Understanding and Evaluating SMT Solvers

Hans-Jörg Schurr
KU Leuven
October 20, 2025

Part I

Understanding SMT Solvers

An Example in a Bottle

1. We produce 1L, 2L, and 3L bottles.
2. The price of a bottle is the volume plus four times the wall thickness (in mm).
3. The price must be less than 4€.
4. If the new machine is broken, we cannot produce 3L bottles, and the wall thickness must be more than 1mm.
5. The new machine is broken.
6. For all bottle sizes, the all thickness can at most be the volume in liters.

An Example in a Bottle

1. We produce 1L, 2L, and 3L bottles.
2. The price of a bottle is the volume plus four times the wall thickness (in mm).
3. The price must be less than 4€.
4. If the new machine is broken, we cannot produce 3L bottles, and the wall thickness must be more than 1mm.
5. The new machine is broken.
6. For all bottle sizes, the all thickness can at most be the volume in liters.

To solve this, we must understand:

- Logic: **and, if then**
- Arithmetic: **four times the wall thickness**
- Universal statements: **for all**

An Example in a Bottle

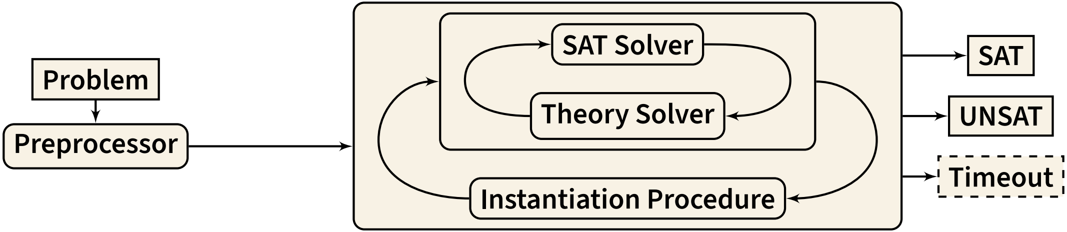
1. We produce 1L, 2L, and 3L bottles.
2. The price of a bottle is the volume plus four times the wall thickness (in mm).
3. The price must be less than 4€.
4. If the new machine is broken, we cannot produce 3L bottles, and the wall thickness must be more than 1mm.
5. The new machine is broken.
6. For all bottle sizes, the all thickness can at most be the volume in liters.

To solve this, we must understand:

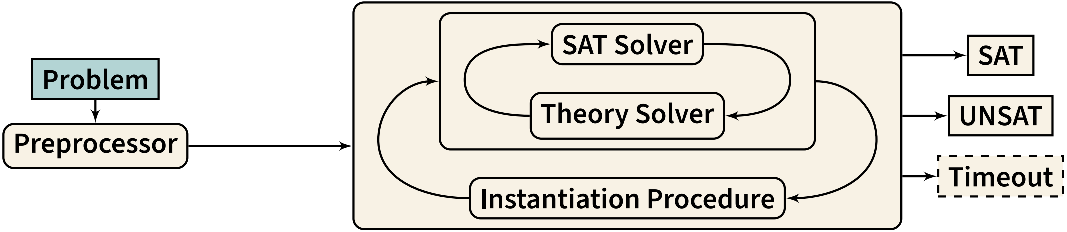
- Logic: **and, if then**
- Arithmetic: **four times the wall thickness**
- Universal statements: **for all**

