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-LIB benchmarks
curated by
Clark Barrett,
Pascal Fontaine,
Cesare Tinelli

SMT-LIB benchmarks
upload
benchmarks

SMT-LIB users
submit
benchmarks

SMT solver developers
upload
solvers

StarExec
maintained by
Aaron Stump

competition results
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

Solver output
sat/unsat

timeout: 20 min

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)
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

Scoring

\[ n = 1 \quad \text{if the solver correctly responds sat or unsat} \]
\[ e = 1 \quad \text{if the solver incorrectly responds sat or unsat} \]

(multiplied by a weight that varies with the benchmark)

timeout: 20 min
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.

Solver input:
```
(set-option :print-success true)
(set-logic ...)
  ...
(check-sat)
  ...
(check-sat)
  ...
(check-sat)
  ...
(check-sat)
(exit)
```

Application Track benchmark:
```
(set-logic ...)
  ...
(check-sat)
  ...
(check-sat)
  ...
(check-sat)
  ...
(check-sat)
(exit)
```

timeout: 40 min

Scoring:
- \( n \) = # correct sat/unsat responses
- \( e = 1 \) if the solver gives an incorrect sat/unsat response
Application Track

Application track benchmarks are fed to the solver \textbf{incrementally} by a trace executor.

Solver input

\begin{verbatim}
(set-option :print-success true)
(set-logic ...)
...
(check-sat)
...
(check-sat)
...
(check-sat)
(check-sat)
(exit)
\end{verbatim}

Solver output

\begin{verbatim}
 sat / unsat
\end{verbatim}

timeout: 40 min
Application Track

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

Application Track benchmark

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

Solver input

(set-option :print-success true)
(set-logic ...)
.
.
(check-sat)
.
.
(check-sat)

Solver output

sat / unsat

timeout: 40 min
Application Track

Application track benchmarks are fed to the solver \textit{incrementally} by a trace executor.

\begin{itemize}
\item Application Track benchmark
  \begin{itemize}
  \item (set-logic ...)
  \item (check-sat)
  \item (check-sat)
  \item (check-sat)
  \item (check-sat)
  \item (exit)
  \end{itemize}
\item Solver input
  \begin{itemize}
  \item (set-option :print-success true)
  \item (set-logic ...)
  \item (check-sat)
  \item (check-sat)
  \item (check-sat)
  \item (check-sat)
  \end{itemize}
\item Solver output
  \begin{itemize}
  \item sat / unsat
  \item sat / unsat
  \end{itemize}
\end{itemize}

\textbf{timeout: 40 min}
Application Track

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

timeout: 40 min

Application Track benchmark

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

Solver input

(set-option :print-success true)
(set-logic ...)
.
.
(check-sat)
.
.
(check-sat)
.
.
(check-sat)

Solver output

sat / unsat
sat / unsat
sat / unsat
Application Track

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

Application Track benchmark

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

Solver input

(set-option :print-success true)
(set-logic ...)
...
(check-sat)
...
(check-sat)
...
(check-sat)

Solver output

sat / unsat
...
sat / unsat
...
sat / unsat

timeout: 40 min
Application track benchmarks are fed to the solver **incrementally** by a trace executor.

---

**Application Track**

**Solver input**

```
(set-option :print-success true)
(set-logic ...)
...
(check-sat)
...
(check-sat)
...
(check-sat)
...
(check-sat)
(exit)
```

**Application Track benchmark**

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

**Solver output**

```
sat / unsat
sat / unsat
sat / unsat
```

**timeout:** 40 min
Application Track

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

Application Track benchmark

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

Solver input

(set-option :print-success true)
(set-logic ...)
...
(check-sat)
...
(check-sat)
...
(check-sat)
...
(check-sat)
...
(check-sat)
...
(check-sat)
...
(check-sat)

Solver output

sat / unsat
sat / unsat
sat / unsat
sat / unsat

timeout: 40 min
Application Track

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

Application Track benchmark

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

Solver input

(set-option :print-success true)
(set-logic ...)
...
(check-sat)
...
(check-sat)
...
(check-sat)
...
(check-sat)
(exit)

Solver output

sat / unsat
...
...
...
sat / unsat
...
...
...
sat / unsat

timeout: 40 min

n = # correct sat / unsat responses
e = 1 if the solver gives an incorrect sat / unsat response
Application track benchmarks are fed to the solver incrementally by a trace executor.

