# 13th International Satisfiability Modulo Theories Competition

## SMT-COMP 2018

Matthias Heizmann  Aina Niemetz
Giles Reger  Tjark Weber

# Outline

- Design and scope
  - Main changes from last year's competition
- Short presentation of solvers
  - Alt-Ergo, Boolector, Ctrl-Ergo, CVC4, OpenSMT, SMTInterpol, SPASS-SATT, Yices
- Selected results

# Design and Scope

# Background

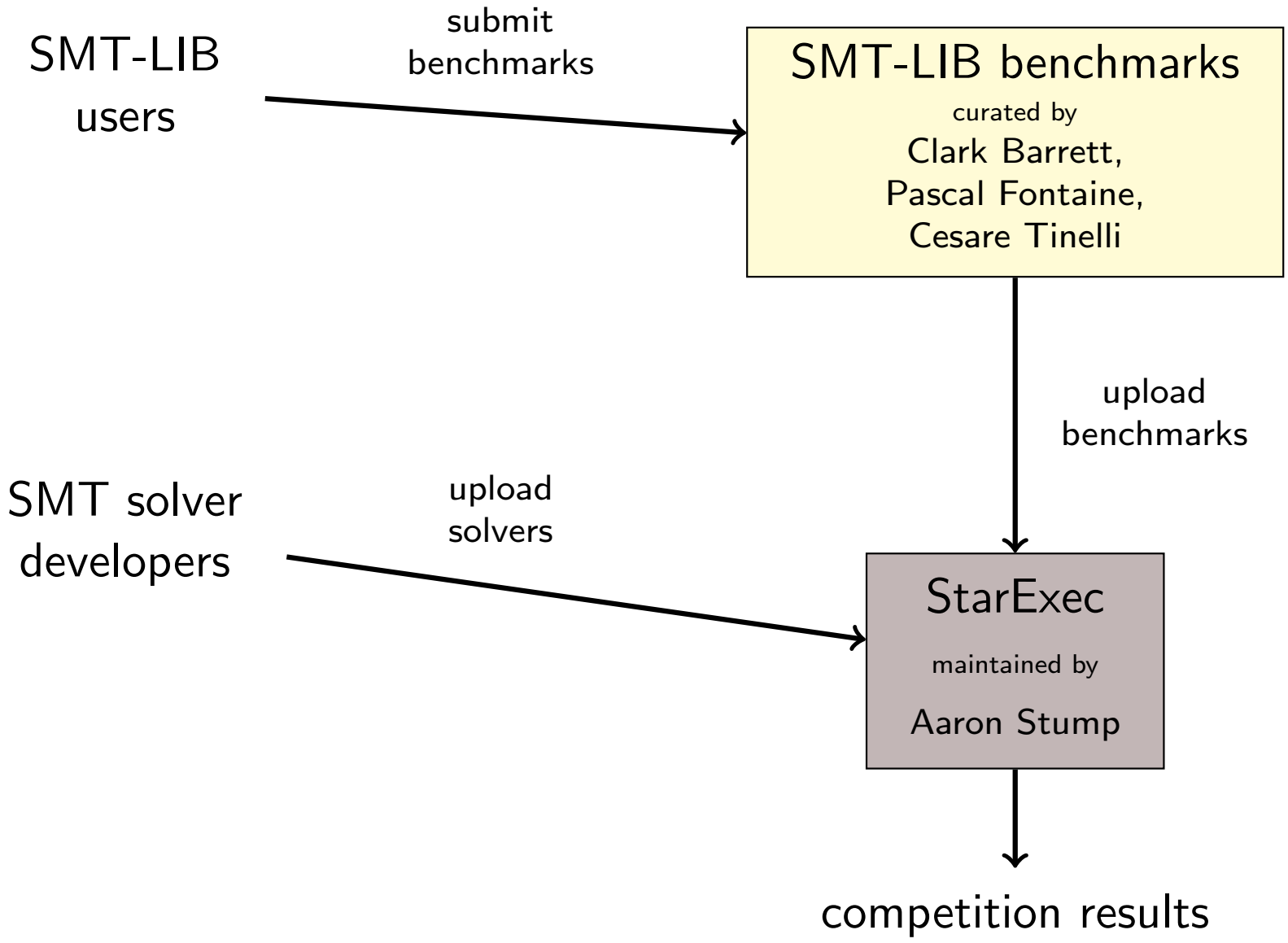
SMT-COMP is an annual competition between SMT solvers.

It was first held in 2005

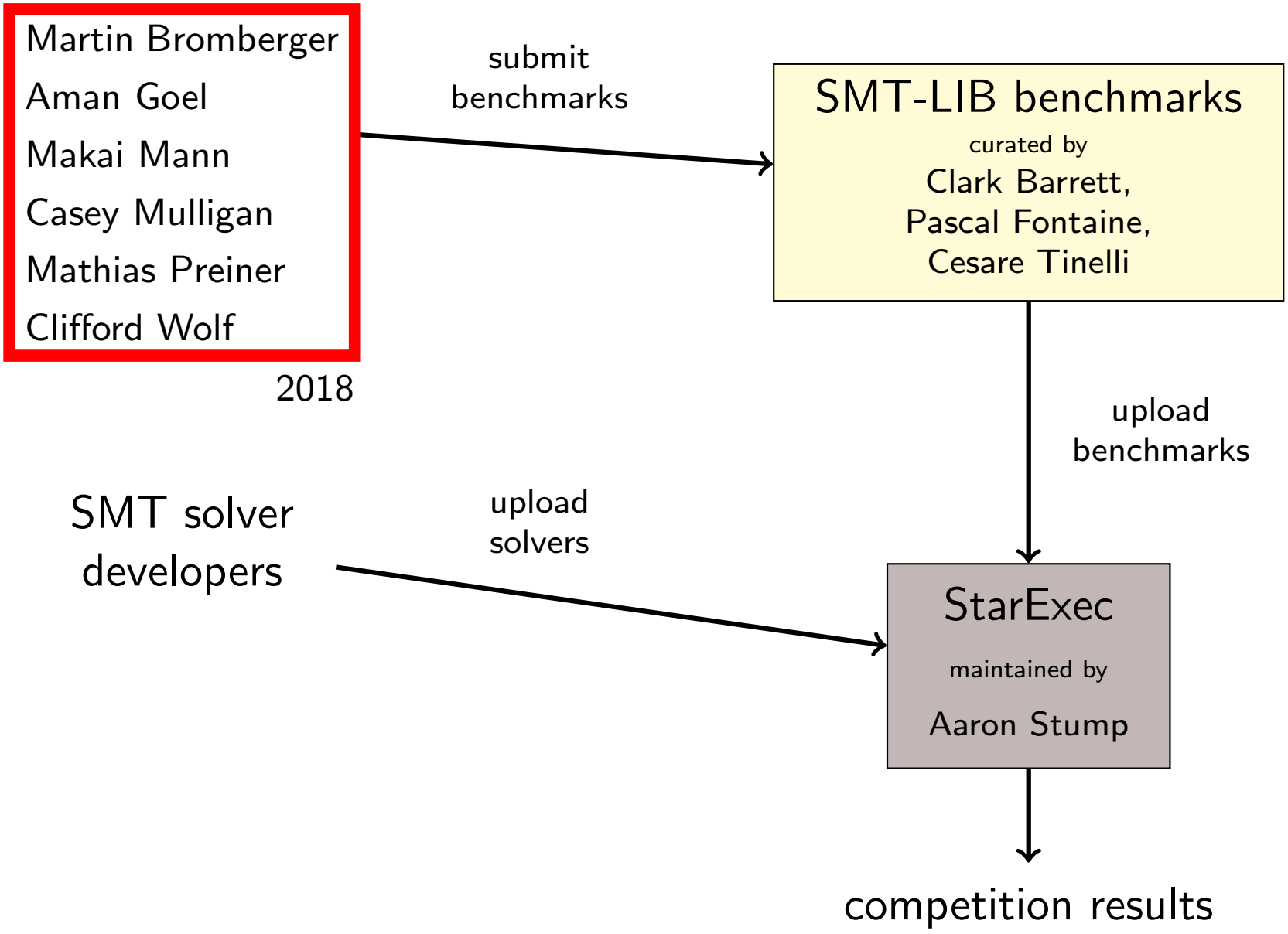
- to spur adoption of the common, community-designed SMT-LIB format, and
- to spark further advances in SMT by stimulating improvement in solver implementations.

It has evolved into the world's largest* ATP competition.

# SMT-COMP – Procedure



# SMT-COMP – Procedure



# Main Track

Main Track benchmark

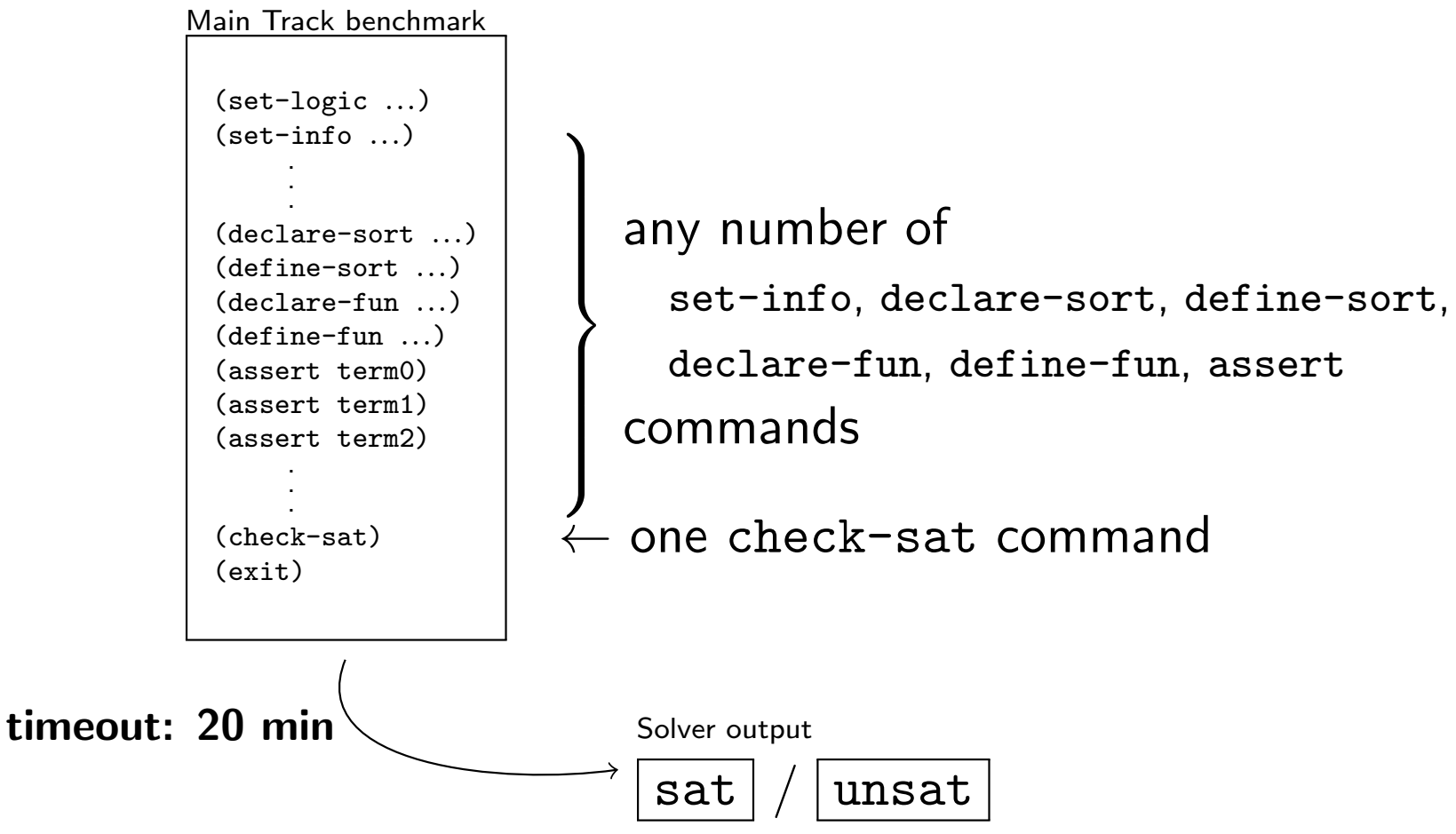
```
(set-logic ...)
(set-info ...)
     .
     .
     .
(declare-sort ...)
(define-sort ...)
(declare-fun ...)
(define-fun ...)
(assert term0)
(assert term1)
(assert term2)
     .
     .
     .
(check-sat)
(exit)
```