This is **Satisfiability Modulo Theories**

SMT Solving As A Diagram



SMT Solving As A Diagram



An Example: Problem Specification

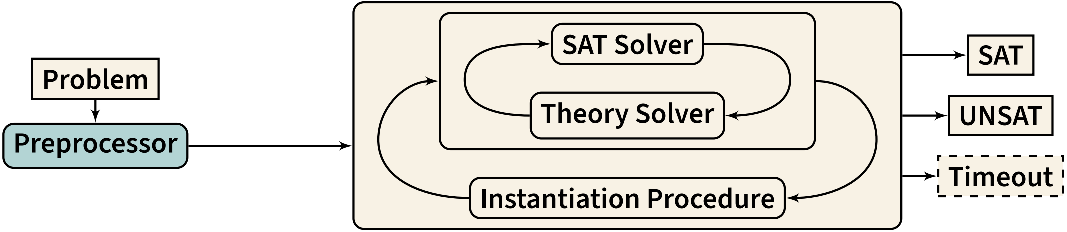
1. We produce 1L, 2L, and 3L bottles.
2. The price of a bottle is the volume plus four times the wall thickness (in mm).
3. The price must be less than 4€.
4. If the new machine is broken, we cannot produce 3L bottles, and the wall thickness must be more than 1mm.
5. The new machine is broken.
6. For all bottle sizes, the wall thickness in millimetre can at most be the volume in liters.

An Example: Problem Specification

1. We produce 1L, 2L, and 3L bottles.
2. The price of a bottle is the volume plus four times the wall thickness (in mm).
3. The price must be less than 4€.
4. If the new machine is broken, we cannot produce 3L bottles, and the wall thickness must be more than 1mm.
5. The new machine is broken.
6. For all bottle sizes, the wall thickness in millimetre can at most be the volume in liters.

1. $v = 1 \lor v = 2 \lor v = 3$
2. $p = v + 2t$
3. $p < 4$
4. $b \to (v \neq 3 \land t > 1)$
5. b
6. $\forall z. v = z \to t \leq z$

SMT Solving As A Diagram



An Example: Preprocessing

1. $v = 1 \lor v = 2 \lor v = 3$
2. $v + 2t < p$
3. $p = 4$
4. $b \to (\neg v = 3 \land t > 1)$

5. b
6. $\forall z.\, v = z \to t \leq z$

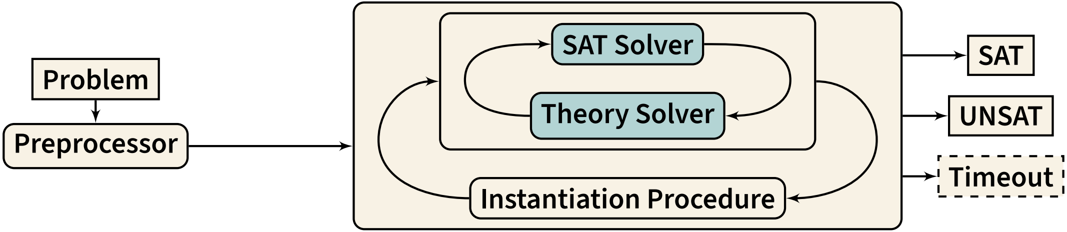
An Example: Preprocessing

1. $v = 1 \lor v = 2 \lor v = 3$
2. $v + 2t < p$
3. $p = 4$
4. $b \rightarrow (\neg v = 3 \land t > 1)$
5. b
6. $\forall z.\, v = z \rightarrow t \leq z$

1. $v = 1 \lor v = 2 \lor v = 3$
2. $v + 2t < 4$
3.
4. $\neg b \lor \neg v = 3$

   $\neg b \lor 1 < t$
5. b
6. $\forall z.\, \neg v = z \lor \neg(z < t)$

SMT Solving As A Diagram



An Example: The Ground Solver

- $v = 1 \lor v = 2 \lor v = 3$
- $v + 2t < 4$
- $\neg b \lor \neg v = 3$
- $\neg b \lor 1 < t$
- b
- $\forall z. \neg v = z \lor \neg(z < t)$

An Example: The Ground Solver

- $v = 1 \lor v = 2 \lor v = 3$
- $v + 2t < 4$
- $\neg b \lor \neg v = 3$
- $\neg b \lor 1 < t$
- b
- $\cancel{\forall z. \neg v = z \lor \neg(z < t)}$

SAT Problem

- $p_1 \lor p_2 \lor p_3$
- p_4
- $\neg b \lor \neg p_3$
- $\neg b \lor p_5$
- b

An Example: The Ground Solver

- $v = 1 \lor v = 2 \lor v = 3$
- $v + 2t < 4$
- $\neg b \lor \neg v = 3$
- $\neg b \lor 1 < t$
- b
- $\cancel{\forall z. \neg v = z \lor \neg(z < t)}$

