12th International Satisfiability Modulo Theories Competition

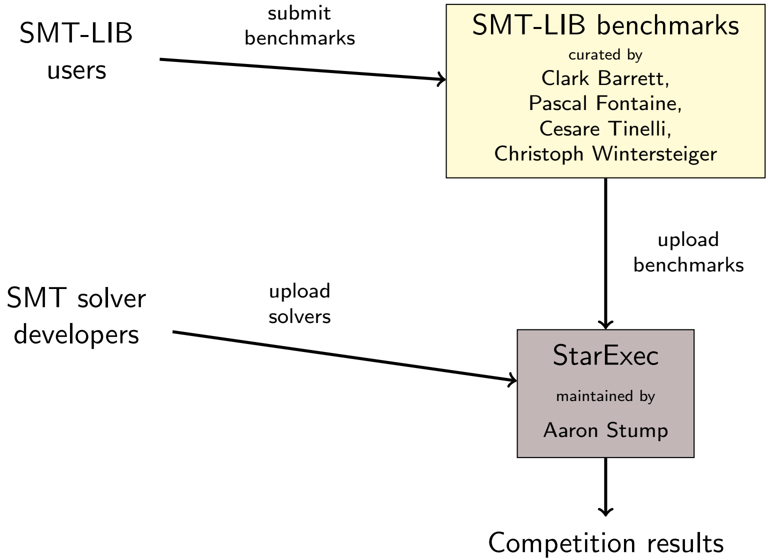
SMT-COMP 2017

Matthias Heizmann (co-organizer)
Giles Reger (co-organizer)
Tjark Weber (chair)

Outline

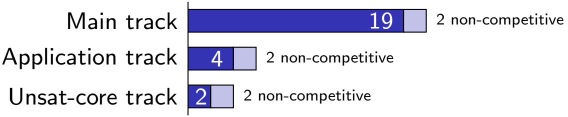
- Main changes over last competition
  - Benchmarks with 'unknown' status
  - Logics with algebraic data-types
    AUFBVDTLIA, AUFDTLIA, QF_DT, UFDT, UFDTLIA
  - Unsat-core Track
- Statistics and selected results of competition
- Short presentation of solvers
  Boolector, COLIBRI, CVC4, SMTInterpol, veriT, Yices

SMT-COMP – Procedure

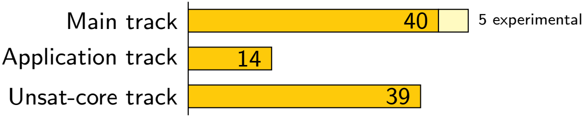


Solvers, Logics, and Benchmarks

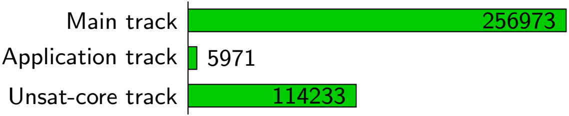
- 15 teams participated
- Solvers:



- Logics:



- Benchmarks:



StarExec

Cluster of machines 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

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`

in any order

$\leftarrow$ one single `check-sat` command

Benchmarks with 'unknown' status

Some benchmarks in SMT-LIB repository do not have a sat/unsat status.

Benchmarks with 'unknown' status in SMT-COMP

not used in competition

2015

2016 separate experimental track

2017 included in Main Track

New logics

Algebraic data-types

- defined in SMT-LIB 2.6 draft
- “experimental” this year (i.e., no winner determined)

	benchmarks	solvers
AUFBVDTLIA	1709	CVC4
AUFDTLIA	728	CVC4, vampire
QF_DT	8000	CVC4
UFDT	4535	CVC4, vampire
UFDTLIA	303	vampire, CVC4

Benchmarks with 'unknown' status

Rules

- we trust the results of the solver(s)
- in case of disagreement we trust solvers that are sound on benchmarks with known status
- if there is disagreement between otherwise sound solvers, we exclude the benchmark

Benchmarks with 'unknown' status

Rules

- we trust the results of the solver(s)
- in case of disagreement we trust solvers that are sound on benchmarks with known status
- if there is disagreement between otherwise sound solvers, we exclude the benchmark

Outcome

- There were 29 benchmarks with unknown status on which solvers disagreed on the result.
- On one benchmark (in BV) the corresponding solvers were sound on all benchmarks with known status.
- On 28 benchmarks (all in QF_FP) the presumably wrong answers were given by unsound solvers.

