The International Satisfiability Modulo Theories Competition (SMT-COMP)

Tjark Weber

UPPSALA
UNIVERSITET

Advancing Verification Competitions as a Scientific Method

February 18, 2019

Formalism

Satisfiability Modulo Theories =

propositional satisfiability + background theories (+ quantifiers)

Formalism

Satisfiability Modulo Theories =

propositional satisfiability + background theories (+ quantifiers)

Example (SMT formula)

$$x \leq y \ \wedge \ y \leq x \ \wedge \ P(f(x) - f(y)) \ \wedge \ \neg P(0)$$

Formalism (cont.)

Background theories:

- EUF $x = y \implies f(x) = f(y)$
- Arithmetic $y < 0 \implies x + y < x$
- Arrays $\mathsf{select}(\mathsf{store}(a, i, x), i) = x$
- Bit-vectors $2 \cdot x = x \ll 1$
- ...

A rich language with lots of applications in program analysis, testing, verification, and other areas.

Problems

- Satisfiability

  For instance, is

  $$x \leq y \ \wedge \ y \leq x \ \wedge \ P(f(x) - f(y)) \ \wedge \ \neg P(0)$$

  satisfiable?

Problems

- Satisfiability

  For instance, is

  $$x \leq y \ \wedge \ y \leq x \ \wedge \ P(f(x) - f(y)) \ \wedge \ \neg P(0)$$

  satisfiable?

- Incremental satisfiability

  Answer multiple satisfiability queries, simulating an on-line interaction with applications that generate and retract formulas on the fly.

Problems

- Satisfiability

  For instance, is

  $$x \leq y \ \wedge \ y \leq x \ \wedge \ P(f(x) - f(y)) \ \wedge \ \neg P(0)$$

  satisfiable?

- Incremental satisfiability

  Answer multiple satisfiability queries, simulating an on-line interaction with applications that generate and retract formulas on the fly.

- Unsatisfiable core generation

  Find a small, but still unsatisfiable subset of input formulas.

Input Format

Problems are presented to solvers in SMT-LIB format.

This text-based language is widely supported by SMT solvers.

SMT-LIB defines

- concrete syntax for input formulas, and
- a command-based scripting language.

Input Format (cont.)

Example (SMT-LIB benchmark)

```
(set-info :smt-lib-version 2.6)
(set-logic QF_UFLIA)
(set-info :status unsat)
(declare-fun x () Int)
(declare-fun y () Int)
(declare-fun f (Int) Int)
(declare-fun P (Int) Bool)
(assert (and (<= x y) (<= y x) (P (- (f x) (f y))) (not (P 0))))
(check-sat)
(exit)
```