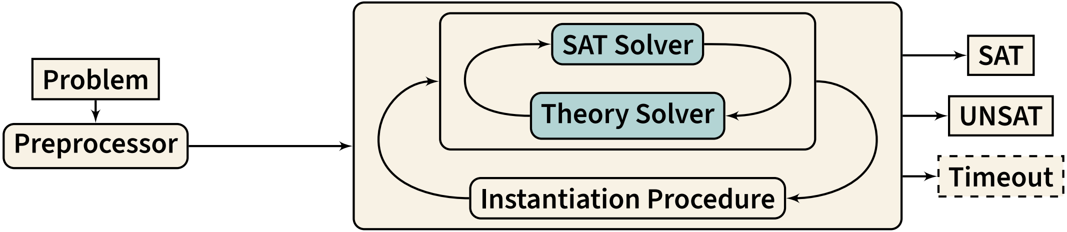
SAT Problem

- $p_1 \lor p_2 \lor p_3$
- p_4
- $\neg b \lor \neg p_3$
- $\neg b \lor p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

SMT Solving As A Diagram



An Example: The SAT Solver and the Theory Solver

SAT Problem

- $p_1 \vee p_2 \vee p_3$
- p_4
- $\neg b \vee \neg p_3$
- $\neg b \vee p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

An Example: The SAT Solver and the Theory Solver

SAT Problem

- $p_1 \vee p_2 \vee p_3$
- p_4
- $\neg b \vee \neg p_3$
- $\neg b \vee p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

SAT Solver

I pick b, p_2, p_4, and p_5 ☺️

An Example: The SAT Solver and the Theory Solver

SAT Problem

- $p_1 \vee p_2 \vee p_3$
- p_4
- $\neg b \vee \neg p_3$
- $\neg b \vee p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

SAT Solver

I pick b, p_2, p_4, **and** p_5 😊

Linear Arithmetic Solver

1. I get $v = 2$, $v + 2t < 4$, **and** $t > 1$

An Example: The SAT Solver and the Theory Solver

SAT Problem

- $p_1 \vee p_2 \vee p_3$
- p_4
- $\neg b \vee \neg p_3$
- $\neg b \vee p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

SAT Solver

I pick b, p_2, p_4, and p_5 😊

Linear Arithmetic Solver

1. I get $v = 2$, $v + 2t < 4$, and $t > 1$
2. Doesn't work:
   $\neg v = 2 \vee \neg(v + 2t < 4) \vee \neg t > 1$ 🤔

An Example: The SAT Solver and the Theory Solver

SAT Problem

- $p_1 \lor p_2 \lor p_3$
- p_4
- $\neg b \lor \neg p_3$
- $\neg b \lor p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

SAT Solver

I pick b, p_2, p_4, and p_5 😊

Linear Arithmetic Solver

1. I get $v = 2$, $v + 2t < 4$, and $t > 1$
2. Doesn't work:
   $\neg v = 2 \lor \neg(v + 2t < 4) \lor \neg t > 1$ 🤔

SAT Solver

I have to pick b, p_1, p_4, and p_5 🤞

An Example: The SAT Solver and the Theory Solver

SAT Problem

- $p_1 \lor p_2 \lor p_3$
- p_4
- $\neg b \lor \neg p_3$
- $\neg b \lor p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

SAT Solver

I pick b, p_2, p_4, and p_5 😊

Linear Arithmetic Solver

1. I get $v = 2$, $v + 2t < 4$, and $t > 1$
2. Doesn't work:
   $\neg v = 2 \lor \neg(v + 2t < 4) \lor \neg t > 1$ 🤔

SAT Solver

I have to pick b, p_1, p_4, and p_5 🤞

Linear Arithmetic Solver

1. I get $v = 1$, $v + 2t < 4$, and $t > 1$

An Example: The SAT Solver and the Theory Solver

SAT Problem

- $p_1 \vee p_2 \vee p_3$
- p_4
- $\neg b \vee \neg p_3$
- $\neg b \vee p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