Competition run of Main Track

- run all job pairs with 10 min timeout
- made preliminary results available
- rerun all job pairs that timed out with 20 min timeout
- made final results available on Friday (21st June)

Main Track – Selected results – QF_ABV

Benchmarks in this division : 15061

Winners:

Sequential Performances	Parallel Performances
Boolector	Boolector

Result table[1]

Solver	Sequential performance			Parallel performance			
	Error Score	Correctly Solved Score	CPU time Score	Errors	Correct Score	CPU Score	WALL Score
Boolector	0.000	14627.624	56.054	0.000	14627.624	56.054	56.101
CVC4	0.000	13136.950	155.681	0.000	13136.950	155.681	162.762
Yices2	0.000	14515.846	65.395	0.000	14515.846	65.395	65.207
mathsat-5.4.1[n]	0.000	13098.157	155.650	0.000	13098.157	155.651	155.727
z3-4.5.0[n]	0.000	13115.900	171.596	0.000	13115.900	171.597	171.665

n. Non-competing.

`http://smtcomp.sourceforge.net/2017/results-QF_ABV.shtml`

Main Track: Competition-Wide Scoring

Rank	Solver	Score (sequential)	Score (parallel)
	Z3	171.99	171.99
1	CVC4	161.38	161.76
2	Yices2	110.63	110.63
3	SMTInterpol	65.96	66.00

Application Track

Logic	Solvers	Benchmarks	Order (parallel performance)
ANIA	2	3	CVC4; z3-4.5.0[n]
QF_ANIA	2	5	z3-4.5.0[n]; CVC4
QF_ALIA	5	44	z3-4.5.0[n]; SMTInterpol; Yices2; mathsat-5.4.1[n]; CVC4
QF_UFNIA	2	1	z3-4.5.0[n]; CVC4
QF_BVFP	1	2	z3-4.5.0[n]
LIA	2	6	z3-4.5.0[n]; CVC4
ALIA	2	24	z3-4.5.0[n]; CVC4
QF_UFLRA	5	3056	Yices2; z3-4.5.0[n]; SMTInterpol; CVC4; mathsat-5.4.1[n]
UFLRA	2	1870	z3-4.5.0[n]; CVC4
QF_UFLIA	5	780	z3-4.5.0[n]; CVC4; Yices2; SMTInterpol; mathsat-5.4.1[n]
QF_NIA	2	10	CVC4; z3-4.5.0[n]
QF_FP	1	2	z3-4.5.0[n]
QF_BV	4	18	mathsat-5.4.1[n]; Yices2; CVC4; z3-4.5.0[n]
QF_LRA	6	10	mathsat-5.4.1[n]; SMTInterpol; Yices2; z3-4.5.0[n]; CVC4; opensmt2
QF_LIA	5	68	Yices2; z3-4.5.0[n]; SMTInterpol; mathsat-5.4.1[n]; CVC4
QF_AUFLIA	5	72	Yices2; z3-4.5.0[n]; SMTInterpol; CVC4; mathsat-5.4.1[n]