A Short History

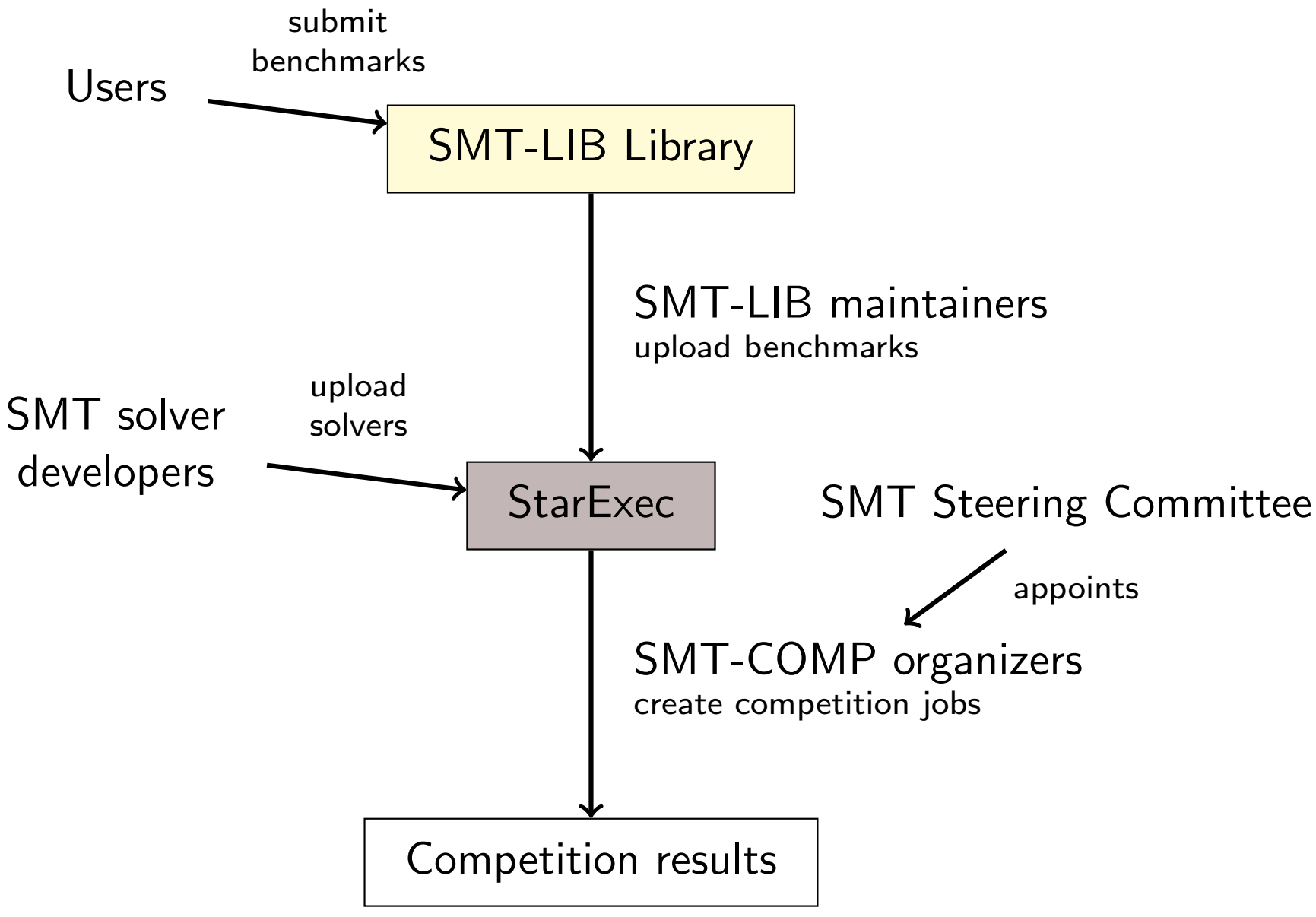
SMT-COMP was instituted in 2005.

It is an annual event[1] that is affiliated with the International Workshop on Satisfiability Modulo Theories.

2019: 14th SMT-COMP

[1] In 2013, an evaluation was conducted instead of a competition.

Organization



Number of Participants, Tracks, Benchmarks

SMT-COMP 2018

Participants: 25 solvers (+6 hors concours)

Tracks: 3 tracks, 115 divisions

Benchmarks: 342,498

Job pairs: 1,776,062

Total wall-clock time: $>$ 7.6 years

Evaluation and Result Validation

Solvers are applied to benchmarks on  StarExec.

Competition scores for each solver are based on the number of (in)correct answers (or the size of unsatisfiable cores), and on the time that it took to find them.

Preliminary results are made publicly available about two weeks before the official result presentation. Solver developers are encouraged to report irregularities.

Dissemination of Results

- StarExec (solvers, benchmarks, raw job data)
- `smt-comp.org` (result tables)
- `smt-comp@cs.nyu.edu` (announcements)
- Presentation at the SMT Workshop
- 2014, 2018: FLoC Olympic Games
- Competition reports

Impact

- Adoption of the SMT-LIB format
- Guidance for users and developers
- Further advances in SMT solving
- Recognition for solver developers

Impact (cont.)

A Virtuous Circle