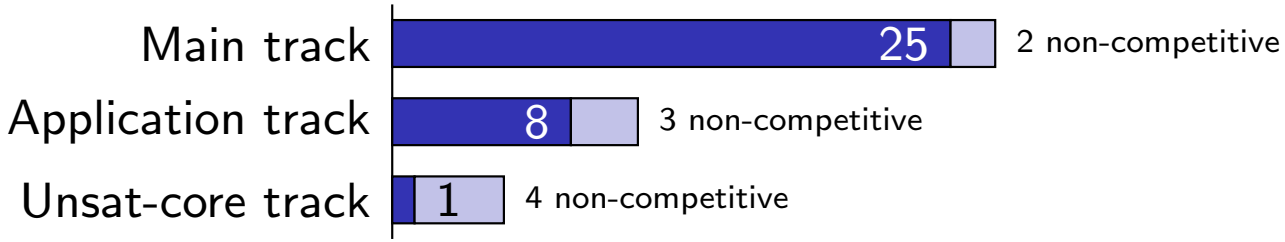
11th International Satisfiability Modulo Theories Competition

SMT-COMP 2016

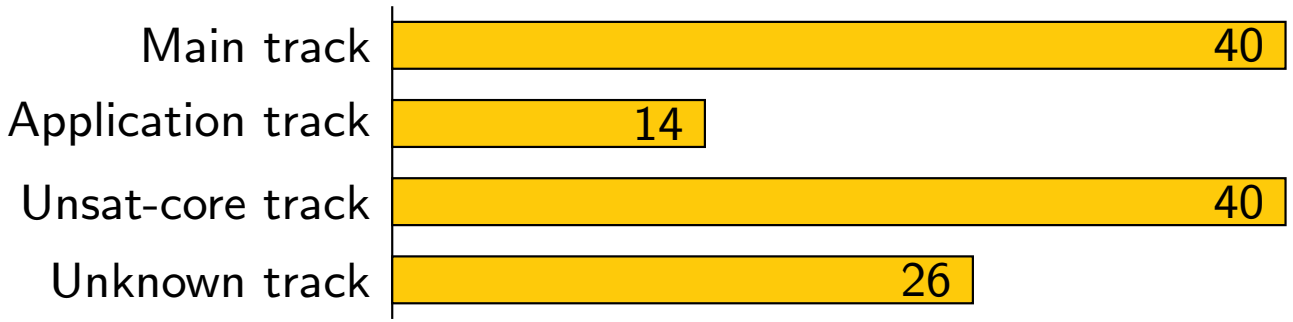
Sylvain Conchon David Déharbe Matthias Heizmann
Tjark Weber

The Numbers

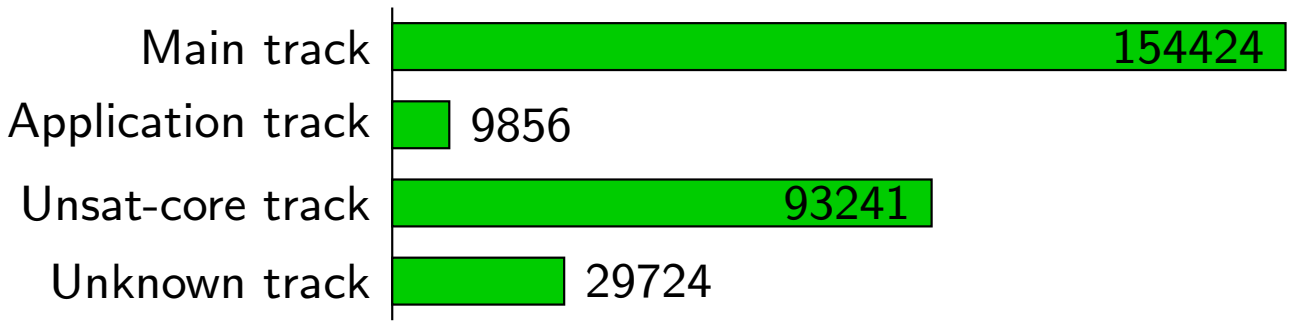
- 17 teams participated
- Solvers:



- Logics:



