Set Theory in the MathSem Program

Luxemburg Andrey

Moscow, Russia

andr_lux@mail.ru

Abstract

Knowledge representation is a popular research field in IT.

As mathematical knowledge is most formalized, its representation is important and interesting.

Mathematical knowledge consists of various mathematical theories.

I’ll present program MATHSEM and deductive system that derives mathematical notions, axioms and theorems of elementary set theory.

All these notions, axioms and theorems can be considered a small mathematical theory.

The term "knowledge representation" usually means representations of knowledge aimed to enable automatic processing of the knowledge base on modern computers, in particular, representations that consist of explicit objects and assertions or statements about them.

We are particularly interested in the following formalisms for knowledge representation:

1. First order predicate logic [1, 4].

2. Deductive (production) systems. In such a system there is a set of initial objects, rules of inference to build new objects from initial ones or ones that are already build, and the whole of initial and constructed objects [5].

- In this presentation we describe a part of the project and a part of the interactive computer application for automated building of mathematical theories.
- Studies in this area are mainly connected with writing programs for automatic theorem proving, the development of the semantic Internet, ontologies.
- First-order theorem proving is one of the most mature subfields of automated theorem proving. On the other hand, it is still semi-decidable, and a number of sound and complete calculi have been developed, enabling fully automated systems. More expressive logics, such as higher order logics, allow the convenient expression of a wider range of problems than first order logic, but theorem proving for these logics is less well developed

PROVERS

- E is a high-performance prover for full first-order logic, but built on a purely equational calculus.
- Otter, developed at the Argonne National Laboratory, is based on first-order resolution and paramodulation.

  Otter has since been replaced by Prover9, which is paired with Mace4.
- SETHEO is a high-performance system based on the goal-directed model elimination calculus. It is developed in the automated reasoning group of Technical University of Munich.
- E and SETHEO have been combined (with other systems) in the composite theorem prover E-SETHEO.
- Vampire is developed and implemented at Manchester University by Andrei Voronkov and Krystof Hoder, formerly also by Alexandre Riazanov.
- Waldmeister is a specialized system for unit-equational first-order logic.
- SPASS is a first order logic theorem prover with equality. This is developed by the research group Automation of Logic, Max Planck Institute for Computer Science

Coq

- Coq is an interactive theorem prover. It allows the expression of mathematical assertions, mechanically checks proofs of these assertions, helps to find formal proofs, and extracts a certified program from the constructive proof of its formal specification.
- Coq works within the theory of the calculus of inductive constructions, a derivative of the calculus of constructions. Coq is not an automated theorem prover but includes automatic theorem proving tactics and various decision procedures.

MATHSEM

- In our project, unlike other provers, where it is necessary to translate the theorem and the axioms needed for its proof in a formal language and directly to the internal language of the system itself, on the contrary, a formal written axioms and theorems are generated automatically by a computer program.
- For a new formula written in the formal language a human expert (mathematician) can translate it into "natural" language (Russian, English etc.), thus we obtain a glossary of basic notions of the system. More complicated formulae are translated into natural language using an algorithm and the glossary. Thus it is possible to construct (new) mathematical objects, concepts, definitions, theorems.

POWER of SET THEORY

- Using the language of set theory and axiomatic set theory can be constructed a significant part of mathematics.
- That is why as the original object taken membership predicate.
- In this presentation, therefore, is considered as an example provide you with the basic concepts of set theory (empty set, subset, membership, inclusion, intersection, union, powerset, Cartesian product).
- With the help of program MathSem one can build the axioms and theorems of set theory. In the future, a deductive system is expected to bring and represent in the form of a semantic net framework of set theory, Euclidean geometry, group theory and graph theory.

Description of the Project

- We define a formal language (close to first-order predicate logic), and a deductive (production) system that builds expressions in this language.
- There are rules for building new objects from initial (atomic) ones and the ones already built. Objects can be either statement (predicates), or definitions (these could be predicates or truth sets of predicates).
- The membership predicate is taken as the atomic formula.
- Rules for building new objects include logical operations (conjunction, disjunction, negation, implication), adding a universal or existential quantifier, and one more rule: building the truth set of a predicate.
- One can consider symbols denoting predicates and sets, and also the predicates and sets themselves (when an interpretation or model is fixed). One more rule allows substitution of an individual variable or a term for a variable. Further, when we have built a new formula, we can simplify it using term-rewriting rules and logical laws (methods of automated reasoning).

- In order to prove theorems one can apply well-known methods of automated reasoning (resolution method, method of analytic tableaux, natural deduction, inverse method), as well as new methods based on the knowledge of «atomic» structure of the formula (statement) that we are trying to prove.
- For a new formula written in the formal language a human expert (mathematician) can translate it into «natural» language (Russian, English etc.), thus we obtain a glossary of basic notions of the system. More complicated formulae are translated into natural language using an algorithm and the glossary.
- The deductive system constructed here is based on classical first-order predicate logic. The initial object is the membership predicate, and the derivations result into mathematical notions and theorems. The computer program (algorithm) builds formulae from atomic ones (makes the semantic net of the derivation).

Description of Deductive System of MathSem

By a deductive (production) system we understand a triple

DS=<O_i , R_j , O_k>,

where O_i is a set of initial objects, R_j are rules for building new objects, and O_k is the set of objects constructed, $O_i \subset O_k$

$$R_j : {O_k}^{n} \rightarrow O_k$$

Examples: <axioms, rules of inference, theorems>, <a line segment; Cartesian product, sewing by the border; a square, a torus, a sphere, etc.>. Deductive system of MathSem is not deductive system in the sense of first example.