**Scoring**

\[ n = \# \text{ correct sat/unsat responses} \]

\[ e = 1 \quad \text{if the solver gives an incorrect sat/unsat response} \]

**Solver input**

```
(set-option :print-success true)
(set-logic ...)
(check-sat)
(check-sat)
(check-sat)
(check-sat)
(exit)
```

**Solver output**

```
sat / unsat
sat / unsat
sat / unsat
sat / unsat
(exit)
```

**timeout:** 40 min
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

(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

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 ...)
(define-fun ...)
(assert (! term0 :named y0))
(assert (! term1 :named y1))
(assert (! term2 :named y2))
...
(check-sat)
(get-unsat-core)
(exit)

Validation script

(set-logic ...)
(set-info ...)
...
(declare-sort ...)
(define-sort ...)
(define-fun ...)
(assert term1)
(assert term2)
(assert term3)
...
(check-sat)
(exit)

Solver output

unsat
(y0 y2)

timeout: 40 min
Unsat-Core Track

Main Track benchmark (unsat)

(set-option :produce-unsat-cores true)
(set-logic ...)
(set-info ...)
(declare-sort ...)
(define-sort ...)
(declare-fun ...)
(define-fun ...)
(assert term0)
(assert term1)
(assert term2)
(...)
(check-sat)
(exit)

Solver input

Solver output

unsat
(y0 y2)

Validation script

Validation solver 1
sat/unsat/unknown
Validation solver 2
sat/unsat/unknown
Validation solver 3
sat/unsat/unknown
Validation solver 4
sat/unsat/unknown

timeout: 40 min

timeout: 2 min each

timeout: 2 min each

Validation
solver 1
Validation
solver 2
Validation
solver 3
Validation
solver 4

timeout: 40 min
Unsat-Core Track

Main Track benchmark
(unsat)

(set-logic ...)
(set-info ...)
...
(declare-sort ...)

Solver input
(set-option :produce-unsat-cores true)
(set-logic ...)
(set-info ...)
...
(declare-sort ...)

Validation script
(set-logic ...)
(set-info ...)
...
(declare-sort ...)

Scoring

\[ n = \# \text{ assert commands} - \text{size of unsatisfiable core} \]

\[ e = 1 \quad \text{if} \quad \begin{cases} \text{wrong check-sat result, or} \\ \text{unsat-core rejected by validating solvers} \end{cases} \]

Solver output

unsat
(y0 y2)

timeout: 40 min

Validation

solver 1
sat/
unsat/
unknown

solver 2
sat/
unsat/
unknown

solver 3
sat/
unsat/
unknown

solver 4
sat/
unsat/
unknown

timeout: 2 min each
17 teams participated

**Solvers:**

- **Main track**: 20 solvers, 4 non-competing
- **Application track**: 4 solvers, 2 non-competing
- **Unsat-core track**: 3 solvers, 2 non-competing

**Logics:**

- **Main track**: 49 logics, 1 experimental
- **Application track**: 21
- **Unsat-core track**: 44

**Benchmarks:**

- **Main track**: 333241
- **Application track**: 9257
- **Unsat-core track**: 130705
1,776,062 job pairs (+ some repeats)
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

\[ \sim 17 \text{ days} \times 120 \text{ nodes} \times 2 \text{ processors/node} \text{ of compute time} \]
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
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
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
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
Quantifier Free Linear Arithmetic

\[ y \leq i + 1 \]

\[ i \leq y \]

\[ y - \text{to_int}(y) < .3 \]

Quantifier Free Uninterpreted Functions

\[ f(b) = v \]

\[ f(a) \neq v \]

Theory combination

\[ b[i] \geq i \]

\[ f(i + y) = 2v \]

\[ f(b) \leq i \]

