Interactive Proof: Applications to Semantics

Gerwin Klein
NICTA, Sydney & University of New South Wales, Australia

This lecture builds on the previous introduction to interactive proof and demonstrates a specific application of interactive proof assistants: the semantics of programming languages. In particular, I will show how to formalise a small imperative programming language in the theorem prover Isabelle/HOL, how to define its semantics in different variations, and how to prove properties about the language in the theorem prover.

The emphasis of the lecture is not on formalising a complex language deeply, but to teach a number of formalisation techniques and proof strategies using simple examples. To this purpose, we will cover big- and small step semantics, typing and type safety, as well as a small machine language with compiler and compiler correctness proof.

The idea is to provide a solid basis from which to experiment with own language features and extensions.

The lecture has no prerequisite for a deep understanding of semantics or interactive theorem proving. The material on semantics is similar in scope to the languages presented by Nielson & Nielson [1,2]. The basics of interactive proof are covered by a separate lecture at the Summer School. A good introduction is the Isabelle tutorial [3].

References

[1] H. R. Nielson, F. Nielson. *Semantics with Applications: A Formal Introduction.* John Wiley & Sons, Inc., New York, NY, USA, 1992.

[2] H. R. Nielson, F. Nielson. *Semantics with Applications: An Appetizer.* Springer, 2007.

[3] T. Nipkow, L. Paulson, M. Wenzel. *Isabelle/HOL – A Proof Assistant for Higher-Order Logic.* LNCS 2283, Springer, 2002.