is the one that describes various aspects of natural languages, including their syntax and denotational semantics.

Moreover, to solve the problem of interoperability introduced at the beginning of the article, it is not enough to focus solely on natural languages. What is needed is a kind of upper ontology for languages in general and their units of meaning (which we will call information constructions). We will formalize the necessary concepts as an ontology of the subject domain of languages and information constructions.

II. State of the art

Currently, the research in Artificial Intelligence covers a wide range of problems. However, there is little compatibility in the conceptual systems of different schools, approaches, and paradigms within the domain, which results in incompatible computer systems being developed on the basis of research findings [1].

Given the complexity of modern software, it is vital to ensure that there is interoperability between different software entities, and that it is possible to re-use previous implementations in a way that they are compatible with current software.

One way to solve this problem is to create software engineering ontologies. Such ontologies should satisfy the following requirements [2]:

- formal semantics should be specified to avoid the ambiguity of definitions and the possibility of invalid interpretations, in order to ensure interoperability;

- it should be possible to apply the ontologies to a different or a more general subject domain, which would help to reduce development cost and increase the quality of the end product;
- it should be possible to perform logical inference over the ontology.

An example of an ontology in this field is COPS [3]. The aim of this ontology was to formalize general concepts in the domain of software engineering in order to simplify the development and use of software.

The issue of compatibility between research results is also of concern in linguistics – a science that has many (often incompatible) theories. Linguists utilize different annotation schemas, different approaches to structuring of corpora, and different ways of representing data in them [4], [5].

In order to solve the problem of incompatibility between annotation schemas, there have been proposed different standards for annotation formats. Among the examples of such efforts are Text Encoding Initiative – a digital data representation consortium [6] – and guidelines proposed by EAGLES (Expert Advisory Group on Language Engineering Standards), for example, – their corpus encoding recommendations [7]. However, none of the standards became widely used by linguists [8, p. 4].

Instead of providing recommendations for linguistic data annotation, a more effective way of solving the aforementioned problems has been proposed – that is, creating linguistic ontologies [9], [10]. Apart from the fact that an upper ontology for the domain of linguistics can provide a link between divergent linguistic theories, such an ontology by its nature is conceived as a formal description of concepts used in linguistics, represented in a machine-readable format, which means that it can be used in computer systems capable of understanding annotated linguistic data, performing intelligent search over language corpora, as well as, potentially, reasoning over linguistic research [4].

As a result, an ontology of the domain of linguistics has been created – The General Ontology of Linguistic Description (GOLD) [11]. This ontology describes the most basic categories and relations used in linguistics, and the ontology itself integrated into a top-level ontology called Suggested Upper Merged Ontology (SUMO) [12]. The authors of GOLD state that they created the ontology first and foremost with a view to solve the problem of interoperability of linguistic typology data so that expert systems could process scientific evidence of natural languages – that is, the ontology was not aimed at solving the problems in the domain of natural language processing *per se* [13, p. 4].

A natural language ontology, aimed at being used in natural language processing tasks is Ontologies of Linguistic Annotation (OLiA) [14]. The main idea of this ontology is to ensure compatibility between linguistic data annotations produced by computer systems while analyzing natural language texts on the one hand, and the corresponding linguistic concepts within the ontology on the other hand. As opposed to other linguistically-motivated ontology, OLiA provides not only an inventory of concepts and relations, but also specifies the way to integrate the elements of the ontology with linguistic data annotations used, for example, in corpora [14, p. 4].

When creating an ontology of natural language, it is necessary to pay attention to the status of specifications of linguistic information within such ontology. J. Bateman suggests to differentiate between three types of ontologies according to the degree with which natural language-specific information is integrated into the ontology [15]:

1) ontologies that are an abstract semantico-conceptual representation of world knowledge, which is used directly as a specification of denotational semantics for syntax and lexicon of a natural language (mixed ontologies);
2) ontologies that have a separate specification of denotational semantics of natural languages, which is used as an interface between natural language syntax and the conceptual ontology itself (interface ontologies);
3) ontologies that are an abstract specification of real world knowledge that pays no directed attention to meanings encoded in natural language (conceptual ontologies).

Especially popular within the field of natural language processing are ontologies of the second type [15, p. 8], because such ontologies (in contrast to the ontologies of the third type) make it possible to formalize more information about natural languages. For example, one of the most popular ontologies used in natural language processing, the Generalized Upper Model [16], is a second-type ontology [15]. P. Buitelaar et al. [17] stress that all formal ontologies have to be linked with linguistic information in order to solve such tasks as extracting information from natural language texts, automated population of ontologies, and natural language text generation.

Since ontological approach to NLP allows to specify the semantics of data obtained as a result of processing a text as well as to potentially raise the quality of NLP-analysis, there has been a shift towards creating ontology-driven NLP systems [18], [19]. Natural language ontologies are actively used in NL-text generation from some domain ontology [20], [21].

Ontological approach is also used in systems for making natural language queries to databases, in which a natural language query is translated into a subject domain ontology query language, with the result of the translation itself then being translated into SQL in order to facilitate user interaction with relational databases [22].

Moreover, specifying linguistic information within an ontology helps in automated ontology design based on natural language texts [23].

More specialized ontologies for specific subdomains of linguistics are being created: for example, an ontology of spacial expressions in natural languages [24], an ontology of temporal expressions [25], ontologies of individual languages [26]. When using ontologies in NLP, it is important to "link" the concepts from an ontology with the lexicon of a specific natural language. This led to the creation of extensions for popular linguistic databases, such as WordNet [27], VerbNet [28] and FrameNet [29], to use the databases together with top-level ontologies (for an example, see [30]). There is an active ongoing development of ontologies of natural language lexica,

which resulted in a variety of formal descriptions of lexica being created [31], [32], [33], [34], [35]. Because popular lexical databases were not designed to serve as an ontology and do not possess the necessary degree of formalization (e.g., WordNet), ontologies that could serve as a sort of "superstructure" over such databases are being created. One such ontology that is widely used is Lemon [36].

Many of the ontologies above are designed using the Semantic Web technology [37], which possesses several disadvantages, namely:

1) there is no rigorous and at the same time simple formal basis for representing information (a kind of kernel, or a representation invariant), which would be universal in the sense that all other systems could be designed based on this kernel. The model that is de facto used in this way is RDF [38] [39], but it does not sufficiently satisfy the requirements of universality and formality.
2) no basic relations have been defined that could be reflected in the syntax of the basic language itself. All relations have to be specified explicitly.

Thus, the technology used to design such ontologies does not allow to structure information space in a formal enough way, streamline information search, ensure compatibility of descriptions provided by different authors – that is, it cannot be viewed as a universal language for representing any kinds of data in knowledge bases.

Moreover, Semantic Web is a technology that is external in relation to existing solutions for natural language processing, which is why such systems have to interact with it using APIs or standardized query languages (in particular, SPARQL) [21].

It is important to highlight the fact that, despite very active development of ontology-driven NLP systems, many popular NLP-libraries (e.g., NLTK [40] and spaCy [41]) do not support the use of ontologies, and the majority of natural language text markup tools use specific formats, which makes it necessary to use parsers and converters specific for such tools in order to utilize them in NLP-related tasks [42, p. 3].

In conclusion, the following problems in state-of-the-art approaches to natural language processing:

1) Lack of unification (standardization) in the approaches described above leads to increased costs of their integration, which significantly complicates the development of various systems on their basis [43], [1].
2) Despite the fact that ontologies can help to solve a wide range of NLP tasks, the majority of ontology driven NLP systems focus on addressing very specific problems (for example, systems can focus only on text generation, or ontology population, or natural language querying).
3) A number of specialized linguistic ontologies have been designed which only formalize a subdomain of linguistics (lexicon, in particular), which is in some sense a consequence of the problem described above. At the same time, the existing upper-level linguistic ontologies (e.g., OLiA) do not completely solve the problem of unification because they have to introduce an intermediate level of representation in order to integrate data collected by an NLP system with the corresponding ontology fragments.

The solution to these problems requires a language with enough expressive power to describe knowledge of any kind, it also requires a technology aimed directly at designing interoperable next-generation intelligent computer systems. The OSTIS technology satisfies both of the requirements. Because of this, natural language interfaces of systems designed using the OSTIS technology (*ostis-systems*) will be able to solve a wide range of NLP-related problems – be it natural language synthesis in general, dialoging using natural languages, NL-driven search, information extraction, etc. While currently such tasks are often performed by specialized tools and require additional effort to ensure potential compatibility with individual computer systems, the OSTIS Technology utilizes a single universal knowledge representation language (called *SC-code*), in which all components of the problem solver as well as the ontology of languages and ontologies of specific subject domains are implemented, which will help to solve the problem of interoperability.

Moreover, a natural language ontology designed using such a technology could be used not only in practical NLP-related applications but also to ensure interoperability of data obtained by linguistic research, which would be a valuable contribution to theoretical linguistics.

Finally, an ontology of natural language can be viewed as a subset of an ontology of languages in general (not only natural ones but also formal and artificial) – something that the aforementioned ontologies do not do. This will make it possible to conceptualize natural languages and programming languages within the same information space, and to unify the concepts used in these different domains to more efficiently solve natural language processing tasts in intelligent computer systems.

The aim of this article is to propose basic means of formal description of syntax and denotational semantics of various languages. This will be done by way of presenting a fragment of the ontology of languages and information constructions created using the OSTIS Technology, which can be used to design next-generation intelligent computer systems.

III. Ontology of languages and information constructions

As a solution to the problem of interoperability, we suggest representing knowledge about various languages (including knowledge of their syntax and semantics) in a unified way. This will help to ensure compatibility of such representations and reduce the cost of developing ontology-driven computer systems.

In our approach, we propose using *the OSTIS Technology* [43], which helps to facilitate interoperability of different problem solver models and reduce the number of modifications introduced when adding, for example, a new model of the problem solver.

Systems developed on the basis of the OSTIS Technology are called ostis-systems. The OSTIS Technology is based on a universal way of semantic representation of information

in the memory of intelligent computer systems, called *SC-code*. SC-code texts are unified semantic networks with a basic set-theoretic interpretation. The elements of such semantic networks are called *sc-elements* (*sc-nodes* and *sc-connectors*, which, in turn, depending on their directivity, can be *sc-arcs* or textitsc-edges). *SC-code alphabet* consists of five main elements, on the basis of which SC-code constructions of any complexity are built, including the introduction of more specific types of sc-elements (for example, new concepts). The memory that stores SC-code constructions is called semantic memory or *sc-memory*.

