SMT Solvers

Program Verification - Laborator

FMI · Denisa Diaconescu · Spring 2022

Z3

- Is a theorem prover from Microsoft Research
- It can be used to check satisfiability of logical formulas over one or more theories.
- An efficient SMT solver
- Check Z3's Github page

Z3

- Many program analysis, verification and test tools solve problems that can be reduced to logical formulas and transformations between logical formulas at their core.
- There are many verifiers built on top of the Z3, e.g. Dafny.

Z3- little engines of proof

Bit-Arith
Simplifier
Simplex
Quant. Instances
Datatypes
SAT Solver
Congruence Closure
Q- Elim
Arrays
Super-position
MBQI
User-Theories

source: Lecture Notes "Modern Satisfiability Modulo Theories Solvers in Program Analysis" by N. Bjørner

Input formats

Text:

- SMT-LIB2 – main exchange format for SMT solvers
- Log – low-level for replay
- Datalog engine – A Datalog format for the fixed-point
- DIMACS – a format for propositional SAT

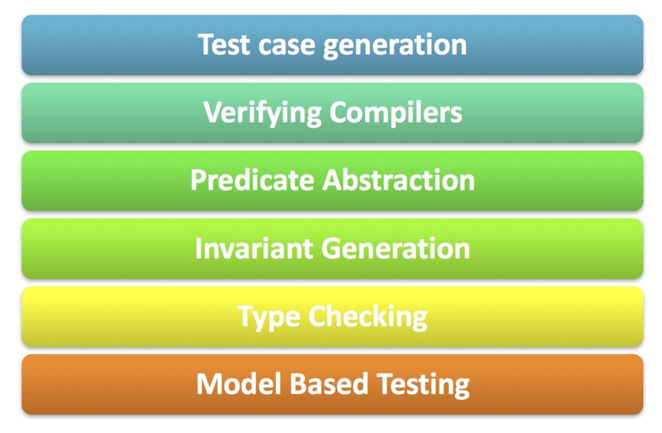
Programmatic:

- C – API functions exposed for C
- Ocaml – Ocaml wrapper around C API
- .NET – .NET wrapper around C API
- Python
- Scala

Applications of Z3

Decision Engine for Software

Applications:



Test case generation

Verifying Compilers

Predicate Abstraction

Invariant Generation

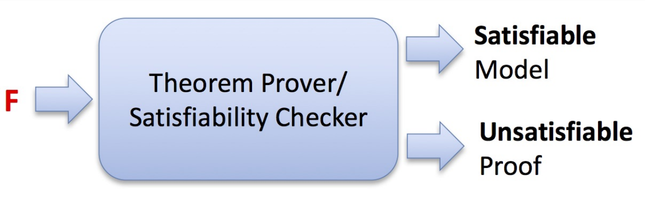
Type Checking

Model Based Testing

source: Lecture Notes "Modern Satisfiability Modulo Theories Solvers in Program Analysis" by N.

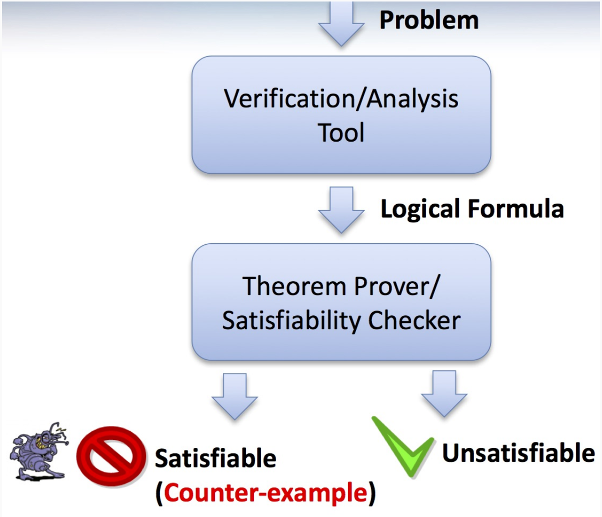
Theorem Provers/Satisfiability checkers

F is valid iff $\neg$F is unsatisfiable



source: Lecture Notes "Modern Satisfiability Modulo Theories Solvers in Program Analysis" by N. Bjørner

Template of analysis tool


Problem
Verification/Analysis Tool
Logical Formula
Theorem Prover/ Satisfiability Checker
Satisfiable (Counter-example)
Unsatisfiable

source: Lecture Notes "Modern Satisfiability Modulo Theories Solvers in Program Analysis" by N. Bjørner

Z3 theories

- Uninterpreted functions
- Arithmetic (linear)
- Bit-vectors
- Algebraic data-types
- Arrays
- Polynomial arithmetic

Z3 tutorial

Exercise 1: Read the Z3 Guide and experiment with Z3.

https://web.archive.org/web/20210119175613/https://rise4fun.com/Z3/tutorial/guide

Read about

- Basic Commands
- Propositional Logic
- Uninterpreted functions and constants
- Arithmetic
- Arrays

You can use the online Z3

https://compsys-tools.ens-lyon.fr/z3/index.php

Z3

Exercise 2: We define the following operation between any two integers:

$$x \ominus y := \max(0, (x - y))$$

Write a SMT-LIB2 formulation in Z3 for checking if the following formula is valid:

$$(x \ominus a) + a \leq x$$

Z3

Exercise 3: Solve the following system of equations over the real numbers:

$$\begin{cases} -x - 3y + 2z = 1 \\ x - y - 6z = 1 \\ 2x + y - 10z = 3 \end{cases}$$

Write a SMT-LIB2 formulation in Z3 for solving this problem.

Further Reading

- Here are some nice examples of problems solved in SMT solvers.