SAT Solver

I pick b, p_2, p_4, and p_5 😊

Linear Arithmetic Solver

1. I get $v = 2$, $v + 2t < 4$, and $t > 1$
2. Doesn't work:
   $\neg v = 2 \vee \neg(v + 2t < 4) \vee \neg t > 1$ 🤔

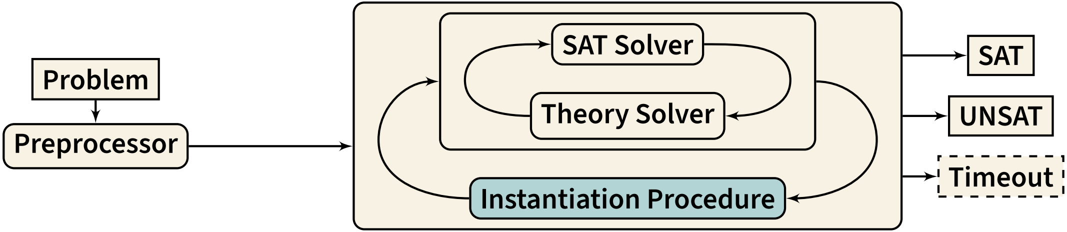
SAT Solver

I have to pick b, p_1, p_4, and p_5 🤞

Linear Arithmetic Solver

1. I get $v = 1$, $v + 2t < 4$, and $t > 1$
2. That works! 🎉

SMT Solving As A Diagram



An Example: Quantifier Instantiation

SAT Problem

- $p_1 \vee p_2 \vee p_3$
- p_4
- $\neg b \vee \neg p_3$
- $\neg b \vee p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

Instantiation Procedure

- I have $\forall z. \neg v = z \vee \neg z < t$

An Example: Quantifier Instantiation

SAT Problem

- $p_1 \vee p_2 \vee p_3$
- p_4
- $\neg b \vee \neg p_3$
- $\neg b \vee p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

Instantiation Procedure

- I have $\forall z. \neg v = z \vee \neg z < t$
- What happens if I pick $z \leftarrow 1$? 😈

An Example: Quantifier Instantiation

SAT Problem

- $p_1 \vee p_2 \vee p_3$
- p_4
- $\neg b \vee \neg p_3$
- $\neg b \vee p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

Instantiation Procedure

- I have $\forall z. \neg v = z \vee \neg z < t$
- What happens if I pick $z \leftarrow 1$? 😈
- That's $\neg v = 1 \vee \neg t > 1$

An Example: Quantifier Instantiation

SAT Problem

- $p_1 \vee p_2 \vee p_3$
- p_4
- $\neg b \vee \neg p_3$
- $\neg b \vee p_5$
- b

Theory Literals

- $p_1 := v = 1, p_2 := v = 2, p_3 := v = 3$
- $p_4 := v + 2t < 4$
- $p_5 := t > 1$

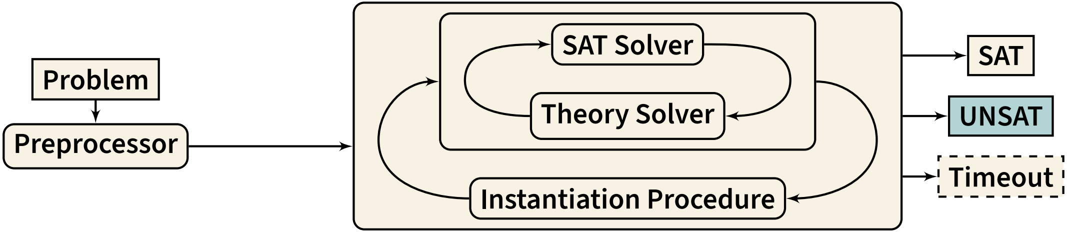
Instantiation Procedure

- I have $\forall z. \neg v = z \vee \neg z < t$
- What happens if I pick $z \leftarrow 1$? 😈
- That's $\neg v = 1 \vee \neg t > 1$

SAT Solver

- That's $\neg p_1 \vee \neg p_5$
- Oh no 😢

SMT Solving As A Diagram



Part II

Using SMT Solvers

Some Solvers You Can Try (a Biased List)