any number of

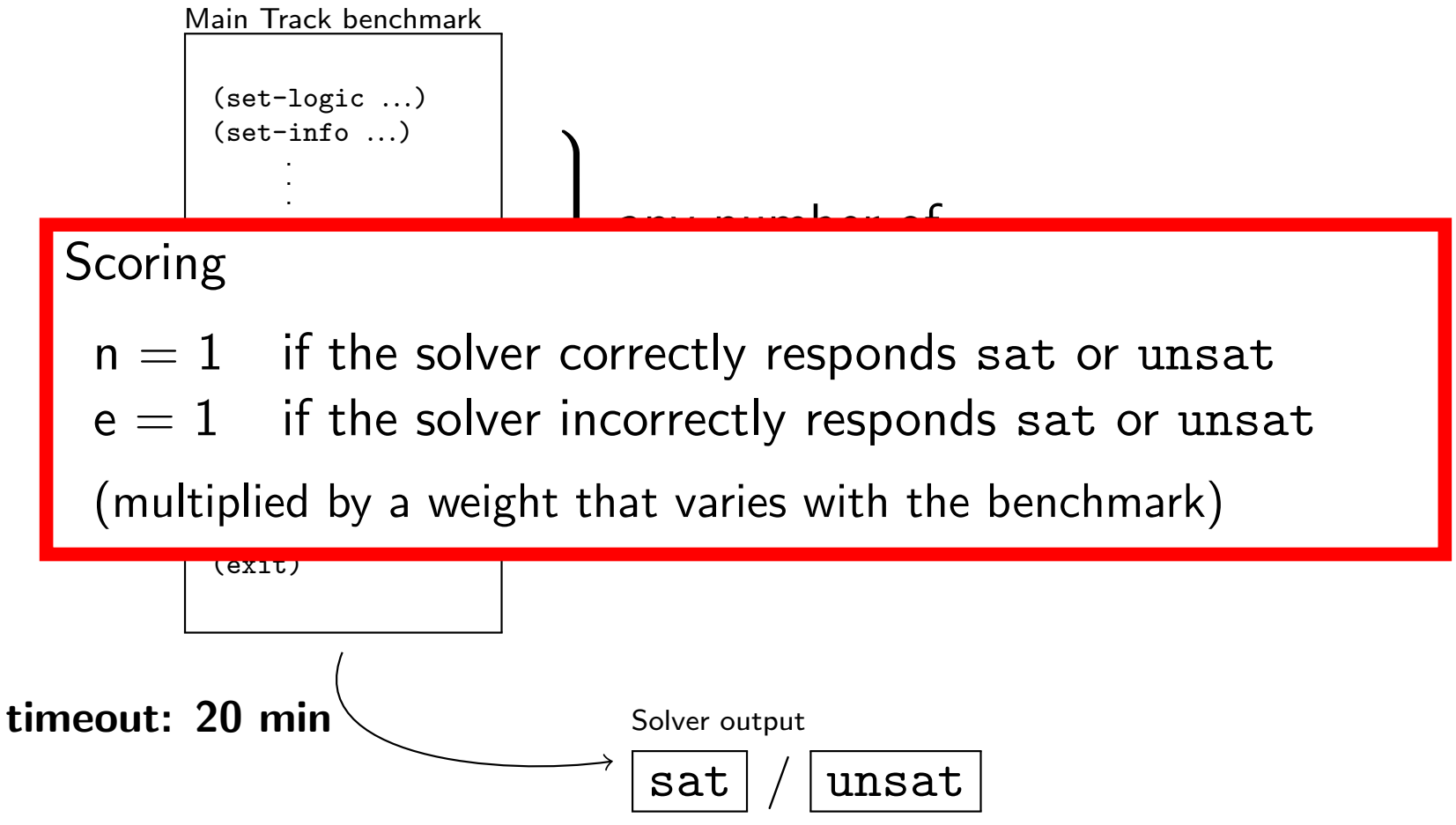
`set-info`, `declare-sort`, `define-sort`, `declare-fun`, `define-fun`, `assert` commands

← one `check-sat` command

# Main Track


Main Track benchmark
(set-logic ...)
(set-info ...)
(declare-sort ...)
(define-sort ...)
(declare-fun ...)
(define-fun ...)
(assert term0)
(assert term1)
(assert term2)
(check-sat)
(exit)
any number of
set-info, declare-sort, define-sort,
declare-fun, define-fun, assert
commands
← one check-sat command
timeout: 20 min
Solver output
sat / unsat

# Main Track


Main Track benchmark
(set-logic ...)
(set-info ...)
(exit)
Scoring
n = 1 if the solver correctly responds sat or unsat
e = 1 if the solver incorrectly responds sat or unsat
(multiplied by a weight that varies with the benchmark)
timeout: 20 min
Solver output
sat / unsat

# Application Track

Application track benchmarks may contain **multiple** `check-sat` commands, as well as `push` and `pop` commands.

Application Track benchmark

```
(set-logic ...)
      .
      .
      .
(check-sat)
      .
      .
      .
(check-sat)
      .
      .
      .
(check-sat)
      .
      .
      .
(check-sat)
(exit)
```

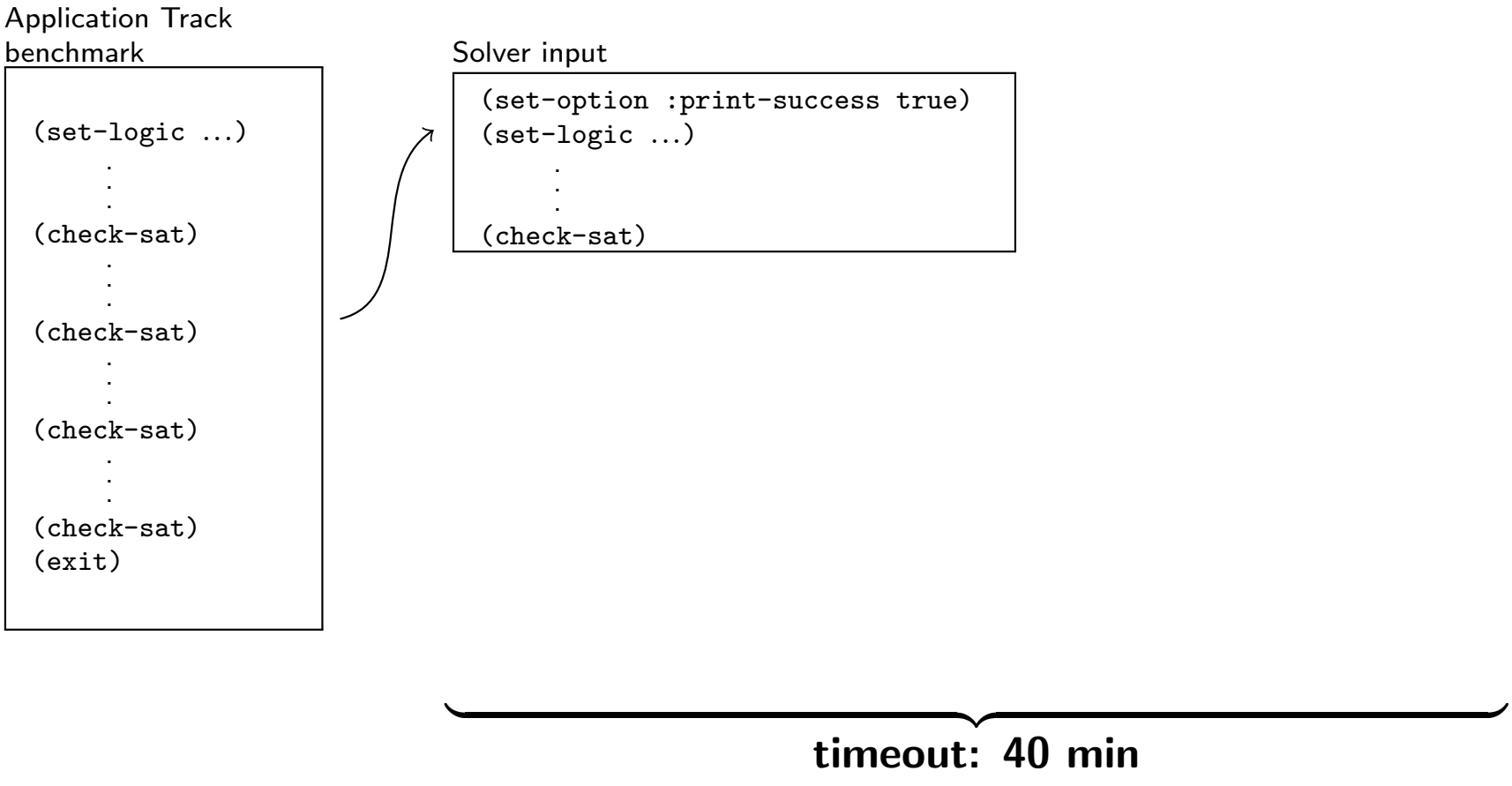
any number of

`set-info`, `declare-sort`, `define-sort`, `declare-fun`, `define-fun`, `assert`, `push`, `pop`, `check-sat`

