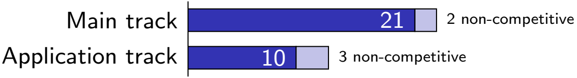
10th International Satisfiability Modulo Theories Competition

SMT-COMP 2015

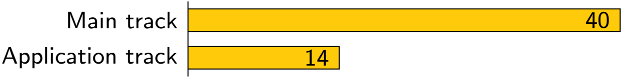
Sylvain Conchon David Déharbe Tjark Weber

The Numbers

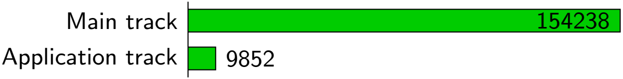
- 11 teams participated
- Solvers:



- Logics:



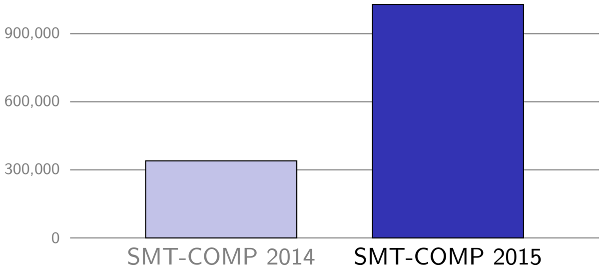
- Benchmarks:



Record numbers of solvers, logics, and benchmarks!

Job Pairs

- 1,028,615 job pairs executed (+ some repeats)
- $\sim$ 5 days $\times$ 150 nodes $\times$ 2 processors/node of compute time



More than 3 times as many job pairs as in 2014!

StarExec

- All job pairs executed on StarExec
- Over 9,000 job pairs/hour completed

StarExec worked great

- Thanks to Aaron Stump for prompt help when problems or questions arose
- $\sim$ 20 feature requests and (minor) bug reports submitted to the StarExec developers

Machine Specifications

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

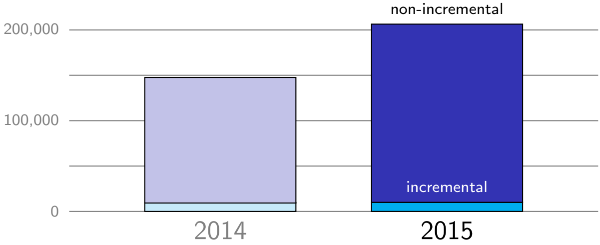
Software:

- Red Hat Enterprise Linux Workstation release 6.3
- Kernel 2.6.32-431, gcc 4.4.6, glibc 2.12 ($\sim$ 2009-2011)
- Virtual machine image available before the competition

Problems with missing libraries (due to dynamic linking) in several solvers resolved during pre-competition testing in early June.

Benchmarks and Logics

- Almost 60,000 new benchmarks added to SMT-LIB, thanks to several contributors:

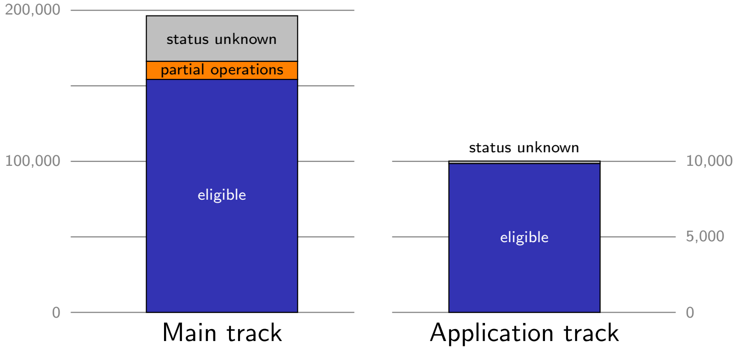


- Six new logics, including two new floating-point logics
- Thanks to Clark Barrett for curation and uploading

Benchmark Curation

- Sanity checks
  - One satisfiability check per benchmark in main track
  - Status information set before satisfiability check
- Verify benchmark signature against logic set
- Remove unused symbols
- Improve logic settings

Eligible Benchmarks



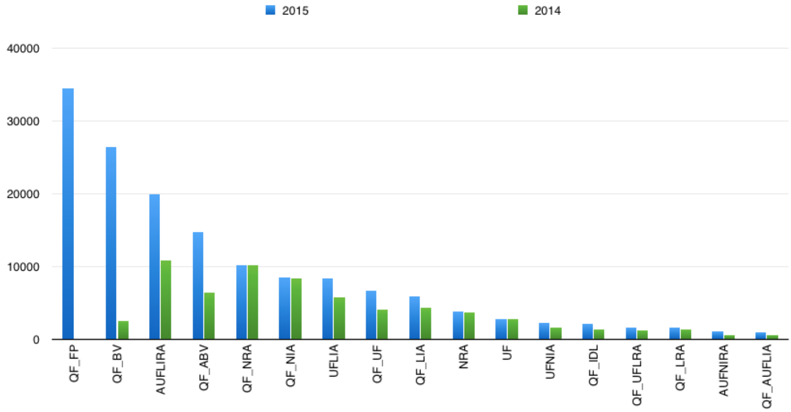
All eligible benchmarks were used for the competition. There was no further selection.