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

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVC4</td>
<td></td>
</tr>
<tr>
<td>SMTInterpol</td>
<td></td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td></td>
</tr>
</tbody>
</table>
Unsat-Core Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVC4</td>
<td>QF_AUFLIA, QF_IDL, QF_LIRA, QF_RDL, QF_UF</td>
</tr>
<tr>
<td>SMTInterpol</td>
<td></td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td></td>
</tr>
</tbody>
</table>
Unsat-Core Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVC4</td>
<td>QF_AUFLIA, QF_IDL, QF_LIRA, QF_RDL, QF_UF</td>
</tr>
<tr>
<td>SMTInterpol</td>
<td>QF_LIA, QF_LRA, QF_UFLIA</td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td></td>
</tr>
</tbody>
</table>
Unsat-Core Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVC4</td>
<td>QF_AUFLIA, QF_IDL, QF_LIRA, QF_RDL, QF_UF</td>
</tr>
<tr>
<td>SMTInterpol</td>
<td>QF_LIA, QF_LRA, QF_UFLIA</td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td>QF_ABV, QF_ALIA, QF_AUFBV, QF_AX, QF_BV, QF_UFBV, QF_UFIDL, QF_UFLRA</td>
</tr>
</tbody>
</table>
Application Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td></td>
</tr>
<tr>
<td>CVC4</td>
<td></td>
</tr>
<tr>
<td>SMTInterpol</td>
<td></td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td></td>
</tr>
</tbody>
</table>
Application Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_UFBV</td>
</tr>
<tr>
<td>CVC4</td>
<td></td>
</tr>
<tr>
<td>SMTInterpol</td>
<td></td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td></td>
</tr>
</tbody>
</table>
Application Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_UFBV</td>
</tr>
<tr>
<td>CVC4</td>
<td>QF_NIA, QF_UFNIA</td>
</tr>
<tr>
<td>SMTInterpol</td>
<td></td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td></td>
</tr>
</tbody>
</table>
Application Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_UFBV</td>
</tr>
<tr>
<td>CVC4</td>
<td>QF_NIA, QF_UFNIA</td>
</tr>
<tr>
<td>SMTInterpol</td>
<td>QF_ALIA, QF_UFLIA</td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td></td>
</tr>
</tbody>
</table>
## Application Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_UFBV</td>
</tr>
<tr>
<td>CVC4</td>
<td>QF_NIA, QF_UFNIA</td>
</tr>
<tr>
<td>SMTInterpol</td>
<td>QF_ALIA, QF_UFLIA</td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td>QF_AUFBV, QF_AUFLIA, QF_BV, QF_LIA, QF_LRA, QF_UFLRA</td>
</tr>
</tbody>
</table>
Main Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF BV seq ABV, QF BV seq</td>
</tr>
<tr>
<td>QF ABV, BV seq</td>
<td></td>
</tr>
<tr>
<td>COLIBRI</td>
<td>QF FP</td>
</tr>
<tr>
<td>CVC4</td>
<td>ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV seq, UF seq, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA</td>
</tr>
<tr>
<td>SMTRAT</td>
<td>QF NIRA</td>
</tr>
<tr>
<td>SPASS-SATT</td>
<td>QF LIA</td>
</tr>
<tr>
<td>Vampire</td>
<td>NRA, UF par, UFNIA</td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td>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</td>
</tr>
</tbody>
</table>
Main Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_BV^\text{seq}, QF_UFBV</td>
</tr>
</tbody>
</table>
Main Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_BV^{seq}, QF_UFBV</td>
</tr>
<tr>
<td>COLIBRI</td>
<td>QF_FP</td>
</tr>
</tbody>
</table>
## Main Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_BV(^{\text{seq}}), QF_UFBV</td>
</tr>
<tr>
<td>COLIBRI</td>
<td>QF_FP</td>
</tr>
<tr>
<td>CVC4</td>
<td>ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA, QF_NIA, UF(^{\text{seq}}), UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA</td>
</tr>
</tbody>
</table>
Main Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_BV&lt;sup&gt;seq&lt;/sup&gt;, QF_UFBV</td>
</tr>
<tr>
<td>COLIBRI</td>
<td>QF_FP</td>
</tr>
<tr>
<td>CVC4</td>
<td>ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA, QF_NIA, UF&lt;sup&gt;seq&lt;/sup&gt;, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA</td>
</tr>
<tr>
<td>Minkeyrink-MT</td>
<td>QF_BV&lt;sup&gt;par&lt;/sup&gt;</td>
</tr>
</tbody>
</table>
Main Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_BV&lt;sup&gt;seq&lt;/sup&gt;, QF_UFBV</td>
</tr>
<tr>
<td>COLIBRI</td>
<td>QF_FP</td>
</tr>
<tr>
<td>CVC4</td>
<td>ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVF, QF_LRA, QF_NIA, UF&lt;sup&gt;seq&lt;/sup&gt;, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA</td>
</tr>
<tr>
<td>Minkeyrink-MT</td>
<td>QF_BV&lt;sup&gt;par&lt;/sup&gt;</td>
</tr>
<tr>
<td>SMTRAT</td>
<td>QF_NIRA</td>
</tr>
</tbody>
</table>
Main Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_BVseq, QF_UFBV</td>
</tr>
<tr>
<td>COLIBRI</td>
<td>QF_FP</td>
</tr>
<tr>
<td>CVC4</td>
<td>ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA,</td>
</tr>
<tr>
<td></td>
<td>LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA,</td>
</tr>
<tr>
<td></td>
<td>QF_NIA, UFseq, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA</td>
</tr>
<tr>
<td>Minkeyrink-MT</td>
<td>QF_BVpar</td>
</tr>
<tr>
<td>SMTRAT</td>
<td>QF_NIRA</td>
</tr>
<tr>
<td>SPASS-SATT</td>
<td>QF_LIA</td>
</tr>
</tbody>
</table>
Main Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_BV$^{\text{seq}}$, QF_UFBV</td>
</tr>
<tr>
<td>COLIBRI</td>
<td>QF_FP</td>
</tr>
<tr>
<td>CVC4</td>
<td>ALIA, AUFDTLIA, AUFLIA, AUFLIRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA, QF_NIA, UF$^{\text{seq}}$, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA</td>
</tr>
<tr>
<td>Minkeyrink-MT</td>
<td>QF_BV$^{\text{par}}$</td>
</tr>
<tr>
<td>SMTRAT</td>
<td>QF_NIRA</td>
</tr>
<tr>
<td>SPASS-SATT</td>
<td>QF_LIA</td>
</tr>
<tr>
<td>Vampire</td>
<td>NRA, UF$^{\text{par}}$, UFNIA</td>
</tr>
</tbody>
</table>
### Main Track