commands

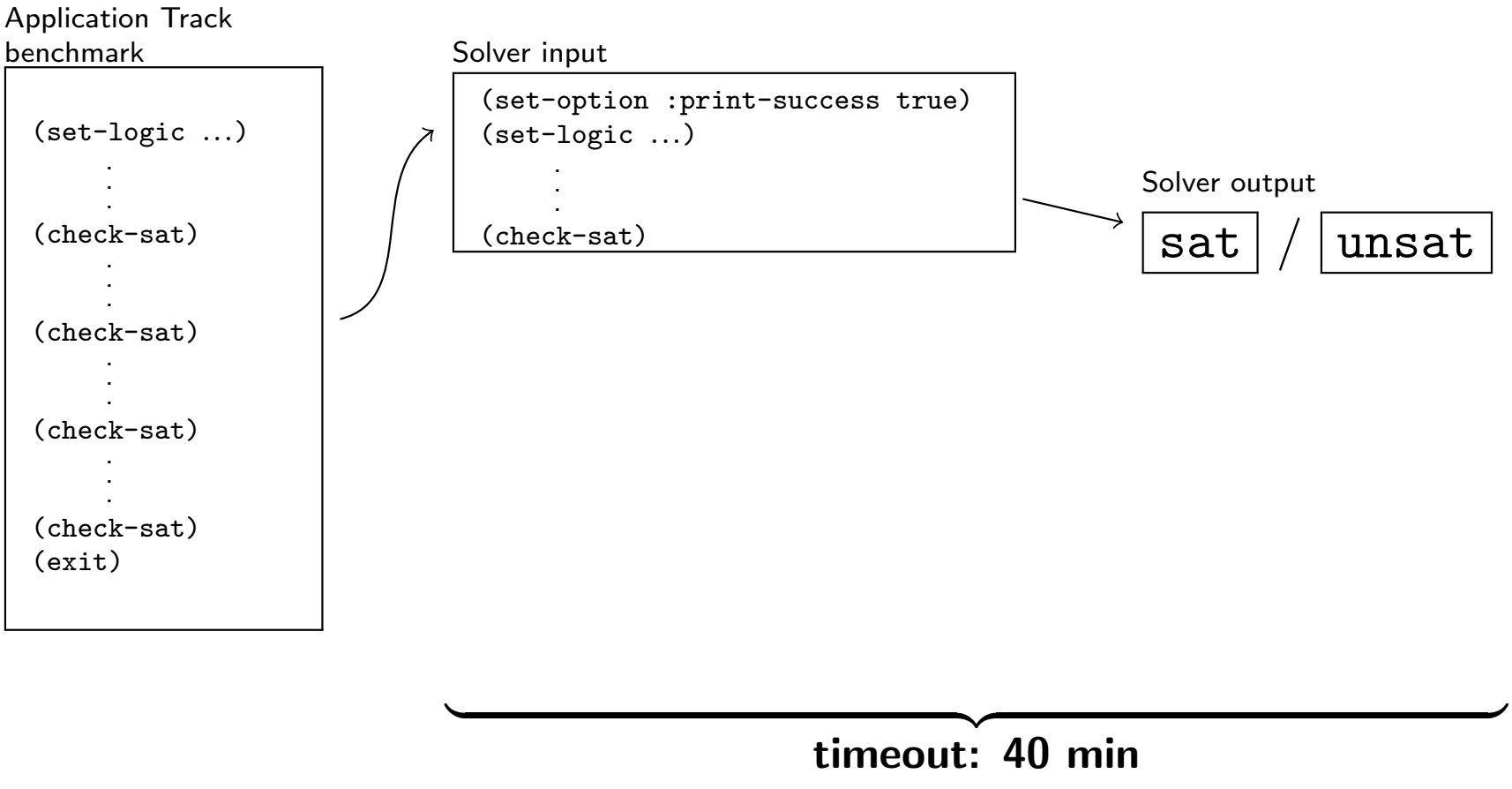
# Application Track

Application track benchmarks are fed to the solver **incrementally** by a trace executor.



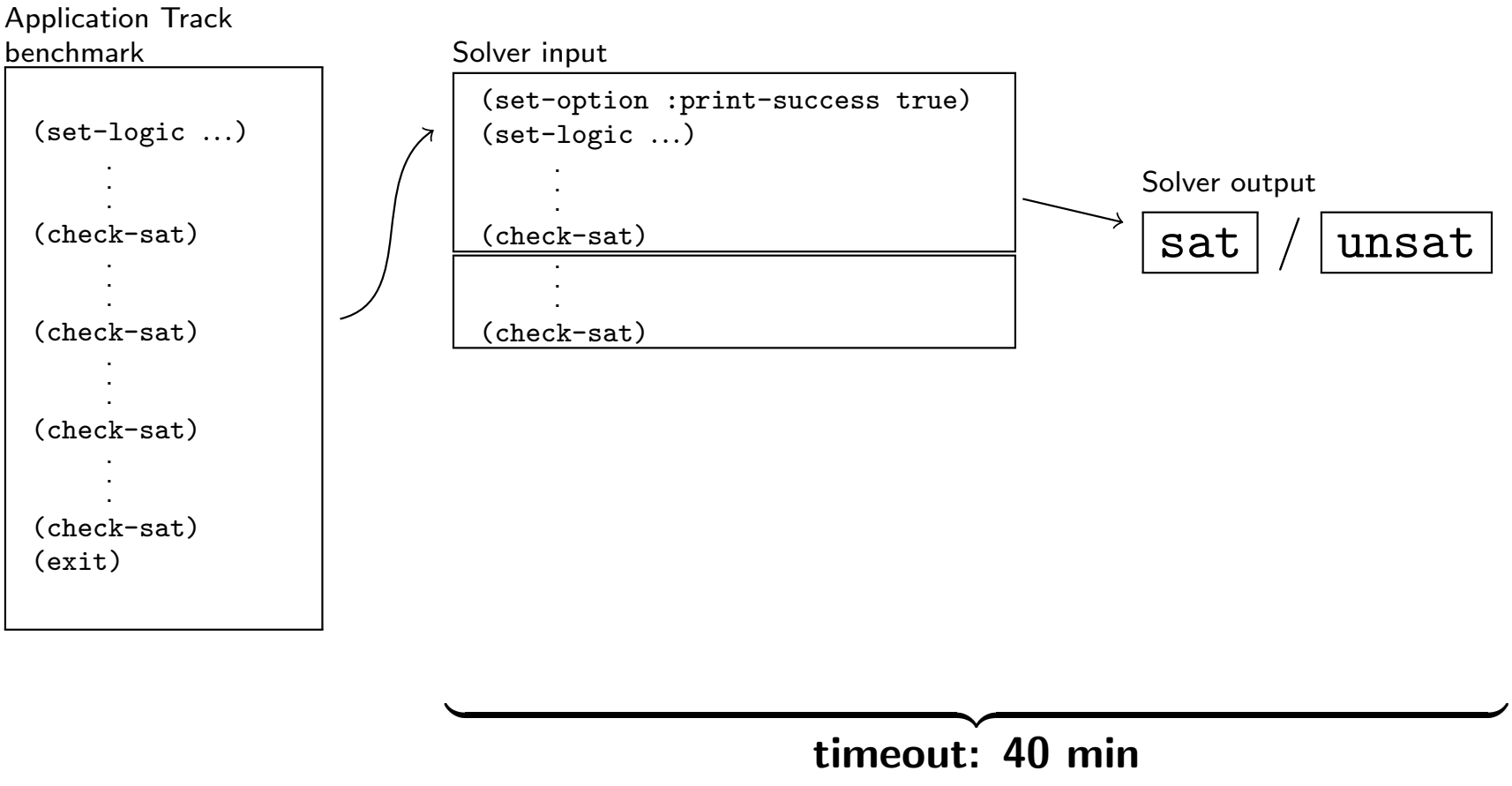
# Application Track

Application track benchmarks are fed to the solver **incrementally** by a trace executor.



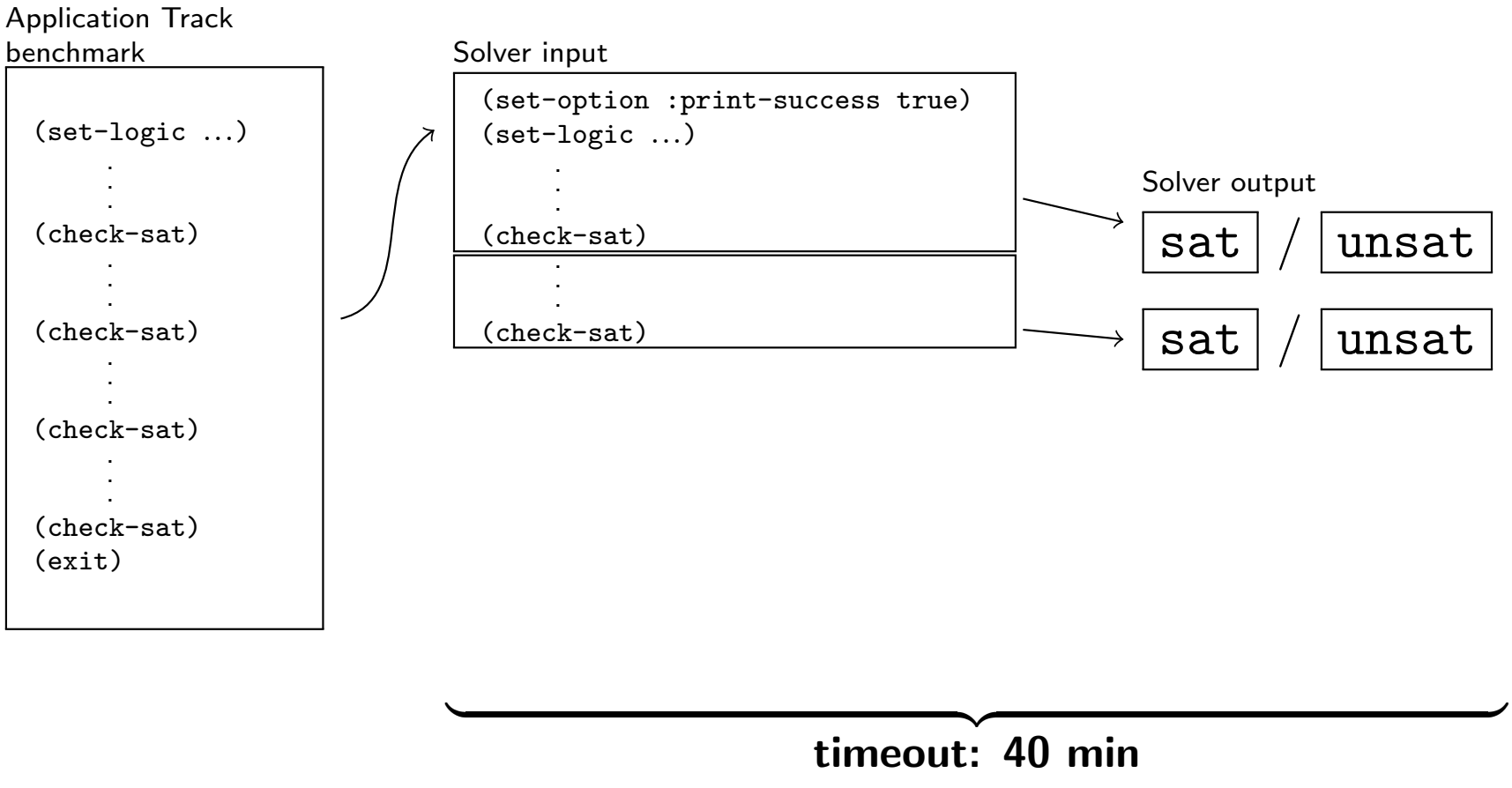
# Application Track

Application track benchmarks are fed to the solver **incrementally** by a trace executor.



