

SAT Modulo Theories: Enhancing SAT with Special-Purpose Algorithms

Robert Nieuwenhuis*

During the last decade SAT techniques have become very successful for practice, with important impact in applications such as electronic design automation. DPLL-based clause-learning SAT solvers work surprisingly well on real-world problems from *many* sources, using a *single, fully automatic, push-button* strategy. Hence, modeling and using SAT is essentially a *declarative* task. On the negative side, propositional logic is a very low level language and hence modeling and encoding *tools* are required. Also, the answer can only be “unsatisfiable” (possibly with a proof) or a model: *optimization* aspects are not as well studied.

For applications such as hard/software verification, more and more complicated and sophisticated encodings into SAT were developed for constraints such as EUF (Equality with Uninterpreted Functions, i.e., congruences), Difference Logic, or other fragments of linear arithmetic.

However, it is nowadays clear that *SAT Modulo Theories* (SMT) is frequently several orders of magnitude faster. The idea is a tight integration of two components: a *theory solver* that can handle conjunctive constraints, and a DPLL-based SAT engine that does the search without knowing the semantics of the literals. Similarly to the constraint propagators in Constraint Programming (CP), the theory solver uses efficient *specialized algorithms* for detecting additional propagations and inconsistencies.

In this talk we first give an overview of our *DPLL(T)* approach to SMT and its implementation in the Barcelogic SMT tool. Then we discuss a longer-term research project, namely the development of SMT technology for hard combinatorial (optimization) problems outside the usual verification applications. Our aim is to obtain the best of several worlds, combining the advantages inherited from SAT: efficiency, robustness and automation (no need for *tuning*) and CP features such as rich modeling languages, special-purpose filtering algorithms (for, e.g., planning, scheduling or timetabling constraints), and sophisticated optimization techniques. We give several examples and discuss the impact of aspects such as first-fail heuristics vs activity-based ones, realistic structured problems vs random or handcrafted ones, and lemma *learning*.

* Technical Univ. of Catalonia (UPC), Barcelona, Spain. Partially supported by Spanish Min. of Science & Innovation, LogicTools-2 project (TIN2007-68093-C02-01). For more details and further references, see 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, 2006.