Sateen: Sat Enumeration Engine for SMT-COMP'05 *

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. The approach leverages recent advances in the efficient enumeration of the satisfying assignments.

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.

Efficiency of this scheme depends crucially on the ability to identify concise explanations for the unsatisfiability in the given theory. We employ new techniques that allow the propositional solver to quickly generate small clauses to be added as refinement to the propositional abstraction. These techniques leverage recent work that we have done in the enumeration of all satisfying assignments of a propositional formula, and in the minimization of the satisfying assignments.

2 Problem Divisions

Real Difference Logic and Integer Difference Logic. Sateen generates a model for a satisfiable formula. Currently it does not produce a proof for an unsatisfiable one. The generated models may be partial, in the sense that they may not specify the values of some variables if those values are immaterial.

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.,

* This work was supported in part by SRC contract 2004-TJ-920.

a conjunction of real 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]
- Minimization of the satisfying assignments

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

- Equality Solver
- Negative cycle detection
- LP Solver [3]

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