Within the framework of the technology, the structure of the *knowledge base* of any ostis-system is described by a hierarchy of subject domains and their corresponding ontologies. At the same time, an ontology is treated as a specification of the corresponding subject domain.

Fragments (substructures) of the subject domains and ontologies, as well as structures related to the models of the knowledge base and problem solver, will be shown below in the form of SC-code texts (sc-texts).

In the following sections we will describe the proposed ontologies.

A. Subject domain of languages and information constructions

sign
⇒ *subdividing**:
{• *sign, which is an element of a discrete information structure*
• *sign, which is a non-atomic fragment of a discrete information structure*
}

Sign is a fragment of an information construction that conventionally represents (depicts) some describable entity, which is called the denotation of the sign.

Since all signs are discrete information constructions, the set of signs is the domain of all relations defined on the set of discrete information constructions.

the relation defined on the set of signs^
∋ *synonymy of signs**

Signs are synonymous if and only if they denote the same entity. At the same time, synonymous signs may or may not be syntactically equivalent.

sign construction
⊂ *discrete information structure*

Sign construction is a discrete information construction, which generally is a configuration of signs and special fragments of an information construction that provide structuring of the configuration of signs.

*Sign** is a binary oriented relation linking a sign construction to the set of all signs included in it.

*Semantic adjacency of sign constructions** is a binary relation linking semantically adjacent sign constructions. The sign constructions Ti and Tj are adjacent if and only if there are synonymous signs ti and tj, one of which is a part of the construction Ti and the other is a part of the construction Tj.

class of sign constructions^
∋ *semantically elementary sign construction*
∋ *semantically coherent sign construction*

Semantically elementary sign construction is a sign construction describing some (one) relationship between some entities.

Semantically coherent sign construction is a sign construction that can be represented as a concatenation of semantically elementary sign constructions, each of which is semantically adjacent to the preceding and subsequent semantically elementary sign construction.

parameter defined on the set of sign constructions^
∋ *semantic coherence of sign constructions^*
∋ *semantically coherent sign construction*
∋ *semantically incoherent sign construction*

Information construction is a construction (structure) containing some information about some entities. The form of representation ("image", "materialization"), the form of structuring (syntactic structure), as well as *meaning** (denotational semantics) of *information constructions* can be very different.

Discrete information construction is an *information construction* whose meaning is defined by:

- the set of elements (syntactically atomic fragments) of this information construction,
- alphabet of these elements - a family of classes of syntactically equivalent elements of the information construction,
- inclusion of each element of the information construction to the corresponding class of syntactically equivalent elements of the information construction,
- configuration of incidence relations between the elements of the information construction.

A consequence of this is that the form of representation of the elements of a discrete information construction does not need to be specified for the analysis of its meaning. It is important to ensure the following:

- the presence of a simple procedure for selecting (segmenting) elements of a discrete information construction,
- specification of a simple procedure for establishing the syntactic equivalence of different elements of a discrete information construction,
- the presence of a simple procedure for determining whether each element of a discrete information construction belongs to the corresponding class of syntactically equivalent elements (i.e. to the corresponding element of the alphabet).

Element of a discrete information construction is a syntactically atomic fragment (symbol) included in a discrete information construction. Since discrete information constructions can have common elements (atomic fragments) and even some of them can be parts of other information constructions, an element of a discrete information construction can be a part of several information constructions at once.

Next we will consider the relations defined on the set of discrete information constructions.

the relation defined on the set of discrete information constructions^

- ∋ *discrete information construction element**
- ∋ *syntactic equivalence of elements of discrete information constructions**
- ∋ *incidence of elements of discrete information constructions**
- ∋ *non-elementary fragment of a discrete information construction**
- ∋ *alphabet of a discrete information construction**
- ∋ *primary syntactic structure of a discrete information construction**
- ∋ *syntactic equivalence of discrete information constructions**
- ∋ *copy of a discrete information construction**
- ∋ *semantic equivalence of discrete information constructions**
- ∋ *semantic extension of a discrete information construction**
- ∋ *syntax of an information construction**
- ∋ *meaning**
- ∋ *operational semantics of an information construction**

*Element of a discrete information construction** is a binary oriented relation, each pair of which links (1) a sign of some discrete information construction and (2) a sign of one of the elements of this discrete information construction*.

*Syntactic equivalence of elements of discrete information constructions** is the relation linking syntactically equivalent elements (atomic fragments) of the same or different discrete information constructions, i.e. elements belonging to the same class of syntactically equivalent *elements of discrete information constructions**.

*Incidence of elements of discrete information constructions** for *linear information constructions* is a sequence of elements (symbols) included in these constructions. For discrete information constructions whose configuration has a non-linear nature, the incident relation of their elements can be broken down into several partial incidence relations, each of which is a subset of the combined incidence relation. For example, for two-dimensional discrete information constructions it is (1) the incidence of elements of information constructions "horizontally" and (2) the incidence of elements of information constructions "vertically".

*Elementary fragment of a discrete information construction** is a binary oriented relation linking a given discrete information construction with a discrete information construction which is a substructure for it, which includes (1) a subset of elements of the given information construction and, respectively, (2) a subset of incidence pairs of elements of the given information construction.

*Alphabet of a discrete information construction** is a binary relation linking a discrete information construction with a family of pairwise non-overlapping classes of syntactically equivalent elements of a given discrete information construction*.

*Primary syntactic structure of a discrete information construction** is a binary oriented relation linking a discrete information construction to a *graphic structure* that fully describes its configuration and which includes: (1) signs of all those classes of syntactically equivalent elements to which the elements of the described discrete information construction belong, (2) signs of all elements (atomic fragments) of the information construction described, (3) pairs describing the incidence of the elements of the information construction described, (4) pairs describing belonging of elements of the information construction described to the corresponding classes of syntactically equivalent elements of this information construction.

*Syntactic equivalence of discrete information constructions**: Discrete information constructions $\boldsymbol{Ti}$ and $\boldsymbol{Tj}$ are syntactically equivalent if and only if there exists an isomorphism between the construction $\boldsymbol{Ti}$ and the construction $\boldsymbol{Tj}$, in which each element of the construction $\boldsymbol{Ti}$ corresponds to a syntactically equivalent element of the construction $\boldsymbol{Tj}$, i.e., an element from the same class of syntactically equivalent elements of the construction $\boldsymbol{Tj}$, i.e. an element belonging to the same class of syntactically equivalent elements of discrete information constructions. And vice versa.

*Copy of a discrete information construction** is a binary oriented relation that links a discrete information construction with a discrete information construction which is not only syntactically equivalent to it, but also contains information about the form of representation of elements of this copied information construction*

copy of a discrete information structure*

- ⊂ *syntactic equivalence of discrete information constructions**

*Semantic equivalence of discrete information constructions**: Information construction $\boldsymbol{Ti}$ and information construction $\boldsymbol{Tj}$ are semantically equivalent if and only if each entity (including each relationship between entities) described in information construction $\boldsymbol{Ti}$ is also described in information construction $\boldsymbol{Tj}$. And vice versa.

*Semantic extension of a discrete information construction**: the information construction $\boldsymbol{Tj}$ is a semantic extension of the information construction $\boldsymbol{Ti}$ if and only if each entity described in $\boldsymbol{Ti}$ is also described in $\boldsymbol{Tj}$, but the reverse is not true.

*Syntax of an information construction** is a description of what parts a given information construction consists of and

how these parts (fragments) are related to each other.

*Meaning** (*denotational semantics of an information construction**) is a binary oriented relation, each pair of which relates some information construction to its explicit (formal) representation of what entities this information construction describes and how these entities are related to each other.

*Operational semantics of an information construction** is a binary oriented relation, each pair of which connects a sign of some information construction with a set of rules of its transformation - a description of what rules can be used to perform actions on transformation (processing, transformation) of a given information construction, leaving it within the class of syntactically and semantically correct information constructions.

operational semantics of an information construction*
⇒ *second domain**:
operational semantics of an information construction

Now let us consider mappings defined on the set of discrete information constructions.

mapping defined on the set of discrete information constructions
∋ *mapping between the syntactic structure of the information construction and the meaning of that construction**
⊂ *mapping**

Mapping defined on the set of discrete information constructions is the set of ordered pairs, the first component of which is an ordered pair consisting of (1) a sign of the syntactic structure of some information construction and (2) a sign of the semantic structure of that construction, and the second component of which is a set of ordered pairs linking fragments of the syntactic structure of a given information construction (which describe either the structure of fragments of a given construction or links between fragments of this construction) with those fragments of the semantic structure of a given information construction that are semantically equivalent to either syntactically represented fragments of a given information construction or syntactically represented links between such fragments.

Discrete information constructions have the following parameters.

parameter defined on the set of discrete information constructions^
∋ *dimensionality of discrete information constructions^*
:= [typology of discrete information constructions based on their dimensionality]
∋ *linear information structure*
∋ *two-dimensional information structure*
∋ *three-dimensional information design*
∋ *four-dimensional information structure*
∋ *graph information structure*

Linear information construction is a discrete information construction, each element of which can have no more than two incident elements (one on the left and one on the right).

Two-dimensional information construction is a discrete information construction, each element of which can have no more than four incident elements (left-right, top-bottom).

Three-dimensional information construction is a discrete information construction, each element of which can have no more than six incident elements (left-right, top-bottom, back-to-front).

Four-dimensional information construction is a discrete information construction, each element of which can have no more than eight incident elements (for example, left-right, top-bottom, front-back, earlier-later).

Graph information construction is a discrete information construction whose set of elements is divided into two subsets - sheaves and nodes. In this case nodes can have an unlimited number of incident sheaves. In some graph information constructions sheaves can have an unlimited number of other sheaves incident to them.

parameter defined on the set of discrete information constructions^
∋ *typology of discrete information constructions, defined by their carrier^*
∋ *non-computer form of representation of discrete information constructions*
⊃ *audio message*
⊃ *an information construction presented in sign language*
⊃ *information construction presented in written form*
∋ *file*

Representation of information constructions in the form of *files* is directed at representation of discrete (!) information constructions. Therefore "file" representation of non-discrete information constructions (for example, various kinds of signals) implies "discretization" of such constructions, i.e. their transformation into discrete ones. This is how audio signals (in particular, speech messages), images, video signals, etc. are converted.

parameter defined on the set of discrete information constructions^
∋ *level of unification of representation of syntactically equivalent elements of discrete information constructions^*
∋ *discrete information construction with a low level of unification of the representation of elements*
⊃ *audio message*
⊃ *an information construction presented in sign language*
⊃ *manuscript or a copy thereof*

∋ *discrete information construction with a high level of unification of element representation*
⊃ *printed text*
⊃ *file*

*Level of unification of representation of syntactically equivalent elements of discrete information constructions^* is the level of "articulateness" of discrete information constructions.

