

# Orchestrating Satisfiability Engines

Leonardo de Moura

Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA  
leonardo@microsoft.com

**Abstract.** Constraint satisfaction problems arise in many diverse areas including software and hardware verification, type inference, static program analysis, test-case generation, scheduling, planning and graph problems. These areas share a common trait, they include a core component using logical formulas for describing states and transformations between them. The most well-known constraint satisfaction problem is *propositional satisfiability*, SAT, where the goal is to decide whether a formula over Boolean variables, formed using logical connectives can be made *true* by choosing *true/false* values for its variables. Some problems are more naturally described using richer languages, such as arithmetic. A supporting *theory* (of arithmetic) is then required to capture the meaning of these formulas. Solvers for such formulations are commonly called *Satisfiability Modulo Theories* (SMT) solvers.

Software analysis and model-based tools are increasingly complex and multi-faceted software systems. However, at their core is invariably a component using logical formulas for describing states and transformations between system states. In a nutshell, symbolic logic is *the calculus* of computation. The state-of-the art SMT solver, Z3, developed at Microsoft Research, can be used to check the satisfiability of logical formulas over one or more theories. SMT solvers offer a compelling match for software tools, since several common software constructs map directly into supported theories.

Z3 comprises of a collection of symbolic reasoning engines. These engines are combined to address the requirements of each application domain. In this talk, we describe the main challenges in orchestrating the different engines, and the main application domains within Microsoft.