

A Model-Constructing Satisfiability Calculus

Leonardo de Moura¹ and Dejan Jovanović²

¹ Microsoft Research

² SRI International

Abstract. Considering the theoretical hardness of SAT, the astonishing adeptness of SAT solvers when attacking practical problems has changed the way we perceive the limits of algorithmic reasoning. Modern SAT solvers are based on the idea of *conflict driven clause learning* (CDCL). The CDCL algorithm is a combination of an explicit backtracking search for a satisfying assignment complemented with a deduction system based on Boolean resolution. In this combination, the worst-case complexity of both components is circumvented by the components guiding and focusing each other. The generalization of the SAT problem into the first-order domain is called satisfiability modulo theories (SMT). The common way to solve an SMT problem is to employ a SAT solver to enumerate the assignment of the Boolean abstraction of the formula. The candidate Boolean assignment is then either confirmed or refuted by a *decision procedure* dedicated to reasoning about conjunctions of theory-specific constraints. This framework is commonly called DPLL(T) and is employed by most of the SMT solvers today. Although DPLL(T) at its core relies on a CDCL SAT solver, this SAT solver is only used as a black-box. This can be seen as an advantage since the advances in SAT easily transfer to performance improvements in SMT. On the other hand, in the last few years the idea of direct model construction complemented with conflict resolution has been successfully generalized to fragments of SMT dealing with theories such as linear real arithmetic, linear integer arithmetic, nonlinear arithmetic, and floating-point. All these procedures, although quite effective in their corresponding first-order domains, have not seen a more widespread acceptance due to their limitations in purely Boolean reasoning and incompatibility with DPLL(T). In this talk we describe a *model-constructing satisfiability calculus* (MCSAT) that encompasses all the decision procedures above, including the decision procedures aimed at DPLL(T), while resolving the limitations mentioned above. The MCSAT framework extends DPLL(T) by allowing assignments of variables to concrete values, while relaxing the restriction that decisions, propagations, and explanations of conflicts must be in term of existing atoms.