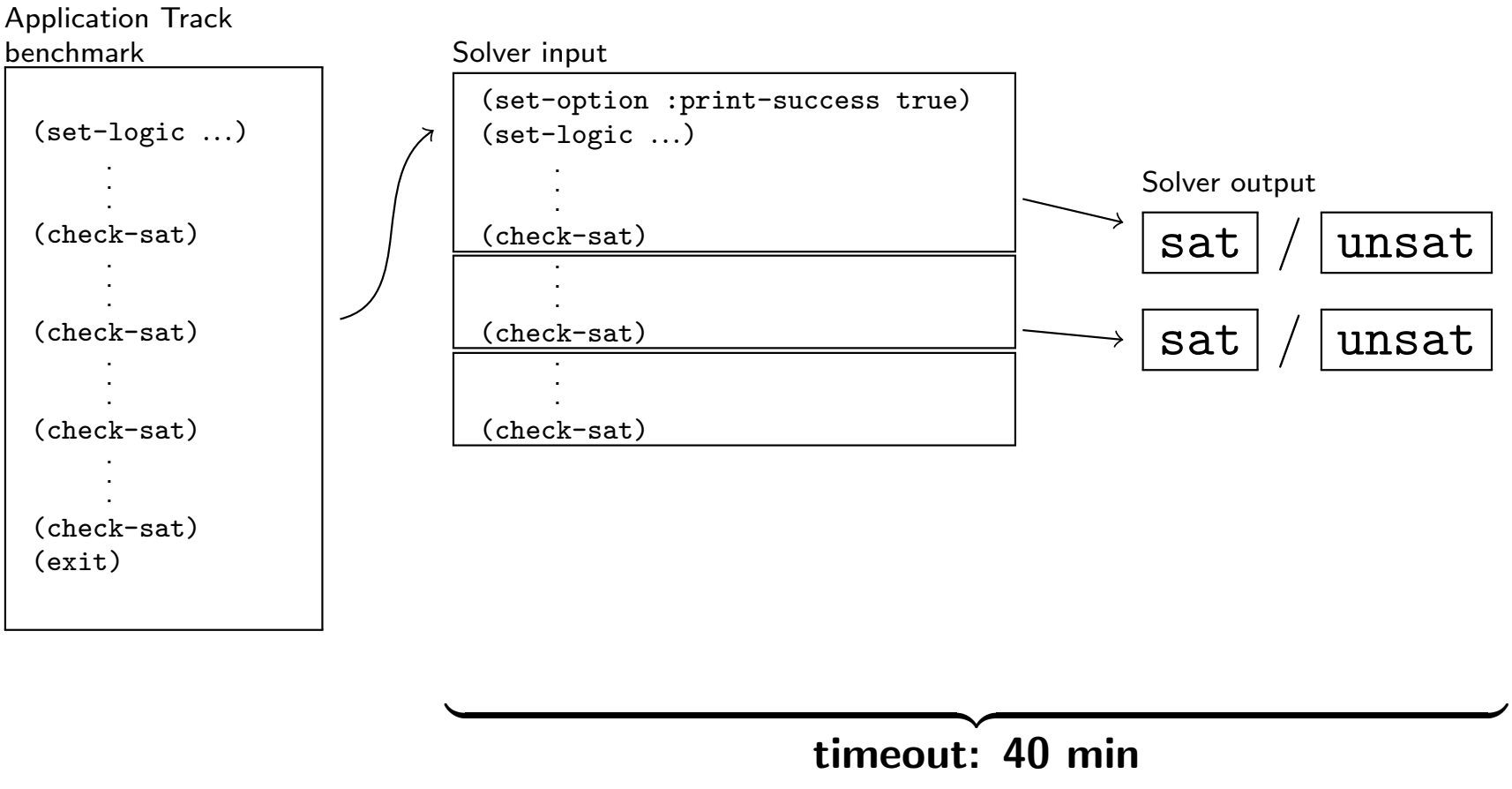
# Application Track

Application track benchmarks are fed to the solver **incrementally** by a trace executor.



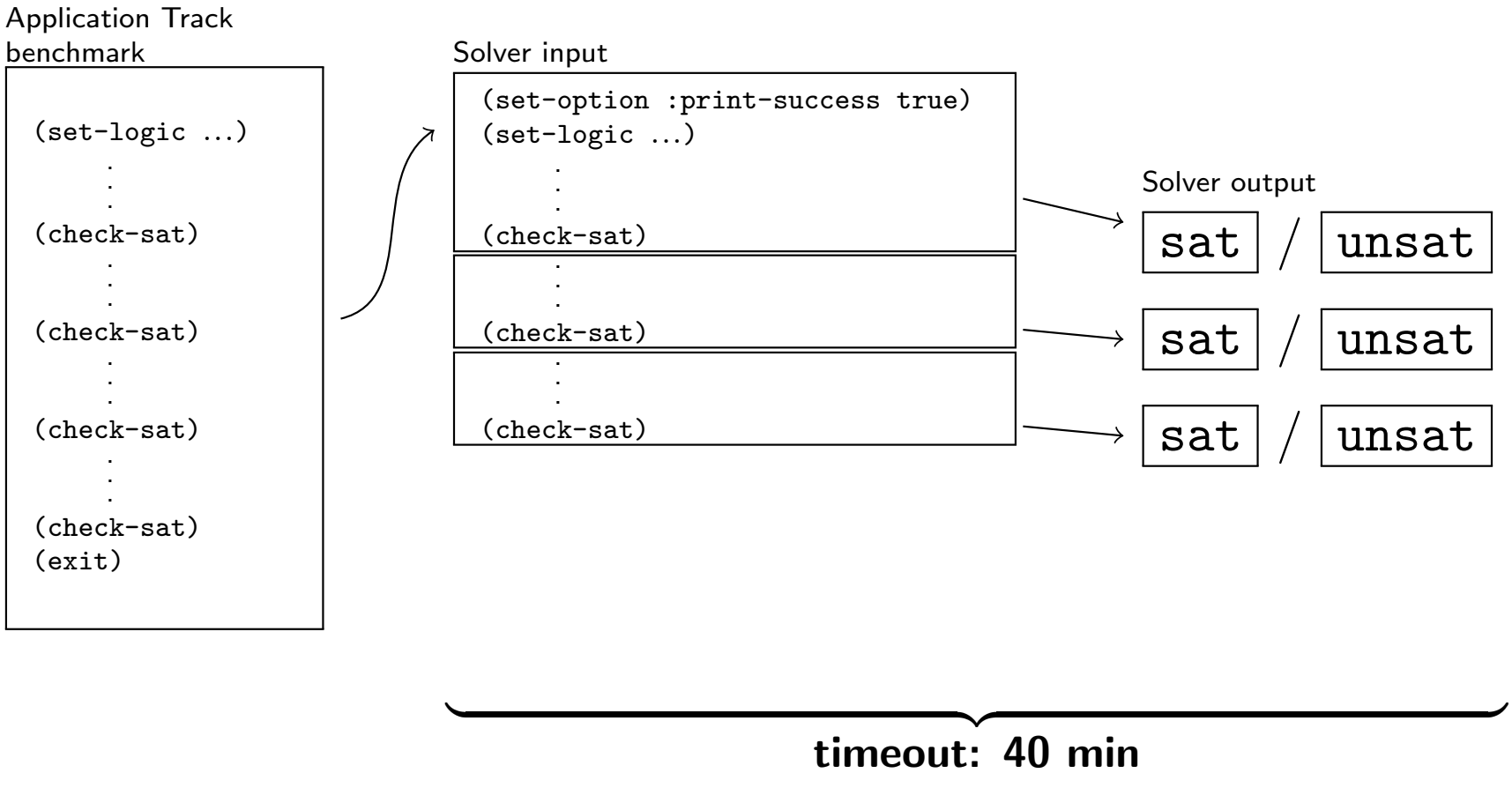
# Application Track

Application track benchmarks are fed to the solver **incrementally** by a trace executor.



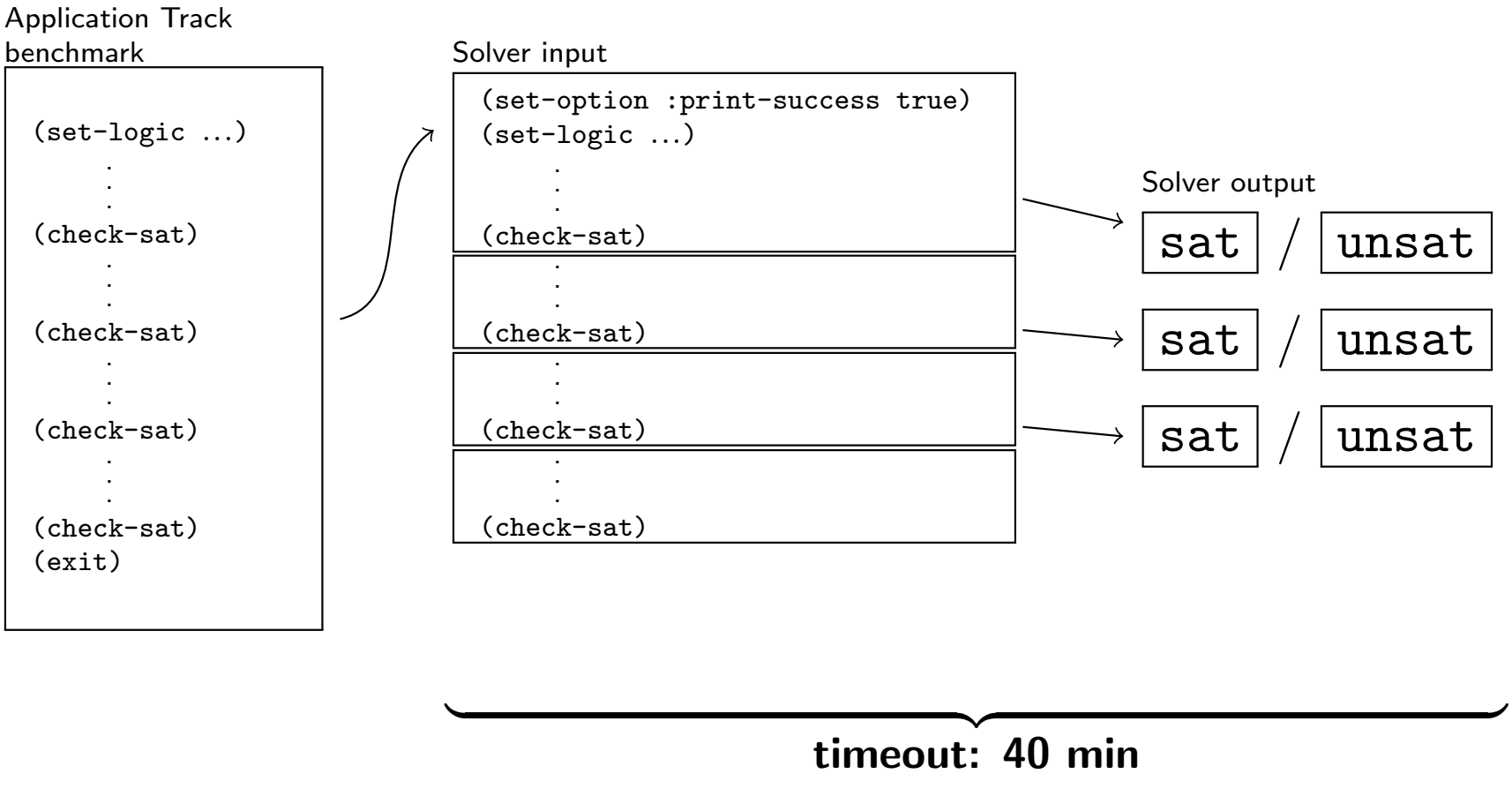
# Application Track

Application track benchmarks are fed to the solver **incrementally** by a trace executor.