Notations

x_i -variables for elements of sets; A_j - variables for sets; P_i denote predicates;

M_i - denote sets of mathematical objects;

$\wedge$ (&), $\vee$,$\neg$,$\Rightarrow$ - logical connectives;

$\forall$-universal quantifier; $\exists$- existential quantifier;

$\in$-denotes membership;

$<x_1 , x_2 \, . \, . \, . \, , x_n>$ stands for a tuple;

$\{ x_1 , x_2\}$ is a set defined by explicitly listing all its elements.

- Here we define objects derivable in the deductive system. They are either formulae that denote predicates, or formulae that describe mathematical objects (notions).
- The main idea. We have the membership predicate. Consider predicates that can be defined via the membership predicate, and truth sets of these predicates.
- The following formulae are considered atomic: $P_0(x_i , A_j) := x_i \in A_j$

Rules for building new formulae:

1. Negation: $P_j: = \neg P_i$

2. Grouping with logical connectives: $P_k := P_i \wedge P_j$, $P_k := P_i \vee P_j$

3. Quantification: for any free variable x in predicate P_i one can build new predicates $P_j := (\forall x) P_i$, $P_k := (\exists x) Pi$

4. Consider a finite number of variables: $x_1, x_2, \ldots, x_n$. One can construct a string $\langle x_1, x_2, \ldots, x_n \rangle$ it is a triple of variables.

5. Building a mathematical object (notion).

Consider a predicate $P(x_1, x_2, \ldots, x_k)$ (here $x_1, x_2, \ldots, x_k$ are free variables,).

One can build the set

$$M(x_{n+1}, \ldots, x_k) := \{ \langle x_1, x_2, \ldots, x_n \rangle \mid P(x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_k) \}$$

It is the truth set of the predicate.

$$\langle x_1, x_2, \ldots, x_n \rangle \in M(x_{n+1}, \ldots, x_k) \Leftrightarrow P(x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_k) \}$$

Or

$$P0(\langle x_1, x_2, \ldots, x_n \rangle, M(x_{n+1}, \ldots, x_k)) \Leftrightarrow P(x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_k)$$

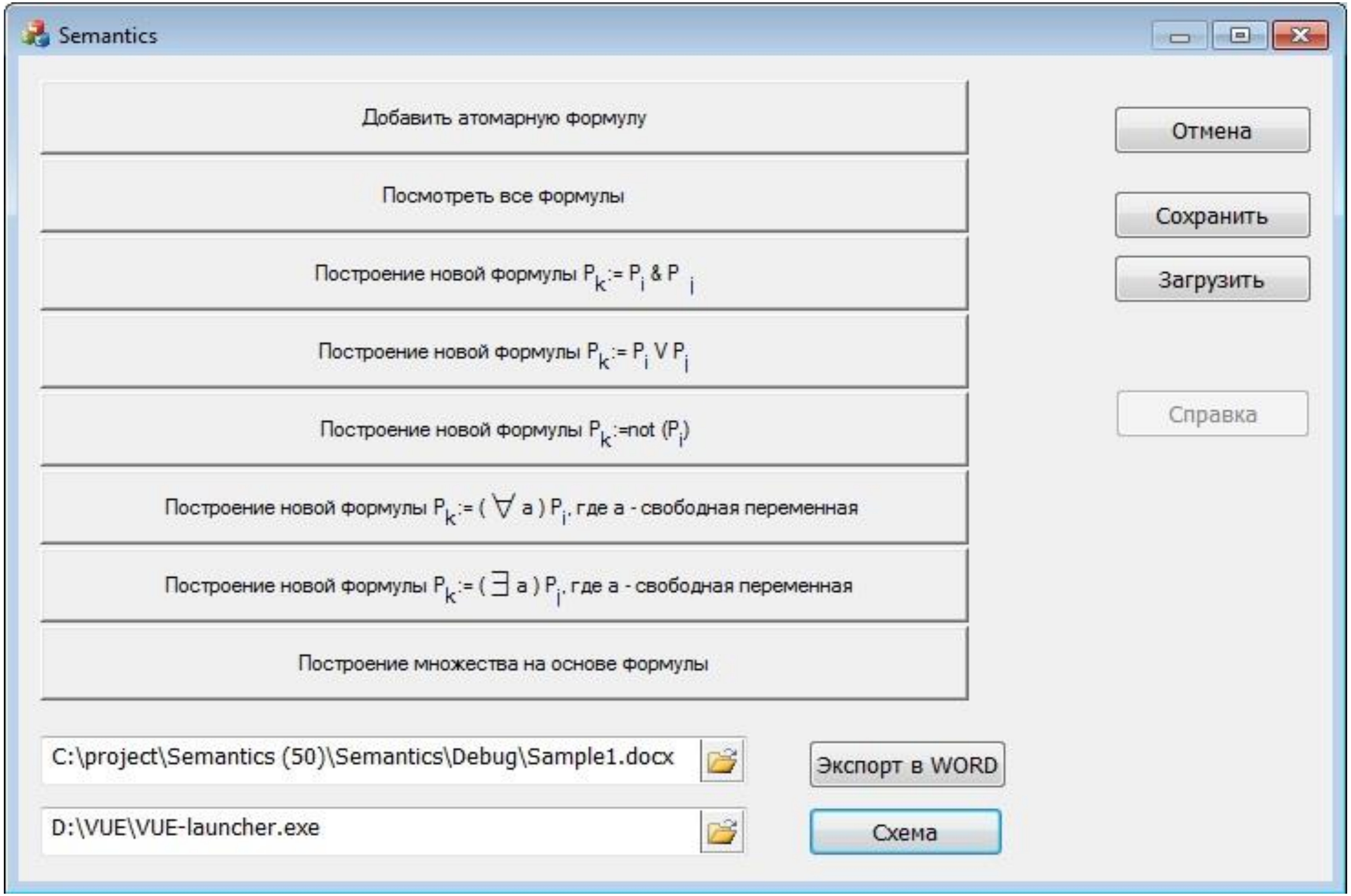
6. Substitution of variables. We can substitute variables in predicate *P* or object *M*. Since mathematical objects are actually sets, we can substitute them for variables into predicates.