CVC5

- Industrial strength
- Supports everything
- `cvc5.github.io`

veriT

- Small solver
- Excellent proofs, good quantifier support
- `www.verit-solver.org`

Z3

- Very established
- Also supports everything
- `https://github.com/Z3Prover/z3`

Bitwuzla

- Specialized on bit-vectors, and floating-points
- Very fast
- `bitwuzla.github.io`

SMT-LIB as a Common Language

```
(set-logic LRA)
```

SMT-LIB as a Common Language

```
(set-logic LRA)
(declare-const v Real) (declare-const t Real)
(declare-const b Bool)
```

SMT-LIB as a Common Language

```
(set-logic LRA)
(declare-const v Real) (declare-const t Real)
(declare-const b Bool)
(assert (or (= v 1) (= v 2) (= v 3)))
(assert (< (+ v (* 2 t)) p))
(assert (= p 4))
(assert (=> b (and (not (= v 3)) (> t 1))))
(assert b)
(assert (forall ((z Real)) (=> (= v z) (<= t z))))
```

SMT-LIB as a Common Language

```
(set-logic LRA)
(declare-const v Real) (declare-const t Real)
(declare-const b Bool)
(assert (or (= v 1) (= v 2) (= v 3)))
(assert (< (+ v (* 2 t)) p))
(assert (= p 4))
(assert (=> b (and (not (= v 3)) (> t 1))))
(assert b)
(assert (forall ((z Real)) (=> (= v z) (<= t z))))
(check-sat)
```

SMT-LIB as a Common Language

```
(set-logic LRA)
(declare-const v Real) (declare-const t Real)
(declare-const b Bool)
(assert (or (= v 1) (= v 2) (= v 3)))
(assert (< (+ v (* 2 t)) p))
(assert (= p 4))
(assert (=> b (and (not (= v 3)) (> t 1))))
(assert b)
(assert (forall ((z Real)) (=> (= v z) (<= t z))))
(check-sat)
```

- 👩‍✈️ Most SMT solvers support SMT-LIB
- 🏅 Annual competition (SMT-COMP)
- 📚 Large benchmark library

What SMT Solvers Can Do

All

- read SMT-LIB
- solve a subset of official theories
  - uninterpreted functions
  - (linear) arithmetic
  - arrays
  - algebraic data-types
  - strings
  - bitvectors
  - floating-point arithmetic
  - quantifiers
- solve some proprietary theories, or extensions
  - bags, sets, higher-order quantifiers, ...

What SMT Solvers Can Do

All

- read SMT-LIB
- solve a subset of official theories
  - uninterpreted functions
  - (linear) arithmetic
  - arrays
  - algebraic data-types
  - strings
  - bitvectors
  - floating-point arithmetic
  - quantifiers
- solve some proprietary theories, or extensions
  - bags, sets, higher-order quantifiers, ...

Most

- give you models
- give you cores
- have some API

Some

- give you proofs
- give you interpolants
- can optimize
- have a high-assurance mode
  - cvc5
- have a tactics language
  - Z3

Part III

Evaluating SMT Solvers

Fun Facts

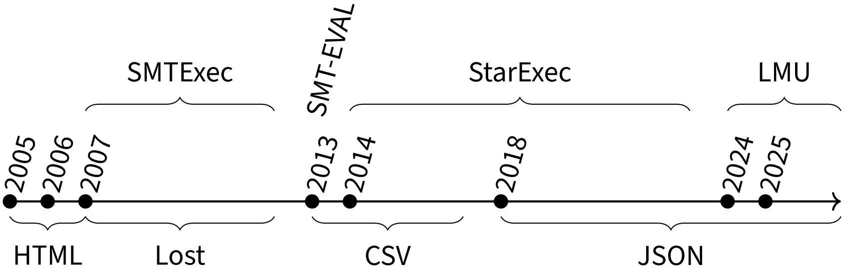
- 495,177 benchmarks
- 34,614,311 queries
- 287 families
- 2,630,828 queries in one benchmark
- a 1.9GB query
- up to 3,515,188 open parentheses

