Programming Constraint Services with Z3

Nikolaj Bjørner

Microsoft Research, Redmond, USA

Many program verification, analysis, testing and synthesis queries reduce to solving satisfiability of logical formulas. Yet, there are many applications where satisfiability, and optionally a model or a proof, is insufficient. Examples of useful additional information include interpolants, models that satisfy optimality criteria, generating strategies for solving quantified formulas, enumerating and counting solutions. The lectures describe logical services from the point of view of the Satisfiability Modulo Theories solver Z3. We cover their foundations, algorithmics and ways to put these features to use.

As an overview we provide a few types of queries below.

Type of Query	Query in symbolic form
Satisfiability	$\varphi \rightsquigarrow$ sat, unsat, timeout
Certificates	$\varphi \rightsquigarrow$ model, proof, unsat core
Interpolation	$\varphi[x, y] \rightarrow I[x] \rightarrow \psi[x, z]$
Optimization	$\max x \mid \varphi$
Consequences	$\varphi \rightarrow \varphi_1 \wedge \ldots \wedge \varphi_n$
Sat subsets	$\psi_1 \wedge \psi_2,\ \psi_1 \wedge \psi_3$
Unsat cores	$\neg(\psi_1 \wedge \psi_2),\ \neg(\psi_1 \wedge \psi_3)$
Model counting	$\lvert\{x \mid \varphi\}\rvert$
All models	$Ideal(\varphi), M_1 \models \varphi, M_2 \models \varphi, \ldots$
Model probability	...

The first type of query is the most typical query posed to SMT solvers: whether a formula φ is satisfiable and a corresponding yes/no/don't know answer. This conveys some information, but applications typically need to retrieve additional output. At the very least they may need a certificate. An assignment of values to variables for satisfiable formulas, e.g., a model is very commonly used. Dually, proofs or cores for unsatisfiability can be used for unsatisfiable formulas. Other queries include asking to find models that optimize objective values, finding formulas that are consequences, count or enumerate models.

We plan to cover several basic topics in theorem proving and constraint solving:

- SAT solving basics Elements of modern SAT solvers: Unit propagation, Conflict Directed Clause Learning, Variable heuristics, indexing data-structures, pre- and in-processing.
- Theory Combination Techniques Nelson-Oppen combination, Model-based theory combination.
- Theories - Algorithms and Integration Arithmetic, Uninterpreted functions, Bit-vectors, arrays, algebraic data-types, strings and sequences, collections.

- Quantifier Reasoning E-matching Based Quantifier Instantiation, Model Based Quantifier Instantiation, Quantifier Elimination, Quantifier Satisfiability, Superposition.
- Horn Clauses Modulo Theories Top-down, bottom-up search methods, PDR/IC3 solvers for Horn clause, Symbolic Datalog engines.
- Backbone and consequence finding, MaxSAT/MaxSMT algorithms, optimization modulo SMT, Core extraction and minimization.

We will complement the background with use cases for the different constraint services and in particular highlight ways to use Z3 to answer some of these classes of queries.

Introductory Reading

- Boolean Satisfiability: From Theoretical Hardness to Practical Success. Sh. Malik, L. Zhang. Communications of the ACM; Vol.52 no.8; 2009.
- Satisfiability Modulo Theories: Introduction and Applications. L. de Moura, N. Bjørner. CACM 2011. `http://dl.acm.org/citation.cfmfiid=1995394`
- Applications of SMT solvers: N. Bjørner, L. de Moura. Notes 2013. `http://research.microsoft.com/en-US/people/nbjorner/smt-application-chapter.pdf`
- Horn Clause Solvers for Program Verification: N. Bjørner, A. Gurfinkel, K. McMillan, A. Rybalchenko. 2015. `http://research.microsoft.com/en-US/people/nbjorner/yurifest.pdf`

Most relevant pointers to reading material is available from:

`http://github.com/z3prover/z3/Documents`