Note that the interpretations plays an important role here. Different interpretations give different semantic values for predicates and different truth sets. To avoid logical paradoxes one can introduce a hierarchy of sets (B. Russel's simple type theory); another option is to choose rules of building new sets that do not allow possible paradoxes.

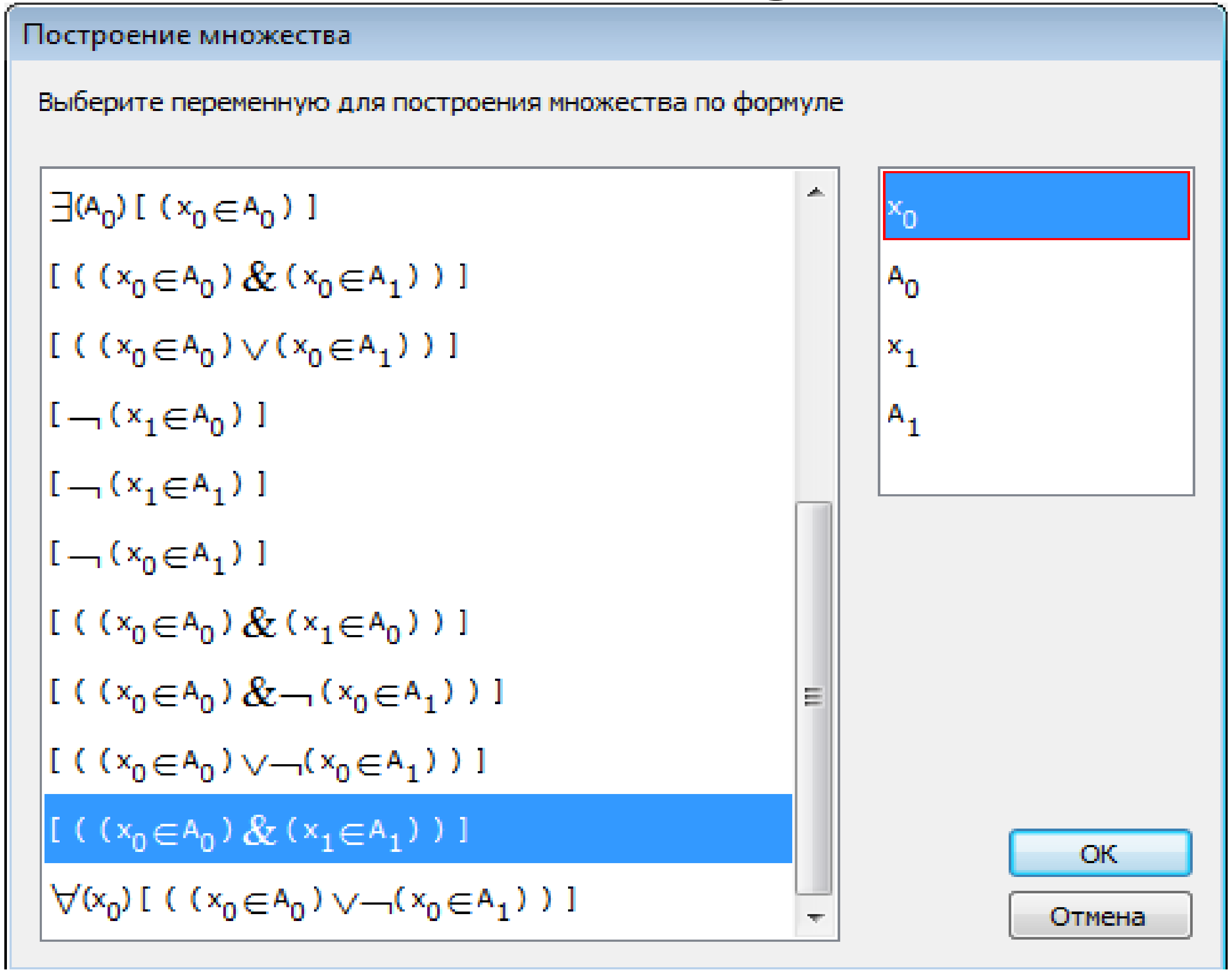
Software Description

- The MATHSEM program is being written by Vitaliy Tatarinsev and Andrey Luxemburg. In this program, complicated formulae are built from atomic ones «manually». The formulae built can be saved in a Word file along with their descriptions. One can also upload formulae from a Word file. Below one can find an example of building circa 30 formulae. Notably, all the signature of set theory is built from formulae with length (number of atomic formulae) not greater than two. Language on screenshots is Russian.