The SMT-LIB Benchmarks

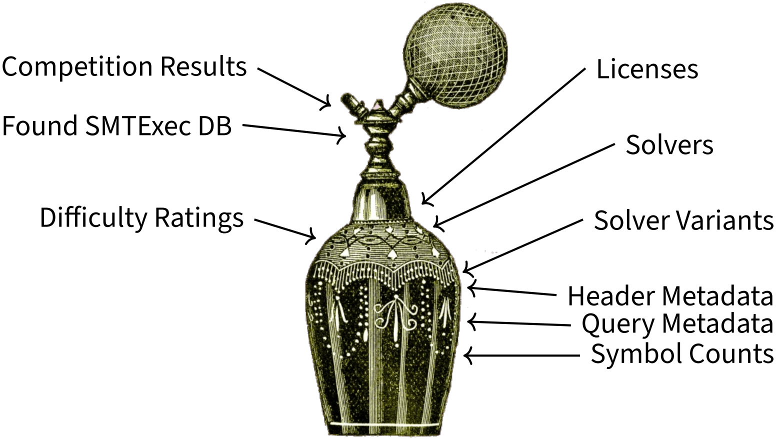
Fun Facts

- 495,177 benchmarks
- 34,614,311 queries
- 287 families
- 2,630,828 queries in one benchmark
- a 1.9GB query
- up to 3,515,188 open parentheses