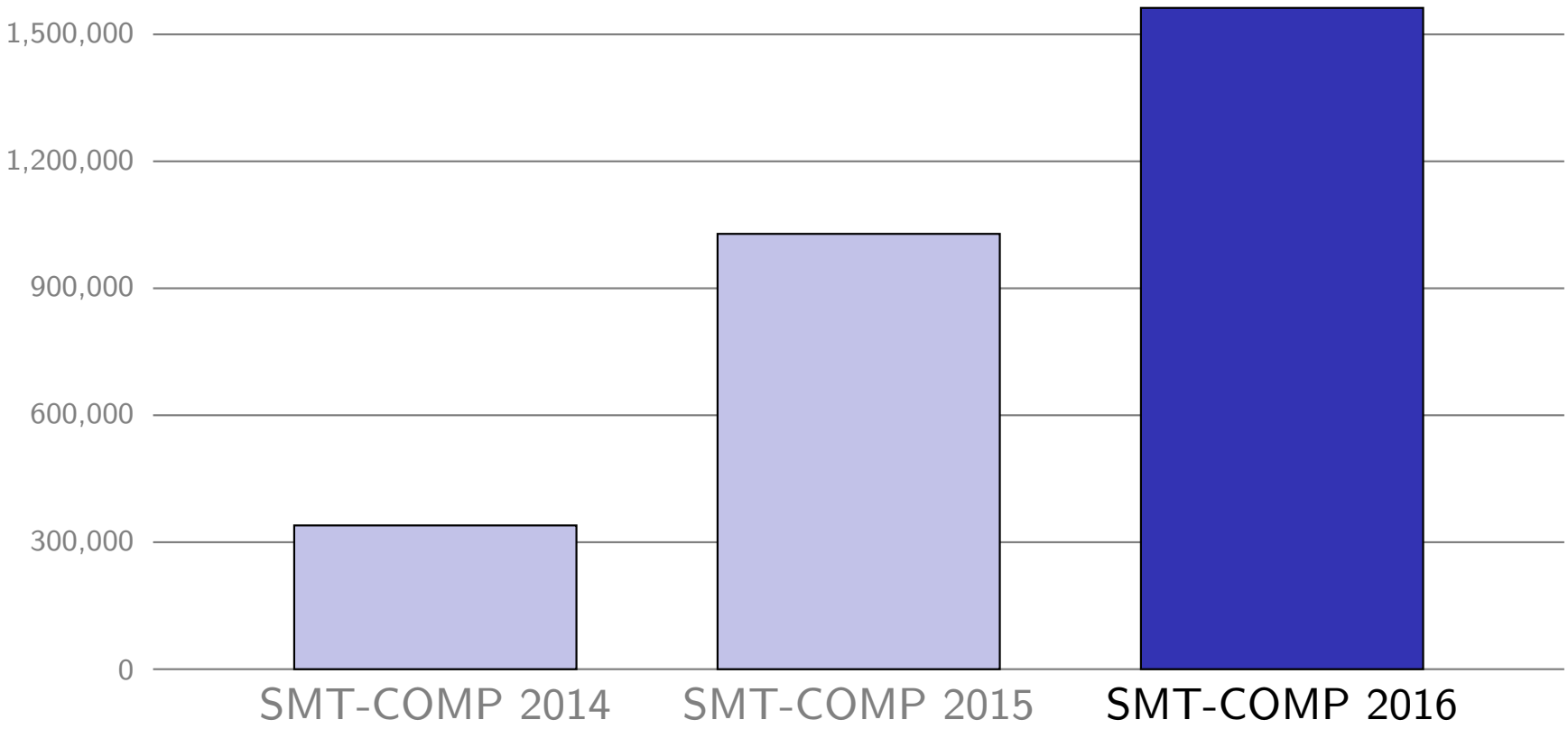
- Benchmarks:



