On a Multi-language Computer Support of a Human Mathematical Activity

Elena Glazunova[0000−0002−0136−4936], Bella Golub[0000−0002−1256−6138], and Alexander Lyaletski[0000−0003−0370−5041]

National University of Life and Environmental Sciences of Ukraine, Ukraine
o-glazunova@nubip.edu.ua bellalg@nubip.edu.ua
a.lyaletski@nubip.edu.ua

Abstract. The work being carried out on the development of Ukrainian and Russian versions of the ForTheL formal natural language, which is the input language of the SAD system for automated deduction (http://nevidal.org/sad.en.html) and simulates the structure of sentences of ordinary English, is described. The implementation of these versions will permit to do a remote access to the SAD system for solving tasks of theorem proving and mathematical text verification for a user who speaks only one of the three languages (English, Russian or Ukrainian). This confirms a perspectivity of providing human mathematical activity with a computer support in languages that, on the one hand, are formal, but, on the other hand, are as close to the languages used by people in their daily practice as possible.

Keywords: Formal natural language, ForTheL language, SAD system, Computer support, Theorem proving, Mathematical text verification

The emergence of the Internet and cloud technologies has reinforced the urgency of the problem of multi-language support of a remote human interaction with a particular computer system. This also applies to various automated reasoning systems that are either already hosted or, potentially, can be hosted in the Internet. As a rule, communication with such systems is performed in a single language, most often in English. However, it is obvious that along with an existing input language of a system, it is desired to have an ability to do such interaction with the system in other languages that are, from a certain point of view, more user-friendly than the existing one.

Existing approaches to solving this problem are usually reduced to the problem of computer understanding of an ordinary human language, which has not yet found a good enough practical solution. That is why the approach proposed by V.M. Glushkov [1] and consisting in the creation of *formal natural languages* (that is, in the creation of such languages that, on the one hand, are uniquely understood by a computer, and on the other hand, are as close to the languages used by people in their daily practice as possible) seems very suitable.

1. An example of a successful enough actions in this direction is the ForTheL language [2] of the SAD system (i.e., [3,4,5]) accessible at "http://nevidal.org/sad.en.html" and intended for theorem proving and verification of self-contained

mathematical texts presented in ForTheL, the structure of sentenses of which simulates the structure of ordinary English sentences. Here, the *description of actions aimed at creating Ukrainian (uaForTheL) and Russian (ruForTheL) versions of the ForTheL language that have the same properties as ForTheL* is given.

Suppose that we wanted to write in ForTheL a proof of the proposition that S is the empty set if and only if S is a subset of any set, and did so in the form of the ForTheL-text, located at the page "http://nevidal.org/cgi-bin/sad.cgi?ty=txt&ln=en&link=small/emptyset.ftl", a part of which is given below and which, being put in a mathematical environment containing all necessary notions and propositions, are easily verified by the SAD system. Leaving aside the verification process itself, we pay attention to the natural presentation of the verified text having the form of an ordinary English text.

```
Definition DefSubset. A subset of S is a set T such that every element of T
   belongs to S.
Definition DefEmpty. S is empty iff S has no elements.
Axiom ExEmpty. There exists an empty set.
Proposition. S is a subset of every set iff S is empty.
Proof.
   Case S is empty. Obvious.
   Case S is a subset of every set.
      Take an empty set E.
      Let z be an element of S.
      Then z is an element of E.
      We have a contradiction.
   end.
qed.
```

Fig.1. ***ForThel-text***

2. Such closeness of a ForTheL-text to an ordinary English text is explained by the fact that the syntax of a ForTheL sentence follows the rules of the English grammar. It is based on the notion of a section.

ForTheL sections are: sentences, sentences with proofs, cases, and top-level sections: axioms, signature extensions, definitions, lemmas, and theorems. A top-level section is a sequence of assumptions concluded by an affirmation.

There exist three kinds of sentences in ForTheL: assumptions, selections, and affirmations. Assumptions serve to declare variables or to provide some hypotheses for a subsequent text. For example, "Any subset of any set is a set." and "Assume that m is less than n." are typical assumptions. Selections state the existence of representatives of notions and also can be used to declare variables. An example of a selection is: "Take an even prime number N.". Affirmations are simply statements, e.g. "If p divides n+p then p divides n".

