System Description: Interface between Theorema and External Automated Deduction Systems*

Temur Kutsia and Koji Nakagawa

Research Institute for Symbolic Computation
Johannes Kepler University Linz
A-4040, Linz, Austria
{kutsia,nakagawa}@risc.uni-linz.ac.at

Software Competence Center Hagenberg
Hauptstrasse, 99
A-4232, Hagenberg, Austria
{teimuraz.kutsia,koji.nakagawa}@scch.at

Abstract. The interface between the Theorema system and external automated deduction systems is described. It provides a tool to access external provers within a Theorema session in the same way as "internal" Theorema provers. Currently 11 external systems are supported. The design of the interface allows combining external systems with each other as well as with "internal" Theorema provers.

1 Introduction

The Theorema system [7] is an integrated environment for proving, solving and computing built on the top of mathematical software system Mathematica [26]. It was designed by Bruno Buchberger [5, 6] and provides a front end for composing formal mathematical text consisting of a hierarchy of axioms, definitions, propositions, algorithms etc. in a common logic frame with user-extensible syntax, a library of both well-established and new provers, solvers and simplifiers for proving, solving and simplifying mathematical formulae.

The interface described in this paper implements a link providing Theorema users with a tool for using automated deduction systems ("external" systems) within Theorema session in the same way as "internal" Theorema provers. Currently, the following 11 external systems are supported: provers - Bliksem [19], EQP [17], E [21], Gandalf [23], Otter [16], Scott [12], Setheo [15], Spass [24], Vampire [20], Waldmeister [4] and a finite model and counter-example searcher Mace [18].

The interface consists of two types of links - direct and indirect - from Theorema to an external system. The indirect link is established with Setheo, Scott and Waldmeister - using first an intermediate translation into TPTP [22] format, and then the TPTP2X converter which converts TPTP format files into a format of the specified external system. All the other provers are linked directly to Theorema, without any intermediate routine, translating input from Theorema

* Supported by the Austrian Science Foundation (FWF) under Project SFB F1302 and by Software Competence Center Hagenberg (Austria) under MathSoft project.

syntax directly into the syntax of an external system. It is relatively easy to add an indirect link to a new prover, but is has also disadvantages compared with the direct link - the user has less control over the system and more intermediate routines are needed.

All the external provers can be used as black box provers within Theorema session, without translating their output into Theorema syntax. The link to Otter has an additional feature - back translator, which transforms an Otter proof into Theorema format. The design of the interface makes it possible to combine various external systems with each other or with Theorema provers.

2 Direct Link

The interface implements two types of direct links from Theorema to external systems: black box style and white box style links. A black box style direct link consists of two parts: the translator component and the linking component. The components and the sequence of operations are illustrated in Figure 1. A white box style direct link consists of the translator, the linking component and the back translator. The architecture is shown in Figure 2.

The black box style link works as follows: first, the translator gets the Theorema goal, knowledge base and options and translates them into the prover format thus preparing the input for the prover call. The linking component gets the translated string and options, writes the string in a temporary file and calls the prover with the options. Finally, the prover output is passed back to Theorema. The user has a full control over the external system using prover options.

The sequence of operations performed by the white box style direct link is the same as those of the black box style link until the external prover output is passed back. Here the back translation component gets the prover output, transforms it into Theorema syntax and places into the Theorema proof object. Finally, the Theorema style proof is shown in the proof notebook. Note that if the prover failed to prove the goal, the back translation component does not affect the output and the link behaves like a black box style link. Currently Theorema has black box style links with Otter, Spass, EQP, Gandalf, Bliksem, Vampire,