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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Divisions won</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolector</td>
<td>QF_ABV, QF_BV^{seq}, QF_UFBV</td>
</tr>
<tr>
<td>COLIBRI</td>
<td>QF_FP</td>
</tr>
<tr>
<td>CVC4</td>
<td>ALIA, AUFDTLIA, AUFLIA, AUFLRA, AUFNIRA, BV, LIA, LRA, NIA, QF_ABVFP, QF_AUFBV, QF_BVFP, QF_LRA, QF_NIA, UF^{seq}, UFDT, UFDTLIA, UFIDL, UFLIA, UFLRA</td>
</tr>
<tr>
<td>Minkeyrink-MT</td>
<td>QF_BV^{par}</td>
</tr>
<tr>
<td>SMTRAT</td>
<td>QF_NIRA</td>
</tr>
<tr>
<td>SPASS-SATT</td>
<td>QF_LIA</td>
</tr>
<tr>
<td>Vampire</td>
<td>NRA, UF^{par}, UFNIA</td>
</tr>
<tr>
<td>Yices-2.6.0</td>
<td>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</td>
</tr>
</tbody>
</table>
# Main Track: Competition-Wide Scoring

<table>
<thead>
<tr>
<th>Rank</th>
<th>Solver</th>
<th>Score (sequential)</th>
<th>Score (parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CVC4</td>
<td>211.99</td>
<td>211.99</td>
</tr>
<tr>
<td>2</td>
<td>Z3</td>
<td>186.19</td>
<td>186.19</td>
</tr>
<tr>
<td>3</td>
<td>Yices-2.6.0</td>
<td>115.26</td>
<td>115.26</td>
</tr>
<tr>
<td>4</td>
<td>SMTInterpol</td>
<td>65.32</td>
<td>65.38</td>
</tr>
</tbody>
</table>