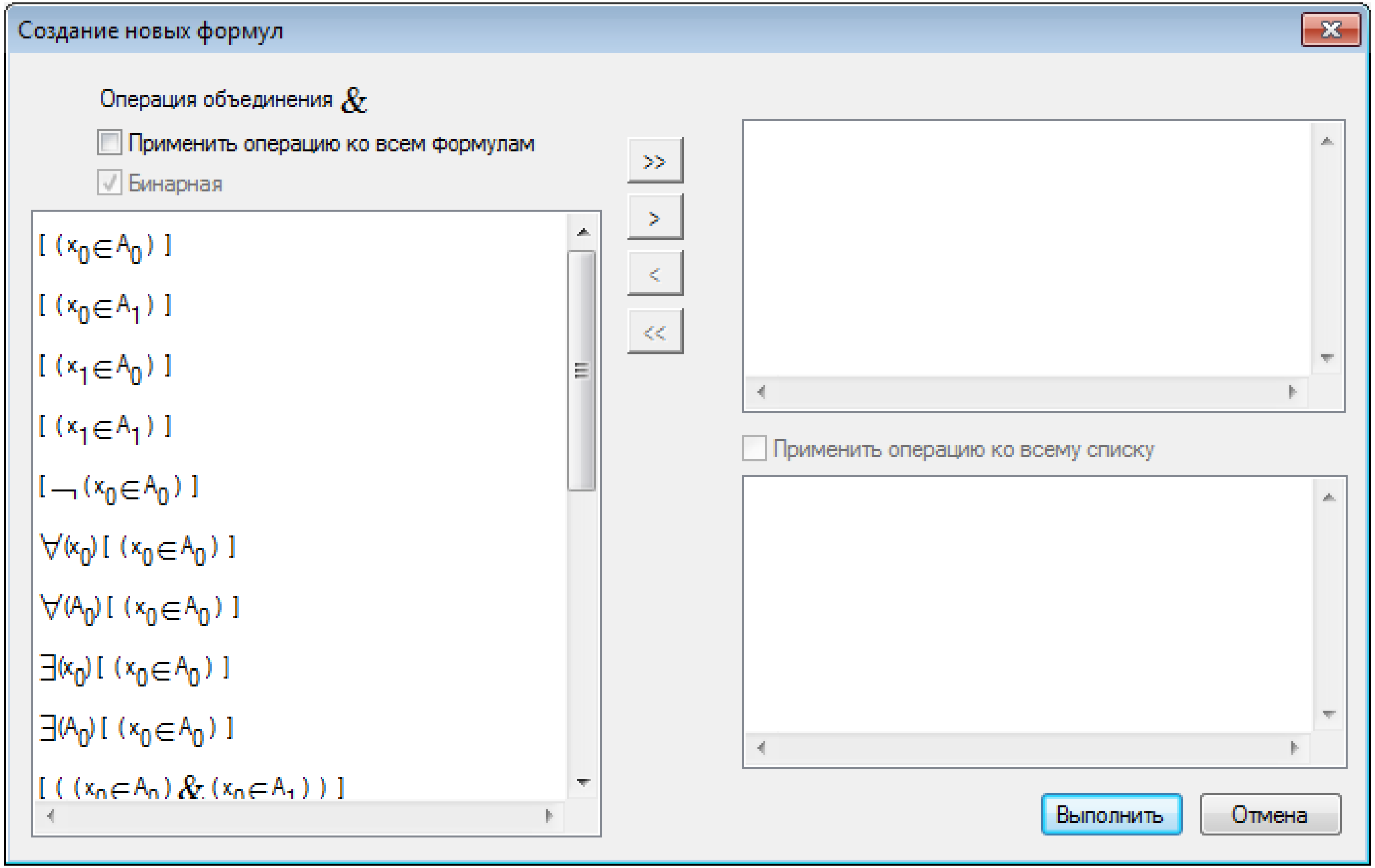
First screen interface. Start dialog window.



Interface for building truth set



Grouping with logical connectives



First part of table with formulae

Список формул

Ном...	Формула	Тип	Свободные...	Описание	Add...	Обозначение
1	$[(x_0 \in A_0)]$	выполнима	$x_0 . A_0$			$P_0 (x_0 . A_0)$
2	$[(x_0 \in A_1)]$	выполнима	$x_0 . A_1$			$P_0 (x_0 . A_1)$
3	$[(x_1 \in A_0)]$	выполнима	$x_1 . A_0$			$P_0 (x_1 . A_0)$
4	$[(x_1 \in A_1)]$	выполнима	$x_1 . A_1$			$P_0 (x_1 . A_1)$
5	$[\neg (x_0 \in A_0)]$	выполнима	$x_0 . A_0$			$P_1 (x_0 . A_0)$
6	$\forall (x_0) [(x_0 \in A_0)]$		A_0	A0 -универсум		$P_2 (A_0)$
7	$\forall (A_0) [(x_0 \in A_0)]$		x_0	false		$P_3 (x_0)$
8	$\exists (x_0) [(x_0 \in A_0)]$		A_0	A0 не пустое		$P_4 (A_0)$
9	$\exists (A_0) [(x_0 \in A_0)]$		x_0	true		$P_5 (x_0)$
10	$\{ x_0 \mid P_0 (x_0 . A_0) \}$		$x_0 . A_0$	A0		$M_0 (A_0)$
11	$\{ A_0 \mid P_0 (x_0 . A_0) \}$		$x_0 . A_0$			$R_0 (x_0)$
12	$[((x_0 \in A_0) \& (x_0 \in A_1))]$		$x_0 . A_0 . A_1$			$P_6 (x_0 . A_0 . A_1)$
13	$[((x_0 \in A_0) \vee (x_0 \in A_1))]$		$x_0 . A_0 . A_1$			$P_7 (x_0 . A_0 . A_1)$
14	$[\neg (x_1 \in A_0)]$	выполнима	$x_1 . A_0$			$P_8 (x_1 . A_0)$
15	$[\neg (x_1 \in A_1)]$	выполнима	$x_1 . A_1$			$P_9 (x_1 . A_1)$
16	$[\neg (x_0 \in A_1)]$	выполнима	$x_0 . A_1$			$P_{10} (x_0 . A_1)$