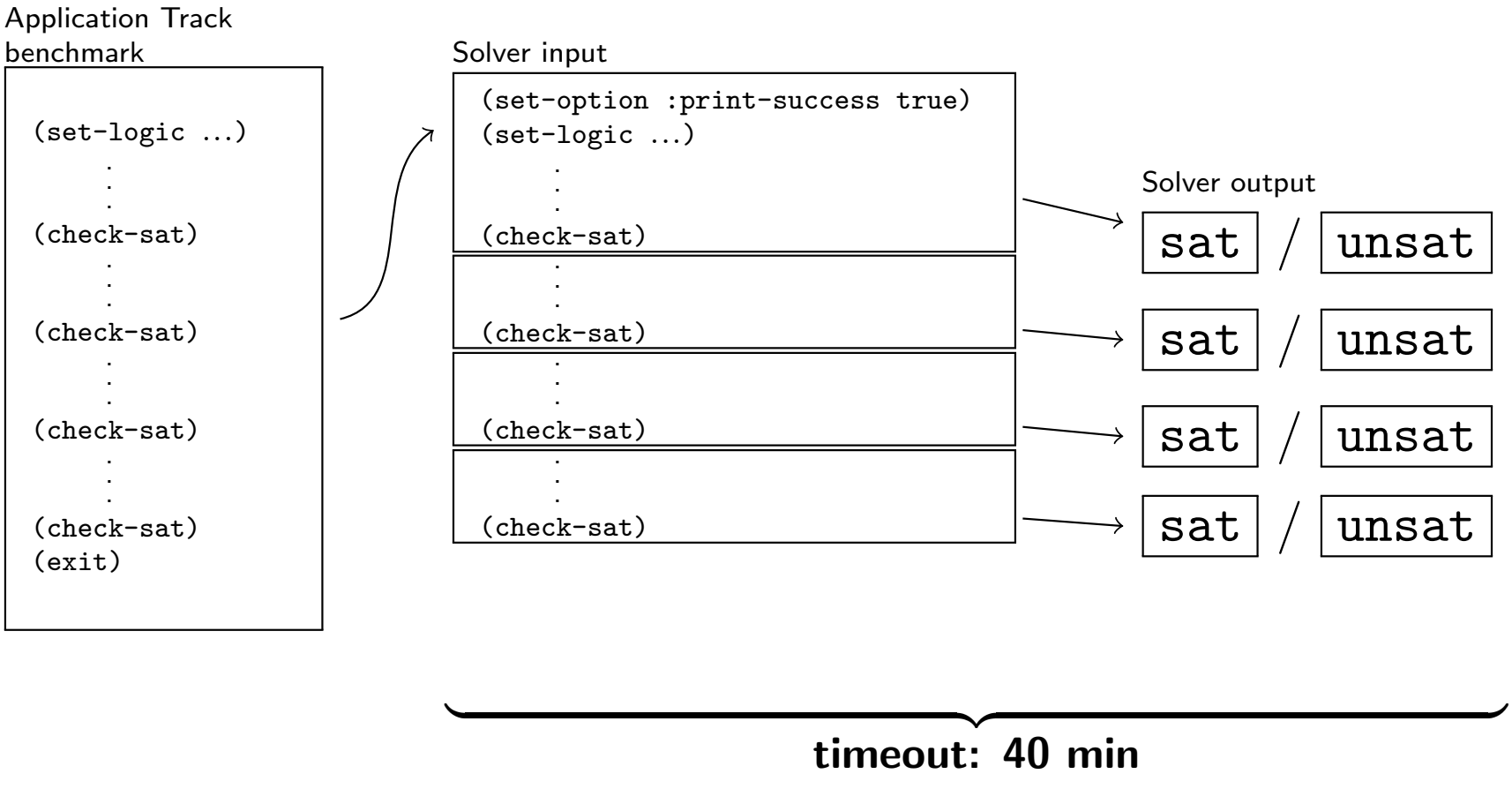
# Application Track

Application track benchmarks are fed to the solver **incrementally** by a trace executor.



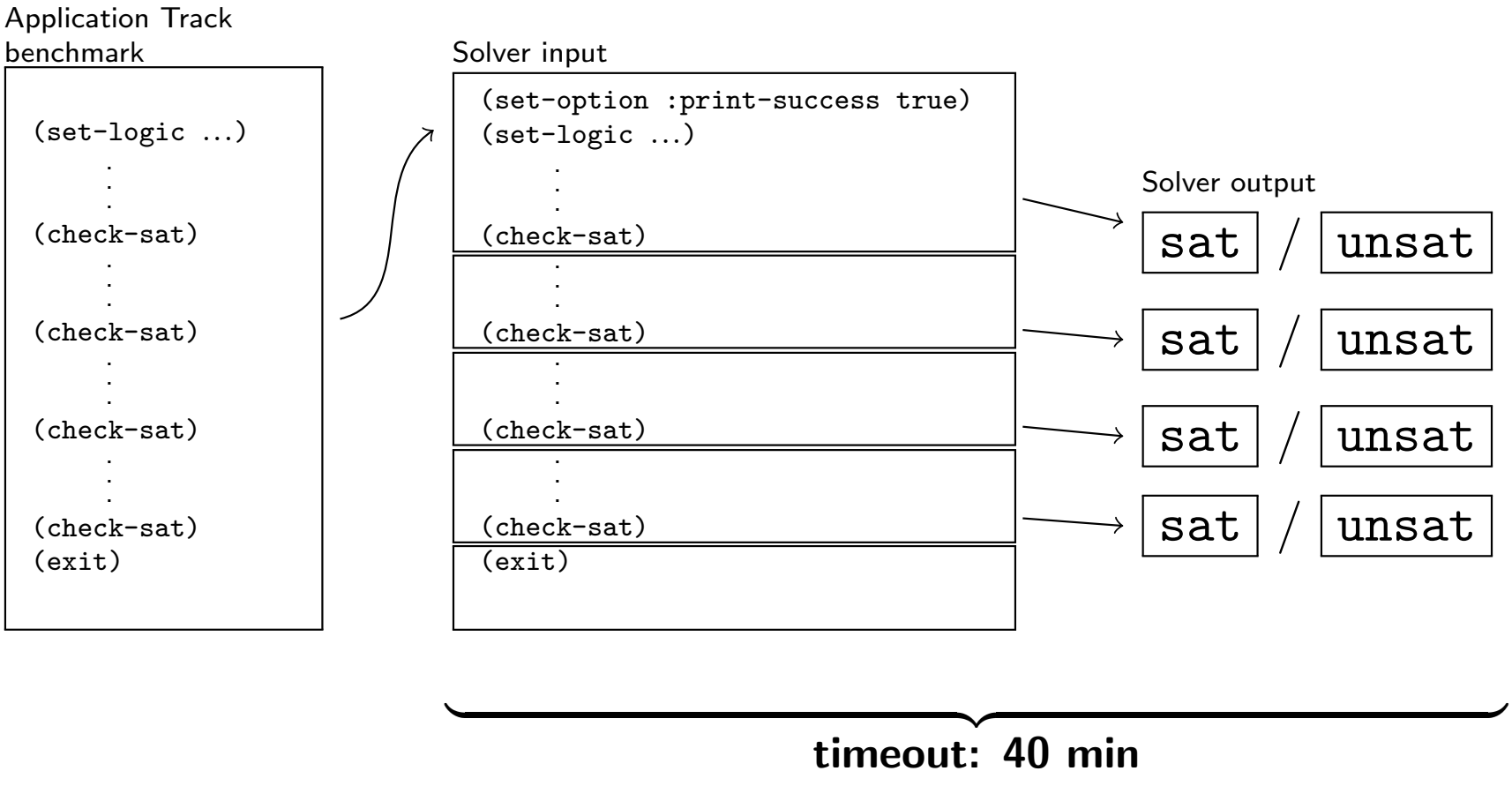
# Application Track

Application track benchmarks are fed to the solver **incrementally** by a trace executor.



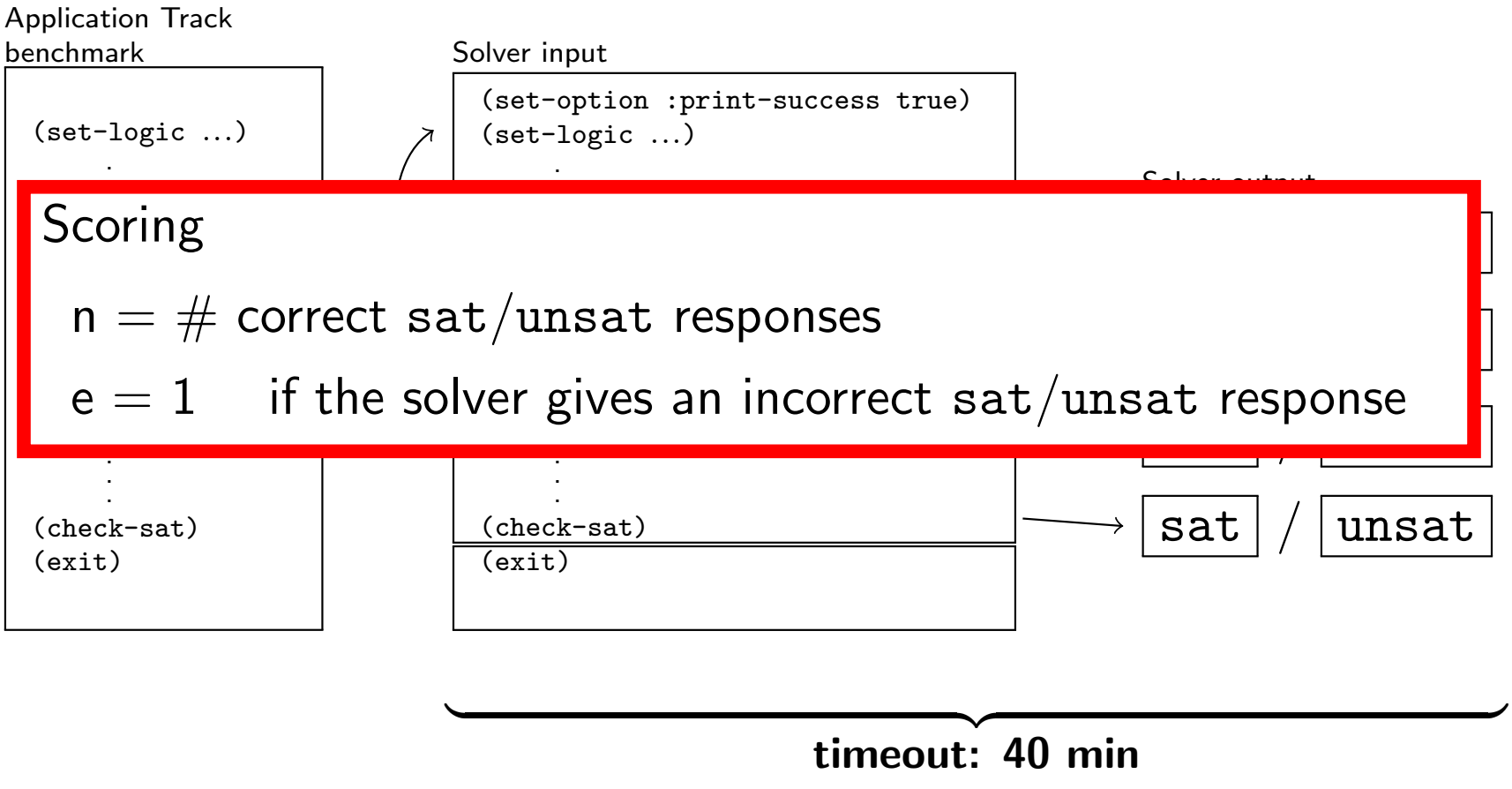
# Application Track

Application track benchmarks are fed to the solver **incrementally** by a trace executor.



# Application Track

Application track benchmarks are fed to the solver **incrementally** by a trace executor.



# Unsat-Core Track

Main Track benchmark
(unsat)

```
(set-logic ...)
(set-info ...)
     .
     .
     .
(declare-sort ...)
(define-sort ...)
(declare-fun ...)
(define-fun ...)
(assert term0)
(assert term1)
(assert term2)
     .
     .
     .
(check-sat)
(exit)
```



Solver input

```
(set-option :produce-unsat-cores true)
(set-logic ...)
(set-info ...)
     .
     .
     .
(declare-sort ...)
(define-sort ...)
(declare-fun ...)
(define-fun ...)
(assert (! term0 :named y0))
(assert (! term1 :named y1))
(assert (! term2 :named y2))
     .
     .
     .
(check-sat)
(get-unsat-core)
(exit)
```

# Unsat-Core Track

Main Track benchmark
(unsat)

```
(set-logic ...)
(set-info ...)
    .
    .
    .
(declare-sort ...)
(define-sort ...)
(declare-fun ...)
(define-fun ...)
(assert term0)
(assert term1)
(assert term2)
    .
    .
    .
(check-sat)
(exit)
```

