Z3 is a Satisfiability Modulo Theories (SMT) solver.
Z3 integrates several decision procedures.
Z3 is used in several program analysis, verification, test-case generation projects at Microsoft.
Z3 1.2 is freely available for academic research:

http://research.microsoft.com/projects/z3
Satisfiability Modulo Theories (SMT)

\[x + 2 = y \Rightarrow f(read(write(a, x, 3), y - 2)) = f(y - x + 1)\]

- Arithmetic
- Array Theory
- Uninterpreted Functions
Main features

- Linear real and integer arithmetic.
- Fixed-size bit-vectors
- Uninterpreted functions
- Extensional arrays
- Quantifiers
- Model generation
- Several input formats (Simplify, SMT-LIB, Z3, Dimacs)
- Extensive API (C/C++, .Net, OCaml)
Z3: Core System Components

Core Theory

Rewriting
Simplification

E-matching

SAT solver

Text

C

.NET

OCaml

Theories

Bit-Vectors

Arithmetic

Arrays

Partial orders

Tuples

Z3: An Efficient SMT Solver
Clients: Program Verification

Hyper-V

Microsoft

Virtualization

Win. Modules

VCC

HAVOC

Boogie

Spec#

Programming System

Bug path

Rustan Leino, Mike Barnet, Michal Moskal, Shaz Qadeer,
Shuvendu Lahiri, Herman Venter, Peter Muller,
Wolfram Schulte, Ernie Cohen

Z3: An Efficient SMT Solver
Quantifiers, quantifiers, quantifiers, ...
- Modeling the runtime
- Frame axioms (“what didn’t change”)
- Users provided assertions (e.g., the array is sorted)
- Prototyping decision procedures (e.g., reachability, heaps, ...)

Solver must be fast in satisfiable instances.

Trade-off between precision and performance.

Candidate (Potential) Models
Clients: Test case generation

Run Test and Monitor

Execution Path

Path Condition

Test Inputs

Known Paths

Constraint System

Unexplored path

Vigilante

Nikolai Tillmann, Peli de Halleux, Patrice Godefroid
Aditya Nori, Jean Philippe Martin, Miguel Castro,
Manuel Costa, Lintao Zhang

Z3: An Efficient SMT Solver
Formulas may be a big conjunction
- Pre-processing step
- Eliminate variables and simplify input format

Incremental: solve several similar formulas
- New constraints are asserted.
  - push and pop: (user) backtracking
  - Lemma reuse

“Small Models”
- Given a formula $F$, find a model $M$, that minimizes the value of the variables $x_0 ... x_n$
Z3 is part of SDV 2.0 (Windows 7)

It is used for:

- Predicate abstraction (c2bp)
- Counterexample refinement (newton)
Z3 & Static Driver Verifier

- All-SAT
  - Fast Predicate Abstraction
- Unsatisfiable cores
  - Why the abstract path is not feasible?
More Microsoft clients

- Bounded model-checking of model programs
- Termination
- Security protocols
- Business application modeling
- Cryptography
- Model Based Testing (SQL-Server)
- *Your killer-application here*
Some Technical goodies

- Model-based Theory Combination
  - *How to efficiently combine theory solvers?*
  - Use models to control Theory Combination.

- E-matching abstract machine
  - Term indexing data-structures for incremental matching modulo equalities.

- Relevancy propagation
  - Use Tableau advantages with DPLL engine

Z3: An Efficient SMT Solver
Given arrays:

bool a1[bool];
bool a2[bool];
bool a3[bool];
bool a4[bool];

All can be distinct.

Add:

bool a5[bool];

Two of a1,..,a5 must be equal.
Future/Current Work

- Coming soon (Z3 2.0):
  - Proofs & Unsat cores
  - Superposition Calculus
  - Decidable Fragments
  - Machine Learning
  - Non linear arithmetic (Gröbner Bases)
  - Inductive Datatypes
  - Improved Array & Bit-vector theories
- Several performance improvements
- More “customers” & Applications
Z3 is a new SMT solver from Microsoft Research.

Z3 is used in several projects.

Z3 is freely available for academic research:
- [http://research.microsoft.com/projects/z3](http://research.microsoft.com/projects/z3)