OK Отмена

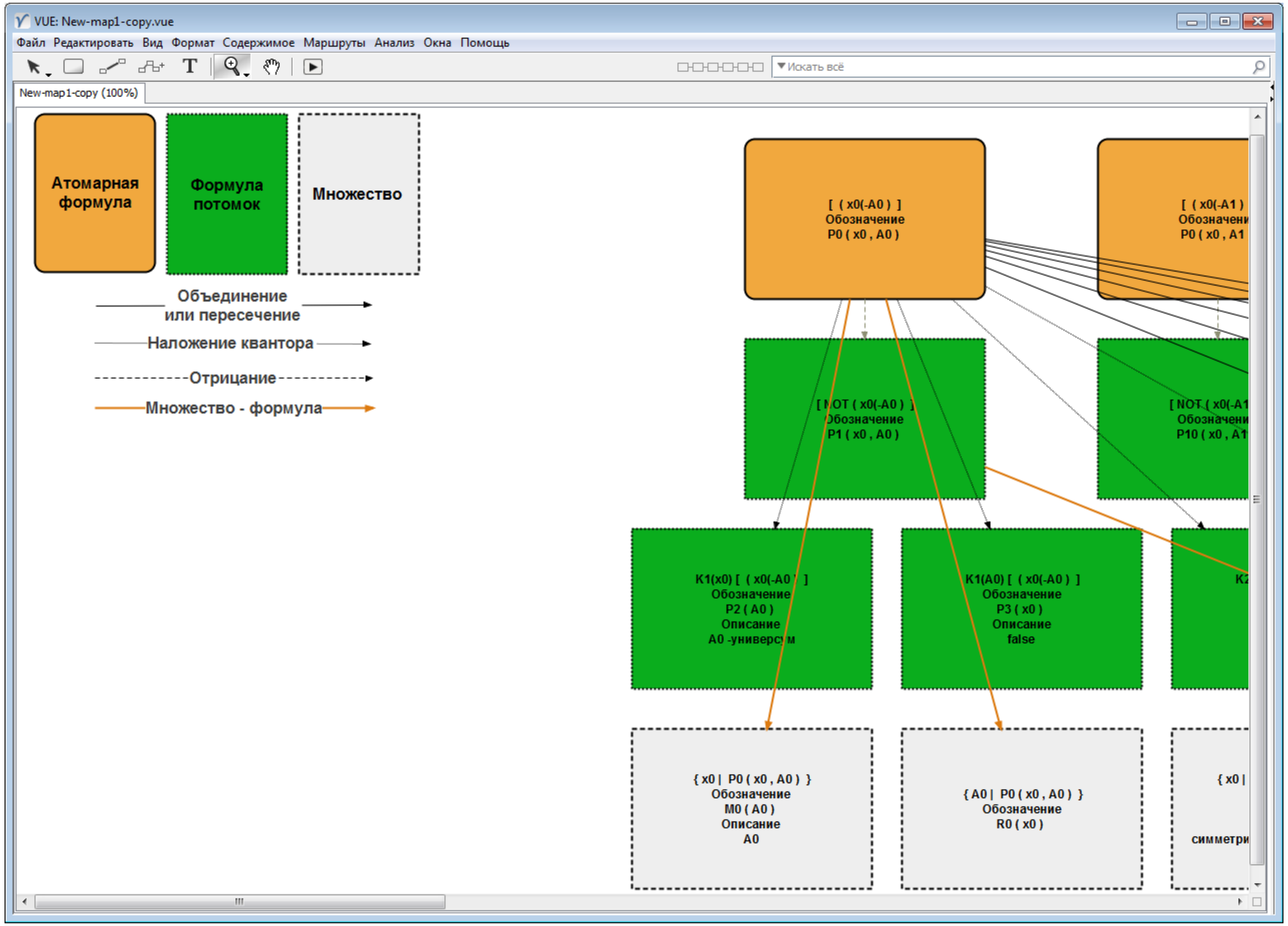
Second part of table with formulae

Список формул

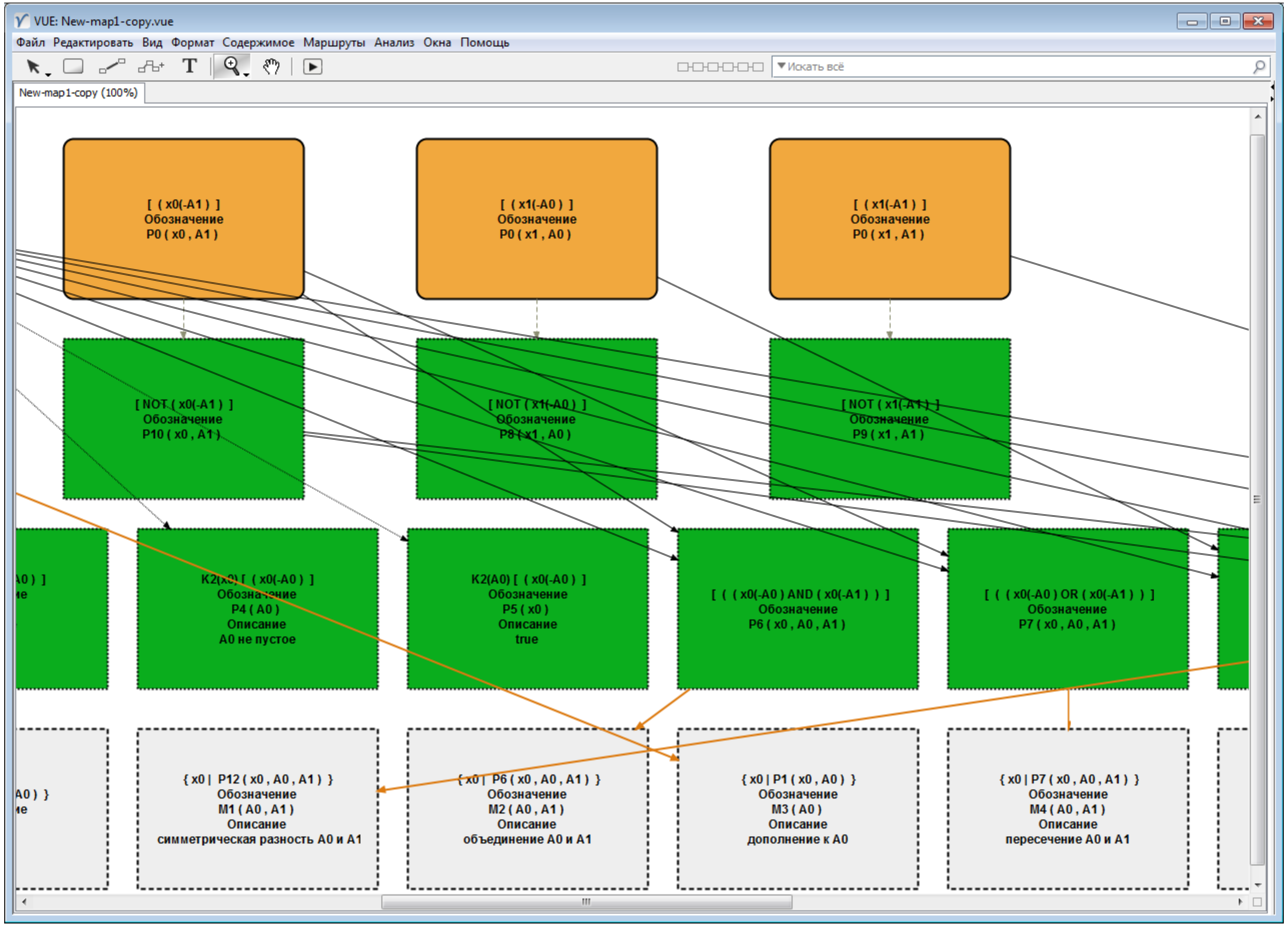
Ном...	Формула	Тип	Свободные...	Описание	Add...	Обозначение
14	$[\neg(x_1 \in A_0)]$	выполнима	$x_1 . A_0$			$P_8(x_1 . A_0)$
15	$[\neg(x_1 \in A_1)]$	выполнима	$x_1 . A_1$			$P_9(x_1 . A_1)$
16	$[\neg(x_0 \in A_1)]$	выполнима	$x_0 . A_1$			$P_{10}(x_0 . A_1)$
17	$[((x_0 \in A_0) \& (x_1 \in A_0))]$		$x_0 . A_0 . x_1$			$P_{11}(x_0 . A_0 . x_1)$
18	$[((x_0 \in A_0) \& \neg(x_0 \in A_1))]$		$x_0 . A_0 . A_1$			$P_{12}(x_0 . A_0 . A_1)$
19	$\{x_0 \mid P_{12}(x_0 . A_0 . A_1)\}$		$x_0 . A_0 . A_1$	симметрическая разность A0 и A1		$M_1(A_0 . A_1)$
20	$\{x_0 \mid P_6(x_0 . A_0 . A_1)\}$		$x_0 . A_0 . A_1$	объединение A0 и A1		$M_2(A_0 . A_1)$