Unsat-core Track

Motivation

- Important application of SMT-LIB
- One step towards verifiable proofs

History

2012 introduced

discontinued

2016 reinstated as experimental track

2017 "regular" track

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)
```

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

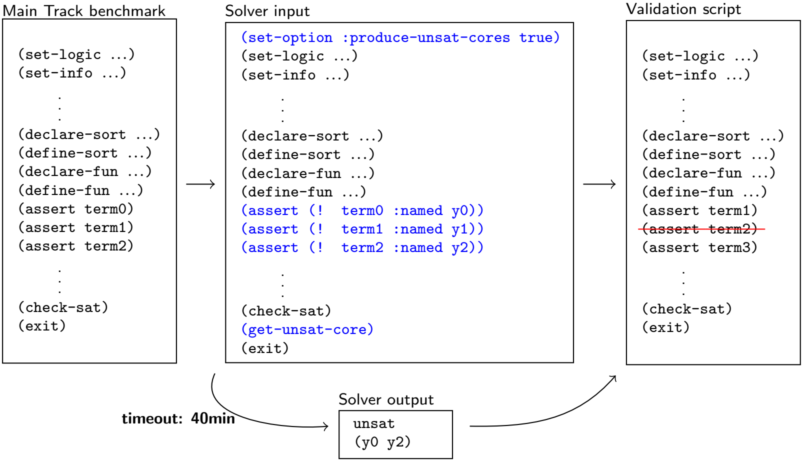
```
(set-option :produce-unsat-cores true)
(set-logic ...)
(set-info ...)
  .
  .
  .
(declare-sort ...)
(define-sort ...)
(declare-fun ...)
(define-fun ...)
(assert (! term0 :named y0))
(assert (! term1 :named y1))
(assert (! term2 :named y2))
  .
  .
  .
(check-sat)
(get-unsat-core)
(exit)
```

timeout: 40min

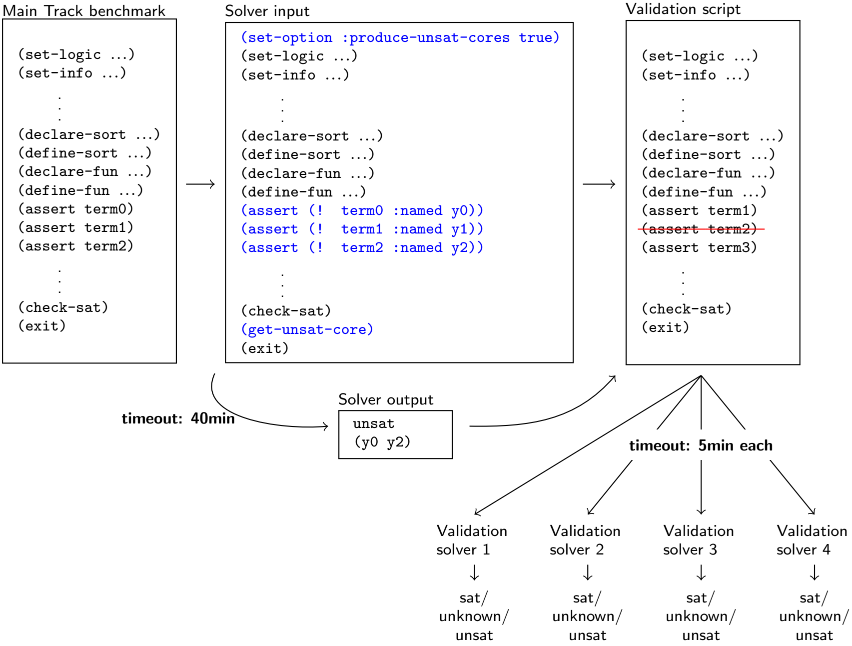
Solver output

```
unsat
(y0 y2)
```

Unsat-core Track



Unsat-core Track



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)
```