Solver input

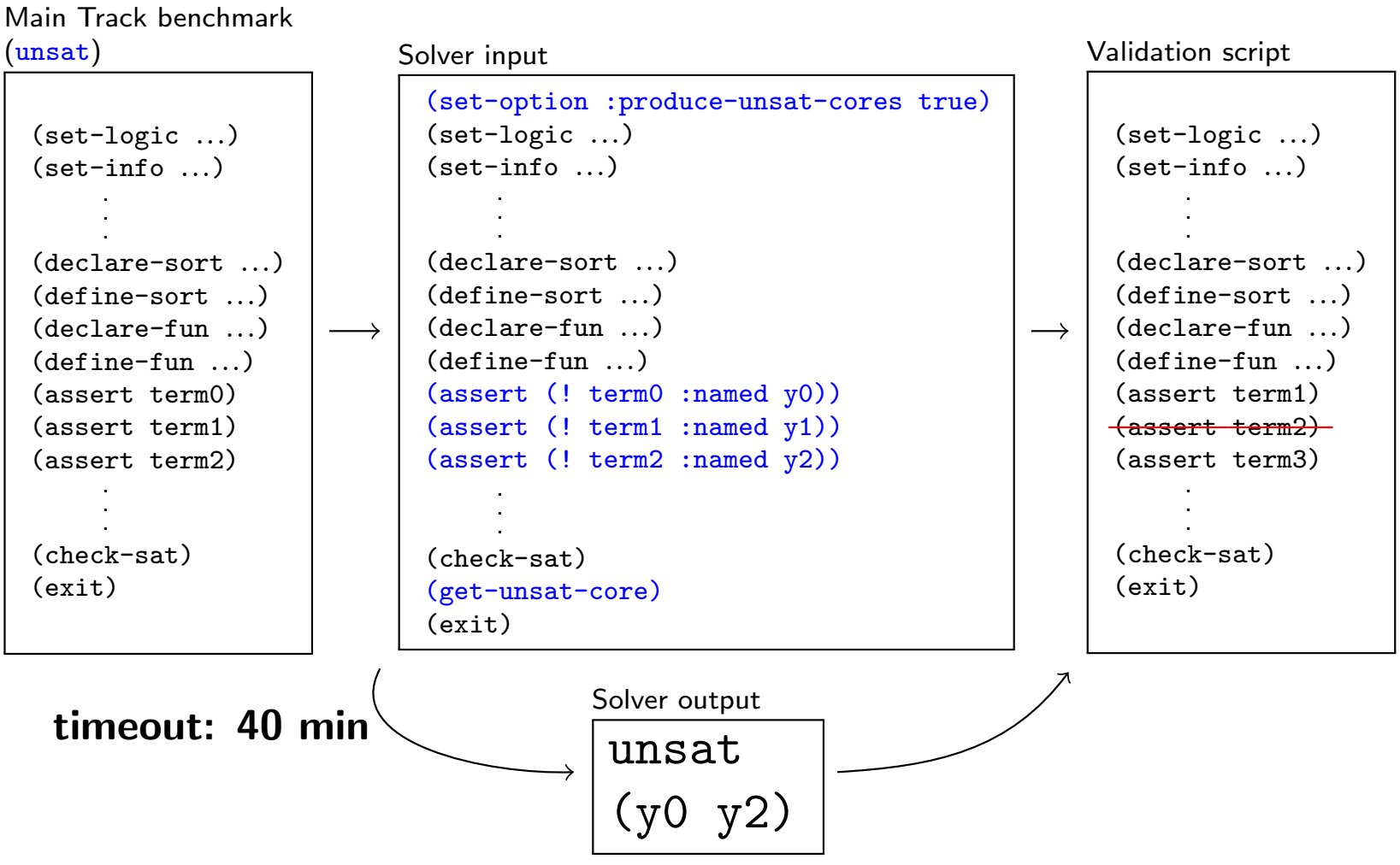
```
(set-option :produce-unsat-cores true)
(set-logic ...)
(set-info ...)
    .
    .
    .
(declare-sort ...)
(define-sort ...)
(declare-fun ...)
(define-fun ...)
(assert (! term0 :named y0))
(assert (! term1 :named y1))
(assert (! term2 :named y2))
    .
    .
    .
(check-sat)
(get-unsat-core)
(exit)
```

