SMT for Higher-Order Logic

Topics

Computer Science, Logic, Automatic and Interactive Theorem Proving, Satisfiability Modulo Theories, Verification

Institution

Inria, Loria, Université de Lorraine (in cooperation with VU Amsterdam)

Location

Nancy, France (in close cooperation with Amsterdam, the Netherlands)

Team/Project

VeriDis

Supervision

Pascal Fontaine, Pascal.Fontaine@loria.fr
Jasmin Blanchette, Jasmin.Blanchette@inria.fr

Background

Many applications, notably in the context of verification (for critical systems in transportation, energy, etc.), rely on checking the satisfiability of logic formulas. Satisfiability-modulo-theories (SMT) solvers [2–4,6] handle large formulas in expressive languages with built-in and custom operators (e.g. arithmetic and data structure operators). These tools are built using a cooperation of a SAT (propositional satisfiability) solver to handle the Boolean structure of the formula and theory reasoners to tackle the atomic formulas (e.g. $x > y+z$ for the theory of arithmetic).

Currently, SMT solvers only handle first-order logic. They cannot reason about bound variables in expressions such as $\sum_{i=1}^{n} i^2$, $\lambda x.\ x+1$, and $\{x \in \mathbb{N} \mid x \in A \lor x \in B\}$, and they generally cannot perform proofs by induction. This is unfortunate, because most interactive verification tools (e.g. Coq [5]), which use SMT solvers as backend reasoning engines, offer higher-order languages.

This PhD subject is proposed in the context of J. Blanchette's ERC Starting Grant Matryoshka,[1] an ambitious five-year project that aims at automatic provers more useful for interactive verification by reducing the gap between the automatic and interactive worlds. One of the concrete goals of the project is to lift up the reasoning capabilities of SMT solvers towards higher order.

Objectives

As an accessible first step towards higher-order SMT, we propose to study for SMT the so-called *applicative encoding* of λ-free higher-order logic, similarly to the preliminary investigation for the superposition calculus (a rival of SMT) [1]. The effect of the encoding on the ability for SMT solvers to tackle benchmarks will

[1] http://matryoshka.gforge.inria.fr/

be evaluated experimentally, and the issues (e.g. in the various decision procedures or in the instantiation modules) will be identified. The PhD student will then have the opportunity to propose new techniques to mitigate these issues, or to better perform on benchmarks using the applicative encoding. The complexity of these techniques will be studied, in particular if they impact on the existing efficient decision procedures. In a second phase of the work, the PhD student will propose techniques to tightly integrate higher-order reasoning within an SMT framework.

Since the ultimate goal is to integrate higher-order SMT solvers within proof assistants (interactive theorem provers), it is crucial to produce proofs that can be understood and replayed by external tools. The PhD student will describe, provide the theoretical foundations, and implement proof reconstruction of higher-order formulas delegated to the SMT solver using the various methods studied and mentioned above.

The PhD subject can and will be adjusted according to the interests of the student.

Requirements

We are looking for excellent candidates with a strong interest for logic, decision procedures and proofs. Some acquaintance with either automated or interactive theorem proving is a plus. Knowledge of French is not required.

Bibliographic references

1. J. C. Blanchette, U. Waldmann and Daniel Wand. A Lambda-Free Higher-Order Recursive Path Order. 2016. `http://people.mpi-inf.mpg.de/~jblanche/lambda_free_rpo_rep.pdf`

2. C. Barrett, R. Sebastiani, S. A. Seshia and C. Tinelli, Satisfiability Modulo Theories. Chapter 26 of the Handbook of Satisfiability, pages 825–885. Volume 185 of Frontiers in Artificial Intelligence and Applications. IOS Press 2009.

3. T. Bouton, D. Caminha B. de Oliveira, D. Déharbe and P. Fontaine. veriT: an open, trustable and efficient SMT-solver. *In Proc. Conference on Automated Deduction (CADE)*, volume 5663 of LNCS, pages 151–156. Springer-Verlag, 2009.

4. P. Fontaine, J.-Y. Marion, S. Merz, L. P. Nieto and A. Tiu. Expressiveness + Automation + Soundness: Towards Combining SMT Solvers and Interactive Proof Assistants. *In Proc. Tools and Algorithm for the Construction and Analysis of Systems (TACAS)*, volume 3920 of LNCS, pages 167–181. Springer-Verlag, 2006.

5. Coq, `http://coq.inria.fr`.

6. veriT, `http://www.veriT-solver.org`.