The higher the level of unification of the representation of the elements of discrete information constructions, the easier it is to implement:

- the procedure for segmenting the elements of a discrete information construction,
- the procedure for establishing syntactic equivalence of these elements,
- the procedure for their recognition, i.e. the procedure of establishing their belonging to the corresponding classes of syntactically equivalent elements.

Having clarified the notions of sign, sign construction, information construction, discrete information construction and having considered the corresponding relations, we can proceed to the formalization of the concept of "language".

Language is a class of sign constructions, for which there are (1) general rules of their construction and (2) general rules of their correlation with those entities and configurations of entities, which are described (reflected) by the specified sign constructions.

*A relation defined on a set of languages^* is a relation whose domain includes a set of all possible languages.

*The text of a given language** is a binary relation linking the language and a syntactically correct (well-formed) sign construction of that language.

*Syntactically correct sign construction for a language** is a binary relation linking the language and a sign construction that does not contain syntactic errors for that language.

a relation defined on the set of languages^

:= [relation, the domain of which includes the set of all possible languages]
∋ *text of a language**
= (*syntactically correct sign construction for a given language** ∩ *syntactically complete sign construction for a given language*)
∋ *syntactically correct sign construction for a language**
∋ *syntactically complete sign construction for a language**
∋ *syntactically incorrect sign construction for a language**
= (*syntactically incorrect sign construction for a language** ∪ *syntactically incoherent sign construction for a language**)
⊃ *syntactically incorrect sign construction for a language**
⊃ *a syntactically incoherent sign construction for a language**
∋ *knowledge presented in a language**
:= [semantically correct text of a language*]
= (*semantically correct text of a language** ∩ *semantically complete text of a language**)
∋ *semantically correct text of a language**
:= [text of a given language that does not contain semantic errors that contradict recognized patterns and facts*]
∋ *semantically coherent text of a language**
:= [the text of a given language containing sufficient information to establish its truth*]
∋ *semantically incorrect text for a language**
= (*semantically incoherent text for a language** ∪ *semantically incoherent text for a language**)
⊃ *semantically incorrect text for a language**
⊃ *semantically incoherent text for a language**
∋ *alphabet**
:= [the alphabet of a given information construction or a given language*]
:= [family of classes of syntactically equivalent elements (elementary fragments) of a given information construction or information constructions of a given language*]
∋ *language**
:= [is a theory of well-formed information constructions belonging to a given language*]
:= [the syntactic rules of a given language*]
:= [Binary oriented relation, each pair of which connects a sign of some language with the description of syntactically distinguished classes of fragments of constructions of the given language, with the description of relations defined on these classes and with the conjunction of quantized statements, which are syntactic rules of the given language, i.e. rules, which all syntactically correct (well-formed) constructions (texts) of the given language should satisfy*.]
⇒ *second domain**: *language syntax*
∋ *a description of the syntactic concepts of the language**
:= [description of syntactically distinguishable classes of fragments of constructions of a given language*]
⇒ *second domain**: *a description of the syntactic concepts of the language*
⇐ *generalized inclusion**: *language syntax*
∋ *syntactic rules of the language**
:= [the syntactic rules of a given language*]
⇒ *second domain**: *syntactic rules of the language*
∋ *denotational semantics of a language**
:= [is a theory of morphisms, linking well-formed information constructions of a given language with described configurations of described

entities*]

∋ *denotational semantics of language**
:= [semantic rules of a given language*]
:= [be the semantic rules of a given language*]
:= [A binary oriented relation, each pair of which connects a sign of some language with a description of basic semantic notions of a given language and a conjunction of quantifier statements, which are semantic rules of a given language, i.e. the rules to which semantically correct meaningful information constructions corresponding (semantically equivalent) to syntactically correct constructions (texts) of a given language should satisfy*]
⇒ *note**:
[When formulating the semantic rules of a given language, concepts introduced in basic ontologies (top-level ontologies) are used.]
⇒ *second domain**:
denotational semantics of language

∋ *description of the semantic concepts of a language**
⇒ *second domain**:
description of the semantic concepts of a language

∋ *semantic rules of a language**
⇒ *second domain**:
semantic rules of a language

∋ *semantic equivalence of languages**
:= [be semantically equivalent languages*]
⇒ *определение**:
[Language ***Li*** and language ***Lj*** will be considered *semantically equivalent languages** if and only if for every text belonging to language ***Li***, there is a *semantically equivalent text** belonging to language ***Lj*** and vice versa.]

∋ *semantic extension of a language**
⇔ *inverse relation**:
*semantic reduction of a language**
⇒ *definition**:
[Language ***Lj*** will be considered a *semantic extension** of language ***Li*** if and only if for every text belonging to language ***Li*** there is a *semantically equivalent text** belonging to language ***Lj***, but the reverse is not true.]

∋ *syntactic extension of a language**
:= [be a semantically equivalent superset of a given language*]
⇒ *definition**:
[The language ***Lj*** will be considered a *syntactic extension** of ***Li*** if and only if
□ ***Lj*** ⊃ ***Li*** (that is, all texts of ***Li*** are also texts of ***Lj***, but the reverse is not true);
□ Language ***Lj*** and language ***Li*** are *semantically equivalent languages**.
]

∋ *syntactic core of a language**
:= [be the syntactic core of a given language*]
:= [be a semantically equivalent subset of a given language with minimal syntactic complexity*]

∋ *the direction of the syntactic extension of the kernel of a given language**
:= [be the rule of transformation of information constructions belonging to a given language, which describes one of the directions of transition from the set of constructions of the core of this language to the set of all information constructions belonging to it*.]

∋ *operational semantics of a language**
:= [A binary oriented relation, each pair of which associates a sign of some language with a set of rules for transforming texts of that language*.]
⇒ *second domain**:
operational semantics of the language

∋ *internal language**
:= [be an internal language for a given information processing-based system or a given set of such systems*.]
:= [be the language of the internal representation of information in the memory of a given information processing-based system or a given class of such systems*.]

∋ *external language**
:= [be an external language for a given information processing-based system or a given set of such systems*.]
:= [be a language used to exchange information of a given information processing-based system, or a given set of such systems, with other information processing-based systems (including those of their own kind)*]

∋ *language used**
= (*internal language** ∪ *external language**)
:= [the language used by a given system based on information processing or a given set of such systems*.]
:= [the language spoken by the system in question, which is based on information processing]

∋ *languages used**

*Parameter defined on a set of languages^* is a family of language equivalence classes, interpreted in the context of some property (characteristic) inherent to the languages.

parameter defined in the set of languages^

∋ *semantic power of language^*
:= [a class of languages that are semantically equivalent to each other]
∋ *universal language*
:= [a class of all possible universal languages]

⇒ *note**:
[Obviously, all universal languages (if they really are, and not just claim to be) are semantically equivalent to each other, i.e. have the same semantic power.]

∋ *the level of syntactic complexity of the representation of the signs in the texts of the language*^
- ∋ *a language in whose texts all signs are represented syntactically by elementary fragments*
- ∋ *a language in whose texts signs are generally represented by syntactically non-elementary fragments*

∋ *the use of delimiters and terminators in language texts*^
- ∋ *language that does not use delimiters and terminators in its texts*
- ∋ *language that uses delimiters and terminators in its texts*

∋ *the level of complexity of the procedure for establishing synonymy of signs in language texts*^
- ∋ *language, within each text of which there are no synonymous signs*
  - ⇒ *explanation**:
    [In texts of such language, the sign of each entity being described is present only once.]
- ∋ *язык, в рамках которого синонимичные знаки представлены синтаксически эквивалентными фрагментами текстов*
- ∋ *inflected language*
  - ≔ [language, within which synonymous signs can be represented by syntactically non-equivalent fragments, but by fragments which are modifications of some "kernel" of these fragments (in the declension and conjugation of these signs).]
- ∋ *a language in which synonymous signs can generally be represented by syntactically non-equivalent text fragments whose structure is unpredictable*

∋ *presence of homonymy in the texts of a language*^
- ∋ *a language whose texts contain homonymic signs*
  - ≔ [language whose texts contain syntactically equivalent, non-synonymous signs]
- ∋ *language, in the texts of which there is no homonymy of signs*

semantically distinguishable class of discrete information constructions

∋ *syntactic structure of the information construction*
- ⇐ *second domain**:
  *syntax of the information construction**
- ⊃ *primary syntactic structure of the information structure*
- ⊃ *secondary syntactic structure of the information structure*

∋ *language syntax*
∋ *a description of the syntactic concepts of a language*
∋ *syntactic rules of a language*
∋ *denotational semantics of a language*
∋ *description of the semantic concepts of a language*
∋ *semantic rules of a language*
∋ *the operational semantics of the language*
∋ *meaning*
- ⇐ *second domain**:
  *meaning**

Meaning - an explicit (formal) representation of the described entities and the relationships between them. The explicit representation of the described entities and the relationships between them requires a significant simplification of the syntactic structure of information constructions.

Ostis-system language is the language for representing information constructions in ostis-systems.

ostis-system language

⊂ *formal language*
⊂ *universal language*
⇐ *languages used**:
ostis-system
∋ *SC-code*
- ≔ [Semantic Computer Code]
- ⇐ *internal language**:
  ostis-system
- ∈ *universal language*

To formally describe various kinds of languages, including the languages we are considering (SCg-code, SCs-code, SCn-code) a number of metalanguage notions are used.

Here are some of them: *identifier*, *class of syntactically equivalent identifiers*, *name*, *simple name*, *expression*, *external identifier**, *alphabet**, *delimiters**, *terminators**, *sentences**

The syntax of *knowledge representation languages in ostis-systems* can be formally described in various ways. For example, it is possible to use Bacus-Naurus meta-language to describe the syntax of SCs-code or its extension to describe the syntax of SCn-code. However, it is much more logical and expedient to describe the syntax of all forms of external sc-text mapping using SC-code itself. This approach will allow ostis-systems to independently understand, analyze and generate texts of the specified languages on the basis of principles common to any form of external representation of information, including non-linear ones.