**timeout: 40 min**

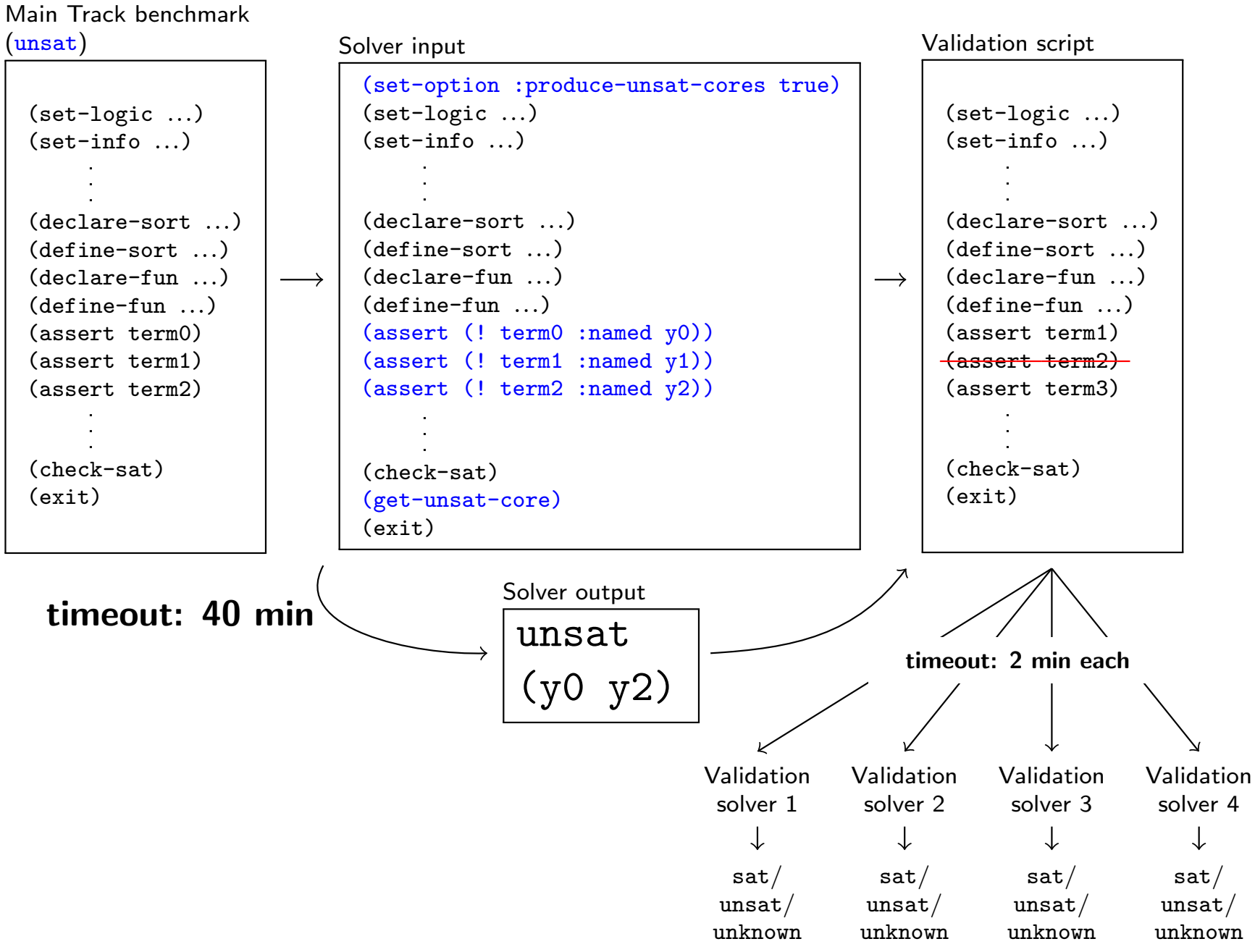
Solver output

```
unsat
(y0 y2)
```

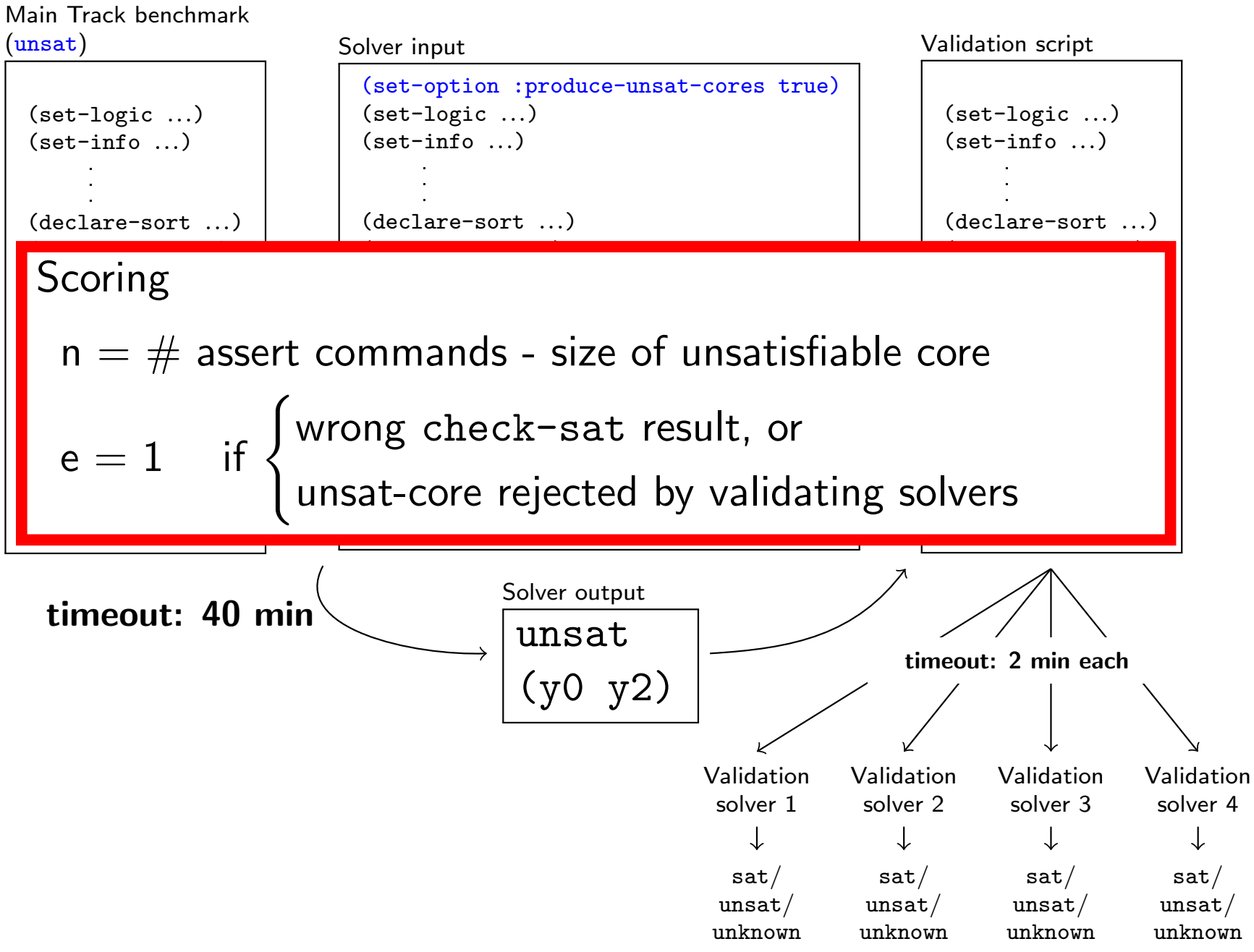
# Unsat-Core Track



# Unsat-Core Track

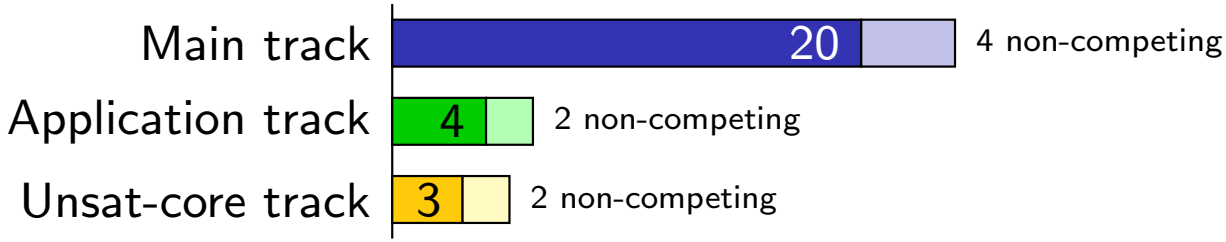


# Unsat-Core Track

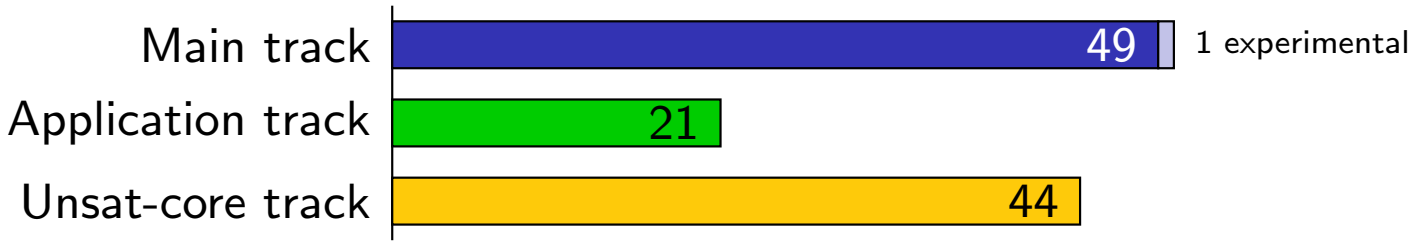


# Solvers, Logics, and Benchmarks

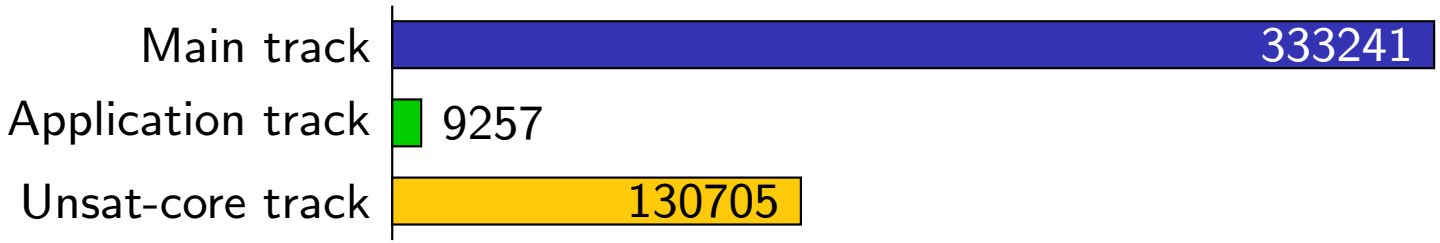
- 17 teams participated
- Solvers:




- Logics:

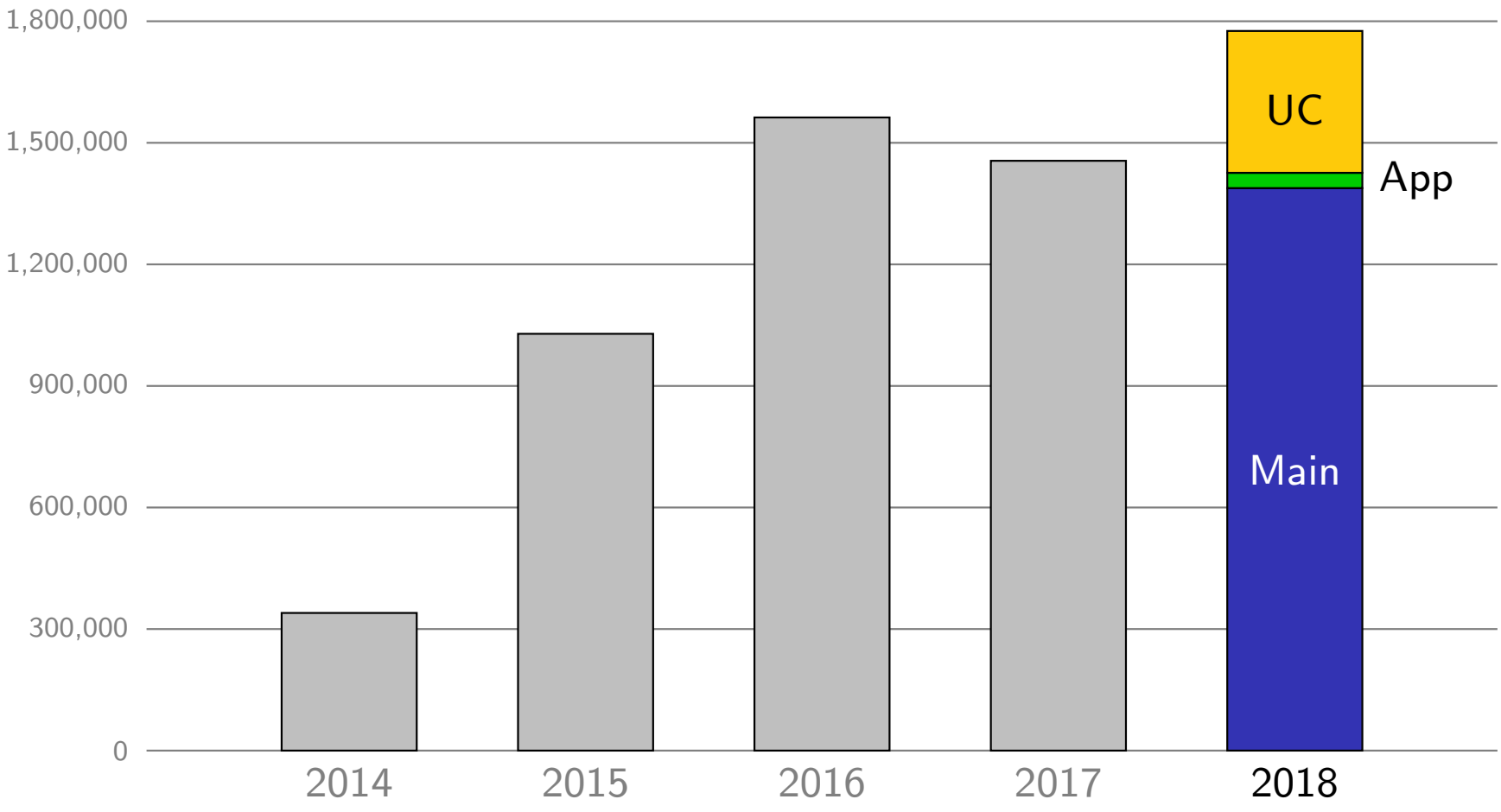



- Benchmarks:



# Job Pairs

1,776,062 job pairs (+ some repeats)



# StarExec

All job pairs were executed on StarExec, a cluster at the University of Iowa.

Hardware:

- Intel Xeon CPU E5-2609 @ 2.4 GHz, 10 MB cache
- 2 processors per node, 4 cores per processor
- Main memory capped at 60 GB per job pair

Software:

- Red Hat Enterprise Linux Server release 7.2
- Kernel 3.10.0-514, gcc 4.8.5, glibc 2.17

$\sim$ 17 days $\times$ 120 nodes $\times$ 2 processors/node of compute time

# Main Changes From 2017

- Datatype (DT) divisions no longer experimental
- Experimental string division (QF_SLIA)
- Unsat-core track: core validation by simple majority vote
- Certificates

(Very) short presentations of

# Solvers

that sent us slides:

Alt-Ergo, Boolector, Ctrl-Ergo, CVC4,
OpenSMT, SMTInterpol, SPASS-SATT, Yices

# Alt-Ergo @ SMT-Comp 2018

- based on version 2.2.0 presented by Albin yesterday,
- improve triggers inference, in particular for multi-triggers,
- allow/propagate more triggers in the backend,
- improve handling of Let-In,
- enable additional heuristics before returning unknown,
- experimental : enable a kind of first-order resolution
- experimental : SAT detection in some situations
- add the ability to run several strategies in parallel

`https://github.com/OCamlPro/alt-ergo`

# Boolector at the SMT-COMP'18

Aina Niemetz, Mathias Preiner, Armin Biere

## Divisions

**Main:** BV QF_BV QF_UFBV QF_ABV QF_AUFBV
**Application:** QF_BV QF_UFBV QF_ABV

## Configuration

- SAT competition 2017 version of CaDiCaL for QF_BV
- SAT competition 2018 version of Lingeling for all other divisions
- Combination of prop.-based local search + bit-blasting for BV, QF_BV
- Minor improvements to array engine and simplifications/rewriting

## New release of Boolector

- Version 3.0
- Now on GitHub: `https://github.com/boolector/boolector`
- MIT license

# Ctrl-Ergo @ SMT-Comp 2018

- a prototype I developed during my thesis to validate our work published at IJCAR'2012
  - Simplex-based Fourier-Motkzin procedure to decide QF_LIA
- pre-processing for QF_LIA **Let-In** and **Ite** expressions
- general Simplex for QF_LRA
- mini-SAT based SAT solver
- extended to be able to run several strategies in parallel

`https://gitlab.com/OCamlPro-Iguernlala/Ctrl-Ergo`

# CVC4 at the SMT Competition 2018

Clark Barrett, Haniel Barbosa, Martin Brain, Duligur Ibeling, Tim King, Paul Meng, Aina Niemetz, Andres Nötzli, Mathias Preiner, Andrew Reynolds, Cesare Tinelli

## Divisions

This year's configuration of CVC4 enters all divisions in all tracks.

## New Features / Improvements

- New: Floating-Point Solver
- New: Novel approach for Quantified Bit-Vectors
- New: Experimental division QF_SLIA (strings)
- Eager Bit-Blasting Solver with CaDiCaL as back end
- Heuristic Approaches for Non-Linear Arithmetic with CaDiCaL as back end
- Improvement of quantifier instantiation

## Experimental Configuration CVC4-experimental-idl-2

- non-competitive
- specialized IDL solver, entered division QF_IDL of the main track

# OpenSMT

A relatively small DPLL(T)-based SMT Solver

Developed at University of Lugano, Switzerland

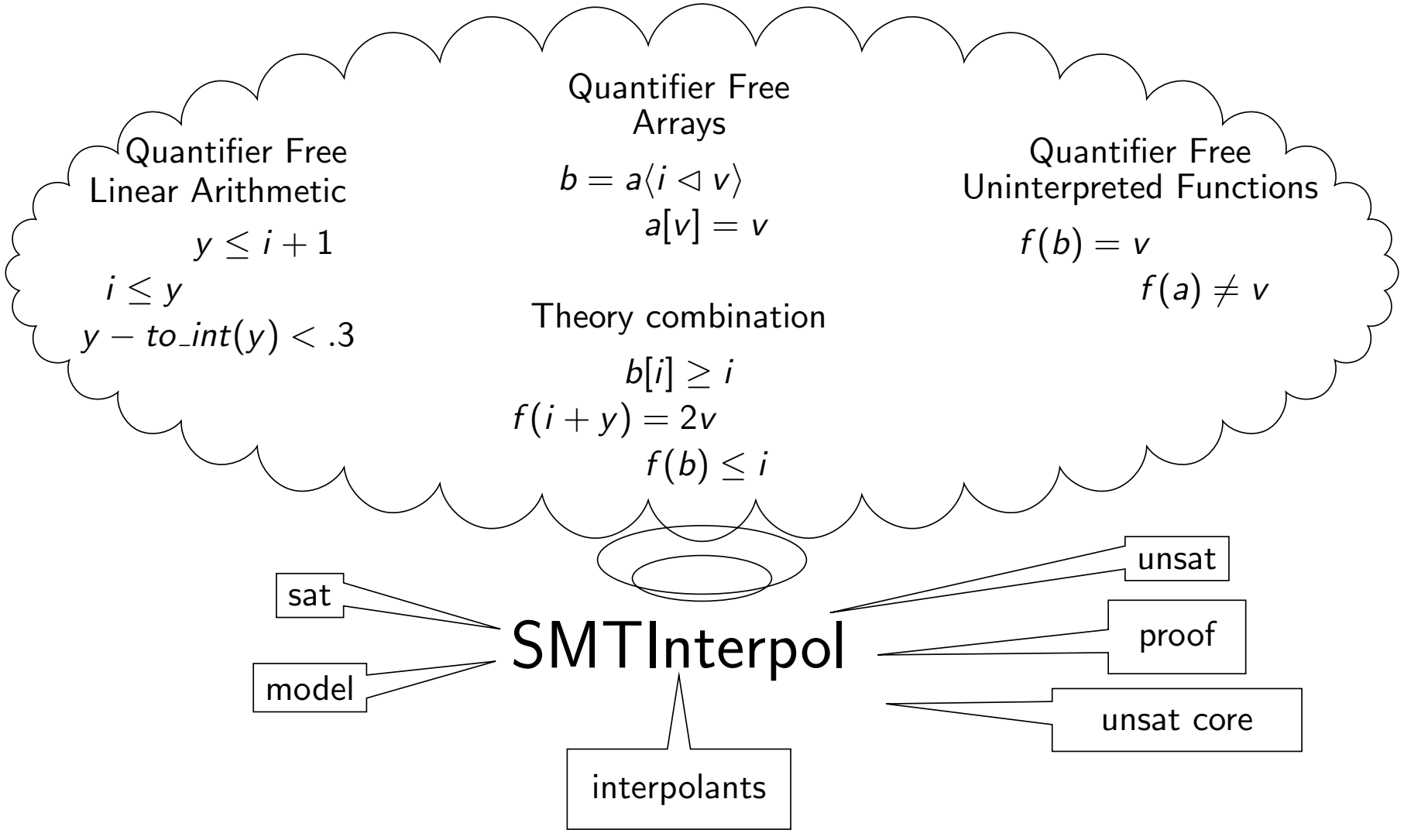
Supports QF_UF, QF_LRA, and to some extent QF_BV