21	$\{x_0 \mid P_1(x_0 . A_0)\}$		$x_0 . A_0$	дополнение к A0		$M_3(A_0)$
22	$\{x_0 \mid P_7(x_0 . A_0 . A_1)\}$		$x_0 . A_0 . A_1$	пересечение A0 и A1		$M_4(A_0 . A_1)$
23	$[((x_0 \in A_0) \vee \neg(x_0 \in A_1))]$		$x_0 . A_0 . A_1$			$P_{13}(x_0 . A_0 . A_1)$
24	$\{x_0 \mid P_{13}(x_0 . A_0 . A_1)\}$		$x_0 . A_0 . A_1$			$M_5(A_0 . A_1)$
25	$[((x_0 \in A_0) \& (x_1 \in A_1))]$		$x_0 . A_0 . x_1 . A_1$			$P_{14}(x_0 . A_0 . x_1 . A_1)$
26	$\{<x_0 . x_1> \mid P_{14}(x_0 . A_0 . x_1 . A_1)\}$		$x_0 . A_0 . x_1 . A_1$	декартовое произведение A0 и A1		$M_6(A_0 . A_1)$
27	$\forall(x_0)[((x_0 \in A_0) \vee \neg(x_0 \in A_1))]$		$A_0 . A_1$	A1 подмножество A0		$P_{15}(A_0 . A_1)$
28	$\{A_1 \mid P_{15}(A_0 . A_1)\}$		$A_0 . A_1$	множество подмножеств A0		$R_1(A_0)$
29	$\{<x_0 . x_1> \mid P_{11}(x_0 . A_0 . x_1)\}$		$x_0 . A_0 . x_1$			$M_7(A_0)$

