The Need for Humans in the Loop: Meeting the Grand Challenge of Full Functional Verification

Matt Kaufmann & J Strother Moore
Department of Computer Science
University of Texas at Austin
Austin, TX 78701
{kaufmann,moore}@cs.utexas.edu

October 22, 2010

Abstract

We consider the grand challenge of verifying any functional property that is true about a digital system component. Fully automatic tools have an important role to play, but so do people who are trained to work with interactive reasoning tools and to write specifications. Experience with ACL2 suggests that a little bit of human guidance can go a very long way. We therefore urge that some attention be given to interactive methods and user interfaces.

Keywords: interactive theorem proving, formal verification, functional verification

Formal verification tools have been usable for a long time, for appropriately trained people who cared enough to use them.

It doesn't take people any more skilled than the (good) software and hardware designers industry already hires, but they have to be skilled in other things.

The easiest way to make verification more automatic is to focus on "glorified type checkers" — engines for verifying a specific class of properties automatically. That's a laudable goal we do not mean to discourage, but the grand challenge is to be able to verify at reasonable cost any functional property that is true about a digital system component.

If a researcher wants to imagine him or herself working on that deep problem while actually working on (near) decision procedures for (almost) decidable properties, then he or she must pay careful attention to interface issues, including tolerance for syntax outside your fragment, incremental construction of the internal database (allowing for communication between the decision method and more general drivers), fast classifiers that identify subgoals that are "likely" to be worth giving to your method, ways to translate your positive conclusions into more general lemmas for use by other systems, and ways to translate your negative conclusions into requests for additional information that might be supplied by the more general host.

Even with careful focus on such issues and great engineering, we do not expect the grand challenge to be solved purely by automatic methods. We anticipate the need for a "human in the loop." While expectations on the human can be drastically reduced from those imposed by today's tools, the experience

with ACL2 leads us to believe that a little bit of human guidance can go a very long way. Furthermore, the most difficult kinds of problems for the automatic tools to solve are generally the ones where the human has the most insight anyway, leading to a natural decomposition of tasks. In a properly engineered system, the human should feel empowered by the tool, with it correctly turning his or her intuitively correct (if not technically perfect) advice into proofs. We therefore urge that some attention be given to interactive methods and user interfaces. In short, do not fall into the trap of thinking that economically viable tools have to be 100% automatic.

A major stumbling block is in having specifications. Indeed, this is often where the user is most helpful and yet where it is too easy to demand too much of him or her.

A promising development is the growth of assertion-based methodologies, in particular in hardware design but also probably in most areas of digital system development. As is well known, these present a significant opportunity for the use of formal verification tools.

While tool improvements are important in order to take advantage of this opportunity, so is the education of the next generation of software engineers and hardware designers. It will be critical to the promulgation of formal technologies that students be trained to write and reason about formal specifications.