*Alphabet** is a binary relation linking a set of texts to a family of maximal sets of syntactically identical elementary (atomic) text fragments belonging to a given set of texts.

Identifier is a structured sign of the corresponding (denoted) entity, which is most often a string – the name of the

corresponding entity. In formal texts (including texts of SC-code, SCg-code, SCs-code, SCn-code) the main identifiers used should not be homonymic, i.e. they should unambiguously correspond to the entities being identified. Consequently, each pair of identifiers having the same structure must denote the same entity.

identifier
⊃ *sc.s-identifier*

name
⊂ *identifier*
:= [string identifier]
:= [identifier, which is a string (chain) of characters]
⇒ *action decomposition**:
{• *simple name*
:= [atomic name]
:= [a name that does not include other names]
• *expression*
:= [non-atomic name]
}
⊃ *sc.s-identifier*

*An external identifier** is a binary oriented relation, each sheaf (sc-arc) of which links some element to a file, the content of which is an external identifier (most often, a name) corresponding to the specified element. The notion of external identifier is a relative notion and important for ostis-systems because internal representation of information (in the form of SC-code texts) operates not with identifiers of described entities, but with signs whose structure does not matter.

B. Subject domain of files, external information constructions and external languages of ostis-systems

There are several languages for the external representation of sc-texts [43]:

- SCn-code;
- SCs-code;
- SCg-code.

SCg-code is an *external language** of ostis-systems, the texts of which are graph structures of a general form with precisely defined *denotational semantics**.

SCg-code
∈ *ostis-system language*
:= [Semantic Code graphical]
⇐ *external language**:
ostis-system
∈ *universal language*

SCs-code is an *external language** of ostis-systems, the texts of which are strings (chains) of characters.

SCs-code
∈ *ostis-system language*
:= [Semantic Code string]
⇐ *external language**:
ostis-system
∈ *universal language*

SCn-code is an *external language** of ostis-systems whose texts are two-dimensional matrices of symbols, which are the result of formatting of two-dimensional structuring of SCs-code texts.

SCn-code
∈ *ostis-system language*
:= [Semantic Code natural]
⇐ *external language**:
ostis-system
∈ *universal language*

To implement *ostis-systems knowledge bases*, all kinds of languages are used: *universal languages* as well as *specialized languages*, both *formal languages* and *natural languages*, both *internal languages* that provide the representation of information in the *ostis-systems* memory, as well as *external languages* providing the representation of input or output information. *Natural languages* are used exclusively to represent *files* stored in the *ostis-system* memory and formally specified within the *knowledge base* of that *ostis-system*.

In order to operate *intelligent computer systems* built on the basis of *SC-code*, besides the method of abstract internal representation of knowledge bases (*SC-code*), several methods of external representation of abstract *sc-texts* will be required, which are user-friendly and used in the design of source texts of *knowledge bases* of said intelligent computer systems and source texts of fragments of those *knowledge bases*, as well as used to display various fragments of *knowledge bases* to users according to user requests. The aforementioned external ostis-system languages (*SCg-code*, *SCs-code* and *SCn-code*) are proposed as such methods of external display of *sc-texts*.

All the main external formal languages used by ostis-systems (*SCg-code*, *SCs-code*, *SCn-code*) are different variants of the external representation of texts written in the internal language of ostis-systems - SC-code. These languages are universal and, therefore, *semantically equivalent languages**.

Moreover, each ostis-system can acquire the ability to use any external language (both universal and specialized, both natural and artificial) if the syntax and denotational semantics of that language are described in the memory of the ostis-system in its internal language (SC-code).

C. Formalization of natural languages

As explained above, in order to utilize insights from linguistics in the design of intelligent computer systems it is necessary to represent linguistic knowledge in a formal way. In this section, we propose a formalization of the basic concepts in linguistics made in a formal knowledge representation language – SC-code.

language
⇒ *subdividing**:
{• *natural language*
⇒ *explanation**:
[A natural language is a language that was not created purposefully.]
• *artificial language*
⇒ *explanation**:
[An artificial language is a language specially designed to achieve certain goals.]
∋ *Esperanto*
∋ *Python*
⊃ *constructed language*
⇒ *explanation**:
[A constructed language is an artificial language designed for human communication.]
∋ *Esperanto*
}
⊃ *international language*
⇒ *explanation**:
[An international language is a natural or artificial language used by people from different countries to communicate.]
∋ *English*
∋ *Russian*

planned language
⇐ *intersection**:
{• *constructed language*
• *international language*
}

language of communication
⇐ *union**:
{• *natural language*
• *constructed language*
}
∋ *English*
∋ *Russian*
∋ *Esperanto*
⇐ *union**:
{• *isolating language*
⇒ *explanation**:
[An isolating language is a language that is characterized by the complete absence of inflection and the presence of grammatical significance of the order of words consisting only of the root.]
∋ *Chinese*
• *agglutinative language*
⇒ *explanation**:
[An agglutinative language is characterized by a developed system of affixes added to the invariable stem of the word, which are used to express the categories of number, case, gender, etc.]
∋ *Japanese*
• *inflected language*
⇒ *explanation**:
[An inflected language is characterized by a developed use of endings to express the categories of gender, number, case, a complex system of verb declension, alternation of vowels in the root, as well as a strict distinction between parts of speech.]
∋ *Russian*
• *polysynthetic languages*
⇒ *explanation**:
[A polysynthetic language is a language that relies on affixes to encode all (or almost all) grammatical meanings.]
}

1) Formalization of natural language lexicon: A *lexeme* is a minimal unit of a language that has a semantic interpretation and denotes a concept that reflects the view of the world of a certain linguistic community [44].

A *grammatical category* is a system of oppositions between grammatical forms with homogeneous meanings. As part of our formalization, it is proposed to represent grammatical categories as classes of role relations, each of which corresponds to a certain grammatical meaning. We will list here only a fragment of the ontology that describes the most basic grammatical categories and relations.

grammatical category
∋ *person*
⇐ *set of subsets**:
role relation
∋ *first person′*
∋ *second person′*
∋ *third person′*
∋ *number*
⇐ *set of subsets**:
role relation
∋ *singular number′*
∋ *plural number′*
∋ *dual number′*
∋ *trial number′*
∋ *paucal number′*
∋ *gender*
⇐ *set of subsets**:
role relation
∋ *masculine gender′*
∋ *neuter gender′*
∋ *feminine gender′*
∋ *case*
⇐ *set of subsets**:
role relation

∋ *nominative case'*
∋ *genitive case'*
∋ *dative case'*
∋ *accusative case'*
∋ *instrumental case'*
∋ *prepositional case'*
∋ *vocative case'*
∋ *absolutive case'*
∋ *ergative case'*

∋ *tense*
⇐ *set of subsets**:
role relation
∋ *present tense'*
∋ *past tense'*
∋ *future tense'*

∋ *mood*
⇐ *set of subsets**:
role relation
∋ *indicative mood'*
∋ *imperative mood'*
∋ *irrealis mood'*

∋ *voice*
⇐ *set of subsets**:
role relation
∋ *active voice'*
∋ *passive voice'*
∋ *middle voice'*
∋ *reflexive voice'*
∋ *reciprocal voice'*

∋ *aspect*
⇐ *set of subsets**:
role relation
∋ *perfective aspect'*
∋ *imperfective aspect'*
∋ *common aspect'*
∋ *progressive aspect'*
∋ *perfect aspect'*

∋ *degree of comparison*
⇐ *set of subsets**:
role relation
∋ *positive degree of comparison'*
∋ *comparative degree of comparison'*
∋ *superlative degree of comparison'*

An example of the the way some of the above relations are formalized in the sc.g-language is shown in Figure 1.

Part of speech is a grammatical category that represents a class of syntactically equivalent natural language signs.

part of speech
⇐ *set of subsets**:
lexeme
∋ *noun*
∋ *adjective*
∋ *verb*
∋ *adverb*

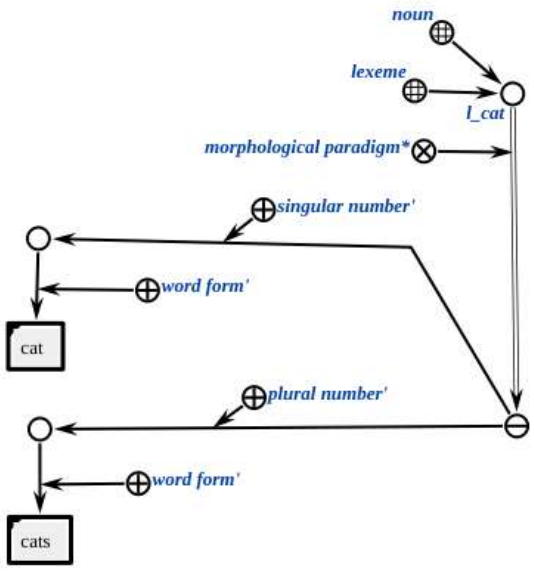

Figure 1. An example of a lexeme specification in the knowledge base

∋ *preposition*
∋ *complementizer*
∋ *auxiliary*
∋ *determiner*

*Morphological paradigm** is a binary oriented relation connecting a lexeme and a set of its word forms.

A *word form* – a subset of a lexeme that contains all tokens of a lexeme that share certain grammatical meanings. Within our ontology, a word form is understood somewhat differently than is customary in linguistics, since all tokens of a lexeme in the OSTIS technology are considered to be instances of files.

2) Formalization of natural language syntax: When formalizing the core of natural language syntax, we drew from the framework of generative linguistics [45], [46], [47], [48].

The *distribution of a sign* is a subset of the syntactic environments that the sign appears in.

A *constituent* is an element of the set $\boldsymbol{C}$ of subsets of the tuple of tokens $\boldsymbol{S}$, which contains as elements both $\boldsymbol{S}$ and all tokens in $\boldsymbol{S}$ in such a way that any two subsets included in $\boldsymbol{C}$ either do not intersect, or one of them is included into another.

An *immediate constituent*: let $\boldsymbol{S}$ be a set of constituents such that it includes constituents $\boldsymbol{A}$ and $\boldsymbol{B}$. $\boldsymbol{B}$ is an immediate constituent of $\boldsymbol{A}$ if and only if $\boldsymbol{B}$ is a constituent of $\boldsymbol{A}$ and there is no constituent $\boldsymbol{C}$ such that $\boldsymbol{C}$ is a constituent of $\boldsymbol{A}$ and $\boldsymbol{B}$ is a constituent of $\boldsymbol{C}$.

