Set Theory in the MathSem Program

Luxemburg Andrey

UMCA Technologies, 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. In this paper we present a 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.

Keywords: Semantic network · semantic net· mathematical logic · set theory · axiomatic systems · formal systems · semantic web · prover · ontology · knowledge representation · knowledge engineering · automated reasoning

1 Introduction

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 paper 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 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.

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. 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 paper, 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.

2 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).

3 Software Description

The MathSem program is being written by Vitaliy Tatarinsev.

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.

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	$\exists(x_0)\ [\ (x_0 \in A_0)\ ]$	$P_4(A_0)$	$A_0 \neq \varnothing$	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	$\{\langle x_0, x_1 \rangle \mid P_{14}(x_0, A_0, x_1, A_1)\}$	$M_6(A_0, A_1)$	$A_0 \text{ x } A_1$	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 $A_0=\{x_0\}$

Table 1.

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. Luxemburg A. A.: Automated Construction of Mathematical Theories (in Russian). Editorial URSS, Moscow (2005)
4. Robinson J.A., Voronkov A. (Eds.): Handbook of Automated Reasoning (in 2 volumes). Elsevier and MIT Press (2001)
5. Maslov S. Yu.: Theory of Deductive Systems and its Applications (in Russian). Radio i svyaz, Moscow (1986)
6. VUE http://vue.tufts.edu/about/index.cfm