**Best newcomer:**

7    SPASS-SATT  14.81  14.81
Main Track: Competition-Wide Scoring

<table>
<thead>
<tr>
<th>Rank</th>
<th>Solver</th>
<th>Score (sequential)</th>
<th>Score (parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CVC4</td>
<td>211.99</td>
<td>211.99</td>
</tr>
<tr>
<td>2</td>
<td>Z3</td>
<td>186.19</td>
<td>186.19</td>
</tr>
<tr>
<td>3</td>
<td>Yices-2.6.0</td>
<td>115.26</td>
<td>115.26</td>
</tr>
<tr>
<td></td>
<td>SMTInterpol</td>
<td>65.32</td>
<td>65.38</td>
</tr>
</tbody>
</table>

Best newcomer:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Solver</th>
<th>Score (sequential)</th>
<th>Score (parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>SPASS-SATT</td>
<td>14.81</td>
<td>14.81</td>
</tr>
</tbody>
</table>
## Main Track: Competition-Wide Scoring

<table>
<thead>
<tr>
<th>Rank</th>
<th>Solver</th>
<th>Score (sequential)</th>
<th>Score (parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CVC4</td>
<td>211.99</td>
<td>211.99</td>
</tr>
<tr>
<td>2</td>
<td>Z3</td>
<td>186.19</td>
<td>186.19</td>
</tr>
<tr>
<td>3</td>
<td>Yices-2.6.0</td>
<td>115.26</td>
<td>115.26</td>
</tr>
<tr>
<td>4</td>
<td>SMTInterpol</td>
<td>65.32</td>
<td>65.38</td>
</tr>
<tr>
<td>7</td>
<td>SPASS-SATT</td>
<td>14.81</td>
<td>14.81</td>
</tr>
</tbody>
</table>

Best newcomer:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Solver</th>
<th>Score (sequential)</th>
<th>Score (parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>SPASS-SATT</td>
<td>14.81</td>
<td>14.81</td>
</tr>
</tbody>
</table>
## Main Track: Competition-Wide Scoring

<table>
<thead>
<tr>
<th>Rank</th>
<th>Solver</th>
<th>Score (sequential)</th>
<th>Score (parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CVC4</td>
<td>211.99</td>
<td>211.99</td>
</tr>
<tr>
<td>2</td>
<td>Z3</td>
<td>186.19</td>
<td>186.19</td>
</tr>
<tr>
<td>3</td>
<td>Yices-2.6.0</td>
<td>115.26</td>
<td>115.26</td>
</tr>
<tr>
<td></td>
<td>SMTInterpol</td>
<td>65.32</td>
<td>65.38</td>
</tr>
</tbody>
</table>

Best newcomer:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Solver</th>
<th>Score (sequential)</th>
<th>Score (parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>SPASS-SATT</td>
<td>14.81</td>
<td>14.81</td>
</tr>
</tbody>
</table>
# Main Track: Competition-Wide Scoring

<table>
<thead>
<tr>
<th>Rank</th>
<th>Solver</th>
<th>Score (sequential)</th>
<th>Score (parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CVC4</td>
<td>211.99</td>
<td>211.99</td>
</tr>
<tr>
<td></td>
<td>Z3</td>
<td>186.19</td>
<td>186.19</td>
</tr>
<tr>
<td>2</td>
<td>Yices-2.6.0</td>
<td>115.26</td>
<td>115.26</td>
</tr>
<tr>
<td>3</td>
<td>SMTInterpol</td>
<td>65.32</td>
<td>65.38</td>
</tr>
</tbody>
</table>

Best newcomer:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Solver</th>
<th>Score (sequential)</th>
<th>Score (parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>SPASS-SATT</td>
<td>14.81</td>
<td>14.81</td>
</tr>
</tbody>
</table>
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%)