# Automated Provers doing (Higher-Order) Proof search: A Case Study in the Verification of Pointer Programs

Farhad Mehta [a] Silvio Ranise [b]

[a] *Institut für Informatik, Technische Universität München (Germany),* mehta@in.tum.de

[b] *LORIA & INRIA-Lorraine, Nancy (France),* ranise@loria.fr

**Abstract**

We would like to present results obtained after doing a case study on the possibilities of doing proof search in a higher-order logic using existing automated proof tools. A commonly occurring type of proof obligation necessary to prove the correctness of the Schorr-Waite algorithm in the interactive prover Isabelle/HOL is given as a problem to the automated prover haRVey. Preliminary experimental results are encouraging.

In [10], Shankar argues that decision procedures for various decidable fragments of a higher-order Logic (HOL) can be put to productive use in the context of interactive theorem provers. Among others, he advocates the use of "ground decision procedures" in PVS [8]. A ground decision procedure is a procedure capable of deciding the satisfiability problem for ground formulae in some (decidable) theory of first-order logic (FOL). According to the seminal work of Boyer and Moore [2], Shankar remarks that integrating decision procedures in more general reasoning activities (such as rewriting or finding suitable instances of quantified variables) is far from being a trivial task. The main difficulty is that proof obligations generated in many verification problems contains user-defined symbols whose meaning is not known to the available decision procedures. A naïve approach would be to treat user-defined function symbols as uninterpreted and use well-known combination schemas (such as Nelson-Oppen [5]) to build procedures for a combination of theories, decided by the available procedure and the theory of equality. Unfortunately, for some interesting verification problems such as program verification, this approach

is not sufficient and the user is required to manually include definitions or lemmas about certain user-defined symbols such that the extended proof obligation falls into the scope of the decision procedure. Needless to say, this can be a difficult and tedious activity.

Automated Theorem Provers (ATPs) for FOL based on resolution and/or superposition (such as Otter [1]) have proved to be quite successful to prove open problems in mathematics such as Robbin's conjecture for which a very efficient proof search mechanism must be available to control explosion of the search space. Furthermore, it has been repeatedly observed [11,13,12] that it is often the case that first-order reasoning is applicable for proof search in a large class of HOL proof obligations. Using automated proof tactics, the user of an interactive proof tool, in theory, has only to come up with the logically interesting, high-level structure of a proof, leaving "obvious" subgoals to be discharged by a automated tactic (possibly an external tool). Automated proof tactics that work in such a way already account for a large chunks of proofs done in such interactive proof systems.

Pushing this idea further, it could be advantageous to combine the automation of ground decision procedures with the the proof search capabilities of ATPs, all in one system. Such an integration is particularly easy to obtain in haRVey [3] which uses a superposition prover. In fact, the system implements satisfiability procedures for a variety of theories axiomatised by a set of clauses (such as lists, arrays, etc) by exhaustively applying the inference rules of the superposition calculus on the input set of ground literals and the axioms of the theory. In [1], it is proved that for many interesting theories, such a process always terminates, thereby yielding a satisfiability procedure because of the refutational completeness of the calculus. If definitions or lemmas about user-defined symbols are also available, then one can simply add these to the axioms of the theory. Even if termination is no more guaranteed, one can hope that the prover terminates in most cases. Furthermore, since the superposition calculus allows us to obtain proofs, we foresee the possibility to translate such proofs back into the interactive prover in order to certify the results of the external system.

There are several issues about using ATPs to do higher-order proof search that we would like to talk about. A central issue, closely related to the degree of success of the ATP, and maybe more importantly, the possibility of exchanging proofs between the two systems, is how proof obligations are translated from HOL to FOL. There are several requirements on the translation which require a trade-off. First, we want to design a translation whose validity is plausible. Secondly, we want the external ATP to perform well on the proof obligations generated by the translation. In particular, we need to avoid a naïve translation of higher-order constructs or some axioms (e.g. extensionality) which can drastically expand the search space of the system for FOL.

[1] http://www-unix.mcs.anl.gov/AR/otter/

Third, we would like proofs in FOL to be easily translated or replayed back into proofs in a HOL.

All these issues will not be talked about in their raw generality but rather sketched while describing our preliminary experiences in using haRVey with the interactive theorem prover Isabelle/HOL [7]. The case study is inspired from the verification of pointer programs [4]. A commonly occurring type of proof obligation in Isabelle/HOL, necessary to prove the correctness of the Schorr-Waite graph marking algorithm [9] is solved using haRVey. We feel this case study to be appropriate since its background theory contains arrays, whose behaviour in haRVey is well studied [1], along with axioms about set theory and relations from the Isabelle/HOL theory library, as well as an application oriented part, i.e. operations on pointers.

The talk will include a summary of the work done, in order to spark interest and discussion on the issues relating to the problems encountered here. Its main aim will be to give a taste of the challenges posed by interactive proof tools on their automated counterparts in the SMT-LIB community.

## References

[1] A. Armando, S. Ranise, and M. Rusinowitch. A Rewriting Approach to Satisfiability Procedures. *Info. and Comp.*, 183(2):140–164, June 2003.

[2] R.S. Boyer and J S. Moore. Integrating Decision Procedures into Heuristic Theorem Provers: A Case Study of Linear Arithmetic. *Machine Intelligence*, 11:83–124, 1988.

[3] D. Déharbe and S. Ranise. Light-Weight Theorem Proving for Debugging and Verifying Units of Code. In *Proc. of the International Conference on Software Engineering and Formal Methods (SEFM03)*, Brisbane, Australia, September 2003. IEEE Computer Society Press.

[4] F. Mehta, T. Nipkow, Proving pointer programs in higher-order logic, In Proc. CADE19 2003. http://www.in.tum.de/~nipkow/pubs/cade03.html.

[5] G. Nelson and D. C. Oppen. Simplification by cooperating decision procedures. *ACM TOPLAS*, 1(2):245–257, 1979.

[6] T. Nipkow, Structured Proofs in Isar/HOL, in: H. Geuvers, F. Wiedijk (Eds.), Types for Proofs and Programs (TYPES 2002), Vol. 2646 of Lect. Notes in Comp. Sci., Springer-Verlag, 2003, pp. 259–278.

[7] T. Nipkow, L. Paulson, M. Wenzel, Isabelle/HOL — A Proof Assistant for Higher-Order Logic, Vol. 2283 of Lect. Notes in Comp. Sci., Springer-Verlag, 2002, http://www.in.tum.de/~nipkow/LNCS2283/.

[8] S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In Deepak Kapur, editor, 11th International Conference on Automated Deduction (CADE), volume 607 of Lecture Notes in Artificial Intelligence, pages 748-752, Saratoga, NY, June 1992. Springer-Verlag.

[9] H. Schorr and W.M. Waite, An Efficient Machine-Independent Procedure for Garbage collection in various List Structures, in: CACM Aug 1967.

[10] N. Shankar. Using Decision Procedures with a Higher-Order Logic. In Proc. of Theorem Proving in Higher Order Logics, 14th International Conference, TPHOLs 2001, Edinburgh, Scotland, UK, September 3-6, 2001. LNCS 2152, Springer 200.

[11] J. Harrison, First Order Logic in Practice, In Proc. FTP 1997.

[12] J. Hurd. First-Order Proof Tactics in Higher-Order Logic Theorem Provers In Proc. STRATA 2003, Rome, Italy.

[13] J. Meng and L. Paulson. Experiments On Supporting Interactive Proof Using Resolution. In: David Basin and Michael Rusinowitch (editors), IJCAR 2004: Second International Joint Conference on Automated Reasoning (Springer LNCS , 2004), in press.