Barcelogic SAT Solver: System description

Robert Nieuwenhuis, Albert Oliveras, Tomás Lloret*
Technical University of Catalonia, Barcelona.
www.lsi.upc.es/~roberto.

June 30, 2006

The Barcelogic SAT Solver is an efficient implementation of the Davis-Putnam-Logemann-Loveland (DPLL) procedure [DP60, DLL62].

Our original aim for developing it was the need of a DPLL-based *engine* for our DPLL(T) approach to *Satisfiability Modulo Theories (SMT)*: deciding the satisfiability of ground first-order formulas with respect to background theories such as equality, linear integer or real arithmetic, arrays, etc. A DPLL(T) system consists of a general DPLL(X) engine, very similar in nature to a SAT solver, whose parameter X can be instantiated with a solver $Solver_T$ for a theory T. Once the DPLL(X) engine has been implemented, this SMT approach becomes extremely flexible: new theories can be dealt with by simply plugging in new theory solvers. These solvers only must be able to deal with *conjunctions* of theory literals and conform to a minimal and simple set of additional requirements [GHN+04]. Our Barcelogic implementation of DPLL(T) won four divisions at the 2005 SMT Competition [BdMS05]).

The Barcelogic SAT Solver includes all the modern enhancements of the DPLL procedure as they can be found in solvers such as Chaff, Berkmin, Siege or MiniSAT [MMZ+01, GN02, ES03, Rya04]. This includes two-watched-literal unit propagation, activity-based decision heuristics and lemma deletion strategies, as well as different learning schemes, including first-UIP and all-UIP and backjump clause simplification methods. Also some degree of clause redundancy detection is performed, during preprocessing and at certain moments during the solving process.

Due to our needs in the context of SMT applications, where the different theory solvers require additional memory, a special attention has been devoted to the memory efficiency of the Barcelogic SAT Solver. Two-literal clauses are stored independently, and are used in several clause simplification techniques. Also a special care has been taken in avoiding unnecessary cache misses.

*Technical Univ. of Catalonia, Barcelona, Partially supported by Spanish Min. of Educ. and Science through the LogicTools project (TIN2004-03382, all three authors), and FPU grant AP2002-3533 (Oliveras), and personal research grant "SMT Solvers for High-Level Hardware Verification" from Intel Corporation (Nieuwenhuis).

The system is written in C++, and has about 5000 (not very dense) lines. Currently the Barcelogic SAT Solver has to be considered as ongoing work. Many ideas still need to be worked out and tried out and re-implemented, and very little work has been done on dynamically adjusting the different search parameters for different problems.

References

[BdMS05] C. Barrett, L. de Moura, and A. Stump. SMT-COMP: Satisfiability Modulo Theories Competition. In K. Etessami and S. Rajamani, editors, *17th International Conference on Computer Aided Verification*, volume 3576 of *Lecture Notes in Computer Science*, pages 20–23. Springer, 2005. See www.csl.sri.com/users/demoura/smt-comp.

[DLL62] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Comm. of the ACM*, 5(7):394–397, 1962.

[DP60] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:201–215, 1960.

[ES03] Niklas Eén and Niklas Sörensson. An Extensible SAT-solver. In *Proceedings of the Sixth International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 502–518, 2003.

[GHN+04] Harald Ganzinger, George Hagen, Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. DPLL(T): Fast Decision Procedures. In R. Alur and D. Peled, editors, *Proceedings of the 16th International Conference on Computer Aided Verification, CAV'04 (Boston, Massachusetts)*, volume 3114 of *Lecture Notes in Computer Science*, pages 175–188. Springer, 2004.

[GN02] E. Goldberg and Y. Novikov. BerkMin: A fast and robust SAT-solver. In *Design, Automation, and Test in Europe (DATE '02)*, pages 142–149, 2002.

[MMZ+01] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an Efficient SAT Solver. In *Proc. 38th Design Automation Conference (DAC'01)*, 2001.

[Rya04] Lawrence Ryan. Efficient Algorithms for Clause-Learning SAT Solvers. Master's thesis, School of Computing Science, Simon Fraser University, 2004.