Sentences are constructed from such syntactical units as statements, predicates, notions (that denote classes of objects) and terms (that denote individual entities). Units are composed of syntactical primitives: nouns which form notions (e.g., "subgroup of") or terms (e.g., "closure of"), verbs, and adjectives, which form predicates (such as "belongs to", "compact" and others), symbolic primitives using a concise symbolic notation for predicates and functions and allowing to construct usual first-order formulas in the form of ForTheL-statements. Naturally, just a little fragment of English is formalized in the syntax of ForTheL.

Like a usual mathematical text, a ForTheL-text consists of definitions, assumptions, affirmations, theorems, proofs, etc. (see details in [6]).

3. The proposed languages uaForTheL and ruForTheL have a similar structure and grammars admitting, in particular, the translation of uaForTheL- and ruForTheL-texts into first-order language formulas. As a result, the current versions of uaForTheL and ruForTheL allow one to write the above-given text as follows (giving the ability to set tasks for the SAD system in different languages with the possibility to translate them from one into another language in the case of the implementation of uaForTheL and ruForTheL).

Визначення DefSubset. Підмножина S є множиною T такою що кожен
  елемент T належить S.
Визначення DefEmpty. S є пустою множиною iff S не має елементів.
Аксіома ExEmpty. Існує пуста множина.
Твердження. S є підмножина будь-якої множини iff S є пуста множина.
Доведення.
  Випадок коли S є пуста множина. Очевидно.
  Випадок коли S есть є підмножина будь-якої множини.
    Візьмемо E у якості пустої множини.
    Нехай z є елемент S.
    Тоді z є елемент E.
    Протиріччя.
  Кінець випадку.
Кінець доведення.

Fig.2. ***uaForTheL-text***

Определение DefSubset. Подмножество S есть множество T такое что
  каждый элемент T принадлежит S.
Определение DefEmpty. S есть пустое множество iff S не имеет элементов.
Аксиома ExEmpty. Существует пустое множество.
Предложение. S есть подмножество любого множества iff S есть пустое
  множество.
Доказательство.
  Случай когда S есть пустое множество. Очевидно.
  Случай когда S есть подмножество любого множества.
    Возьмем E в качестве пустого множества.
    Пусть z есть элемент S.
    Тогда z есть элемент E.
    Противоречие.
  Конец случая.
Конец доказательства.

Fig.3. ***ruForTheL-text***

We see that the uaForTheL- and ruForTheL-texts being formal can be considered as texts written in ordinary Ukrainian and Russian. Therefore, we can say that we have at least partially achieved our aim. Besides, there appears the possibility to construct the next bidirectional translators from one formal natural language to another: ForTheL-texts ↔ uaForTheL-texts, ForTheL-texts ↔ ruForTheL-texts, and uaForTheL-texts ↔ ruForTheL-texts, which leads to the multi-language interface both with the SAD system and with a computer service requiring a user-friendly interaction in different languages.

The above-said shows that on the basis of the proposed approach, it is possible to achieve a sufficiently good solution of the problem under consideration. Additionally, it can be noted that the outlined approach can be used not only in the case of solving tasks of automated theorem proving and verification of mathematical texts, presented in different languages, but also in the case of a multi-language support of (e-)learning and testing a knowledge gained by a person in the process of studying mathematical disciplines.

Finally, the authors hope that this research will give an impulse to the development of ForTheL-like languages and lead to the creation of an info-structure for the remote multilingual presentation and complex processing of mathematical knowledge and it will be useful in both academic and teaching daily activities of a person.

References

1. Glushkov, V. M.: Some problems in automata theory and artificial intelligence. Cybernetics and System Analysis, **6**(2), Springer, 17-27 (1970).
2. Vershinin, K., Paskevich, A.: ForTheL — the language of formal theories. International Journal of Information Theories and Applications, **7**(3), 120-126 (2000).
3. Lyaletski, A., Verchinine, K., Degtyarev, A., Paskevich, A.: System for Automated Deduction (SAD): linguistic and deductive peculiarities. Advances in Soft Computing: Intelligent Information Systems 2002 — Proceedings of the IIS'2002 Symposium, Sopot, Poland, Physica-Verlag, 413-422 (2002).
4. Lyaletski, A., Paskevich, A., Verchinine, K.: SAD as a mathematical assistant — how should we go from here to there? Journal of Applied Logic, **4**(4), 560–591 (2006).
5. Verchinine, K., Lyaletski A., Paskevich A.: System for Automated Deduction (SAD): a tool for proof verification. Lecture Notes in Computer Science, vol. 4603, 398-403 (2007).
6. ForTheL Reference: http://www.nevidal.org/dload.en.html