

SAT Competition 2017

Overview and Results

Tomáš Balyo¹ Marijn Heule² Matti Järvisalo³

¹ Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Germany

² Department of Computer Science, The University of Texas at Austin, USA

³ HIIT, Department of Computer Science, University of Helsinki, Finland

August 31, 2017 @ SAT'17, Melbourne

SAT Solver Competitions

Goals

- identify new challenging benchmarks
- promote SAT solvers & their development
- “snapshot” evaluation of current solvers

Long tradition, starting from 1992

- 3 competitions in the 90s (1992, 1993, 1996)
- 11 SAT Competitions (2002–)
- 4 SAT Races (2006, 2008, 2010, 2015)
- 1 SAT Challenge (2012)

Key rules

- Certified UNSAT using DRAT proof logging
- Disqualification of buggy solvers
 - Provided model incorrect
 - Report UNSAT on know-to-be-satisfiable instance
 - Proof check fails on UNSAT instance → “timeout”
transition-period rule, will likely be changed
- Mandatory solver descriptions + open source

New for 2017

- Ranking scheme: PAR-2
 - Favors solvers that are faster (not only count solved instances)
- BYOB — Bring your own beer benchmarks
 - Each submitter *must* submit 20 benchmarks
- Proofs of unsatisfiability certified by a theorem prover
 - Proofs were converted into LRAT and checked with ACL2
- Updated IPASIR interface (for incremental track)
 - Learned clauses can be extracted from the solvers (implemented via callback function)
- Many new benchmarks in the Incremental Track

Tracks

Tracks

Track	Benchmarks	Solvers	Limits	Cluster
Main (sequential)	350 main app + crafted	28	5000 s, 1 core, 24 GB 20 000 s DRAT	StarExec
Parallel	350 main	10	5000 s / 64 GB 24 cores / 48 threads	TACC
Incremental	apps & inputs	4	300 s / 24 GB	KIT
Agile	5000 SMT	16	60 s / 24 GB	StarExec
Random SAT	(planted) <i>k</i> -SAT	6	5000 s / 24 GB	StarExec
No-limits	270 new main	18	5000 s / 24 GB	StarExec

Total number of solvers (solver versions) submitted: 82

Tracks

Track	Benchmarks	Solvers	Limits	Cluster
Main (sequential)	350 main app + crafted	28	5000 s, 1 core, 24 GB 20 000 s DRAT	StarExec
Parallel	350 main	10	5000 s / 64 GB 24 cores / 48 threads	TACC
Incremental	apps & inputs	4	300 s / 24 GB	KIT
Agile	5000 SMT	16	60 s / 24 GB	StarExec
Random SAT	(planted) <i>k</i> -SAT	6	5000 s / 24 GB	StarExec
No-limits	270 new main	18	5000 s / 24 GB	StarExec

Total number of solvers (solver versions) submitted: 82

Towards a leaner competition?

Interest in the Agile, Random SAT, and No-limits tracks appears limited.

Benchmarks

Main: Several new benchmark domains/sets submitted, including Rubik's Cube, Equivalence Checking, Bounded Model Checking, Pseudo Industrial, Cryptography (SHA-1), Latin Square, Balanced Random, Block Puzzle, Subshape in Grid, and Polynomial Multiplication.

Incremental

- Instances of SAT-based applications under various inputs (assumptions)
- average rank for each application determines winner

Agile: Bit-blasted Z3 instances. Selected 5000 from a suite submitted last year, a significant overlap with last year's selection.

Random: Satisfiable k -SAT. Three types: medium size close to the phase transition, huge and somewhat below the phase transition, and hard planted SAT.

Results

Results

Note: Medals will be posted from Germany.

All medalists should provide us a current mailing address via email.

Random Track: Top-3

Random Track: Top-3

-
-
3. **Score2SAT** (2075884.23)
by Shaowei Cai, Chuan Luo

Random Track: Top-3

-
2. **tch glucose3** (1894840.68)
by Seongsoo Moon, Inaba Mary
3. **Score2SAT** (2075884.23)
by Shaowei Cai, Chuan Luo

Random Track: Top-3

1. **YalSAT** (1794277.02)
by Armin Biere
2. **tch glucose3** (1894840.68)
by Seongsoo Moon, Inaba Mary
3. **Score2SAT** (2075884.23)
by Shaowei Cai, Chuan Luo

Random Track: Top-3

1. **YalSAT** (1794277.02)
by Armin Biere
2. **tch glucose3** (1894840.68)
by Seongsoo Moon, Inaba Mary
3. **Score2SAT** (2075884.23)
by Shaowei Cai, Chuan Luo

Dimetheus, the strongest solver in recent years, did not participate.