SMT Competitions



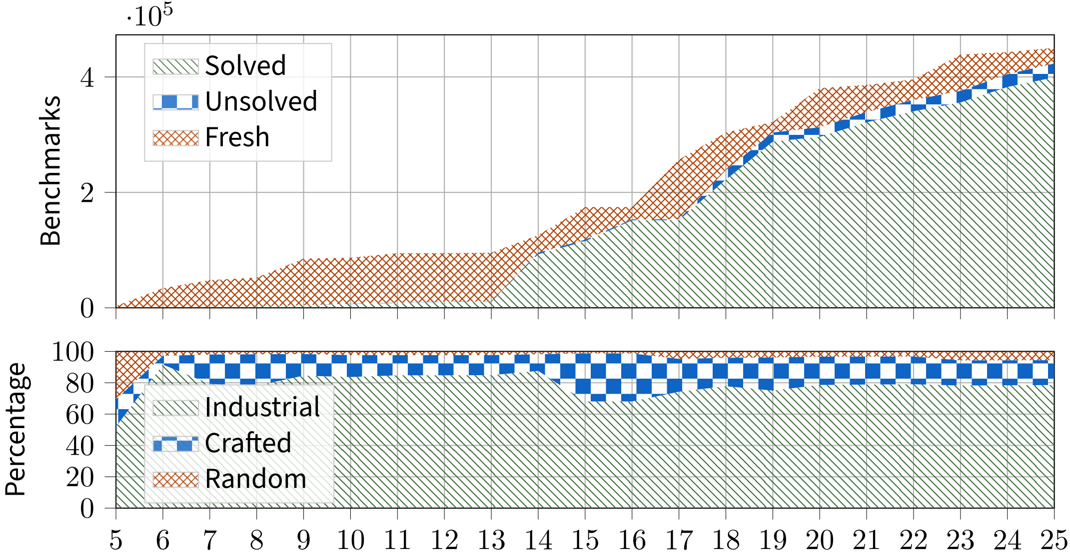
Botteling Everything Up



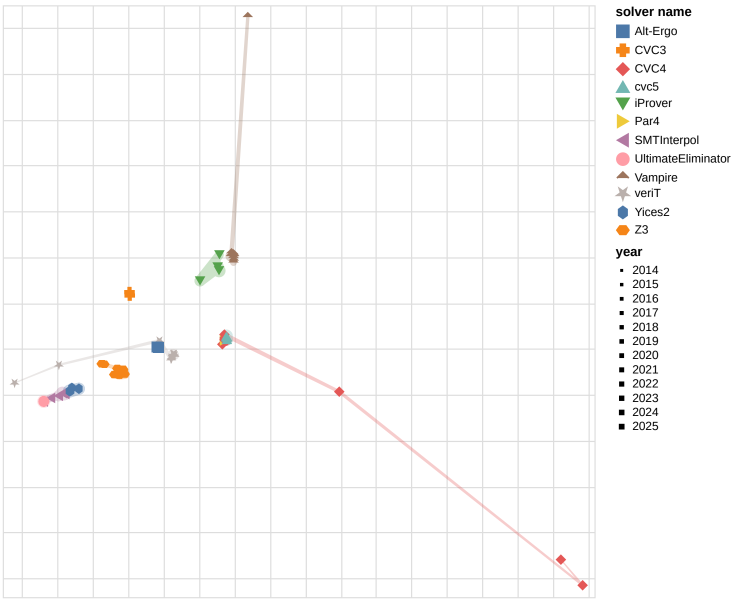
SQLite

`explore.smt-lib.org`

The SMT-LIB Benchmarks Over Time



Isomap: UF



solver name
Alt-Ergo
CVC3
CVC4
cvc5
iProver
Par4
SMTInterpol
UltimateEliminator
Vampire
veriT
Yices2
Z3
year
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025

Conflicting Instances! (Reynolds et al. 2016, Barbosa et al. 2017)

Idea

1. Given: ground model M
2. $F := t = u$ or $F := t \neq u$ with variables V
3. Find ground substitution σ on V s.t.
4. $M \wedge F\sigma \models_{EUF} \bot$

+ Like a theory lemma!
+ Generalization of E-matching.
− Often fails.

Example

1. Model: $a = b, g(a) \neq f(b)$
2. and $\forall x.\, g(x) = f(x)$

Conflicting Instances! (Reynolds et al. 2016, Barbosa et al. 2017)

Idea

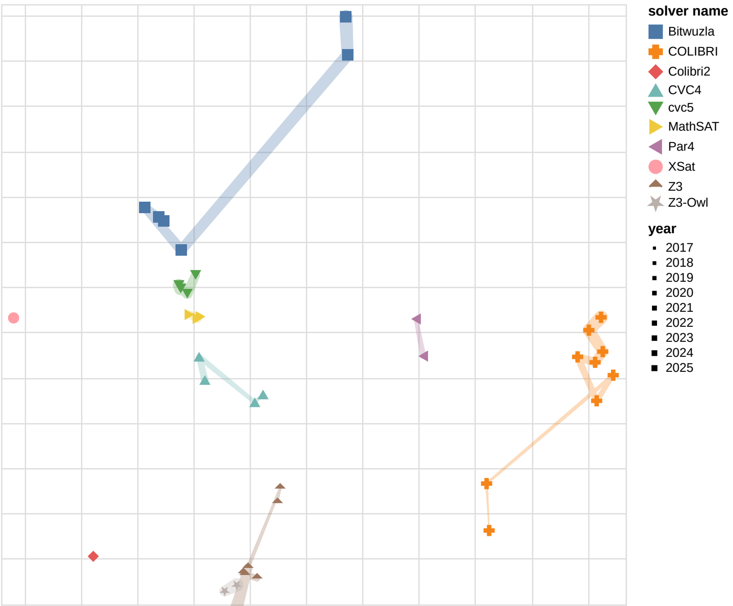
1. Given: ground model M
2. $F := t = u$ or $F := t \neq u$ with variables V
3. Find ground substitution σ on V s.t.
4. $M \wedge F\sigma \models_{EUF} \perp$

+ Like a theory lemma!
+ Generalization of E-matching.
– Often fails.

Example

1. Model: $a = b$, $g(a) \neq f(b)$
2. and $\forall x.\, g(x) = f(x)$
3. $\sigma = \{x \leftarrow a\}$
4. gives us: $g(a) = f(a)$

Isomap: QF_FP

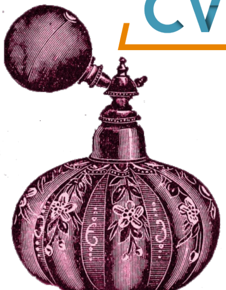


solver name
Bitwuzla
COLIBRI
Colibri2
CVC4
cvc5
MathSAT
Par4
XSat
Z3
Z3-Owl
year
2017
2018
2019
2020
2021
2022
2023
2024
2025

Thank You!

Questions? Benchmarks?

CVC5



IOWA