Theory refinement

Interpolation

Integration to our model checker HiFrog

Available from http://verify.inf.usi.ch/opensmt




`http://ultimate.informatik.uni-freiburg.de/smtinterpol`

http://www.spass-prover.org/spass-satt


## Developers:

Martin Bromberger, Mathias Fleury, Fabian Kunze, Dominik Wagner, Christoph Weidenbach

## Ground Linear Arithmetic Solver:

- newest tool in the SPASS Workbench
- combines our theory solver SPASS-IQ and our unnamed SAT solver
- supports QF_LIA, QF_LRA, (and QF_LIRA)
- complete but efficient theory solver [IJCAR2018]
- uses fast cube tests [IJCAR2016, FMSD2017]
- SAT decisions based on theory solver information
- uses many more well-known techniques for linear arithmetic

# Yices 2.6 in SMTCOMP 2018

## Yices 2

- Supports linear and non-linear arithmetic, arrays, UF, bitvectors
- Includes two types of solvers: classic DPPL($T$) + MC-SAT
- `https://github.com/SRI-CSL/yices2`

## New in 2018

- Unsat cores
- Incremental MC-SAT

## Entered in all the divisions that Yices supports

- Main/application track: Quantifier-free logics including linear and nonlinear arithmetic, bitvectors, and combination with UF and Arrays.
- Unsat core track: Same logics, except that unsat cores are not yet supported by MC-SAT (i.e., nonlinear arithmetic)

Acknowledgments: thanks to Aman Goel (UMich) for help with unsat cores

# Selected Results

# Unsat-Core Track

- 3 competing solvers: CVC4, SMTInterpol, Yices-2.6.0
- 16 competitive divisions (out of 44)

| Solver | Divisions won |
|---|---|
| CVC4 | |
| SMTInterpol | |
| Yices-2.6.0 | |

# Unsat-Core Track

- 3 competing solvers: CVC4, SMTInterpol, Yices-2.6.0
- 16 competitive divisions (out of 44)

| Solver | Divisions won |
|---|---|
| CVC4 | QF_AUFLIA, QF_IDL, QF_LIRA, QF_RDL, QF_UF |
| SMTInterpol | |
| Yices-2.6.0 | |

# Unsat-Core Track

- 3 competing solvers: CVC4, SMTInterpol, Yices-2.6.0
- 16 competitive divisions (out of 44)

| Solver | Divisions won |
|---|---|
| CVC4 | QF_AUFLIA, QF_IDL, QF_LIRA, QF_RDL, QF_UF |
| SMTInterpol | QF_LIA, QF_LRA, QF_UFLIA |
| Yices-2.6.0 | |

# Unsat-Core Track

- 3 competing solvers: CVC4, SMTInterpol, Yices-2.6.0
- 16 competitive divisions (out of 44)

| Solver | Divisions won |
|---|---|
| CVC4 | QF_AUFLIA, QF_IDL, QF_LIRA, QF_RDL, QF_UF |
| SMTInterpol | QF_LIA, QF_LRA, QF_UFLIA |
| Yices-2.6.0 | QF_ABV, QF_ALIA, QF_AUFBV, QF_AX, QF_BV, QF_UFBV, QF_UFIDL, QF_UFLRA |

# Application Track

- 4 competing solvers: Boolector, CVC4, SMTInterpol, Yices-2.6.0
- 12 competitive divisions (out of 21)

| Solver | Divisions won |
|---|---|
| Boolector | |
| CVC4 | |
| SMTInterpol | |
| Yices-2.6.0 | |

# Application Track

- 4 competing solvers: Boolector, CVC4, SMTInterpol, Yices-2.6.0
- 12 competitive divisions (out of 21)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_UFBV |
| CVC4 | |
| SMTInterpol | |
| Yices-2.6.0 | |

# Application Track

- 4 competing solvers: Boolector, CVC4, SMTInterpol, Yices-2.6.0
- 12 competitive divisions (out of 21)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_UFBV |
| CVC4 | QF_NIA, QF_UFNIA |
| SMTInterpol | |
| Yices-2.6.0 | |

# Application Track

- 4 competing solvers: Boolector, CVC4, SMTInterpol, Yices-2.6.0
- 12 competitive divisions (out of 21)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_UFBV |
| CVC4 | QF_NIA, QF_UFNIA |
| SMTInterpol | QF_ALIA, QF_UFLIA |
| Yices-2.6.0 | |

# Application Track

- 4 competing solvers: Boolector, CVC4, SMTInterpol, Yices-2.6.0
- 12 competitive divisions (out of 21)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_UFBV |
| CVC4 | QF_NIA, QF_UFNIA |
| SMTInterpol | QF_ALIA, QF_UFLIA |
| Yices-2.6.0 | QF_AUFBV, QF_AUFLIA, QF_BV, QF_LIA, QF_LRA, QF_UFLRA |

# Main Track

- 20 competing solvers
- 41 competitive divisions (out of 50)

| Solver | Divisions won |
|---|---|

# Main Track

- 20 competing solvers
- 41 competitive divisions (out of 50)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_BV$^{seq}$, QF_UFBV |

# Main Track

- 20 competing solvers
- 41 competitive divisions (out of 50)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_BV$^{seq}$, QF_UFBV |
| COLIBRI | QF_FP |

# Main Track

- 20 competing solvers
- 41 competitive divisions (out of 50)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_BV$^{seq}$, QF_UFBV |
| COLIBRI | QF_FP |
| CVC4 | ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA, QF_NIA, UF$^{seq}$, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA |

# Main Track

- 20 competing solvers
- 41 competitive divisions (out of 50)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_BV$^{seq}$, QF_UFBV |
| COLIBRI | QF_FP |
| CVC4 | ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA, QF_NIA, UF$^{seq}$, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA |
| Minkeyrink-MT | QF_BV$^{par}$ |

# Main Track

- 20 competing solvers
- 41 competitive divisions (out of 50)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_BV$^{seq}$, QF_UFBV |
| COLIBRI | QF_FP |
| CVC4 | ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA, QF_NIA, UF$^{seq}$, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA |
| Minkeyrink-MT | QF_BV$^{par}$ |
| SMTRAT | QF_NIRA |

# Main Track

- 20 competing solvers
- 41 competitive divisions (out of 50)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_BV$^{seq}$, QF_UFBV |
| COLIBRI | QF_FP |
| CVC4 | ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA, QF_NIA, UF$^{seq}$, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA |
| Minkeyrink-MT | QF_BV$^{par}$ |
| SMTRAT | QF_NIRA |
| SPASS-SATT | QF_LIA |

# Main Track

- 20 competing solvers
- 41 competitive divisions (out of 50)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_BV$^{seq}$, QF_UFBV |
| COLIBRI | QF_FP |
| CVC4 | ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA, QF_NIA, UF$^{seq}$, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA |
| Minkeyrink-MT | QF_BV$^{par}$ |
| SMTRAT | QF_NIRA |
| SPASS-SATT | QF_LIA |
| Vampire | NRA, UF$^{par}$, UFNIA |

# Main Track

- 20 competing solvers
- 41 competitive divisions (out of 50)

| Solver | Divisions won |
|---|---|
| Boolector | QF_ABV, QF_BV$^{seq}$, QF_UFBV |
| COLIBRI | QF_FP |
| CVC4 | ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA, QF_NIA, UF$^{seq}$, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA |
| Minkeyrink-MT | QF_BV$^{par}$ |
| SMTRAT | QF_NIRA |
| SPASS-SATT | QF_LIA |
| Vampire | NRA, UF$^{par}$, UFNIA |
| Yices-2.6.0 | QF_ALIA, QF_AUFLIA, QF_AX, QF_IDL, QF_LIRA, QF_NRA, QF_RDL, QF_UF, QF_UFIDL, QF_UFLIA, QF_UFLRA, QF_UFNIA, QF_UFNRA |

# Main Track: Competition-Wide Scoring

| Rank | Solver | Score (sequential) | Score (parallel) |
|---|---|---|---|
| Best newcomer: | | | |
| 7 | SPASS-SATT | 14.81 | 14.81 |

# Main Track: Competition-Wide Scoring

| Rank | Solver | Score (sequential) | Score (parallel) |
|---|---|---|---|
| 3 | SMTInterpol | 65.32 | 65.38 |
| Best newcomer: | | | |
| 7 | SPASS-SATT | 14.81 | 14.81 |

# Main Track: Competition-Wide Scoring

| Rank | Solver | Score (sequential) | Score (parallel) |
|---|---|---|---|
| 2 | Yices-2.6.0 | 115.26 | 115.26 |
| 3 | SMTInterpol | 65.32 | 65.38 |
| Best newcomer: | | | |
| 7 | SPASS-SATT | 14.81 | 14.81 |

# Main Track: Competition-Wide Scoring

| Rank | Solver | Score (sequential) | Score (parallel) |
|---|---|---|---|
| | Z3 | 186.19 | 186.19 |
| 2 | Yices-2.6.0 | 115.26 | 115.26 |
| 3 | SMTInterpol | 65.32 | 65.38 |
| Best newcomer: | | | |
| 7 | SPASS-SATT | 14.81 | 14.81 |

# Main Track: Competition-Wide Scoring

| Rank | Solver | Score (sequential) | Score (parallel) |
|---|---|---|---|
| 1 | CVC4 | 211.99 | 211.99 |
| | Z3 | 186.19 | 186.19 |
| 2 | Yices-2.6.0 | 115.26 | 115.26 |
| 3 | SMTInterpol | 65.32 | 65.38 |
| Best newcomer: | | | |
| 7 | SPASS-SATT | 14.81 | 14.81 |

Teams:

- Congratulations on your accomplishments!
- Thanks for your participation!

FLoC Olympic Games Award Ceremony

tomorrow at 14:00 in room L3 (Mathematical Institute)

# Backup Slides

# Incorrect Answers

Main track:

- 125 incorrect answers (0.01%) by 6 solvers (25%)
- No disagreements between sound solvers on benchmarks with unknown status

Application track:

- No incorrect answers

Unsat-core track:

- No incorrect `check-sat` answers
- 443 incorrect unsat cores (0.1%) by 1 solver (20%)