Competition Tools Improved

- Fixed an issue where the trace executor would sometimes not count correct solver responses on partially solved incremental benchmarks. (Thanks to Kshitij Bansal for reporting this.)
- Fixed several issues in the benchmark scrambler that caused invalid output in the presence of variable shadowing.

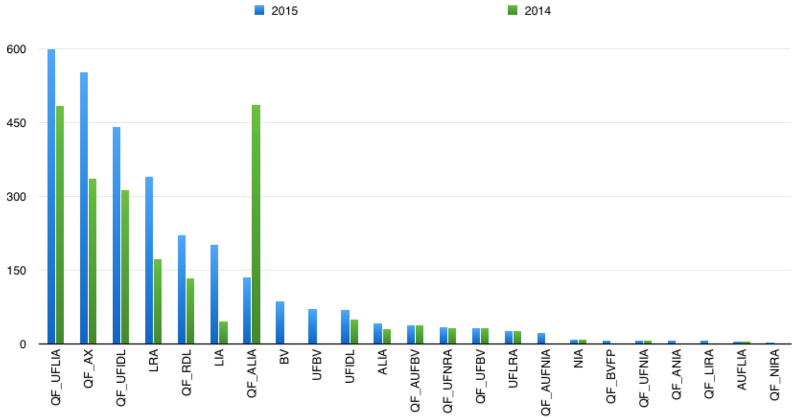
Evolution of Benchmarks: Breakdown

Tier 1 (> 1000 Benchmarks)



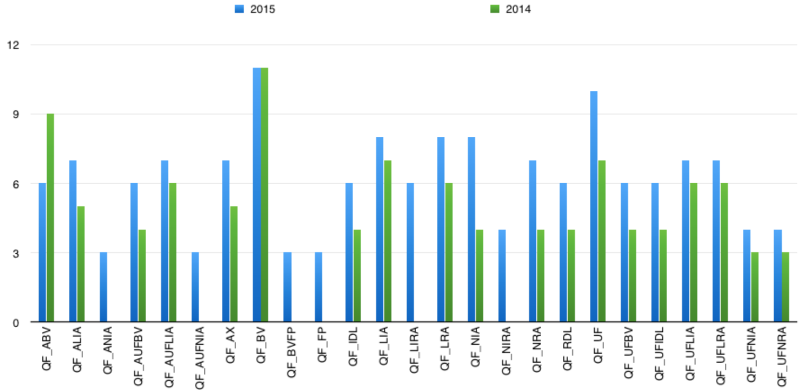
Evolution of Benchmarks: Breakdown

Tier 2 (< 1000 Benchmarks)



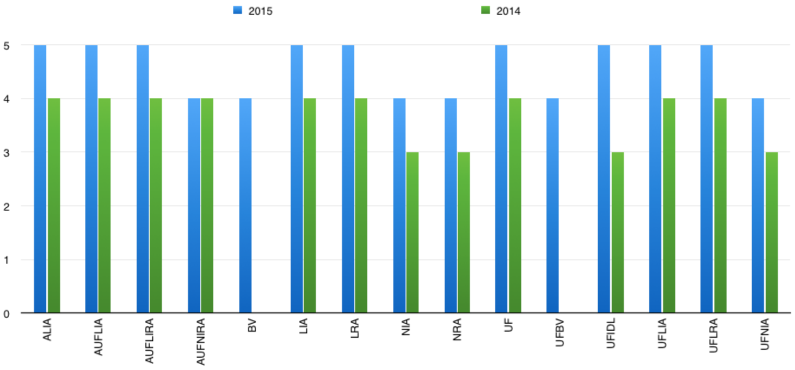
Evolution of Tool Participation: Breakdown

Quantifier-Free Logics



Evolution of Tool Participation: Breakdown

Logics with Quantifiers



Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Teams

- CVC4
- Yices2
- SMTInterpol
- veriT
- STP-MiniSat
- STP-CryptoMiniSat
- openSMT2
- AProVE
- Boolector
- raSAT
- SMT-RAT