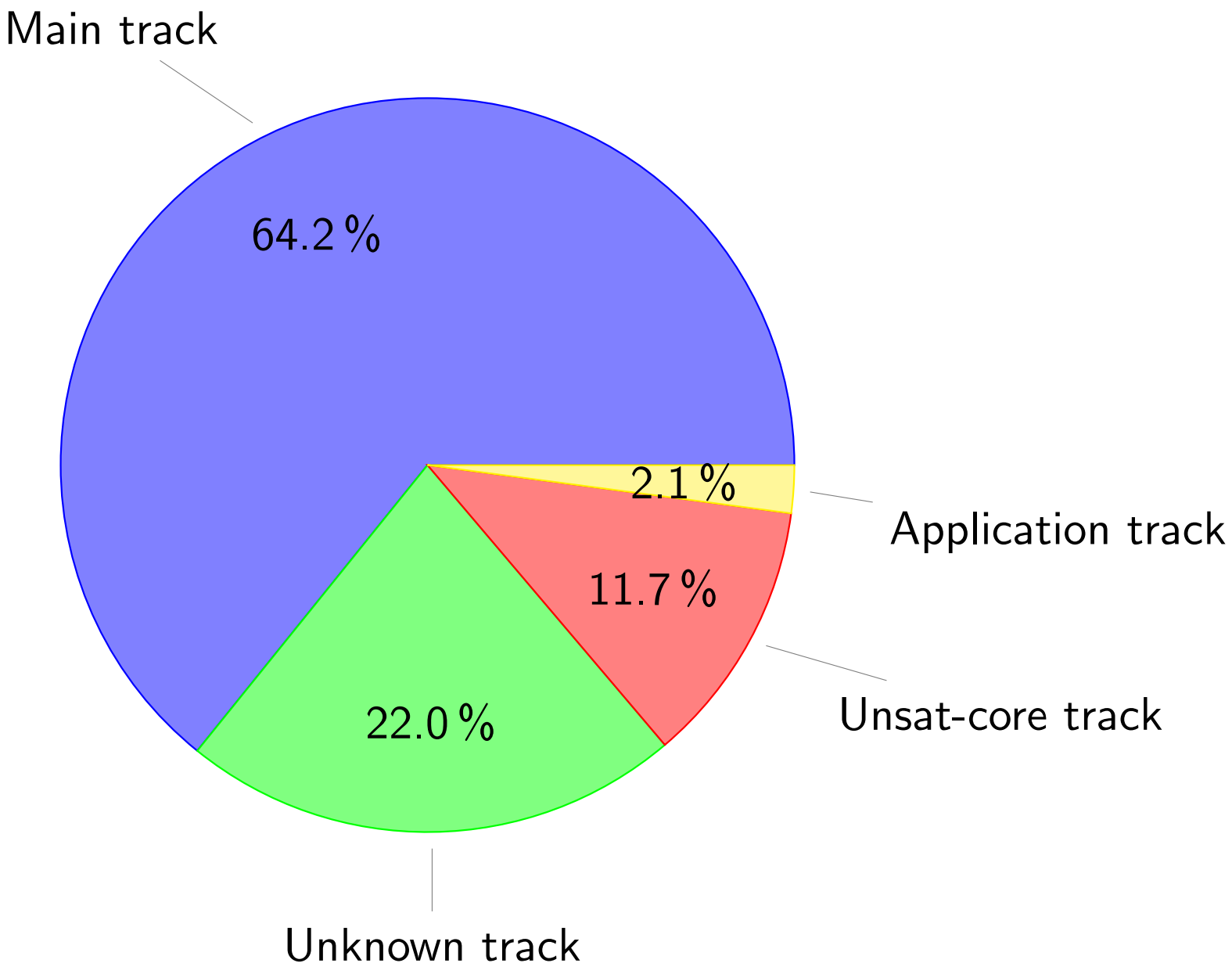
Record numbers of solvers and benchmarks!

Job Pairs

- 1,562,544 job pairs executed (+ some repeats)



Job Pairs by Track



StarExec

- All job pairs executed on StarExec
- Timeout: 40 minutes (unknown track: 10 minutes)
- $\sim$ 12 days $\times$ 100 nodes $\times$ 2 processors/node of compute time

StarExec worked even better than last year

- Thanks to Aaron Stump for prompt help when problems or questions arose
- Only very few (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