An *ultimate constituent* is an element $\boldsymbol{U}$ of a tuple of tokens $\boldsymbol{T}$ such that $\boldsymbol{U}$ is an immediate constituent in a set of constituents $\boldsymbol{C}$ and $\boldsymbol{U}$ itself has no immediate constituents.

A *phrase* is a class of constituents, which includes constituents with heads of the same part of speech. Phrases are either singletons (minimally including a single head) or an ordered pair consisting of a head and another phrase.

A *head* is a constituent whose distribution is equivalent to the distribution of the entire phrase.

constituent
⇒ *subdividing**:
{• *phrase*
• *head*
}

phrase
⇒ *subdividing**:
{• *noun phrase*
⇒ *explanation**:
[A *noun phrase* is a phrase headed by a noun.]
• *verb phrase*
⇒ *explanation**:
[A *verb phrase* is a phrase headed by a verb.]
• *adjective phrase*
⇒ *explanation**:
[An *adjective phrase* is a phrase headed by an adjective.]
• *adverb phrase*
⇒ *explanation**:
[An *adverb phrase* is a phrase headed by an adverb.]
• *prepositional phrase*
⇒ *explanation**:
[A *prepositional phrase* is a phrase headed by a preposition.]
• *complementizer phrase*
⇒ *explanation**:
[A *complementizer phrase* is a phrase headed by a complementizer.]
• *tense phrase*
⇒ *explanation**:
[A *tense phrase* is a phrase headed by an auxiliary or a modal verb.]
• *determiner phrase*
⇒ *explanation**:
[A *determiner phrase* is a phrase headed by a determiner.]
}
⇒ *subdividing**:
{• *maximal projection of a head*
• *intermediate projection of a head*
}

More specific classes can be inferred as a result of intersection between the sets of classes listed above, e.g. *maximal projection of a determiner phrase head*

maximal projection of a determiner phrase head
⇐ *intersection**:
{• *determiner phrase*
• *maximal projection of a head*
}

An example of syntactic structure of a sentence, parsed using the concepts above is shown in Figure 2.

Phrase structure is not arbitrary – elements within a phrase can only border certain sets of elements. The possible structures of phrases are given below. The sign "->" should be read as "consists of". Optional elements are shown in parentheses.

Determiner phrase:

- Maximal projection of a determiner phrase head -> (Maximal projection of a determiner phrase head) Intermediate projection of a determiner phrase head
- Intermediate projection of a determiner phrase head -> Determiner phrase head (Maximal projection of a noun phrase head)

Noun phrase:

- Maximal projection of a noun phrase head -> (Maximal projection of a determiner phrase head) Intermediate projection of a noun phrase head
- Intermediate projection of a noun phrase head -> (Maximal projection of an adjective phrase head) Intermediate projection of a noun phrase head OR Intermediate projection of a noun phrase head (Maximal projection of a prepositional phrase head)
- Intermediate projection of a noun phrase head -> Noun phrase head (Maximal projection of a prepositional phrase head)

Verb phrase:

- Maximal projection of a verb phrase head -> Intermediate projection of a verb phrase head
- Intermediate projection of a verb phrase head -> Intermediate projection of a verb phrase head (Maximal projection of a prepositional phrase head) OR Intermediate projection of a verb phrase head (Maximal projection of an adverb phrase head)
- Intermediate projection of a verb phrase head -> Verb phrase head (Maximal projection of a noun phrase head)

Adverb phrase:

- Maximal projection of an adverb phrase head -> Intermediate projection of an adverb phrase head
- Intermediate projection of an adverb phrase head -> (Maximal projection of an adverb phrase head) Intermediate projection of an adverb phrase head
- Intermediate projection of an adverb phrase head -> Adverb phrase head (Maximal projection of a prepositional phrase head)

Adjective phrase:

- Maximal projection of an adjective phrase head -> Intermediate projection of an adjective phrase head
- Intermediate projection of an adjective phrase head -> (Maximal projection of an adverb phrase head) Intermediate projection of an adjective phrase head
- Intermediate projection of an adjective phrase head -> Adjective phrase head (Maximal projection of a prepositional phrase head)

Prepositional phrase:

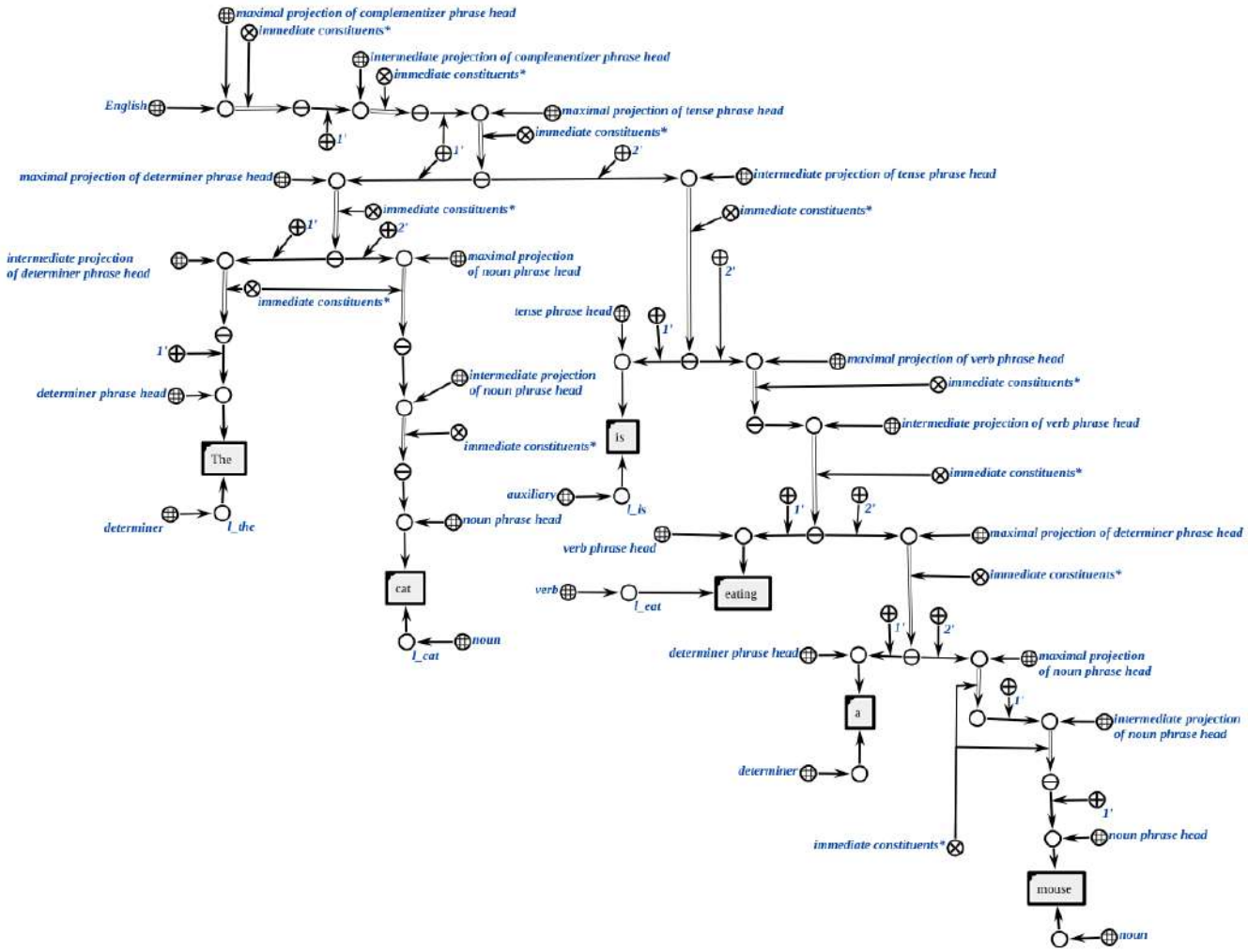

Figure 2. Syntactic structure example.

- Maximal projection of a prepositional phrase head -> Intermediate projection of a prepositional phrase head
- Intermediate projection of a prepositional phrase head -> Intermediate projection of a prepositional phrase head (Maximal projection of a prepositional phrase head) OR (Maximal projection of an adverb phrase head) Intermediate projection of a prepositional phrase head
- Intermediate projection of a prepositional phrase head -> Prepositional phrase head (Maximal projection of a noun phrase head)

Tense phrase:

- Maximal projection of a tense phrase head -> (Maximal projection of a determiner phrase head) Intermediate projection of a tense phrase head
- Intermediate projection of a tense phrase head -> Tense phrase head (Maximal projection of a verb phrase head)

Complementizer phrase:

- Maximal projection of a complementizer phrase head -> (Maximal projection of some phrase head) Intermediate projection of a complementizer phrase head
- Intermediate projection of a complementizer phrase head -> Complementizer phase head Maximal projection of a tense phrase head

These rules can be formalized as follows (Figure 3).

A *complement* is a phrase that is a sister of a head. Sisters are constituents that are in an immediate constituent relation with the same constituent.

An *adjunct* is a phrase that is a daughter (immediate constituent) of an intermediate projection and a sister of the intermediate projection of the head of the same phrase.

A *specifier* is a phrase that is a daughter of a maximal projection and a sister of an intermediate projection.

The phrase structure rules can be generalized and reduced to three more abstract ones.

Specifier rule: XP -> (YP) X'

Adjunct rule: X' -> X' (ZP) | X' -> (ZP) X'

Complement rule: X' -> X (WP)

These rules are formalized in the same manner as in Figure 3.

