STP at the SMTCOMP 2022

Various

1 Background

STP[1, 2] is an 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 to CNF for solving. Array axioms are added as needed during an abstraction-refinement phase.

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 [3]. By default STP uses CryptoMiniSat [4], but also can use MiniSat [5], Riss [6] or CaDiCAL [7].

2 Recent Developments to STP

In the last year contributors to STP have:

- further improved its build system, so that it builds reliably on more platforms.
- added better domain analysis for strength reductions. When this simplification determines, for example, that the most significant bits of the operand to a signed division are known, it replaces it with an unsigned division.
- Reimplemented the variable elmination simplification. This simplification substitutes through to remove variables from the problem.
- Added a simplification to create fresh variables when extracts from a variable do not overlap.

Acknowledgements

Vijay Ganesh, Dan Liew, Mate Soos and Ryan Govostes contributed substantially to the STP code base.

References

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

[2] Hansen, T.: A constraint solver and its application to machine code test generation. PhD thesis, Department of Computing and Information Systems, The University of Melbourne, Melbourne, Australia (2012)

[3] 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

[4] Soos, M.: GitHub repository for CryptoMiniSat (June 2021) https://github.com/msoos/cryptominisat.

[5] Niklas Sörensson, N.E.: GitHub repository for MiniSat (June 2021) https://github.com/niklasso/minisat.

[6] Manthey, N.: GitHub repository for Riss (June 2021) https://github.com/conp-solutions/riss.

[7] Biere, A., Fazekas, K., Fleury, M., Heisinger, M.: CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020. In Balyo, T., Froleyks, N., Heule, M., Iser, M., Järvisalo, M., Suda, M., eds.: Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions. Volume B-2020-1 of Department of Computer Science Report Series B., University of Helsinki (2020) 51–53