Automated reasoning: the best of two worlds

Topics

Computer Science, Logic, Automatic Theorem Proving, Superposition, Satisfiability Modulo Theories, Superposition, Verification

Institution

Inria, Loria, Université de Lorraine (in cooperation with DHBW Stuttgart)

Location

Nancy, France (in close cooperation with DHBW Stuttgart, Germany)

Team/Project

VeriDis

Supervision

Pascal Fontaine, Pascal.Fontaine@loria.fr
Stephan Schulz, schulz@eprover.org

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 [1,3,7] 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). The veriT SMT solver [7], developed in Nancy, is a state-of-the-art reasoner of this kind.

When it comes to handle pure quantified first-order logic with equality, the superposition calculus (see e.g. [4]) gives best results. This calculus is a complete set of rules extending resolution to first-order logic with equality. It is extremely good at finding intricate proofs, even on large sets of axioms. Superposition-based saturation automated theorem proving has been steadily progressing for decades. One of the best provers of this kind is the E-prover [5,6], developed in Stuttgart.

The main objective of this subject is to use the best of both SMT and superposition for efficient reasoning on large logic formulas with quantifiers, in presence of interpreted symbols. There are both theoretical and practical aspects related to this question.

The work will be conducted under the supervision of Stephan Schulz (DHBW) and Pascal Fontaine (Loria), and will be mainly located at Loria.

Objectives

We already have first experiments both integrating the superposition calculus in SMT and making use of SAT and SMT techniques in the context of superposition. We suggest that, as an introductory step, the PhD student get knowledge of these prototypes, and evaluate their strength and weaknesses on known sets of benchmarks. Another direction is to get acquainted with the Avatar [] technique successfully used in the Vampire prover.

As a second step, we envision incrementally generalizing the approach in the prototypes for a tighter integration of both reasoning techniques. In parallel, the PhD student will design an abstract calculus to study completeness issues, and termination for decidable fragments (notably for the Bernays-Schönfinkel-Ramsey fragment). Depending on the interest of the student for formally verified

algorithms, this calculus might be the object of formal verification using a proof assistant, along the lines of [2].

Our ultimate goal is to propose powerful reasoners including SMT and superposition techniques to discharge proof obligations from proof assistants (interactive theorem provers). It is thus 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 formulas delegated to the hybrid Superposition-SMT solver.

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, and automated reasoning. Some acquaintance with either automated or interactive theorem proving is a plus. Knowledge of English is mandatory, French or German is not required.

Bibliographic references

1. 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.
2. J. C. Blanchette, M. Fleury, C. Weidenbach. A Verified SAT Solver Framework with Learn, Forget, Restart, and Incrementality. IJCAI 2017: 4786-4790
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. R. Nieuwenhuis, A. Rubio. Paramodulation-Based Theorem Proving. Handbook of Automated Reasoning 2001: 371-443.
5. S. Schulz. System Description: E 0.81. In Proc. International Joint Conference on Automated Reasoning (IJCAR), volume 3097 of LNCS, pages 223–228. Springer, 2004.
6. E-prover, `www.eprover.org`.
7. veriT, `http://www.veriT-solver.org`.