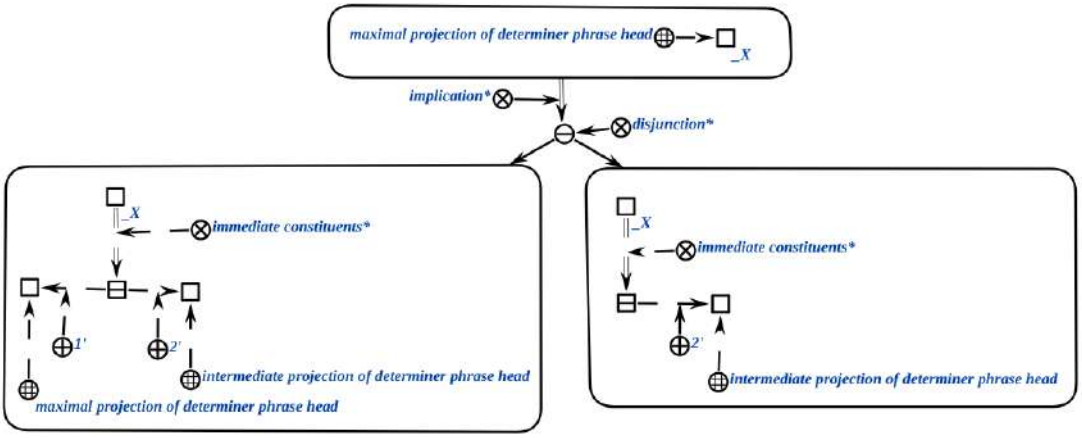

Figure 3. Phrase structure rule example.

3) Formalization of natural language denotational semantics: The denotational semantics of a language specifies the interpretation of the syntactic elements of that language and is a set of formulas that describe how the sign constructions of the language are associated with the entities they denote and the configurations of relations between these entities.

The denotational semantics of natural languages must be compositional, i.e. the interpretation of an entire construction must be derived from the interpretation of its individual parts. Thus, it is necessary to provide a formal description of the interpretation of NL syntax elements presented in the previous section, as well as a description of the rules for combining the interpretation of individual elements to obtain the meaning of the entire construction.

In this article, we propose a variant of the formalization of the denotational semantics of natural languages within the framework of the OSTIS technology, for which the ideas of model-theoretic formal semantics [49], [50], [51] were used.

Below are examples of rules that implement the denotational semantics of English. The rules must be applied one after the other and allow us to obtain the meaning of a natural language sentence from its syntactic structure, "climbing" the syntactic tree from heads to maximal projections.

Figure 4 shows the rule that is used to interpret heads of noun phrases and adjective phrases. We assume that the meaning of such heads is a class of elements (written here in the sc.g language), e.g. the adjective "black" is associated with a corresponding set of black objects, while the noun "cat" is associated with a set of cats. The construction to the right specifies that an entity X (represented by a variable sc.g-node) is linked via a *meaning** relation to some struct that includes a class, i.e. this entity X is a class.

Figure 5 shows the rule of interpreting a noun phrase, the maximum projection of which also includes an adjective phrase. As mentioned above, to apply this rule, you must first apply the rule shown in figure 4. The meaning of such constructions is a class that is an intersection of classes obtained as a result of interpreting the heads of the adjective and noun phrases

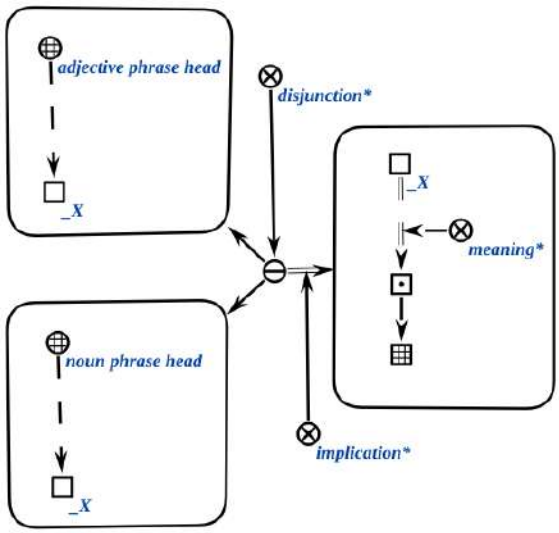

Figure 4. Noun phrase head and adjective phrase head semantic interpretation rule.

separately. For example: the meaning of "black cat" is a set of black cats, i.e. the intersection of a set of cats and a set of black objects.

Figure 6 shows the rule according of interpreting a verb phrase. We need to include the entire branch of the verb phrase in the premise of the rule because this is how we can determine the type of the verb – this rule is intended for the interpretation of intransitive verbs. The meaning of this construction is a class of actions.

Figure 7 shows the rule of interpreting a determiner phrase headed by an indefinite article. The meaning of such a construction is the existence of an element of the class that corresponds to the meaning of the noun phrase included in this determiner phrase.

Figure 8 shows the rule of interpreting an intermediate projection of a tense phrase head, consisting of an auxiliary verb and a full verb. The auxiliary verb in this case specifies

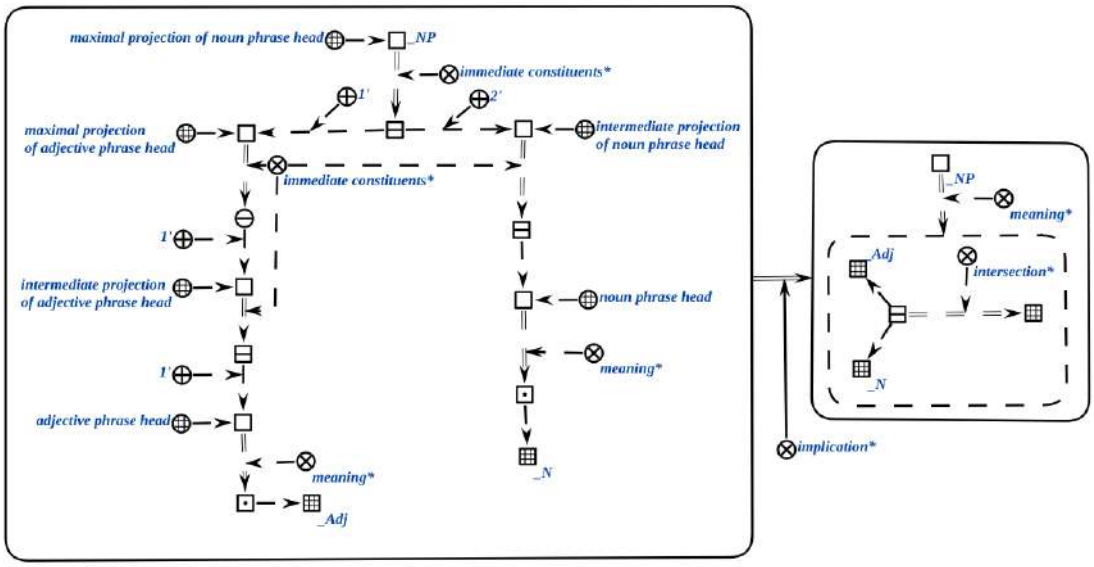

Figure 5. Semantic interpretation rule for maximal projections of noun phrase heads.

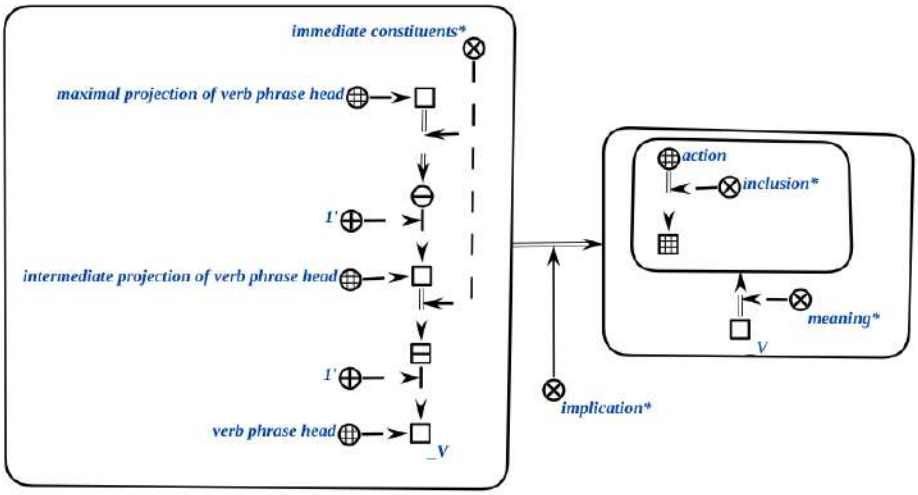

Figure 6. Semantic interpretation rule for maximal projections of intransitive verbs.

the class of actions with respect to time (whether it is planned, in progress, already completed, etc.).

Figure 9 shows the rule of interpreting a maximal projection of a tense phrase head. In this case, the maximal projection includes the subject, represented by the determiner phrase in the specifier position. The resulting meaning is a combination of the meaning of the determiner phrase obtained at an earlier stage of analysis and the meaning of a tense phrase. The determiner phrase is interpreted as the subject of the action denoted by the verb.

Figure 10 shows the rule of interpreting a maximal projection of a complementizer phrase head. This rule specifies the interpretation of a sentence with a transitive verb and is obtained as a result of applying all previously listed rules.

IV. Conclusion

In this article the linguistic means of formal description of syntax and denotational semantics of different languages in intelligent computer systems of the new generation have been described. A formal ontology of various languages has been proposed. The treatment of such notions as language, sign, information construction, sign construction, syntax, semantics, meaning, value, etc. has been clarified. A formalization of the subject domains of syntax and denotational semantics of natural languages has been proposed.

All developed means of describing the syntax and denotational semantics of languages are presented in a unified form, which allows to ensure their compatibility and significantly

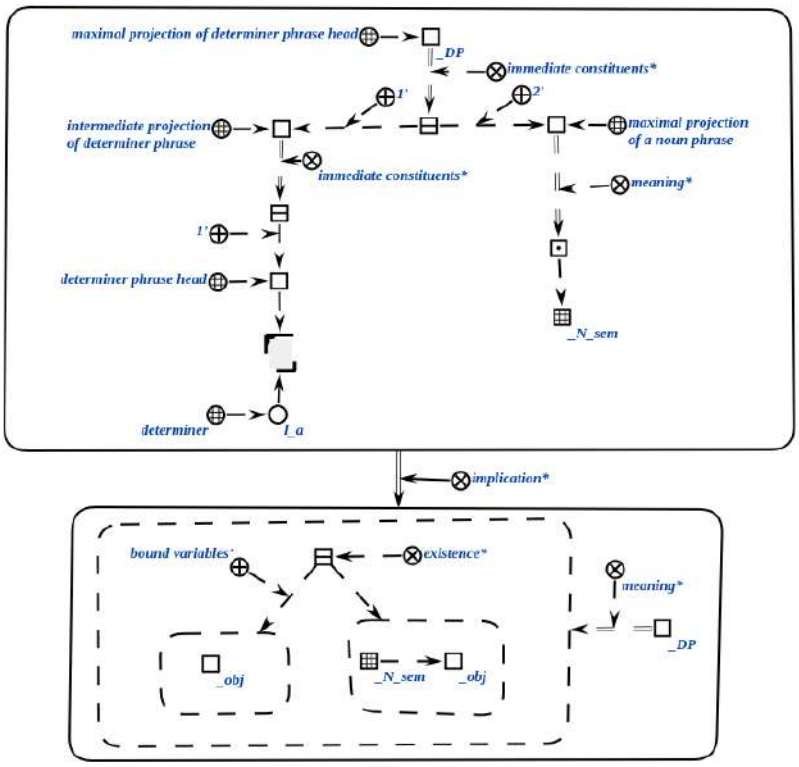

Figure 7. Semantic interpretation rule for maximal projections of determiner phrase heads.

