STP

Vijay Ganesh, Trevor Hansen, Dan Liew, Ryan Govostes, Khoo Yit Phang, Mate Soos

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.

The version of STP submitted to STMCOMP 2014 is revision fcfb30e8664 of STP's publicly available source code repository [2]. It was compiled with a slightly tuned version of CryptoMiniSat [3] revision 7ae6c5123 available from its own public repository [4].

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 [5].

We found many defects using Robert Brummayer and Armin Biere's fuzzing and delta debugging tools [6] in both STP and CryptoMiniSat. Although these tools are not a proper replacement for unit or integration testing, they provide a healthy sanity check against some forms of bugs. In particular, fuzzing does not test against performance regressions, instead it can only detect where certain optimizations are wrongly implemented such as to produce incorrect results.

In the past year, STP has been actively developed on GitHub.

Acknowledgements

We would like to thank everyone who submitted bug reports, pull requests, and other useful data such as test cases.

References

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

2. Ganesh, V., Hansen, T., Liew, D., Govostes, R., Soos, M.: GitHub repository for STP (2014) https://github.com/stp/stp.
3. Soos, M., Nohl, K., Castelluccia, C.: Extending SAT solvers to cryptographic problems. In Kullmann, O., ed.: SAT. Volume 5584 of Lecture Notes in Computer Science., Springer (2009) 244–257
4. Soos, M.: GitHub repository for CryptoMiniSat (2014) https://github.com/msoos/cryptominisat.
5. 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
6. Brummayer, R., Biere, A.: Fuzzing and delta-debugging smt solvers. In: Proceedings of the 7th International Workshop on Satisfiability Modulo Theories. SMT '09, New York, NY, USA, ACM (2009) 1–5