Incremental Track: Top-3

Incremental Track: Top-3

-
2. **Riss** (average rank: 2.000)
by Norbert Manthey

Incremental Track: Top-3

2. **Glucose** (average rank: 2.000)
by Gilles Audemard, Laurent Simon
2. **Riss** (average rank: 2.000)
by Norbert Manthey

Incremental Track: Top-3

1. **AbcdSAT** (average rank: 1.875)
by Jingchao Chen
2. **Glucose** (average rank: 2.000)
by Gilles Audemard, Laurent Simon
2. **Riss** (average rank: 2.000)
by Norbert Manthey

Incremental Track: Ranking for each application

Benchmark App	AbcdSAT	Glucose	Riss
Essential Variables	2	1	3
Longest Path Search	1	2	3
Partial MaxSAT	1	2	3
Pigeon Hole Principle	3	1	1
Automated Planning	1	2	3
ijthead (QBF solver)	2	3	1
CBMC (C verification)	3	2	1
SATPin (axiom pinpointing)	2	3	1
average rank	1.875	2.000	2.000

Parallel Track: Top-3

-
-
3. **Painless MapleCOMSPS** (1368420.48)
by Ludovic Le Frioux, Souheib Baarir, Julien Sopena, Fabrice Kordon

Parallel Track: Top-3

-
2. **Plingeling** (1266163.50)
by Armin Biere
3. **Painless MapleCOMSPS** (1368420.48)
by Ludovic Le Frioux, Souheib Baarir, Julien Sopena, Fabrice Kordon

Parallel Track: Top-3

1. **Syrup24** (1229297.31)
Syrup48 (1334230.93)
by Gilles Audemard, Laurent Simon
2. **Plingeling** (1266163.50)
by Armin Biere
3. **Painless MapleCOMSPS** (1368420.48)
by Ludovic Le Frioux, Souheib Baarir, Julien Sopena, Fabrice Kordon

Parallel Track: Top-3

1. **Syrup24** (1229297.31)
Syrup48 (1334230.93)
by Gilles Audemard, Laurent Simon
2. **Plingeling** (1266163.50)
by Armin Biere
3. **Painless MapleCOMSPS** (1368420.48)
by Ludovic Le Frioux, Souheib Baarir, Julien Sopena, Fabrice Kordon

Less is more

Solvers that did not use hyper-threading were faster

Agile Track: Top-3

-
-
3. **Glucose 4.1** (263772.21)
by Gilles Audemard, Laurent Simon

Agile Track: Top-3

-
2. **Glu_VC** (257997.93)
by Jingchao Chen
3. **Glucose 4.1** (263772.21)
by Gilles Audemard, Laurent Simon

Agile Track: Top-3

1. **CaDiCal Agile** (240983.15)
CaDiCal NoProof (248613.05)
by Armin Biere
2. **Glu_VC** (257997.93)
by Jingchao Chen
3. **Glucose 4.1** (263772.21)
by Gilles Audemard, Laurent Simon

Agile Track: Top-3

1. **CaDiCal Agile** (240983.15)
CaDiCal NoProof (248613.05)
by Armin Biere
2. **Glu_VC** (257997.93)
by Jingchao Chen
3. **Glucose 4.1** (263772.21)
by Gilles Audemard, Laurent Simon

Few new benchmarks!

No-Limits Track: Top-3

No-Limits Track: Top-3

-
-
3. **CaDiCal NoProof** (1841142.37)
by Armin Biere

No-Limits Track: Top-3

2. **MapleCOMSPS LRB VSIDS 2** (1758801.49)
MapleCOMSPS LRB VSIDS (1789914.81)
by Jia Hui Liang, Chanseok Oh, Vijay Ganesh, Krzysztof Czarnecki,
Pascal Poupart
3. **CaDiCal NoProof** (1841142.37)
by Armin Biere

No-Limits Track: Top-3

1. **COMiniSatPS Pulsar** (1758011.46)
by Chanseok Oh
2. **MapleCOMSPS LRB VSIDS 2** (1758801.49)
MapleCOMSPS LRB VSIDS (1789914.81)
by Jia Hui Liang, Chanseok Oh, Vijay Ganesh, Krzysztof Czarnecki,
Pascal Poupart
3. **CaDiCal NoProof** (1841142.37)
by Armin Biere

No-Limits Track: Top-3

1. **COMiniSatPS Pulsar** (1758011.46)
by Chanseok Oh
2. **MapleCOMSPS LRB VSIDS 2** (1758801.49)
MapleCOMSPS LRB VSIDS (1789914.81)
by Jia Hui Liang, Chanseok Oh, Vijay Ganesh, Krzysztof Czarnecki,
Pascal Poupart
3. **CaDiCal NoProof** (1841142.37)
by Armin Biere

No “real” no-limits solvers?