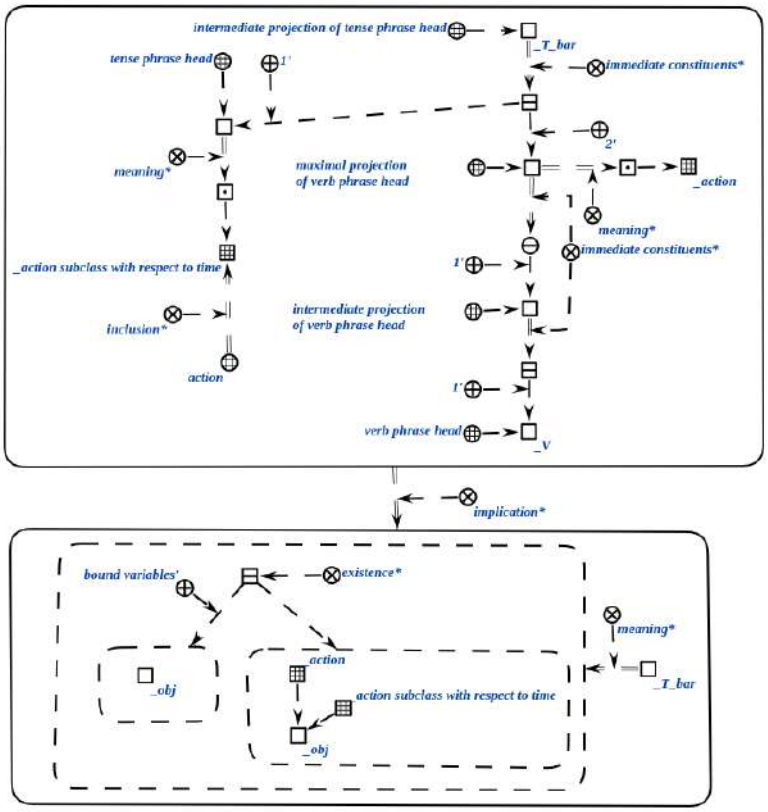

Figure 8. Semantic interpretation rule for an intermediate projection of a tense phrase head.

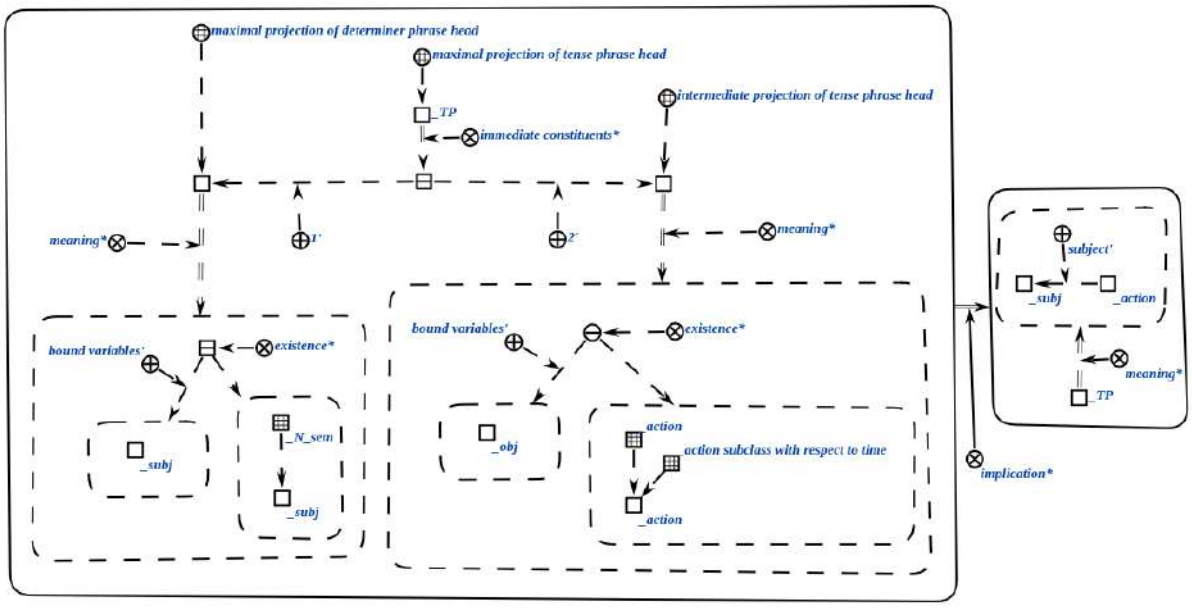

Figure 9. Semantic interpretation rule for a maximal projection of a tense phrase head (with an intransitive verb).

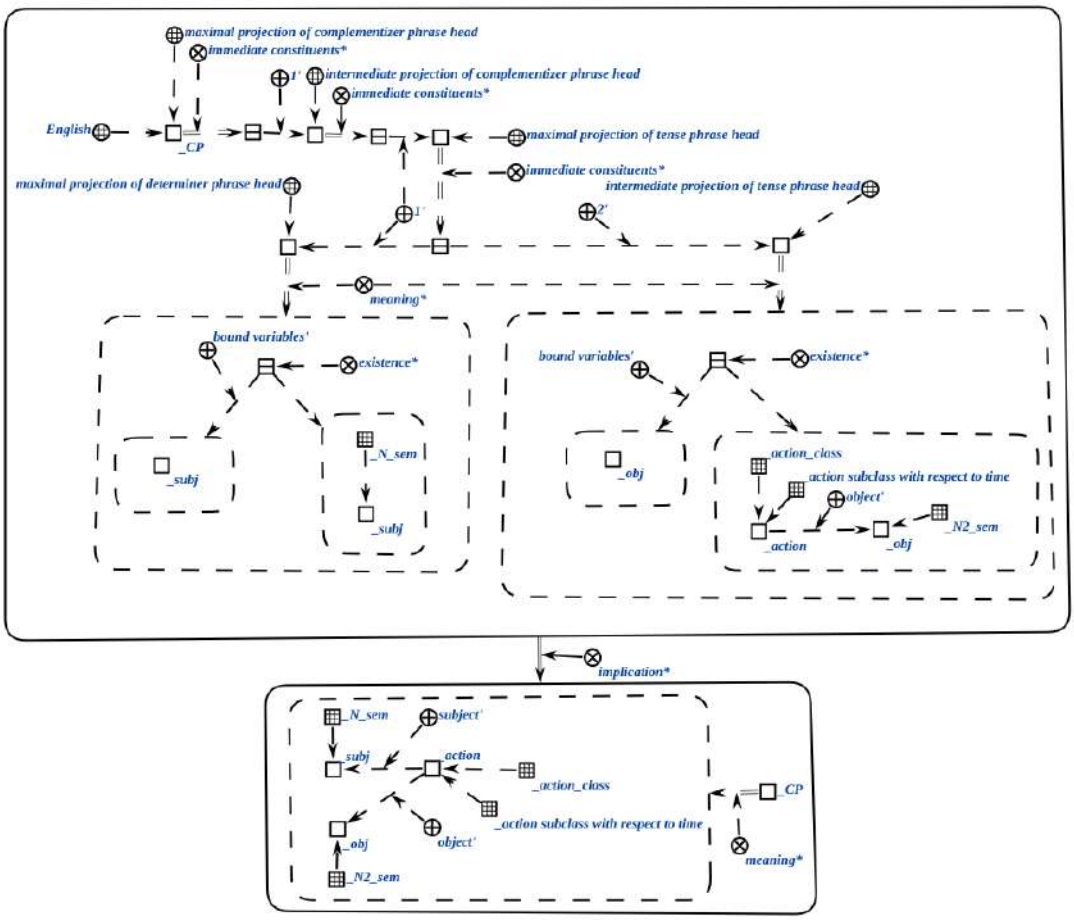

Figure 10. Semantic interpretation rule for sentences with transitive verbs

reduce the overhead in the development of various systems that use them.

As a result a fragment of the upper-level ontology has been obtained, using which it becomes possible to formalize subject domains of specific languages, both natural and artificial, for their further use in natural-language interfaces of next-generation intelligent computer systems.

The fragment of ontology provided above is by no means complete. Given that, one of the directions of further study on this topic is extending the ontology, which would include at least the following:

- creating an ontology of lexical meanings in natural languages;
- describing the mechanism of matching tokens with lexemes in the knowledge base.

To add the ability to process a new language by a system designed on the basis of the approach proposed in this paper, it is necessary to create an ontology of this language, based on the presented upper-level ontology, which would include the following:

- vocabulary with lexemes of that language;
- specific features of syntax and vocabulary for that language;
- features of semantic interpretation specific to that language.

Acknowledgment

The authors would like to thank the research teams of the Department of Intellectual Information Technologies of BSUIR and the Department of Theory and Practice of Translation #1 of MSLU for their help and valuable comments. This research was partially supported by the BRFFR (BRFFR-RFFR No. F21RM-139).

References

[1] Golenkov, V. V., "Methodological problems of the current state of works in the field of artificial intelligence," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 17–24, 2021.

[2] S. Pileggi, A. Lopez-Lorca, and G. Beydoun, "Ontologies in software engineering," 11 2018.

[3] P. Lando, A. Lapujade, G. Kassel, and F. Fürst, "Towards a general ontology of computer programs." 01 2007, pp. 163–170.

[4] S. Farrar, W. D. Lewis, and T. Langendoen, "A common ontology for linguistic concepts," 2002.

[5] C. Chiarcos, *Interoperability of Corpora and Annotations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 161–179. [Online]. Available: https://doi.org/10.1007/978-3-642-28249-2_16

[6] "Text Encoding Initiative," Available at: https://tei-c.org/, (accessed 2022, October).