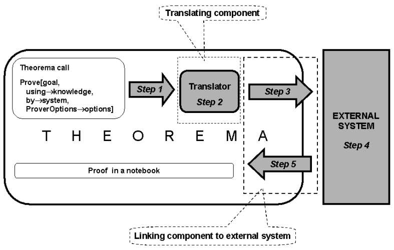

Fig. 1. Black box style direct link

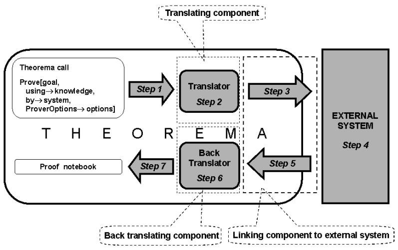

Fig. 2. White box style direct link

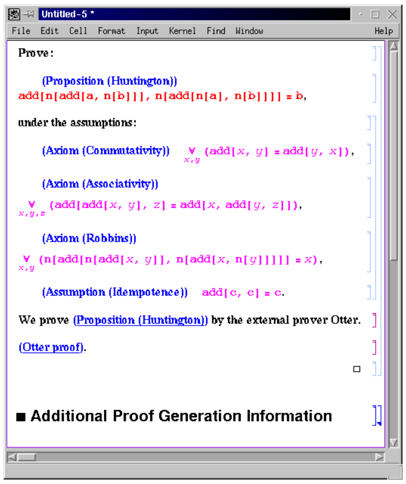

Fig. 3. Black box style proof notebook

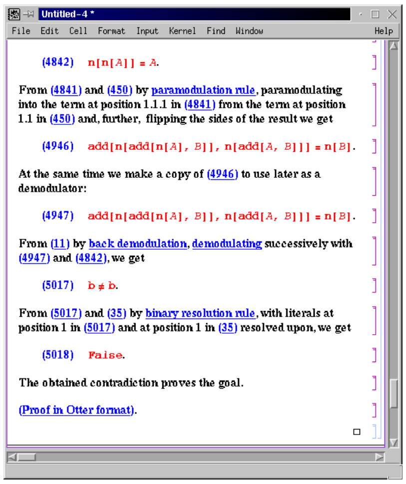

Fig. 4. White box style proof notebook

E and Mace and the white box style link with Otter only. Thus, Otter has both types of direct links to Theorema, with the black box style link as the default option. The notebooks on the figures above show the black box and the white box outputs of the Otter call to prove that every Robbins algebra satisfying idempotence of addition is a Boolean algebra ([25]). In both cases, the output in Otter syntax can be shown in a new notebook which pops up if one clicks on the corresponding hyperlink in the proof notebook. In the back translated output each inference rule used in the proof has a hyperlink to the definition of it.

3 Indirect Link

All indirect links from Theorema to external systems are of the black box style. An indirect link consists of three parts: the translator into TPTP format, the linking component to TPTP2X and the linking component to the external prover. We have chosen the TPTP format for the intermediate translation because of the tptp2X converter, which can convert a TPTP format file into a format of many automated provers. The indirect link works as follows: the Theorema `Prove` call invokes the translator into TPTP format which translates the goal and knowledge base into TPTP format and puts the result into a temporary file. Next, the linking component to TPTP2X calls the script tptp2X provided with the TPTP library on the file with the command line options relevant to the prover. The obtained result is passed to the linking component to the prover together with the options. The linking component calls the prover, gets the result back and locates it into the Theorema proof object. It is relatively easy to add an indirect link to a new prover, but is has also disadvantages comparing with the direct link - the user has less control over the system and an intermediate translation/linkage is needed. In future, We intend to replace indirect links with direct ones. The indirect link is established with Setheo, Scott and Waldmeister.

4 Combination of Systems

The design of the interface allows combining various external systems with each other or with "internal" Theorema provers in the similar way as the "internal" provers can be combined with each other. To demonstrate this capability we combined Otter and Mace into the Theorema user prover OtterMace (not the prover from Argonne), which first runs Otter on a problem, and if it fails to prove the goal, Mace tries to find a countermodel. Another example of such a combination is the Theorema experimental user prover PLOtter, which combines Otter and the predicate logic prover of Theorema.

5 Related Work and Conclusion

The problem of integrating software/reasoning systems has been studied intensively in the recent years (e.g. [2], [3], [10], [11], [13], [14]). There are several platforms which try to provide a general solution to the problem ([1], [8], [9]). The interface described in this paper is not an attempt to provide such a general solution. We tried to make an interface which is simple, easy to implement, maintain and use and which links reasoning systems directly, without intermediate routines (we still have indirect links for experimental reasons but we intend to replace them with direct ones). The interface allows using external systems in a Theorema session as "internal" Theorema provers.

References

1. Armando, A. and Zini, D. Towards Interoperable Mechanized Reasoning Systems: the Logic Broker Architecture. In: Proc. of the AI*IA-TABOO Joint Workshop "Dagli Oggetti agli Agenti", Parma, Italy, May 29-30 (2000).
2. Ballarin, C., Homann, K. and Calmet, J. Theorems and Algorithms: An Interface between Isabelle and Maple. In: Proc. of Int. Symposium on Symbolic and Algebraic Computation, Berkeley, USA (1995) 150-157.
3. Ballarin, C. and Paulson, L. A Pragmatic Approach to Extending Provers by Computer Algebra - with Applications to Coding Theory. Fundamenta Informaticae **34** (1999) 1-20.
4. Buch, A. and Hillenbrand, Th. Waldmeister: Development of a High Performance Completion-Based Theorem Prover, SR-**96-01**. Universitaet Kaiserslautern, Germany (1996).
5. Buchberger, B. Symbolic Computation: Computer Algebra and Logic. In: F. Baader and K.U. Schulz (eds.), Frontiers of Combining Systems, Kluwer Academic Publishers (1996) 193-220.
6. Buchberger, B. Using Mathematica for Doing Simple Mathematical Proofs. Invited paper in: Proc. of the 4th Mathematica Users' Conference, Tokyo, November 2, 1996, Wolfram Media Publishing (1996) 80-96.

