

Towards a Translation of Computer Algebra Algorithms into Tactics

Volker Sorge
Fachbereich Informatik
Universität des Saarlandes
D-66041 Saarbrücken, Germany
sorge@cs.uni-sb.de
<http://jswww.cs.uni-sb.de/~sorge>

1 Introduction

Mechanized reasoning systems (MRS) and computer algebra systems (CAS) apparently have different objectives. Their integration is, however, highly desirable, since both different tasks, proving and calculating, have to be performed in many formal proofs. Consequently several concepts of combining these systems have recently been developed, see [CZ92, HT93, BHC95] for examples. However using CAS for term rewriting leads to proofs that are no longer completely within the ND-calculus and thus no longer verifiable by a simple proof checker. Furthermore we might obtain incorrect proofs if there are any computational errors in an algorithm¹.

The SAPPER-system [Sor96], presented in section 2 of this extended abstract, uses a different approach by exploiting the mathematical knowledge which is implicitly contained in the algorithms of a CAS. It extracts proof plans that correspond to the mathematical computation of an algorithm using a simple tactic mechanism. In section 3 the author proposes a combination of a meta-theoretical framework for SAPPERS tactics with a formal framework for computer algebra algorithms in order to translate algorithms to tactics.

2 SAPPER

SAPPER (System for Algorithmic Proof Plan Extraction and Reasoning) [KKS96] can be seen as a generic interface connecting the MRS Ω -MKRP with one or several CAS (cf. figure 1). The interface itself can be roughly divided into two parts: the *translation part*, and the *tactic generator*. The former performs syntax translations between Ω -MKRP and a CAS in both directions while the latter only transforms verbose output of the CAS to Ω -MKRP proof plans. The interface minimizes the required changes to an existing CAS, while maintaining the possibility of using the CAS stand-alone. The only requirement we make for integrating a particular CAS is that it has to produce enough protocol information so that a proof plan can be generated from this information. The proof plan in turn can be expanded into an ND-proof verifying the actual computation.

¹Here and in the rest of this paper the term algorithm always refers to computer algebra algorithms.

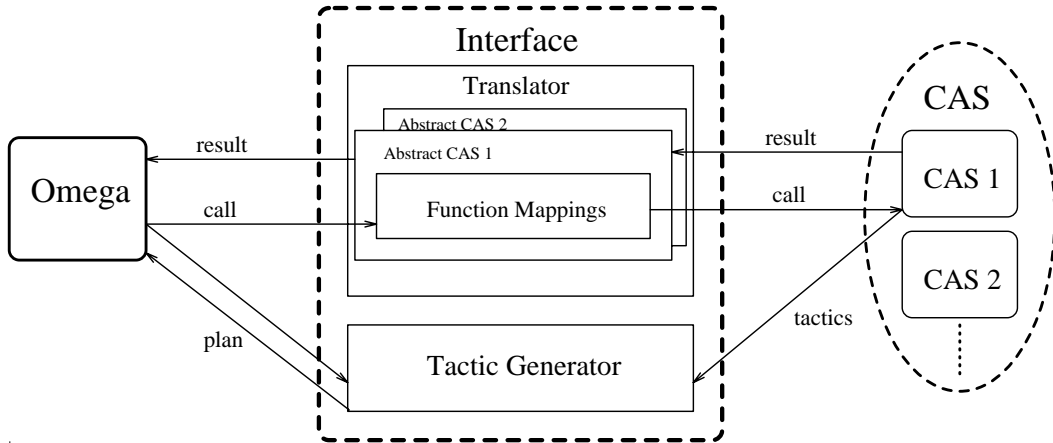


Figure 1: System architecture of SAPPER

Unlike other proof planners a CAS does not have to search for a plan but only to assemble one as the algorithms have implicit knowledge of the actual computation. Thus SAPPER can use a relatively simple tactic mechanism for constructing proof plans. It consists of a set of hierarchically structured tactics:

- *simple tactics* corresponding to the application of one hypotheses in a proof.
- *complex tactics* describing computational steps of computer algebraic algorithms; they are compositions of simple tactics with tacticals.