[7] "EAGLES Recommendations for the Morphosyntactic Annotation of Corpora," Available at: https://home.uni-leipzig.de/burr/Verb/htm/LinkedDocuments/annotate.pdf, (accessed 2022, October).

[8] N. Ide and J. Pustejovsky, "What does interoperability mean , anyway ? toward an operational definition of interoperability for language technology," 2010.

[9] A. C. Schalley, "Ontologies and ontological methods in linguistics," *Language and Linguistics Compass*, vol. 13, no. 11, p. e12356, 2019, e12356 LNCO-0634.R3. [Online]. Available: https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/lnc3.12356

[10] J. P. McCrae, P. Labropoulou, J. Gracia, M. Villegas, V. Rodríguez-Doncel, and P. Cimiano, "One ontology to bind them all: The meta-share owl ontology for the interoperability of linguistic datasets on the web," in *The Semantic Web: ESWC 2015 Satellite Events*, F. Gandon, C. Guéret, S. Villata, J. Breslin, C. Faron-Zucker, and A. Zimmermann, Eds. Cham: Springer International Publishing, 2015, pp. 271–282.

[11] "The General Ontology of Linguistic Description," Available at: http://linguistics-ontology.org/info/about, (accessed 2022, October).

[12] A. Pease, I. Niles, and J. Li, "The suggested upper merged ontology: A large ontology for the semantic web and its applications," 01 2002.

[13] S. Farrar and D. Langendoen, "A linguistic ontology for the semantic web," *Glot International*, vol. 7, pp. 97–100, 03 2003.

[14] C. Chiarcos, "Ontologies of linguistic annotation: Survey and perspectives," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 303–310. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/911_Paper.pdf

[15] "The Theoretical Status of Ontologies in Natural Language Processing," Available at: https://arxiv.org/abs/cmp-lg/9704010, (accessed 2022, October).

[16] J. A. Bateman and G. Fabris, "The generalized upper model knowledge base: Organization and use," 2002.

[17] P. Buitelaar, P. Cimiano, P. Haase, and M. Sintek, "Towards linguistically grounded ontologies," in *The Semantic Web: Research and Applications*, L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, and E. Simperl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 111–125.

[18] T. Kostareva, S. Chuprina, and A. Nam, "Using ontology-driven methods to develop frameworks for tackling nlp problems," in *AIST*, 2016.

[19] O. Nevzorova and V. Nevzorov, "Ontology-driven processing of unstructured text," in *Artificial Intelligence*, S. O. Kuznetsov and A. I. Panov, Eds. Cham: Springer International Publishing, 2019, pp. 129–142.

[20] P. Cimiano, J. Lüker, D. Nagel, and C. Unger, "Exploiting ontology lexica for generating natural language texts from RDF data," in *Proceedings of the 14th European Workshop on Natural Language Generation*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 10–19. [Online]. Available: https://aclanthology.org/W13-2102

[21] N. Bouayad-Agha, G. Casamayor, and L. Wanner, "Natural language generation in the context of the semantic web," *Semantic Web*, vol. 5, pp. 493–513, 01 2014.

[22] D. Saha, A. Floratou, K. Sankaranarayanan, U. F. Minhas, A. R. Mittal, and F. Özcan, "Athena: An ontology-driven system for natural language querying over relational data stores," *Proc. VLDB Endow.*, vol. 9, no. 12, p. 1209–1220, aug 2016. [Online]. Available: https://doi.org/10.14778/2994509.2994536

[23] M. Shamsfard and A. A. Barforoush, "Learning ontologies from natural language texts," *International Journal of Human-Computer Studies*, vol. 60, no. 1, pp. 17–63, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071581903001368

[24] J. A. Bateman, J. Hois, R. Ross, and T. Tenbrink, "A linguistic ontology of space for natural language processing," *Artificial Intelligence*, vol. 174, no. 14, pp. 1027–1071, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370210000858

[25] M. Moens and M. Steedman, "Temporal ontology in natural language," in *Proceedings of the 25th Annual Meeting on Association for Computational Linguistics*, ser. ACL '87. USA: Association for Computational Linguistics, 1987, p. 1–7. [Online]. Available: https://doi.org/10.3115/981175.981176

[26] A. Dobrov, A. Dobrova, P. Grokhovskiy, M. Smirnova, and N. Soms, "Computer ontology of tibetan for morphosyntactic disambiguation," in *Digital Transformation and Global Society*, D. A. Alexandrov, A. V. Boukhanovsky, A. V. Chugunov, Y. Kabanov, and O. Koltsova, Eds. Cham: Springer International Publishing, 2018, pp. 336–349.

[27] "WordNet: A Lexical Database for English," Available at: https://wordnet.princeton.edu, (accessed 2022, October).

[28] "VerbNet: A Computational Lexical Resource for Verbs," Available at: https://verbs.colorado.edu/verbnet/, (accessed 2022, October).

[29] "FrameNet," Available at: http://framenet.icsi.berkeley.edu, (accessed 2022, October).

[30] A. Pease and C. Fellbaum, *Formal ontology as interlingua: the SUMO and WordNet linking project and global WordNet*, ser. Studies in Natural Language Processing. Cambridge University Press, 2010, p. 25–35.

[31] T. Matsukawa and E. Yokota, “Development of the concept dictionary implementation of lexical knowledge,” in *Lexical Semantics and Knowledge Representation*, 1991. [Online]. Available: https://aclanthology.org/W91-0219
[32] N. Calzolari, “Acquiring and representing semantic information in a lexical knowledge base.” 06 1991, pp. 235–243.
[33] P. Buitelaar, T. Declerck, A. Frank, S. Racioppa, M. Kiesel, M. Sintek, R. Engel, M. Romanelli, D. Sonntag, B. Loos, V. Micelli, R. Porzel, and P. Cimiano, “LingInfo: Design and Applications of a Model for the Integration of Linguistic Information in Ontologies,” 2006. [Online]. Available: http://smartweb.dfki.de/Vortraege/OntoLex2006.pdf
[34] P. Cimiano, P. Haase, M. Herold, M. Mantel, and P. Buitelaar, “Lexonto: A model for ontology lexicons for ontology-based nlp,” 2007.
[35] P. Buitelaar, M. Sintek, and M. Kiesel, “A multilingual/multimedia lexicon model for ontologies,” in *The Semantic Web: Research and Applications*, Y. Sure and J. Domingue, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 502–513.
[36] J. McCrae, G. Cea, P. Buitelaar, P. Cimiano, T. Declerck, A. Gomez-Perez, J. Gracia, L. Hollink, E. Montiel-Ponsoda, D. Spohr, and T. Wunner, “Interchanging lexical resources on the semantic web,” *Language Resources and Evaluation*, vol. 46, pp. 701–719, 12 2012.
[37] ““semantic web”: W3c’s vision of the web of linked data.” Available at: https://www.w3.org/standards/semanticweb/, (accessed 2022, October).
[38] “RDF: a standard model for data interchange on the Web.” Available at: https://www.w3.org/RDF/, (accessed 2022, October).
[39] “The Resource Description Framework. Abstract syntax (a data model) which serves to link all RDF-based languages and specifications.” Available at: https://www.w3.org/TR/rdf11-concepts/, (accessed 2022, October).
[40] “NLTK NLP library,” Available at: https://www.nltk.org/, (accessed 2022, October).
[41] “spaCy NLP library,” Available at: https://spacy.io/, (accessed 2022, October).
[42] T. N. Erekhinskaya, M. Tatu, M. Balakrishna, S. Patel, D. Strebkov, and D. I. Moldovan, “Ten ways of leveraging ontologies for rapid natural language processing customization for multiple use cases in disjoint domains,” *Open J. Semantic Web*, vol. 7, pp. 33–51, 2020.
[43] V. V. Golenkov, N. A. Gulyakina, D. V. Shunkevich, *Open technology for ontological design, production and operation of semantically compatible hybrid intelligent computer systems*, G. V.V., Ed. Minsk: Bestprint, 2021.
[44] “SIL: Glossary of Linguistic Terms,” Available at: https://glossary.sil.org, (accessed 2022, October).
[45] D. Adger, *Core Syntax: A Minimalist Approach*, ser. Core linguistics. Oxford University Press, 2003. [Online]. Available: https://books.google.by/books?id=GMJ1QgAACAAJ
[46] R. Jackendoff and R. Jackendoff, *X Syntax: A Study of Phrase Structure*, ser. Linguistic inquiry monographs. MIT Press, 1977. [Online]. Available: https://books.google.by/books?id=ALf6PgAACAAJ
[47] L. Haegeman, *Introduction to Government and Binding Theory*, ser. Blackwell Textbooks in Linguistics. Wiley, 1994. [Online]. Available: https://books.google.by/books?id=_fvUInHLUTwC
[48] A. Carnie, *Syntax: A Generative Introduction*, ser. Introducing Linguistics. Wiley, 2012. [Online]. Available: https://books.google.by/books?id=MFZ1UV3YGtgC
[49] I. Heim and A. Kratzer, *Semantics in Generative Grammar*, ser. Blackwell Textbooks in Linguistics. Wiley, 1998. [Online]. Available: https://books.google.by/books?id=jAvR2DB3pPIC
[50] Y. Winter, *Elements of Formal Semantics*. Edinburgh: Edinburgh University Press, 2016. [Online]. Available: https://doi.org/10.1515/9780748677771
[51] P. Portner and B. Partee, *Formal Semantics: The Essential Readings*, ser. Linguistics: The Essential Readings. Wiley, 2008. [Online]. Available: https://books.google.by/books?id=ptgUWREtAkMC

Средства формального описания синтаксиса и денотационной семантики различных языков в интеллектуальных компьютерных системах нового поколения

Гойло А. А., Никифоров С. А.

Статья посвящена языковым средствам формального описания синтаксиса и денотационной семантики различных языков в интеллектуальных компьютерных системах нового поколения. Предложена формальная онтология различных языков. Уточнена трактовка таких понятий как язык, знак, информационная конструкция, знаковая конструкция, синтаксис, семантика, смысл, значение и др. Формализованы предметные области синтаксиса и денотационной семантики естественных языков. В результате получена онтология верхнего уровня, с использованием которой становится возможной формализация более частных предметных областей конкретных языков как естественных, так и искусственных для их дальнейшего применения в естественно-языковых интерфейсах интеллектуальных систем нового поколения.

Received 28.10.2022