→

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)
```

→

Validation script

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

Scoring scheme

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

$$e = \begin{cases} 1 & \text{result erroneous} \\ 0 & \text{otherwise} \end{cases}$$

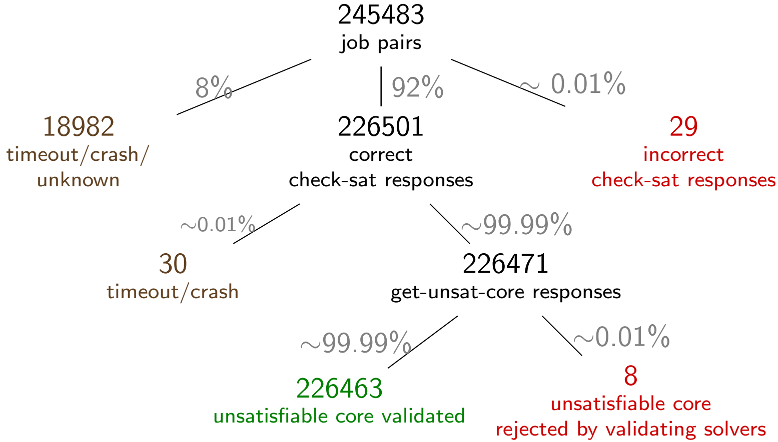
result erroneous if

- ▷ wrong check-sat result or
- ▷ unsat-core rejected by validating solvers

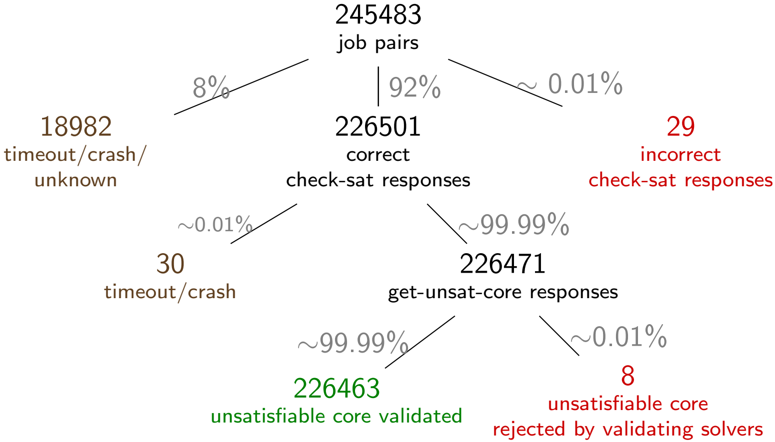
ch

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

Unsat-core Track – Statistics



Unsat-core Track – Statistics



- 19 times there was no consensus among the validating solvers ($\rightsquigarrow$ majority decision)
- 27525 (∼12%) times no independent validating solver approved the correctness of the unsatisfiable core

(Very) short presentations of

Solvers

that sent us slides.

Boolector, COLIBRI, CVC4, SMTInterpol, veriT, Yices

Boolector at the SMTCOMP'17

Aina Niemetz, Mathias Preiner, Armin Biere

Divisions

BV QF_BV QF_UFBV QF_ABV QF_AUFBV

Changes since 2016 (QF_BV)

- combination of prop.-based local search + bit-blasting now default
- experimental configuration with new SAT solver CaDiCaL

Major Improvements

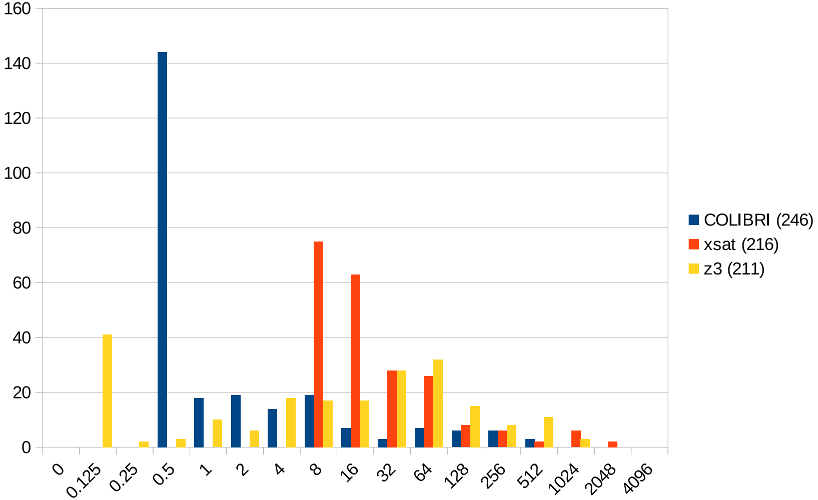
- **BV engine at SMT-COMP 2016** was a prototype implementation

  $\longrightarrow$ **rewrite** of BV engine with major improvements

COLIBRI (QF_FP, QF_BVFP)

- 776 errors due to a wrong backward propagation for `fp.fma`
- 1 error due to a mix between $-0.$ and $+0.$

On the 308, non Wintersteiger benchmark of QF_FP in sec.



CVC4 1.5

Clark Barrett (Stanford), Martin Brain (Oxford), Guy Katz (Stanford), Tim King (Google), Paul Meng (U Iowa), Aina Niemetz (Stanford), Mathias Preiner (Stanford), Andres Nötzli (Stanford), Andrew Reynolds (U Iowa), Cesare Tinelli (U Iowa)

