Dissertation Abstract

First-order Theorem Proving for Program Analysis and Theory Reasoning

Candidate:
MSc. Ioan Dumitru Dragan

Reviewers:
Priv.-Doz. Dr. Laura Kovács
Prof. Dr. Andrei Voronkov
Prof. Dr. Armin Biere

Analyzing and verifying computer programs is an important and challenging task. Banks, hospitals, companies, organizations and individuals heavily depend on very complex computer systems, such as Internet, networking, online payment systems, and autonomous devices. These systems are integrated in an even more complicated environment, using various computer devices. Technically, software systems rely on software implementing complicated arithmetic and logical operations over the computer memory. If this software is not reliable, the costs to the economy and society can be huge. Software development practices therefore need rigorous methods ensuring that the program behaves as expected.

Formal verification provides a methodology for making reliable and robust systems, by using program properties to hold at intermediate points of the program and using these properties to prove that programs have no errors. Providing such properties manually requires a considerable amount of work by highly skilled personnel and makes verification commercially not viable. Formal verification therefore requires non-trivial automation for generating valid program properties, such as loop invariants.

In this thesis we study the use of first-order theorem proving for generating and proving program properties. Our thesis provides a fully automated tool support, called Lingva, for generating quantified invariants of programs over arrays, and shows experimentally that the generated invariants summarize the behavior of the considered loops. Our work is based on the recently introduced symbol elimination method for invariant generation, using a saturation-based
first-order theorem prover.

As program properties involve both logical and arithmetical operations over unbounded data structures, such as arrays, generating and proving program properties requires efficient methods for reasoning with both theories and quantifiers. Another contribution of this thesis comes with the integration of the bound propagation method for solving systems of linear inequalities in the first-order theorem prover Vampire. Our work provides an automated tool support for using Vampire for deciding satisfiability of a system of linear inequalities over the reals or rationals. We experimentally show that bound propagation in Vampire performs well when compared to state-of-the-art satisfiability modulo theory solvers on hard linear optimization problems.

Our arithmetic solver is limited to conjunction of linear inequalities, while arithmetic program properties usually have a more complex boolean structure, using a combination of logical conjunction, disjunction and negation. To make our work applicable for handling such complex arithmetic properties, another contribution of this thesis is the integration of boolean satisfiability (SAT) solvers within Vampire. Our work exploits the recently introduced AVATAR framework for separating the first-order reasoning part of a problem from its boolean structure. We describe our technical and implementation challenges for integrating the best performing SAT solvers within Vampire, and use our implementation to evaluate the AVATAR framework on a large set of problems coming from the TPTP library of automated theorem provers.