Scoring

Raw Scores

A solver's raw score for each benchmark is $\langle e, n, \mathit{wall}, \mathit{cpu} \rangle$, with

- $e \in \{0, 1\}$, the number of erroneous results
- $0 \leq n \leq N$, the number of correct results (N is the number of `check-sat` commands in the benchmark)
- *wall* is the wall-clock (or real) time
- *cpu* is the CPU time
  $\rightarrow$ For programs running in parallel, *cpu* is the sum of CPU times devoted to each task

Track Scoring

Main track

- *Timeouts*, *aborts* (no answer), `unknown`: $\langle 0, 0, \mathit{wall}, \mathit{cpu} \rangle$
- *Incorrect* answers: $\langle 1, 0, \mathit{wall}, \mathit{cpu} \rangle$
- *Correct* answers: $\langle 0, 1, \mathit{wall}, \mathit{cpu} \rangle$

Application track (multiple `checksat` per benchmark)

- *Any incorrect* result : $\langle 1, 0, \mathit{wall}, \mathit{cpu} \rangle$
- Otherwise : $\langle 0, n, \mathit{wall}, \mathit{cpu} \rangle$

Sequential Performances

Given a wall-clock time limit T and a raw score $\langle e, n, wall, cpu \rangle$, we derive a sequential score to evaluate sequential performances:

- If $cpu > T$ then $\langle 0, 0, T \rangle$
- Otherwise $\langle e, n, cpu \rangle$

Division Scoring

For each division, scores are summed component-wise:

- **Sequential performances** = sum all **sequential scores**
- **Parallel performances** = sum all **raw scores**

We compute :

- **Sequential** and **parallel** performances for main track divisions
- Only **parallel** performances for application track divisions

Division scores are compared lexicographically :

Fewer errors takes precedence over more correct solutions, which takes precedence over less wall-clock time taken, which takes precedence over less CPU time taken

Competition Wide Scoring

We define the competition wide score of each solver for the **main track**, separately for sequential and parallel performances

For each *competitive* division i, let N_i be the total number of benchmarks in that division and $\langle e_i, n_i, ...\rangle$ the raw (resp. sequential) score of the solver for i

The competition-wide score of a solver is :

$$\sum_i (\text{if } e_i = 0 \text{ then } (n_i/N_i)^2 \text{ else } -e_i) \times logN_i$$

Results

Results : Main Track

40 divisions but only 28 declared as competitive

Sequential performances (parallel perfs. are identical)

Solver	# Divisions won	Divisions
CVC4 (2 versions)	12	ALIA, AUFLIA, AUFLIRA, LIA, LRA QF_AUFBV, QF_LIA, QF_LRA, QF_NIRA UF, UFIDL, UFLIA
Yices (2 versions)	11	QF_ALIA, QF_AUFLIA, QF_AX, QF_IDL QF_LIRA, QF_NRA, QF_RDL, QF_UF QF_UFIDL, QF_UFLIA, QF_UFLRA
Boolector (2 versions)	3	QF_ABV, QF_BV, QF_UFBV
AProVE	1	QF_NIA
CVC3	1	UFLRA

Results: Application Track

14 divisions but only 7 declared as competitive

Solver	# Divisions won	Divisions
Yices	6	QF_ALIA, QF_AUFLIA, QF_BF, QF_LIA QF_LRA, QF_UFLRA
CVC4	1	QF_UFLIA

Results : Competition-Wide Scoring

Main Track:

Rank	Solver	Seq. Score	Paral. Score
-	[Z3]	159.36	159.36
1	CVC4	144.67	144.74
2	CVC4 (exp)	140.47	140.51
3	Yices	101.91	101.91
-	[MathSat]	79.77	79.77
4	veriT	70.68	70.68

Other recognitions

Open Source Solvers:

- In all divisions, except QF_NIA, winners are *all open source*
- In QF_NIA, the first open source solvers is **raSAT 0.2**

Industrial performances:

- Makes no difference, except for QF_LIA and UFLRA
- **Yices2** is best performing on industrial benchs for QF_LIA
- **veriT** is best performing on industrial benchmarks for UFLRA

New Entrant:

- Two new entrants in 2015
- **SMT-RAT 2.0** obtained the best scores

Breadth of logics:

- **CVC4** covers the most theories and logics

Further Thoughts

Benchmarks:

- Still more benchmarks needed, especially for small divisions
- Resolve semantics of partial operations, e.g., `bvdiv`, `fp.min`

Solvers:

- Parallelism

Competition:

- Relative weight of benchmarks and benchmark families
- Separate measure of performance on quick jobs
- Additional tracks, e.g., unsat-core, proofs

Teams:

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