Barcelogic SAT Solver: System description

Roberto Asín, Morgan Deters, Germain Faure, Robert Nieuwenhuis
Albert Oliveras, Enric Rodríguez-Carbonell[*]

Technical University of Catalonia, Barcelona.

April 7, 2008

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 $\mathsf{DPLL}(T)$ 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 $\mathsf{DPLL}(T)$ system consists of a general $\mathsf{DPLL}(X)$ 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 $\mathsf{DPLL}(X)$ 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 [NOT06].

The Barcelogic SAT Solver includes modern enhancements of the DPLL procedure as they can be found in solvers such as Chaff, Berkmin, MiniSAT, Rsat or PicoSAT, [MMZ$^+$01, GN02, ES03, PD07, Bie08]. This includes two-watched-literal unit propagation, activity-based decision heuristics and lemma deletion strategies, a first-UIP learning scheme, frequent restarts, and relatively powerful techniques for lemma shortening. Each time a new lemma is added we use it for subsumption and subsumption resolution on all other clauses. Also some degree of clause redundancy detection is performed, during preprocessing and at certain moments during the solving process. For not too large instances, we also use SatELite as a preprocessor [EB05].

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.

[*]All authors partially supported by Spanish Min. of Educ. and Science through the LogicTools-2 project (TIN2007-68093-C02-01).

The system, submitted as a 32-bit binary, is written in C/C++ and has about 7500 (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

[Bie08] A. Biere. PicoSAT essentials. *Journal on Satisfiability, Boolean Modeling and Computation*, 2008. submitted.

[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.

[EB05] Niklas Eén and Armin Biere. Effective preprocessing in sat through variable and clause elimination. In F. Bacchus and T. Walsh, editors, *SAT*, volume 3569 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2005.

[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.

[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.

[NOT06] Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. Solving SAT and SAT Modulo Theories: from an Abstract Davis-Putnam-Logemann-Loveland Procedure to DPLL(T). *Journal of the ACM*, 53(6):937–977, November 2006.

[PD07] Knot Pipatsrisawat and Adnan Darwiche. Rsat 2.0: Sat solver description. Technical Report D–153, Automated Reasoning Group, Computer Science Department, UCLA, 2007.