SMT 2017, July 22, 2017

CVC4 1.5: Recent Developments

- A new theory of sets with cardinality and relations.
- A new theory of strings.
- A new theory of separation logic constraints.
- Support for many new heuristics for reasoning with quantifiers, including finite model finding.
- Improved heuristics for reasoning about non-linear arithmetic.
- Support for proofs for uninterpreted functions, arrays, bitvectors, and their combinations.
- Support for unsat cores.
- Native support for syntax-guided synthesis (sygus).

Contact

We aim for CVC4 to be a versatile research platform for SMT and are open to collaborators and contributors.

For more information:

- Contact one of the project leaders:
  - Clark Barrett `barrett@cs.stanford.edu`
  - Cesare Tinelli `cesare-tinelli@uiowa.edu`
- Visit the website: `cvc4.stanford.edu`

Quantifier Free
Linear Arithmetic

$y \leq i + 1$

$i \leq y$

Quantifier Free
Arrays

$b = a\langle i \lhd v\rangle$

$a[v] = v$

Quantifier Free
Uninterpreted Functions

$f(b) = v$

$f(a) \neq v$

SMTInterpol

decides **S**atisfiability **M**odulo **T**heory
computes Craig **Interpol**ants

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

Quantifier Free
Linear Arithmetic

$y \leq i + 1$

$i \leq y$

Quantifier Free
Arrays

$b = a\langle i \lhd v \rangle$

$a[v] = v$

Quantifier Free
Uninterpreted Functions

$f(b) = v$

$f(a) \neq v$

SMTInterpol

decides **S**atisfiability **M**odulo **T**heory
computes Craig **Interpol**ants

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

Quantifier Free
Linear Arithmetic

$y \leq i + 1$

$i \leq y$

$y - to\_int(y) < .3$

mixed int/real

Quantifier Free
Arrays

