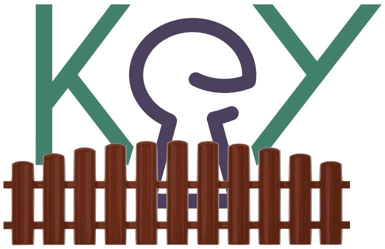
Karlsruhe Institute of Technology
Institute for Theoretical Computer Science

Studienarbeit / Bachelorarbeit

Detecting Flaws in Verification Conditions Using Bounded Verification Techniques



The KeY verification system (developed at our institute) can be used to prove that a Java program satisfies its specification. However, the interactive verification attempt may fail. In such a case one or more reasons can be the cause of the unsuccessful attempt: There can be a bug in the program, the specification may be wrong or the verification tool is not capable to achieve the task automatically and needs the user's assistance.

In practice, it is very difficult to find out which of these cases applies to "stuck" proof situation. But practice shows also that if a bug exists, it often manifests itself in a counterexample of rather small size. Bounded verification techniques exploit this observation: They set boundaries to problem sizes and generate proof obligations over finite domains, making them decidable.

In order to exploit the capability of satisfiability modulo theories (SMT) solvers of automatic reasoning about first order specifications involving commonly used theories in programming, we have developed an experimental translation of KeY proof obligations into the SMT-Lib language (common input language for SMT solvers). Because this technique was intended for proofs, the resulting SMT proof obligation is unbounded.

The task of this thesis is to encode KeY proof situations into a first order logic over bounded domains. This includes coming up with suitable domain boundaries, the translation of formulas and inference rules into the input language of a SMT solver. In case the solver finds and reports a counterexample, this is then to be analysed and to be presented to the KeY user in such a manner they can learn more about the situation. The user should be able to understand the counterexample in order to find out whether it is spurious or indicates a flaw in the program and/or the specification. The approach should be implemented within the KeY system.

You should have attended the course "Formale Systeme" or a similar course covering first order logic and sequent calculus deduction. Familiarity with KeY, SMT solvers, and/or JML is preferred, but not necessary. You should have gathered some experiences with the Java programming language since this is the programming language used in the KeY system.

This thesis is a joint project between the groups Automatic Software Analysis (led by Mana Taghdiri) and Logic and Formal Methods (led by Peter H. Schmitt).

If interested, please contact Mattias Ulbrich (`ulbrich@kit.edu`) or Aboubakr Achraf El Ghazi (`elghazi@kit.edu`)