OK | Отмена

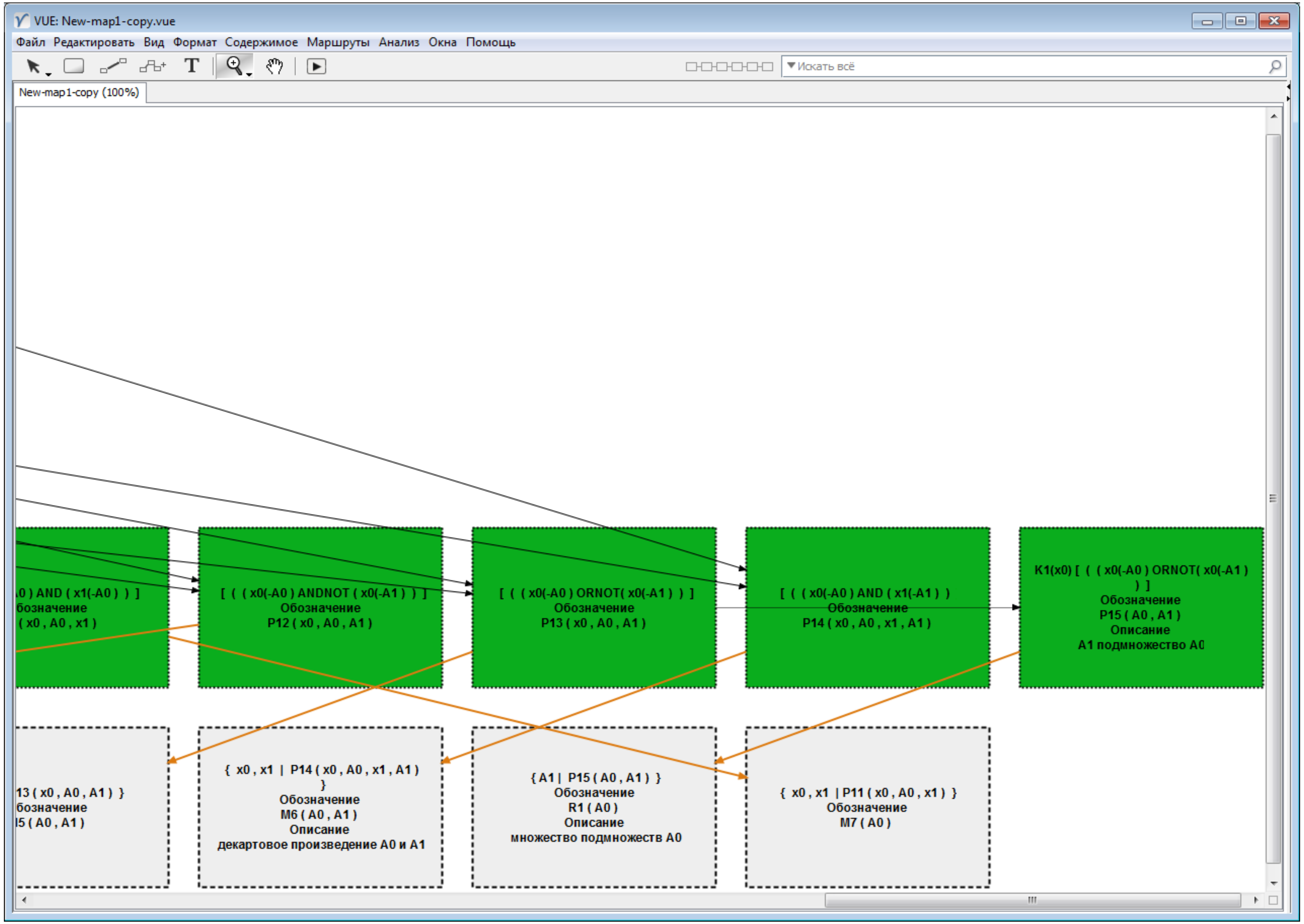
Construction of the semantic net. In the first row there are atomic formulae. In the second row – their negations



