The SAD System: a Current State and Future Work

Alexander Lyaletski[a], Alexandre Lyaletsky[b], and Konstantin Verchinine[c]

[a]Kyiv National University of Trade and Economics, Ukraine
[b]Taras Shevchenko National University of Kyiv, Ukraine
[c]Universite Paris-Est Creteil Val de Marne, France
forlav@mail.ru foraal@mail.ru karadagger@gmail.com

Abstract. The purpose of this communication is to outline briefly the current state of the so-called SAD system (System for Automated Deduction), present its architecture, and discus some possible ways of its further development. At that, the main attention is paid to the development of multiple language support of the SAD system as well as to the construction of SAD proof search methods, toolkit, and engine for making deduction in different first-order logics.

Keywords: Evidence Algorithm, SAD system, TL language, ForTheL language, TPTP library, automated reasoning, automated theorem proving, mathematical text verification, prover, computer algebra system

Key Terms: MachineIntelligence

1 Introduction

The current version of the System for Automated Deduction (SAD system) was designed and implemented in the framework of the so-called evidential paradigm [1] intended for the presentation and complex processing of formal mathematical knowledge [2] and being a current vision of the Evidence Algorithm programme (EA) initiated by V.M. Glushkov [3][1].

By now there were implemented two versions of the SAD system: the Russian and English versions. Firstly the Russian SAD system appeared (it was announced in 1978 and published in [6]) and much later, in 2002, its English modification was first presented at the IIS'2002 symposium [7]. English SAD can be considered as the further development of ideas laid in Russian SAD oriented to automated theorem proving and possessing such additional property as the ability to verify formalized mathematical texts. Also note that opposite to Russian SAD having a restricted (formal) Russian as its input language (called TL [8]), English SAD is equipped with a restricted formal English as its input language (called ForTheL [9]).

[1] A detailed enough description of most of the investigations made in the EA framework can be found in [4] (see also [5]).

The current version of the English SAD system is implemented on the Linux platform and can be reached via Internet (http://nevidal.org/sad.en.html).

The objective of this communication is to outline briefly the current state of the English SAD system and discus some of the possible ways of the development of its linguistic and deductive tools.

2 SAD System: a Current State

While building the English SAD system, the architecture given in Fig. 1 and reflecting the current state of English SAD was designed. Note that in the design process, the objective was to construct a system able to accept and analyze formalized natural texts, translate them into first-order formulas, and, after this, solve the automated theorem proving/verification tasks by using a native prover or one of the famous first-order provers and/or computer algebra systems.

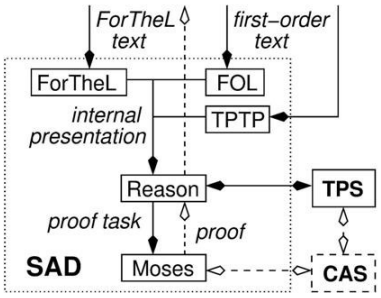

Fig. 1. SAD architecture

This architecture can be considered as a tree level structure containing internal (native) linguistic, reasoning, and deductive modules and having possibility to use external theorem-proving (TPS) and computer algebra (CAS) systems.

At the *first (linguistic) level*, the parser module (ForTheL) first analyzes an input ForTheL-text, its structure defined with the help of ForTheL markups, and its logical content encoded in ForTheL-statements. After this, it translates the text into its internal presentation. The result of translation gives a series of goal statements for deducing them from their predecessors. FOL denotes a parser for a "dialect" of the first-order language, which can be used for solving the task of establishing the deducibility of a first-order formula/sequent in classical logic in the case of necessity. The module TPTP provides the ability to connect with the famous library TPTP (Thousand Problems for Theorem Provers) [10], if a SAD user will decide to try to solve one of the TPTP problems.

At the *second (reasoning) level*, the goal statements are processed one-by-one by the foreground reasoner Reason. This module is intended to reduce a given proof task to a number of subtask for a prover. It works in a dialog with the prover: It may split the main goal to several simpler subgoals or propose an alternative subgoal. This module becomes redundant when English SAD solves a task connected with automated theorem proving.

Inference search tasks are resolved by the background native prover Moses at the *third (deductive) level*. Moses is based on a special goal-driven sequent calculus for classical first-order logic with equality. The original notion of an admissible substitution used in the calculus permits to preserve the initial signature of a task under consideration so that accumulated equations can be sent to a specialized solver, e.g. an external computer algebra system. Note that English SAD was implemented in such a way that at present time it can be connected with one of first-order prover, such as Otter [11], SPASS [12], or Vampire [13].

English SAD reports the result of its work after completing it. The protocol of its successful or unsuccessful work can be printed as desired by a user.

