Don't care in SMT—Building flexible yet efficient abstraction/refinement solvers[1]

Andreas Bauer*, Martin Leucker**, Christian Schallhart**, Michael Tautschnig**

*Computer Sciences Laboratory, Australian National University
** Institut für Informatik, Technische Universität München, Germany

Abstract. This paper describes a method for combining "off-the-shelf" SAT and constraint solvers for building an efficient *Satisfiability Modulo Theories* (SMT) solver for a wide range of theories. Our method follows the abstraction/refinement approach to simplify the implementation of custom SMT solvers. The expected performance penalty by *not* using an interweaved combination of SAT and theory solvers is reduced by *generalising* a Boolean solution of an SMT problem first via assigning *don't care* to as many variables as possible. We then use the generalised solution to determine a thereby smaller constraint set to be handed over to the constraint solver for a background theory. We show that for many benchmarks and real-world problems, this optimisation results in considerably smaller and less complex constraint problems.

The presented approach is particularly useful for assembling a practically viable SMT solver quickly, when neither a suitable SMT solver nor a corresponding incremental theory solver is available. We have implemented our approach in the ABSOLVER framework and applied the resulting solver successfully to an industrial case-study: The verification problems arising in verifying an electronic car steering control system impose non-linear arithmetic constraints, which do not fall into the domain of any other available solver.

1 Introduction

Satisfiability modulo theories (SMT) is the problem of deciding whether a formula in quantifier-free first-order logic is satisfiable with respect to a given *background* theory. For example, one is interested whether the formula $\phi \equiv (i \geq 0) \wedge (\neg(2i + j < 10) \vee (i + j < 5))$ is satisfiable in the theory of integers. In recent years, research on SMT has attracted a lot of attention. SMT solvers for dedicated theories have been developed, such as Yices (Rushby, 2006b), MathSAT (Bozzano et al., 2005), or CVC (Barrett and Berezin, 2004). The growing efficiency of these solvers in their respective domains is witnessed in the annual SMT competition (`http://www.smtcomp.org`).

Amongst others, SMT has its applications in the area of model checking and abstraction (Lahiri et al., 2006), (symbolic) test case generation (Roorda and Claessen, 2006), or in the verification of hybrid control systems (Bauer et al., 2007; Rushby, 2006a), to name just a

[1] Supported by the DFG research grant FORTAS (VE 455/1-1)