Integrating SAT solvers with domain-specific reasoners

Fausto Giunchiglia[1,2], Roberto Sebastiani[1], Paolo Traverso[2]

[1] DISA, Università di Trento, via Inama 5, I-38100 Trento, Italy
[2] ITC-IRST, Loc. Pantè, I-38050 Povo, Trento, Italy
fausto@irst.itc.it, rseba@cs.unitn.it, leaf@irst.itc.it

In the last years we have witnessed an impressive advance in the efficiency of SAT techniques, which has brought large previously untractable problems at the reach of state-of-the-art solvers. As a consequence, some hard real-world problems have been successfully solved by encoding them into SAT. Propositional planning [KMS96] and boolean model-checking [BCCZ99] are among the best achievements.

Unfortunately, simple boolean expressions are not expressive enough for representing many real-world problems. For example, problem domains like resource planning, temporal reasoning, verification and model checking of analog circuits require also handling constraints on real-valued quantities. Moreover, some problem domains like model checking often require an explicit representation of integers and arithmetic operators, which simple boolean expressions are not good at representing efficiently.

In 1996 we have proposed a new general approach, called SAT based, for building domain-specific decision procedures on top of SAT solvers [GS96b, GS96a]. We assume that the input is a boolean combination of domain-specific expressions –which cannot be further decomposed propositionally. We call the latter domain-specific *atoms*. Examples of such atoms are boxed formulas like $\Box\phi$ and inequalities like $(x_i \leq y_i + c_i)$. The basic idea is to decompose the search into two orthogonal components, one purely propositional component and one purely domain-specific component, so that to use a SAT solver for the former and a pure domain-specific procedure for the latter.

The SAT based approach is based on a generate-and-check paradigm. The input formula is encoded as a propositional formula on both boolean and domain-specific atoms, and a SAT solver is run over it. Each time an assignment is found which propositionally satisfies the input, its consistency is verified wrt. the domain-specific semantics of the non-boolean atoms by a proper procedure; this is done until either one domain-consistent assignment is found –i.e., the input formula is consistent– or no more assignments are available –i.e., the input formula is inconsistent.

So far the SAT based approach proved very effective in various problem domains like, e.g., modal logics [GS96a], temporal reasoning [ACG99], resource planning [WW99]. In particular, the latter two proved rather efficient in combining SAT reasoning with Linear Programming (LP). Now we want to focus our research on combining SAT non only with LP systems, but also with math constraint solvers and/or computer algebra systems.

The research directions are manifold. First, in our approach the SAT solver is no more used to find one single assignment, but rather to generate up to an exhaustive set of assignments. Thus, new techniques and heuristic ad hoc for this new task have to be studied. Second, to maximize the effects of techniques like backjumping and forward checking, the SAT solver has to get as much information as it can from any failed assignment. Thus, the research should be directed on maximizing the effectiveness of the information exchange between the SAT solver and the math constraint solver. Finally, the math constraint solver should not be repeatedly run on similar subproblems. This requires studying techniques for incremental constraint solving and to maximize the integration of the two solvers.

References

[ACG99] A. Armando, C. Castellini, and E. Giunchiglia. SAT-based procedures for temporal reasoning. In *Proc. ECP-99*, 1999.

[BCCZ99] A. Biere, A. Cimatti, E. Clarke, and Y. Zhu. Symbolic model checking without BDDs. In *Proc. CAV'99*, 1999.

[GS96a] F. Giunchiglia and R. Sebastiani. Building decision procedures for modal logics from propositional decision procedures - the case study of modal K(m). Technical Report 9611-06, IRST, 1996. To appear on *Information and Computation*, Vol. 160, 2000.

[GS96b] F. Giunchiglia and R. Sebastiani. A SAT-based decision procedure for ALC. In *Proc. KR'96*, 1996.

[KMS96] H. Kautz, D. McAllester, and Bart Selman. Encoding Plans in Propositional Logic. In *Proc. KR'96*, 1996.

[WW99] S. Wolfman and D. Weld. The LPSAT Engine and its Application to Resource Planning. In *Proc. IJCAI'99*, 1999.