STP in the SMTCOMP 2021

Various

1 Background

STP[1] 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 them 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 [2]. By default STP uses CryptoMiniSat [3], but also uses MiniSat [4] and Riss [5].

2 Recent Developments to STP

In the last year:

Andrew V. Jones and Mate Soos have considerably improved the build system. Mate Soos and Norbert Manthey have got the distributed and parallel versions of STP working. Trevor Hansen has sped up some simplifications.

3 Parallel Track Version

For the parallel track, STP is used to generate a CNF formula. Once this formula is generated, a parallel portfolio of SAT solvers is used. This portfolio calls external SAT solver binaries, in order:

1. CryptoMiniSat [3]
2. MergeSat [6]
3. Riss [5]
4. MiniSat [4]

Before starting these solvers, the number of available compute resources is detected. In case less equal four compute resources are detected, the given number of solvers in the above order is started. In case we detect more than four cores, we assign the remaining cores to CryptoMiniSat. Then, this solver will run in its internal portfolio mode with knowledge sharing enabled.

4 Cloud Track Version

The cloud track version of the solver uses STP only to translate the SMTLIB2 problem to CNF. Then it uses the MPI version of CryptoMiniSat to solve the CNF. This solver synchronizes clients to each other through the main process. The main process uses MPI broadcast to send the CNF to all clients. The clients then try to solve the CNF, and send an update with all 0-level variable values and all new binary clauses to the main process every 30k conflicts. The main process then sends to each client individually the cumulative information after it has received more than half the clients' data. Clients indicate to the main in case they found a solution, which in turn informs all clients, prints the solution, and exits. Each client runs its own multi-threaded solver, synchronized every 10k conflicts, in-memory, without the use of MPI.

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

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

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

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