Scalable Proof Production and Checking in SMT

Cesare Tinelli

The University of Iowa, Iowa City, IA, USA

Abstract

Solvers for Satisfiability Modulo Theories (SMT) have become crucial components in safety- or mission-critical formal methods applications, in particular model checking, verification, and security analysis. Since state-of-the-art SMT solvers are large and complex systems, they are prohibitively difficult to prove correct. Hence, proof production is essential as a way to demonstrate instead the correctness of their responses, making those responses amenable to independent verification. Historically, the main challenges for proof production in SMT have been solver performance and proof coverage, often leading to the disabling of many sophisticated solving techniques when running in proof-production mode, or to coarse-grained, and harder to check, proofs.

The first part of this talk presents a flexible proof-production architecture designed to handle the complexity of versatile, industrial-strength SMT solvers, and discusses how it has been leveraged to produce detailed proofs, even for sophisticated reasoning components. The architecture, implemented in the state-of-the-art SMT solver cvc5, allows proofs to be produced modularly, as needed, and with various safeguards for correctness. The architecture supports the generation of textual proof certificates in different formats, for offline proof checking by external tools, as well as a rich API, which is useful for online integration of the SMT solver into other reasoning tools such as, for instance, skeptical proof assistants. Extensive experimental evaluations with both SMT-LIB benchmarks and benchmarks provided by industrial partners have shown that the new architecture results in greater proof coverage than previous approaches, imposes a small runtime overhead, and supports fine-grained proofs in the great majority of cases.

The second part of the talk gives an overview of a new generic language for expressing SMT proof certificates that builds on almost two decades of work and experience in proof generation and checking in SMT and combines the benefits of several previous efforts on the topic. While developed to express cvc5's proof certificates, the language is meant to be useful to other SMT solvers as well. It is in fact a logical framework, based on the syntax and semantics of the upcoming Version 3 of the SMT-LIB standard, that can be customized, as in the case of cvc5, with the specific proof system used by the solver through the definition of new symbols, binders and proof rules. In addition, it features an intuitive syntax for representing natural-deduction-style proofs and the ability to integrate other proof formats (such as, for instance, those currently used by SAT solvers) via the use of oracles. The talk discusses an initial evaluation of the proof language, obtained with a companion checker for it and an instantiation to cvc5's proof system. The evaluation shows the viability of high-performance, fine-grained proof production and checking for SMT.

The talk concludes with a brief overview of future work and new potential applications enabled by scalable proof certificate production and checking.

2012 ACM Subject Classification Theory of computation → Automated reasoning; Theory of computation → Logic and verification; Theory of computation → Proof theory; Theory of computation → Type theory

Keywords and phrases Satisfiability Modulo Theories, Proof generation and certification

Digital Object Identifier 10.4230/LIPIcs.SAT.2024.2

Category Invited Talk

Supplementary Material *Software (Source code of cvc5 SMT solver)*: `https://github.com/cvc5/cvc5`, archived at `swh:1:dir:2f245f4f16b574408f7861a2d7a4a763e47e5a7f`

Funding The work described in this talk was funded in part by Amazon Web Services and by DARPA award FA8750-24-2-1001.

27th International Conference on Theory and Applications of Satisfiability Testing (SAT 2024).
Editors: Supratik Chakraborty and Jie-Hong Roland Jiang; Article No. 2; pp. 2:1–2:2
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements This work has involved many collaborators over the course of almost 4 years. Significant contributors are (in alphabetical order): Leni Aniva, Haniel Barbosa, Clark Barrett, Tomaz Gomes, Harun Khan, Gereon Kremer, Hanna Lachnitt, Abdalrhman Mohamed, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Andrew Reynolds, Hans-Jörg Schurr, Arjun Viswanathan, and Yoni Zohar.