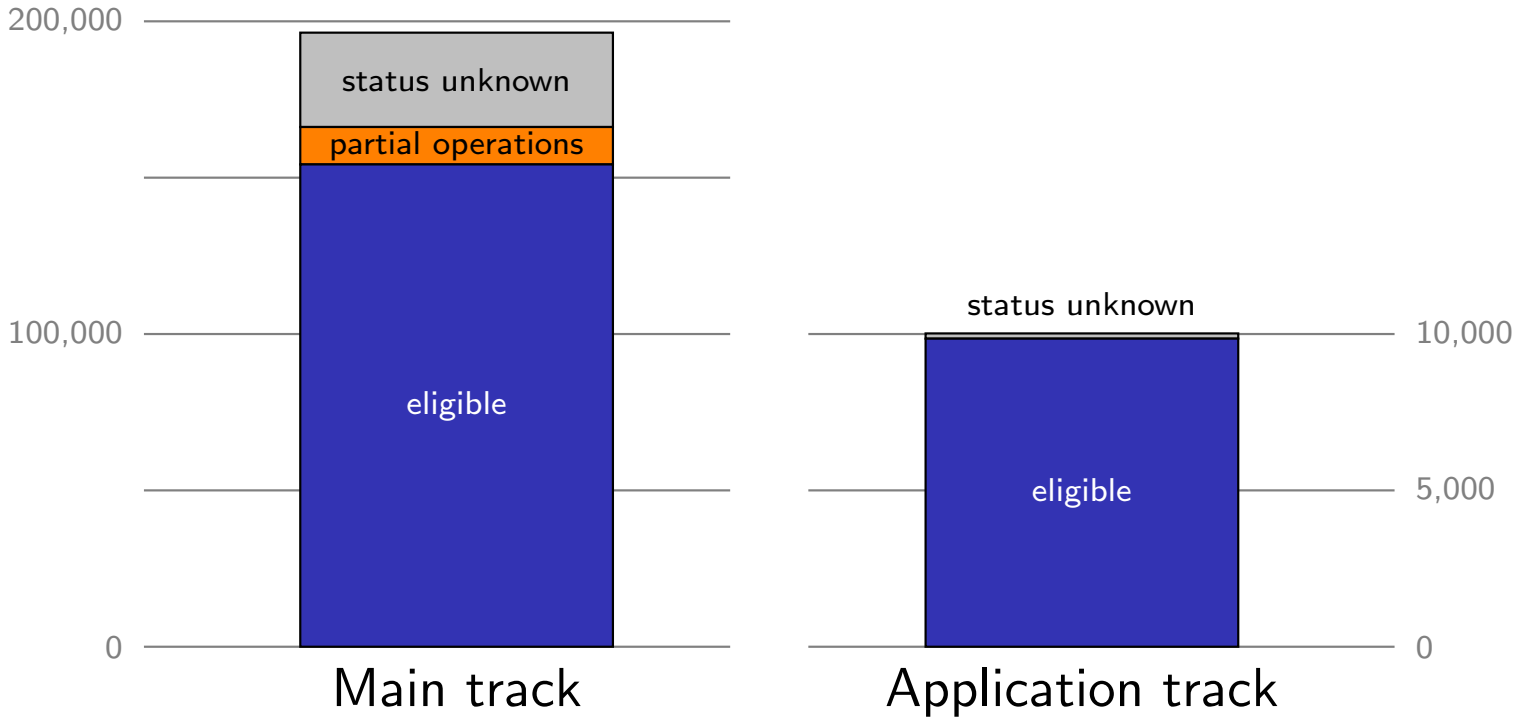
Software (upgraded in 2016):

- Red Hat Enterprise Linux Server release 7.2
- Kernel 3.10.0-327, gcc 4.8.5, glibc 2.17
- Virtual machine image available before the competition

Benchmarks and Logics

- Number of benchmarks in SMT-LIB almost unchanged since 2015
  - Very few new benchmarks
  - Some non-conforming benchmarks were removed
- No new logics
- Thanks to Clark Barrett for curation and uploading

Eligible Benchmarks



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

Important Rule Changes

- SMT-LIB 2.5 instead of 2.0
  - SMT-LIB not fully migrated yet
  - Fortunately, largely backwards-compatible
- Size-based weighting of benchmark families within divisions:

$$1 + \log_e |F|$$

  Small benchmark families are more important than before.
- Unsat-core track reinstated

Competition Tools Improved

- New unsat-core track tools (scrambler and post-processor)
- New scrambling algorithm that makes it harder to identify the original benchmark (cf. yesterday's talk)

Solvers

... primarily a (non-)termination and complexity bounds prover, but also ...

- SMT-LIB 2 front-end for QF_NIA
- use bit-blasting for binary arithmetic, back-end: MiniSat
- fixed bit-length for unknowns
- bit-length for constants, sums, products etc. **as needed**
- details on SAT encoding:
  [Fuhs, Giesl, Middeldorp, Schneider-Kamp, Thiemann, Zankl, *SAT '07*]
- back-end for proof techniques for termination and complexity bounds, search space & time-out fixed in "tactics"
- approach for SMT-COMP
  - start with small search space
  - **if** MiniSat says **satisfiable**: return with model
  - **else**: retry with larger search space until satisfiable (or out of resources)

OpenSMT2

- OpenSMT2 is an MIT-licensed SMT solver written in C++,
  Developed at Università della Svizzera Italiana, Switzerland
  - By Antti, Leo & Matteo
  - Check it out from http://verify.inf.usi.ch/opensmt
- Version 2 has been under development since 2012
  - Currently supports QF_UF and QF_LRA
  - Labeled interpolation on Boolean, QF_UF and QF_LRA with proof compression
  - Multicore and cluster/cloud based parallelization
  - Provides C and Python API through a library
  - Support for incrementality
  - Compact size (55 000 LoC)
  - Compact representation and efficient memory management for the data types
  - An object-oriented design which (hopefully) makes the development of theory support easier

raSAT – an SMT Solver for Polynomial Constraints

Vu Xuan Tung, Mizuhito Ogawa @ JAIST, To Van Khanh @ VNU-UET

- raSAT: ICP + Testing + Intermediate Value Theorem (IVT).