3 Towards a 1-to-1 Representation of Tactics and Algorithms

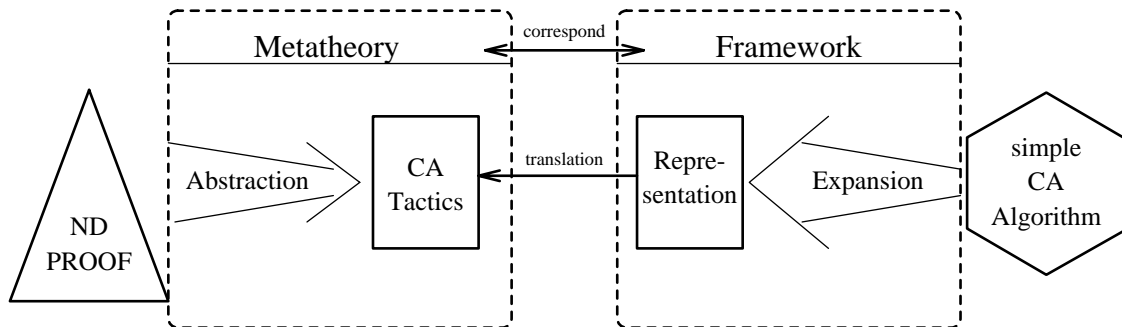


Figure 2: Translations from tactics to algorithms

The author has realized a first working implementation of the SAPPER-system connecting Ω -MKRP with a small prototypical CAS capable of handling polynomial manipulations. But in order to build a stronger system it would be necessary both to add more algorithms and to extend the set of tactics available for them. Doing this for every algorithm separately would not only require time but it would also be very tedious work. Therefore it is worth considering the automation of this process.

Figure 2 shows an overall scenario for translating simple computer algebra algorithms into corresponding tactics and thus obtaining calculus-level proofs. The goal is to develop a framework in which algorithms can automatically be expanded to a representation which in turn can be translated into tactics.

On the one side it is already known that tactics can be handled in a metatheory by abstracting them from actual natural deduction proofs [GT92]. In this metatheory it is not only possible to reason about tactics but also to compile program code from them directly [GT94]. On the other side there is a necessity to develop rules for an expansion of simple computer algebra algorithms so they can be represented on the same level of abstraction as tactics can be. As the same step of computation does not always correspond to the same inference steps in a proof some reconstruction of the logical computation has to be done inside the tactics. Therefore several instances of the computation would have to be considered in order to represent an algorithm in an expanded way.

4 Conclusion and Future Work

SAPPER has already been successfully applied to economical optimization problems and has thus shown the usefulness of the approach. Therefore upgrading the system using a larger scaled CAS would be advisable. Automatizing the process of developing new tactics for new algorithms would dramatically enlarge the amount of algorithms available in such a system.

It would be necessary to carry out some case studies on algorithms in order to develop a framework as proposed by the author in section 3. Discovering parallels between algorithms and their corresponding tactics could lead to rules for an expansion of algorithms and thus to general patterns of translating algorithms into tactics.

References

- [BHC95] C. Ballarin, K. Homann, and J. Calmet. Theorems and Algorithms: An Interface between Isabelle and Maple. In A. H. M. Levelt, editor, *Proceedings of International Symposium on Symbolic and Algebraic Computation (ISSAC'95)*, pages 150–157. ACM Press, 1995.
- [CZ92] E. Clarke and X. Zhao. Analytica-A Theorem Prover in Mathematica. In *Automated Deduction-CADE-II*, pages 761–763, 11th International Conference on Automated Deduction, Saratoga Springs, New York, 15-18 Juni 1992.
- [GT92] F. Giunchiglia and P. Traverso. A Metatheory of a Mechanized Object Theory. Technical Report 9211-24, IRST, 1992.
- [GT94] F. Giunchiglia and P. Traverso. Program Tactics and Logic Tactics. In *LPAR'94, 5th International Conference on Logic Programming and Automated Reasoning*, Kiev, Ukraine, July 16-21 1994.
- [HT93] J. Harrison and L. Théry. Reasoning About the Reals: The Marriage of HOL and Maple. In A. Voronkov, editor, *Proceedings of the 4th International Conference on Logic Programming and Automated Reasoning (LPAR'93)*, volume 698 of *LNAI*, pages 351–353, St. Petersburg, Russia, July 1993. Springer Verlag.
- [KKS96] M. Kerber, M. Kohlhase, and V. Sorge. Integrating Computer Algebra with Proof Planning. To appear in: DISCO96 Conference Proceedings, 1996.
- [Sor96] V. Sorge. Integration eines Computeralgebrasystems in die logische Beweisumgebung Ω -MKRP. Master's thesis, Universität des Saarlandes, 1996. To appear.