

Implementing an Instantiation-based Theorem Prover for First-order Logic

Konstantin Korovin
The University of Manchester, England
korovink@cs.man.ac.uk

The basic idea behind instantiation-based theorem proving is to combine clever generation of instances of clauses with satisfiability checking of ground formulas. There are a number of approaches developed and implemented in recent years: Ordered Semantic Hyper Linking of Plaisted and Zhu, Disconnection Calculus of Letz and Stenz implemented in DCTP, Model Evolution Calculus of Baumgartner and Tinelli implemented in Darwin, and instantiation approach of Claessen implemented in Equinox.

One of the distinctive features of the approach we have been developing is a modular combination of ground reasoning with instantiation. In particular, this approach allows us to employ any off-the-shelf propositional satisfiability solver in a general context of first-order reasoning. In our previous work (together with Harald Ganzinger) we developed a theoretical background for this instantiation method and have shown completeness results together with general criteria for redundancy elimination. In order to evaluate the practical applicability, we implemented our method in iProver.

This talk focuses on implementation issues of an instantiation-based theorem prover, based upon our experience with iProver. We show how our abstract framework can be turned into a concrete implementation. We discuss how well-studied technology of implementing resolution theorem provers can be adapted for implementing iProver and issues specific to instantiation. We show some concrete redundancy criteria for instantiation and how they are implemented in iProver. Finally, we discuss some future directions.