3 SAD System: a Future Work

The above-given description of the English SAD system demonstrates that the system has original possibilities and satisfies the existing approaches and requirements to intelligent computer services intended for the complex processing of formalized mathematical knowledge. But its trial operation as well as a number of investigations made in automated reasoning last years have shown the desirability and possibilities of improving the capabilities of English SAD in the following directions (studied and not implemented).

On the linguistic level. The nearest objective is to incorporate the existing Englsh ForTheL language into the LaTeX-environment in order to reach the reading of ForTheL-texts in the form closest to usual mathematical texts. (Now this task is under consideration.) Besides, there are drafts of the Russian and Ukrainian versions of the (English) ForTheL language. Therefore, there exists the possibility to construct the next bidirectional translators: English ForTheL-texts $\leftrightarrow$ Russian ForTheL-texts, English ForTheL-texts $\leftrightarrow$ Ukrainian ForTheL-texts, and Russian ForTheL-texts $\leftrightarrow$ Ukrainian ForTheL-texts, which will give the opportunity for using such a multilingual extension of English SAD by a person who knows only one or two of three just-mentioned languages, as well as for making automatic translation of a ForTheL-text written in one of these languages into a ForTheL-text written in another. (Of course, one can try to construct a French, German, and/or other version of ForTheL language, thereby strengthening such a multilingual SAD component.)

On the reasoning level. It is planned to increase the heuristic possibilities of the system by incorporating in it the human-like reasoning methods depending on the subject domain under consideration concentrating main attention on inductive theorem proving methods. Besides, tools for interfacing with some of the famous computer algebra systems are going to be developed and implemented.

On the deductive level. On the basis of the research made on computer-oriented proof search in classical and non-classical sequent logics (see, for example, [14]), one can try to construct a toolkit giving the possibility to "puzzle" one or another ("native") proof search method depending on a desire of a SAD user or a subject domain under consideration. (This possible feature of such an extended system will play an important role in the case, when the application

of non-classical reasoning becomes necessary element for successful decision of a task under consideration.)

Finally, the authors hope that the described development of the English SAD system will lead to the creation on its basis of an info-structure for the remote multilingual presentation and complex processing of mathematical knowledge and it will be useful in both academical and teaching daily activity of a person.

References

1. A. Lyaletski and M. Morokhovets. Evidential paradigm: a current state. *Programme of the International Conference "Mathematical Challenges of the 21st Century"*. University of California, Los Angeles, USA, P. 48, 2000.
2. A. Lyaletski, A. Lyaletsky, and A. Paskevich. Evidential paradigm as formal knowledge presentation and processing, *Proceedings of the 12th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI 2016)*, Kyiv, Ukraine, P. 25-32, 2016.
3. V. M. Glushkov. Some problems in automata theory and artificial intelligence. *Cybernetics and System Analysis*, Vol. 6, No. 2, Springer, P. 17-27, 1970.
4. A. Lyaletski, M. Morokhovets, and A. Paskevich. Kyiv school of automated theorem proving: a historical chronicle. In book: *Logic in Central and Eastern Europe: History, Science, and Discourse*, University Press of America, USA, P. 431-469, 2012.
5. A. Lyaletski and K. Verchinine. Evidence Algorithm and System for Automated Deduction: A retrospective view (In honor of 40 years of the EA announcement). *Lecture Notes in Computer Science: Intelligent Computer Mathematics*, Vol. 6167, P. 411-426, 2010.
6. Yu. V. Kapitonova, K. P. Vershinin, A. I. Degtyarev, A. P. Zhezherun and A. V. Lyaletski. System for processing mathematical texts. *Cybernetics and System Analysis*, 15(2), Springer, P. 209-210, 1979.
7. A. Lyaletski, K. Verchinine, A. Degtyarev, and A. Paskevich. 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, P. 413-422, 2002.
8. K. P. Vershinin. Remarks on formal languages for writing proofs. *Cybernetics and System Analysis*, 8(5), Springer, P. 790-792, 1972.
9. K. Vershinin and A. Paskevich. ForTheL — the language of formal theories. *International Journal of Information Theories and Applications*, 7(3), 2000, P. 120-126.
10. G. Sutcliffe, C. B. Suttner, and T. Yemenis. The TPTP problem library. *Lecture Notes in Computer Science: Automated Deduction — CADE-12)*, Vol. 814, Springer, P. 252-266, 1994.
11. Otter prover: https://www.cs.unm.edu/mccune/otter/
12. SPASS Homepage: http://www.spass-prover.org/
13. Vampire's Home Page: http://www.vprover.org/
14. A. Lyaletski. Mathematical text processing in EA-style: a sequent aspect. *Journal of Formalized Reasoning (Special Issue: Twenty Years of the QED Manifesto)*, Vol. 9, No. 1, P. 235-264, 2016.