Derivation of Functional Programs In Isabelle

Abdelwaheb Ayari and David A. Basin
Max-Planck-Institut für Informatik, Im Stadtwald
D-66123, Saarbrücken, Germany
Email: {abdu, basin}@mpi-sb.mpg.de
Phone: (49) (681) 302-5435 Fax: (49) (681) 302-5401

We are interested in the general question of how formal metalogics and higher-order resolution can be used as a framework for deductive program development. This question includes how can useful calculi for program development be derived in conservative extensions of standard logics and how can derived rules be applied to build programs during their correctness proofs. We are also interested in how such a framework can be used to simplify and improve previously proposed approaches and paradigms to program development.

Our starting point here is the *deductive tableau* of Manna and Waldinger [1] which is proposed as a special kind of first-order proof system suited for the synthesis of functional programs. We have taken this calculus and shown how it can be recast as a conservative extension of higher-order logic. In particular, deductive tableau proofs construct witnessing functions for proofs of $\forall/\exists$ formulae and we show how these proofs can be faithfully simulated where higher-order metavariables stand-in for the witnessing functions and are filled out into concrete executable functions during the proof. Our Isabelle theory is, however, more general than Manna and Waldinger's framework in that it allows more flexible means of proof and program construction. For example, the explicit use of metavariables allows us to perform splitting operations on subgoals that are not possible in Manna and Waldinger's setting (this forces their use of non-clausal resolution and simplification to, essentially, operate under positively occurring conjunctions or negatively occuring disjunctions). Another example is that the use of higher-order logic and the Isabelle inductive data-type package allows us to construct recursive programs using well-founded induction where we use resolution to construct well-founded orderings during proofs (these arises in showing the termination of synthesized functions). This leads to a more general approach to induction than possible in the deductive tableau setting where inductions are restricted to a fixed axiomatized set of relations. Our experience using Isabelle in this work has been positive. Theory development (definitions, derivation of rules, supporting tactics) took only a few days and we could immediately apply it to problems of interest. We have reconstructed some of Manna and Waldinger's published examples, and in particular have synthesized many of the standard sorting algorithms (quick-sort, merge-sort, insertion sort, etc.). Note, in contrast, that although such examples had previously been published by Manna and Waldinger and their students

these proofs were all done by hand (sometimes using large macro-steps) as they lacked, for many years, an implementation.

References

[1] Z. Manna and R. Waldinger. Fundamentals of the deductive program synthesis. *IEEE Transactions on Software Engineering*, January 1992.