Predicates and sets from elementary set theory



Edges are parent-child relations



N	Formula	Notation	Symbol	Natural language
1	$[\ (x_0 \in A_0)\ ]$	$P_0(x_0, A_0)$		
2	$[\ (x_0 \in A_1)\ ]$	$P_0(x_0, A_1)$		
3	$[\ (x_1 \in A_0)\ ]$	$P_0(x_1, A_0)$		
4	$[\ (x_1 \in A_1)\ ]$	$P_0(x_1, A_1)$		
5	$[\neg (x_0 \in A_0)\ ]$	$P_1(x_0, A_0)$		
6	$\forall(x_0)\ [\ (x_0 \in A_0)\ ]$	$P_2(A_0)$	$A_0 = I$	A_0 -universe
7	$\forall(A_0)\ [\ (x_0 \in A_0)\ ]$	$P_3(x_0)$		
8	$\exists(x_0)\ [\ (x_0 \in A_0)\ ]$	$P_4(A_0)$	$A_0 \neq \emptyset$	A_0 not empty set
9	$\exists(A_0)\ [\ (x_0 \in A_0)\ ]$	$P_5(x_0)$		
10	$\{ x_0 \mid P_0(x_0, A_0) \}$	$M_0(A_0)$	A_0	A_0
11	$\{ A_0 \mid P_0(x_0, A_0) \}$	$R_0(x_0)$		Ri are sets consisting of sets (comments)
12	$[\ ((x_0 \in A_0)\ \&\ (x_0 \in A_1))\ ]$	$P_6(x_0, A_0, A_1)$		
13	$[\ ((x_0 \in A_0) \vee (x_0 \in A_1))\ ]$	$P_7(x_0, A_0, A_1)$		
14	$[\neg (x_1 \in A_0)\ ]$	$P_8(x_1, A_0)$		
15	$[\neg (x_1 \in A_1)\ ]$	$P_9(x_1, A_1)$		
16	$[\neg (x_0 \in A_1)\ ]$	$P_{10}(x_0, A_1)$		
17	$[\ (\ (x_0 \in A_0)\ \&\ (x_1 \in A_0)\ )\ ]$	$P_{11}(x_0, A_0, x_1)$		
18	$[\ (\ (x_0 \in A_0)\ \&\neg (x_0 \in A_1)\ )\ ]$	$P_{12}(x_0, A_0, A_1)$		
19	$\{ x_0 \mid P_{12}(x_0, A_0, A_1) \}$	$M_1(A_0, A_1)$	$A_0 \setminus A_1$	difference of A_0 and A_1
20	$\{ x_0 \mid P_6(x_0, A_0, A_1) \}$	$M_2(A_0, A_1)$	$A_0 \cap A_1$	intersection of A_0 and A_1
21	$\{ x_0 \mid P_1(x_0, A_0) \}$	$M_3(A_0)$		the complement to A_0
22	$\{ x_0 \mid P_7(x_0, A_0, A_1) \}$	$M_4(A_0, A_1)$	$A_0 \cup A_1$	union of A_0 and A_1
23	$[\ (\ (x_0 \in A_0) \vee \neg (x_0 \in A_1)\ )\ ]$	$P_{13}(x_0, A_0, A_1)$		
24	$\{ x_0 \mid P_{13}(x_0, A_0, A_1) \}$	$M_5(A_0, A_1)$		
25	$[\ (\ (x_0 \in A_0)\ \&\ (x_1 \in A_1)\ )\ ]$	$P_{14}(x_0, A_0, x_1, A_1)$		
26	$\{ < x_0, x_1 > \mid P_{14}(x_0, A_0, x_1, A_1) \}$	$M_6(A_0, A_1)$	A_0 x A_1	Cartesian product of A_0 and A_1
27	$\forall(x_0)\ [\ ((x_0 \in A_0) \vee \neg (x_0 \in A_1))\ ]$	$P_{15}(A_0, A_1)$	$A_1 \subset A_0$	A_1 subset A_0
28	$\{ A_1 \mid P_{15}(A_0, A_1) \}$	$R_1(A_0)$		the powerset of A_0
29	$\exists(x_0)\ \exists(A_0)\ [\ (x_0 \in A_0)\ ]$	$P_{16}()$		TRUE $A_0 = \{ x_0 \}$

Extending language

- Starting from the predicate of membership, we have obtained inclusion, intersection, union, complement, relative complement, Cartesian product of sets, subset relation, powerset.
- $\langle$ **Set**;$\in$ $\rangle \rightarrow \langle$ **Set;** $\in$, $\cap$, $\cup$, $\times$, $\subset$ $\rangle$
- We have built all the signature of set theory from the membership predicate in a combinatorial way. Usually in a first order language the signature is fixed in the beginning, so here we extend our language.
- Inclusion, intersection, union, complement etc. are semantic units of set theory. We have built relations on this units based on derivation. That is why it’s syntax and semantic net at the same time.

New notations are entered into the table (the fourth column) by a human, since the computer doesn’t know them. Then they are copied to the first column, and the program continues to work with the new notations. Using the MathSem program, one can build, for example, the following theorems:

$[(A \subset B \wedge B \subset C) \Rightarrow A \subset C]$=TRUE ; $[(A_0 \cap A_1) \subset A_1]$=TRUE; $[(A_0 \cup A_1) \subset A_1 = A_0 \subset A_1]$; $[A_1 \subset (A_0 \cup A_1)]$=TRUE; $[A_1 \subset (A_1 \cap A_0) = A_1 \subset A_0]$.

Our system’s interpretation is defined on a countable or finite set of x_i, A_i, M_i, R_i. For proving theorems we use classical logic and this logic's laws and rules of inference. In addition, we can substitute a truth set as a term into a predicate.

For first-order predicate logic there exist standard algorithms of automatic proof.[2,6]

Conclusion

- We have built a semantic network of concepts and statements from set theory. It is interesting to trace the connection of the constructed system with the axiomatics of ZFC set theory[4]. Some of the ZFC axioms are syntactically derived in our system.
- MathSem computer program can be used as a computer practicum, for example, in discrete mathematics. Using this program, students can study mathematical logic, set theory, relationship theory, graph theory, group theory. It introduces students to such modern trends in science as semantic networks, knowledge representation, ontologies, logical inference.

References

1. Vereshchagin N. K., Shen A.: Languages and Calculi (in Russian), MCCME, Moscow (2008).
2. Vagin V. N., Golovina E. Yu., Zagoryanskaya A. A., Fomina M. V.: Exact and Plausible Reasoning in Intellectual Systems (in Russian). Fizmatlit, Moscow, (2004)..
3. Zakharov V. K.: Local Set Theory. Math. Notes, Vol. 77, Issue 1-2, pp. 177–193.Moscow (2005)
4. Lavrov I. A.: Mathematical Logic (in Russian). Academia, Moscow (2006)
5. Luxemburg A. A.: Automated Construction of Mathematical Theories (in Russian). Editorial URSS, Moscow (2005)
6. Robinson J.A., Voronkov A. (Eds.): Handbook of Automated Reasoning (in 2 volumes). Elsevier and MIT Press (2001)
7. Maslov S. Yu.: Theory of Deductive Systems and its Applications (in Russian). Radio i svyaz, Moscow (1986)
8. Davydenko I. T., Zhitko V. A., Zalivako S. S., Koronchik D. N., Moshenko S. G., Savelyeva O. Yu., Startsev S. S., Shunkevich D. V: An intellectual reference system for geometry (in Russian).In: Proc. of OSTIS-2011, Minsk (2011)
9. VUE http://vue.tufts.edu/about/index.cfm