Main Track: Top-3

-
-
3. **COMiniSatPS Pulsar** (1798300.38)
by Chanseok Oh

Main Track: Top-3

2. **MapleCOMSPS LRB VSIDS 2** (1780711.47)
MapleCOMSPS LRB VSIDS (1805445.41)
by Jia Hui Liang, Chanseok Oh, Vijay Ganesh, Krzysztof Czarnecki,
Pascal Poupart
3. **COMiniSatPS Pulsar** (1798300.38)
by Chanseok Oh

Main Track: Top-3

1. **Maple LCM Dist** (1610934.19)
Maple LCM (1640696.51)
MapleLRB LCMoccRestart (1654244.83)
MapleLRB LCM (1676517.65)
by Fan Xiao, Mao Luo, Chu-Min Li, Felip Manyà, Zhipeng Lü
2. **MapleCOMSPS LRB VSIDS 2** (1780711.47)
MapleCOMSPS LRB VSIDS (1805445.41)
by Jia Hui Liang, Chanseok Oh, Vijay Ganesh, Krzysztof Czarnecki,
Pascal Poupart
3. **COMiniSatPS Pulsar** (1798300.38)
by Chanseok Oh

Main Track: Top-3

1. **Maple LCM Dist** (1610934.19)
Maple LCM (1640696.51)
MapleLRB LCMoccRestart (1654244.83)
MapleLRB LCM (1676517.65)
by Fan Xiao, Mao Luo, Chu-Min Li, Felip Manyà, Zhipeng Lü
2. **MapleCOMSPS LRB VSIDS 2** (1780711.47)
MapleCOMSPS LRB VSIDS (1805445.41)
by Jia Hui Liang, Chanseok Oh, Vijay Ganesh, Krzysztof Czarnecki, Pascal Poupart
3. **COMiniSatPS Pulsar** (1798300.38)
by Chanseok Oh

No solver announced as Glucose hack

Impact of PAR-2

Penalized average runtime (PAR)

- PAR- x : penalized timeouts by $x \cdot \text{TIMEOUT}$
- SCR, solution-count ranking: PAR- x as $x \rightarrow \infty$.
- x balances average succesful runtimes and number of solved instances

In 2017: little differences between PAR-2 and SCR.

Impact of PAR-2

Penalized average runtime (PAR)

- PAR- x : penalized timeouts by $x \cdot \text{TIMEOUT}$
- SCR, solution-count ranking: PAR- x as $x \rightarrow \infty$.
- x balances average succesful runtimes and number of solved instances

In 2017: little differences between PAR-2 and SCR.

Determined winner in the parallel track:

PAR-2

1. **Syrup24** (1229297.31)
by Audemard and Simon
2. **Plingeling** (1266163.50)
by Armin Biere
3. **Syrup48** (1334230.93)
by Audemard and Simon

SCR

- (1). **Plingeling** (239)
by Armin Biere
- (2). **Syrup24** (237)
by Audemard and Simon
- (3). **Syrup48** (227)
by Audemard and Simon

Disclaimer: Results Depend Heavily on the Benchmarks

Benchmark selection is the achilles heel of the competition.

Disclaimer: Results Depend Heavily on the Benchmarks

Benchmark selection is the achilles heel of the competition.

- Few new benchmarks submitted.

Disclaimer: Results Depend Heavily on the Benchmarks

Benchmark selection is the achilles heel of the competition.

- Few new benchmarks submitted.
- Forcing solver submitters to provide benchmarks slightly improves the situation but mostly reduces solver submissions.

Disclaimer: Results Depend Heavily on the Benchmarks

Benchmark selection is the achilles heel of the competition.

- Few new benchmarks submitted.
- Forcing solver submitters to provide benchmarks slightly improves the situation but mostly reduces solver submissions.
- Armin Biere submitted a suite that if included would have guaranteed him winning the competition. He warned us in advance.

Disclaimer: Results Depend Heavily on the Benchmarks

Benchmark selection is the achilles heel of the competition.

- Few new benchmarks submitted.
- Forcing solver submitters to provide benchmarks slightly improves the situation but mostly reduces solver submissions.
- Armin Biere submitted a suite that if included would have guaranteed him winning the competition. He warned us in advance.
- It is easy to win the competition by submitting benchmarks that favor your solver and people are still reluctant to submit.

Disclaimer: Results Depend Heavily on the Benchmarks

Benchmark selection is the achilles heel of the competition.

- Few new benchmarks submitted.
- Forcing solver submitters to provide benchmarks slightly improves the situation but mostly reduces solver submissions.
- Armin Biere submitted a suite that if included would have guaranteed him winning the competition. He warned us in advance.
- It is easy to win the competition by submitting benchmarks that favor your solver and people are still reluctant to submit.

This part of the competition must become more scientific, but how?

Final Remarks

Full details (to be available) at
<https://baldur.iti.kit.edu/sat-competition-2017/>

- Detailed per-instance per-solver results
- Proceedings: solver and benchmark descriptions
- These slides

Final Remarks

Full details (to be available) at
<https://baldur.iti.kit.edu/sat-competition-2017/>

- Detailed per-instance per-solver results
- Proceedings: solver and benchmark descriptions
- These slides

Many thanks to

- all solver submitters and developers
- all benchmark submitters
- Aaron Stump and StarExec
- TACC for the Lonestar5 resources
- SAT Association for support for awards

Thank you for your attention!