Sateen: Sat Enumeration Engine for SMT-COMP'06 *

Hyondeuk Kim, HoonSang Jin, and Fabio Somenzi

University of Colorado at Boulder
{Hyondeuk.Kim,HoonSang.Jin,Fabio}@Colorado.EDU

1 Description

Sateen is a satisfiability solver that combines the propositional reasoning engine with the theory-specific procedure. It uses the lazy approach that relies on incremental refinements of a propositional abstraction of the given formula during the enumeration of its solutions.

We treat the theory-specific procedures as "black boxes" and concentrate on the propositional aspect of the procedure. If a propositional assignment is found to be consistent in the given theory, then a model for the original formula has been found. Otherwise, a refinement of the propositional abstraction is extracted from the proof of inconsistency and the search is resumed. It is advantageous to call the theory-specific procedures also on partial assignments to terminate the fruitless search of part of the state space, or to learn so-called *theory consequences* [3]. Our implementation follows this approach, though the full check for inconsistencies due to disequalities is applied only to complete assignments.

Our approach does not decompose equalities and their negations; rather, it converts the problem of checking satisfiability of a conjunction of arithmetic atomic formulae into a set of propositional satisfiability checks. This approach greatly impoves the efficiency of our decision procedure for problem instances in which disequalities play a significant role.

2 Problem Divisions

Integer Difference Logic. Sateen generates a model for a satisfiable formula. Currently, it does not produce a proof for an unsatisfiable one.

3 Programming Language(s)

Sateen is written in C. An ANSI C compiler and GNU make are required to build it.

4 Software Architecture of the System

Sateen consists of a SAT solver that enumerates all solutions to a propositional formula and a theory solver that decides whether a conjunction of literals in the theory (e.g., a

* This work was supported in part by SRC contract 2005-TJ-920.

conjunction of integer difference constrains) is satisfiable. The theory solver produces a model if the constraints are satisfiable and a proof of unsatisfiability otherwise. The proof is a set of unsatisfiable constraints—typically a subset of the constraints passed to it.

The propositional SAT solver part is characterized by:

- All SAT Enumeration [2]

The current theory solver part is based on layering [1]. It relies in particular on five layers:

- Equality Solver
- Negative cycle detection
- Nelson-Oppen framework [4]
- Finite Instantiations
- Theory consequence [3]

GMP-4.2.1 library is used for infinite-precision problem [6].

References

[1] M. Bozzano, R. Bruttomesso, A. Cimatti, T. Junttila, P. van Rossum, S. Schulz, and R. Sebastiani. An incremental and layered procedure for the satisfiability of linear arithmetic logic. In *International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'05)*, Edinburgh, UK, Apr. 2005. To appear.

[2] H. Jin, H. Han, and F. Somenzi. Efficient conflict analysis for finding all satisfying assignments of a Boolean circuit. In *International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'05)*, pages 287–300, Apr. 2005. LNCS 3440.

[3] R. Nieuwenhuis and A. Oliveras. DPLL(T) with exhaustive theory propagation and its application to difference logic. In *Seventeenth Conference on Computer Aided Verification (CAV'05)*, pages 321–334. Springer-Verlag, Berlin, July 2005. LNCS 3576.

[4] S. Lahiri and M. Musuvathi. An efficient Nelson-Oppen decision procedure for difference constraints over rationals. In *Third International Workshop on Pragmatical Aspects of Decision Procedures in Automated Reasoning (PDPAR'05)*, pages 2–9, Edinburgh, UK, July 2005. To appear in ENTCS.

[5] Url: http://elib.zib.de/pub/packages/mathprog/linprog/lp-solve.

[6] Url: http://www.swox.com/gmp.