7. B. Buchberger, B., Dupre, C., Jebelean, T., Kriftner, F., Nakagawa, K., Vasaru, D. and Windsteiger, W. Theorema: A Progress Report. In: M. Kerber and M. Kohlhase (eds.), Proc. of the 8th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning, St. Andrews, Scotland, August 6-7 (2000) 100-115.
8. Dennis, L.A., Collins, G., Norrish, M., Boulton. R., Slind, K., Robinson, G., Gordon, M. and Melham, T. The PROSPER Toolkit. In: Proc. of the 6th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems, Berlin, Germany, April (2000).
9. Franke, A. and Kohlhase, M. System description: MathWeb, an Agent-Based Communication Layer for Distributed Automated Theorem Proving. In: Proc. CADE-16, H. Ganzinger (ed.), LNAI **1632**, Springer (1999) 217-221.
10. Giunchiglia, F., Pecchiari, P. and Talcott, C. Reasoning Theories: Towards an Architecture for Open Mechanized Reasoning Systems. In: F. Baader and K.U. Schulz (eds.), Frontiers of Combining Systems, Kluwer Academic Publishers (1996).
11. Harrison, J. and Thery, L. A sceptic's Approach to Combining HOL and Maple. J. Automated Reasoning **21(3)** (1998) 279-294.
12. Hodgson, K. and Slaney, J. Semantic Guidance for Saturation-based Theorem Proving, TR-**ARP-04-2000**, Automated Reasoning Project, Australian National University, Canberra, Australia (2000).
13. Homann, K. and Calmet, J. Combining Theorem proving and Symbolic Mathematical Computing. In: J. Calmet, J.A. Campbell (eds.), Proc. of 2nd Int. Conference on Artificial Intelligence and Symbolic Mathematical Computing, LNCS **958**, Springer (1995) 18-29.
14. Kerber, M, Kohlhase, M. and Sorge, V. Integrating Computer Algebra into Proof Planning. J. Automated Reasoning **21(3)** (1998) 327-355.
15. Letz, R., Schumann, J., Bayerl, S. and Bibel, W. SETHEO: A High-Performance Theorem Prover. J. Automated Reasoning, **8(2)** (1992) 183-212.
16. McCune, W. Otter 3.0 Reference Manual and Guide. **ANL-94/6**, Argonne National Laboratory, Argonne, USA (1994).
17. McCune, W. EQP (1999) http://www.mcs.anl.gov/AR/eqp/
18. McCune, W. MACE (2000) http://www.mcs.anl.gov/AR/mace/
19. de Nivelle, H. Bliksem (1999) http://www.mpi-sb.mpg.de/~bliksem/
20. Riazanov, A. and Voronkov, A. Vampire, In: Proc. CADE-16, H. Ganzinger (ed.), LNAI, **1632**, Springer, (1999) 292-296.
21. Schulz, S. E Equational prover (2000) http://wwwjessen.informatik.tu-muenchen.de/~schulz/WORK/
22. Suttner, C. and Sutcliffe, G., The TPTP Problem Library (TPTP v2.1.0), TR-**AR-97-01**, Institut fuer Informatik, Technische Universitaet Muenchen, Germany (1997).
23. Tammet, T. Gandalf (1997) http://www.cs.chalmers.se/~tammet/gandalf/
24. Weidenbach, C., Afshordel, B., Brahm, U., Cohrs, C., Engel, T., Keen, E., Theobalt, C. and Tpoic, D. System Description: SPASS version 1.0.0, In: Proc. CADE-16, H. Ganzinger (ed.), LNAI **1632**, Springer (1999) 378-382.
25. Winker, S. Robbins Algebra: Conditions That Make a Near-Boolean Algebra Boolean, J. Automated Reasoning **6(4)** (1990) 465-489.
26. Wolfram, S. The Mathematica Book. Fourth edition. Wolfram Media and Cambridge University Press (1999).