STP in the SMTCOMP 2020

Various

1   Introduction

STP[1] is an efficient open source solver for QF_BV and arrays without extensionality. STP recursively simplifies bit-vector constraints, solves linear bit-vector equations, and then eagerly encodes them to CNF for solving. Array axioms are added as needed during an abstraction-refinement phase.

2   Development history

STP was originally developed by Vijay Ganesh under the supervision of Professor David Dill. Later releases were developed by Trevor Hansen under the supervision of Peter Schachte and Harald Søndergaard. STP handles arbitrary precision integers using Steffen Beyer's library. STP encodes into CNF via the and-inverter graph package ABC of Alan Mishchenko [2]. STP supports different SAT backends, by now MiniSat [3], CryptoMiniSat [4] and Riss [5].

3   Recent Developments to STP

Since SMT-COMP 2019 we have: * Improved the build scripts - Christian Cadar, Andrew Vaughan, Mate Soos,BP Foley, * Improved how speculative simplifications are applied - Trevor Hansen

4   Recent Developments to the Underlying SAT Solvers

The underlying SAT solver, CryptoMiniSat, has been significantly improved by including state-of-the-art techniques from past SAT competition winners and it has been tuned for the SMT use case. The way a SAT solver is used in a typical SAT competition scenario, i.e. when the solve() function is called only once, is very different than when used in a typical SMT scenario. Returning solutions quickly and close to one another is essential when the SAT solver is used from within an SMT solver. Hence CryptoMiniSat has been tuned to worked well in this sphere as well as in the regular, SAT competition, use case.

The SAT solver MergeSAT is a fork of the winner of the SAT Competition 2018, MapleL-CMDistChronoBT. MergeSAT ports features that have been proposed in other solvers into that solver, enables incremental solving again, as well as makes the solver behavior reproducible again.

Acknowledgements

References

[1] Ganesh, V.: Decision Procedures for Bit-Vectors, Arrays and Integers. PhD thesis, Computer Science Department, Standford University, CA, United States (2007)

[2] Brayton, R., Mishchenko, A.: Abc: An academic industrial-strength verification tool. In: Proceedings of the 22Nd International Conference on Computer Aided Verification. CAV'10, Berlin, Heidelberg, Springer-Verlag (2010) 24–40

[3] Niklas Sörensson, N.E.: GitHub repository for MiniSat (may 2019) https://github.com/niklasso/minisat.

[4] Soos, M.: GitHub repository for CryptoMiniSat (may 2019) https://github.com/msoos/cryptominisat.

[5] Manthey, N.: GitHub repository for Riss (may 2019) https://github.com/conp-solutions/riss.