Mechanized Semantics, with Applications to Program Proof and Compiler Verification

Xavier Leroy
INRIA Paris-Rocquencourt, France

The goal of this lecture is to show how modern theorem provers – I will use the Coq proof assistant – can be used to mechanize the specification of programming languages and their semantics, and to reason over individual programs as well as over generic program transformations, as typically found in compilers.

The topics covered include:

- Operational semantics: small-step, big-step, definitional interpreters.
- Axiomatic semantics.
- Generation of verification conditions; application to program proof.
- Examples of program transformations/compilation and their correctness proofs.
- Examples of static analyses and their correctness proofs.
- Compiler verification "in the large"(if time permits).

References

1. Y. Bertot. *Coq in a hurry*. 2008.
   Available from `http://cel.archives-ouvertes.fr/inria-00001173/`
2. T. Nipkow. *Winskel is (almost) right: Towards a Mechanized Semantics Textbook*. In "Formal Aspects of Computing", 10:171-186, 1998.
   Available from `http://www4.in.tum.de/~nipkow/pubs/fac98.html`
3. X. Leroy, H. Grall. *Coinductive big-step Operational Semantics*. "Information and Computation", 207(2):285-305, 2009.
   Available from `http://gallium.inria.fr/~xleroy/publi/coindsem-journal.pdf`