  Inequality *Equality*

  - ✓ICP: Interval Constraint Propagation = Interval Arithmetic + Constraint Propagation + Box Decomposition.
  - ✓Testing to boost SAT detection of inequality.
  - ✓Generalized IVT for (non-constructive) SAT detection of equality.

- Sound, but incomplete.
  - ✓Outward rounding (ICP), confirmation by iRRAM (testing)

Download: http://www.jaist.ac.jp/~s1310007/raSAT/, or google “raSAT SMT”

veriT

http://www.veriT-solver.org

Haniel Barbosa, David Déharbe
and Pascal Fontaine

Loria, INRIA, Université de Lorraine (France), ClearSy and UFRN (Brazil)

What is new:

- cleaning, efficiency improvements, e.g. UF (space for improvement)
- (much) improved quantifier handling
- Other w.i.p.: (N|L)RA (Redlog), quantifier handling, proofs

Goals:

- clean, small SMT for UF(N|L)IRA with quantifiers and proofs
- for verification platforms B, TLA+

Selected Results

Results: QF_BV (Main Track)

Solver	Error Score	Solved Score (Parallel)	Unsolved
Boolector (pre)	**0.000**	**24473.995**	**149**
Boolector	0.000	24468.395	150
Minkeyrink	0.000	24434.194	193
smt-cms-mt	0.000	24244.599	216
smt-cms-st	0.000	24165.007	214
CVC4	0.000	23820.707	231
Z3	0.000	23732.215	304
smt-cms-exp	0.000	23640.669	270
ABC_glucose	0.000	23078.931	477
Yices2	0.000	22687.777	638
MathSat5	0.000	22496.779	544
MapleSTP-mt	0.000	22487.264	395
MapleSTP	0.000	21764.885	450
smt-minisat-st	0.000	20582.614	1058
ABC_default	0.000	18528.788	1354
Q3B	719.723	10397.757	4430

Results: Competition-Wide Scoring (Main Track)

Rank	Solver	Score (sequential)	Score (parallel)
	Z3	185.09	185.09
1	CVC4	180.95	181.19
2	Yices	119.29	119.29
3	veriT	75.11	75.11

Best newcomer:

5	Vampire_parallel	65.36	65.62

Results: Application Track (Summary)

Logic	Order
ANIA	Z3; CVC4
QF_ANIA	Z3; CVC4
QF_ALIA	Z3; SMTInterpol; Yices2; MathSat5; CVC4
QF_UFNIA	Z3; CVC4
LIA	Z3; CVC4
ALIA	Z3; CVC4
QF_UFLRA	Z3; Yices2; SMTInterpol; CVC4; MathSat5
UFLRA	Z3; CVC4
QF_UFLIA	Z3; CVC4; Yices2; SMTInterpol; MathSat5
QF_NIA	CVC4; Z3
QF_BV	MathSat5; Yices2; smt-cms-st; smt-cms-mt; smt-cms-exp; CVC4; MapleSTP; MapleSTP-mt; smt-minisat-st; Z3
QF_LRA	MathSat5; SMTInterpol; Z3; Yices2; CVC4
QF_LIA	Yices2; Z3; SMTInterpol; MathSat5; CVC4
QF_AUFLIA	Yices2; Z3; SMTInterpol; MathSat5; CVC4

Selected Results: Unsat-Core Track

Solver	Errors	Reductions
SMTInterpol	0	1166535
toysmt	0	35886
veriT	26	68811
MathSat5	190	1527159
Z3	17079	4597883

- 182,367 job pairs
- In total, 83,450 (45.8%) unsat cores generated
- . . . but also 17,097 (9.4%) wrong sat answers
- Each unsat core was checked with three solvers (CVC4, MathSat5 and Z3). 198 cores (2.4%) were found satisfiable by at least one solver.

Selected Results: Unknown Track

Most benchmarks solved:

Solver	Benchmarks solved	Benchmarks attempted
Yices2	18593	20473
Minkeyring	16724	17504
CVC4	16646	29509

In total, 21,542 benchmarks (72.5%) were solved.

However, disagreements on 79 benchmarks!

Further Thoughts

Benchmarks:

- Still more benchmarks needed, especially for small divisions
- Resolve semantics of partial operations, e.g., `bvdiv`, `fp.min`
- Benchmark curation deserves better tool support

Competition:

- Benchmark weights—good or bad?
- Integration of benchmarks with unknown status?
- Trophies? (T-shirts? Dinner? Funding?!)

Teams:

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