Z3 0.1: an SMT Solver

SMT-COMP 2007

Leonardo de Moura and Nikolaj Bjørner

{leonardo, nbjorner}@microsoft.com.

Microsoft Research

Introduction

- *Z3 is a new SMT solver developed at Microsoft Research.*
- It is still under development.
- Version 0.1 is the first external release.
  - Not all features are enabled.
- New external releases are coming soon.
- Managed (.Net) & Unmanaged (C and OCaml) APIs will be available.
- *It replaced Simplify as the default prover for Spec#/Boogie.*
- We are currently integrating Z3 with Pex, SAGE, and SLAM.

Supported Features

- Uninterpreted Functions.
- Linear real and integer arithmetic.
- Extensional Arrays.
- Fixed-size bit-vectors.
- Quantifiers.
- Model generation.
- Coming soon:
  - Improved support for non linear arithmetic.
  - Improved SAT solver.
  - Improved support for quantifiers.
  - Sets & Reachability.

Architecture

- A modern DPLL-based SAT solver.
- A *core theory solver* that handles equalities and uninterpreted functions.
- *Satellite solvers* (for arithmetic, arrays, etc).
- An *E-matching abstract machine* (for quantifiers).
- Very modular: new theories can be added without modifying the core.

Model based theory combination

- *Z3 uses a new theory combination method that incrementally reconciles models maintained by each theory.*
- Use a candidate model M_i for one of the theories $\mathcal{T}_i$ and propagate all equalities implied by the candidate model, hedging that other theories will agree.

  if $M_i \models \mathcal{T}_i \cup \Gamma_i \cup \{u = v\}$ **then** propagate $u = v$.
- If not, use backtracking to fix the model.
- This approach is particularly important in benchmarks with quantifiers.
  - Reason: quantifier instantiation may produce a lot of shared variables.

Quantifiers

- Z3 uses E-graph matching.
- *Z3 uses new algorithms that identify matches on E-graphs incrementally and efficiently.*
  - E-matching code trees.
  - Inverted path index.
- Z3 garbage collects clauses, together with their atoms and terms, that were useless in closing branches.
- Experimental results show substantial performance improvements on ESC/Java & Boogie benchmarks.

Don’t cares

- DPLL(T) based solvers assign a boolean value to potentially all atoms appearing in a goal.
- In practice, several of these atoms are *don’t cares*.
- *Z3 ignores don’t care atoms for expensive inference rules and theories, such as, quantifier instantiation.*

Theories

- Linear arithmetic: based on the algorithms used in Yices.
- Arrays: lazy instantiation of the array axioms.
  - *Don't cares* are used to minimize the number of instantiations.
- Bit-vectors: bit-blast all bit-vector operations but equality.
  - Bit-vector atoms marked as *don't cares* are ignored.
  - Careful encoding of multiplication and division operations.

Pre-processor

- *Z3 uses an efficient and modular pre-processor.*
- Very important for the bit-vector benchmarks.
- Some simplification rules are missing.
  - They will be included in the next releases.

Oops, is there a bug in Z3?

- Z3 uses the Microsoft bignum package.
  - This package is not available for Linux.
  - Z3 uses GMP when compiled on a Linux machine.
- Z3 0.1 has a bug in the interface with GMP.
  - This bug affected some of the bit-vector benchmarks.
- We only noticed this problem after the submission deadline.
  - The bug only occurs on the Linux version.
- We submitted a (fixed) hors concurs version of Z3 to SMT-COMP'07.

Conclusion

- Z3 is an efficient and modular SMT solver.
- It is going to be used in several projects at Microsoft.
- It can solve 99.7% of the benchmarks in SMT-LIB.
- New releases of Z3 will be available at:
  `http://research.microsoft.com/projects/z3`.