$b = a\langle i \lhd v\rangle$

$a[v] = v$

Quantifier Free
Uninterpreted Functions

$f(b) = v$

$f(a) \neq v$

SMTInterpol

decides **S**atisfiability **M**odulo **T**heory
computes Craig **Interpol**ants

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

Quantifier Free Arrays

Linear Arithmetic

Uninterpreted Functions

$b = a\langle i \lhd v \rangle$

$y \leq i + 1$

$a[v] = v$

$f(b) = v$

$i \leq y$

$f(a) \neq v$

$y - to\_int(y) < .3$

$b[i] \geq i$

$f(i + y) = 2v$

SMTInterpol

decides **S**atisfiability **M**odulo **T**heory
computes Craig **Interpol**ants

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

Quantifier Free Arrays

Linear Arithmetic

Uninterpreted Functions

$b = a\langle i \lhd v \rangle$

$y \leq i + 1$

$a[v] = v$

$f(b) = v$

$i \leq y$

$f(a) \neq v$

$y - to\_int(y) < .3$

$b[i] \geq i$

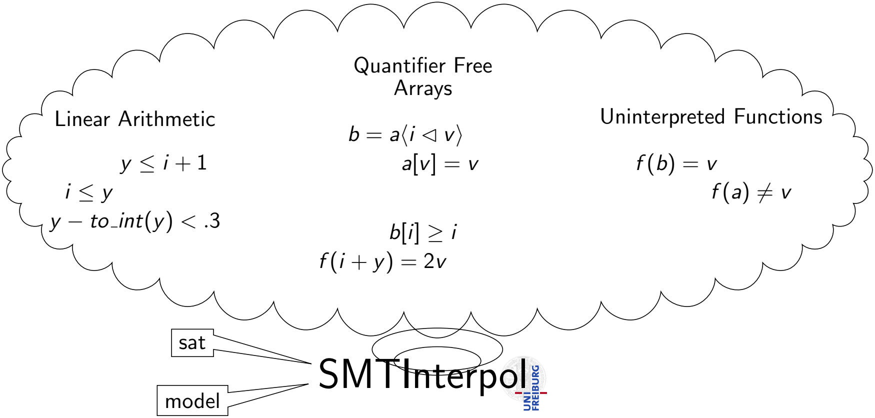
$f(i + y) = 2v$

sat

SMTInterpol

decides **S**atisfiability **M**odulo **T**heory
computes Craig **Interpol**ants

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



decides **S**atisfiability **M**odulo **T**heory
computes Craig **Interpol**ants

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

Quantifier Free Arrays

Linear Arithmetic

Uninterpreted Functions

$b = a\langle i \lhd v\rangle$

$a[v] = v$

$y \leq i + 1$

$i \leq y$

$y - to\_int(y) < .3$

$f(b) = v$

$f(a) \neq v$

$b[i] \geq i$

$f(i + y) = 2v$

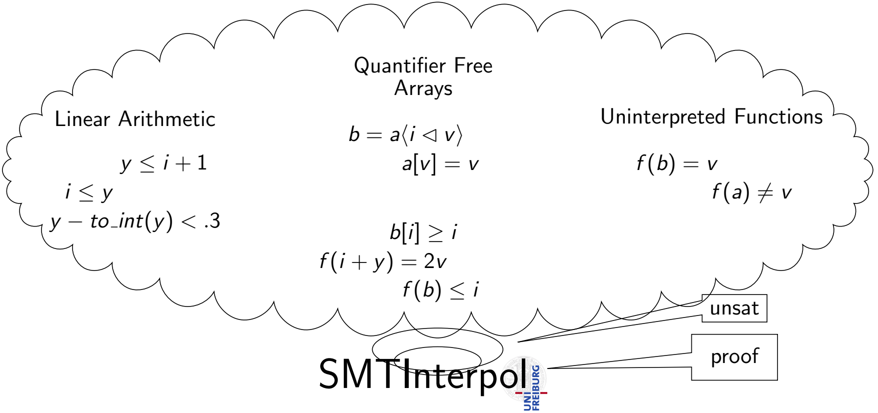
$f(b) \leq i$

unsat

SMTInterpol

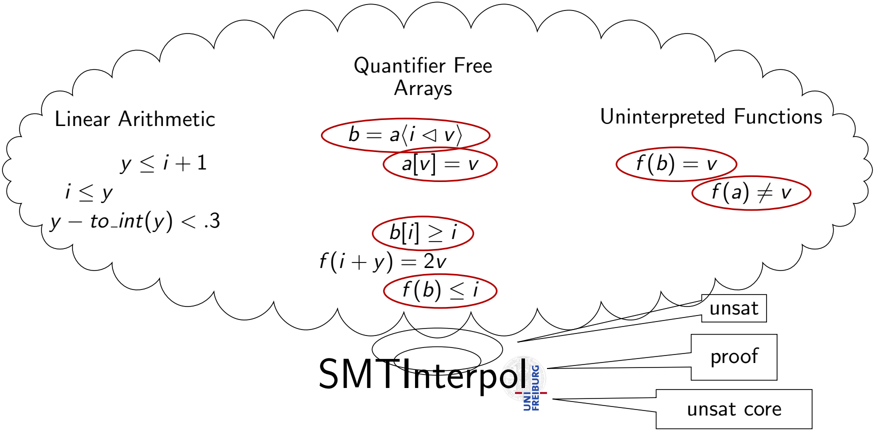
decides **S**atisfiability **M**odulo **T**heory
computes Craig **Interpol**ants

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



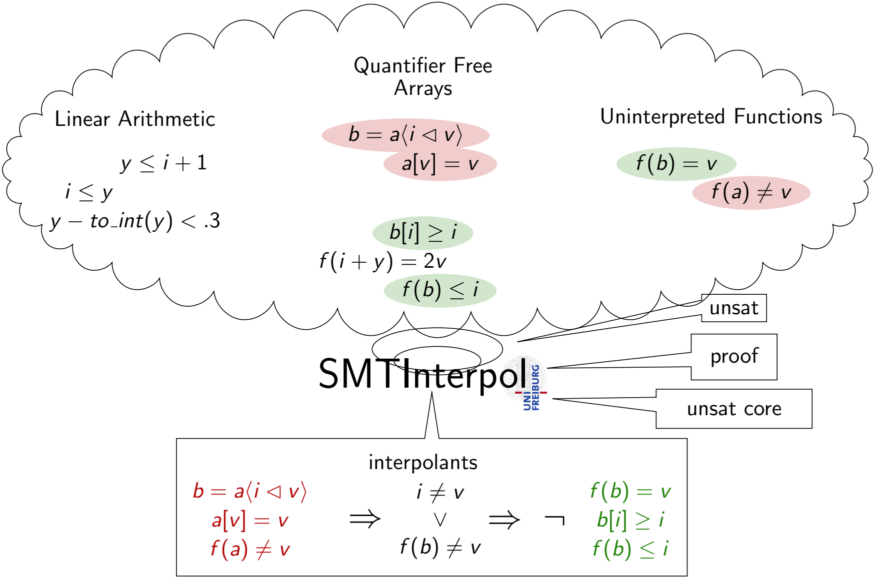
decides **S**atisfiability **M**odulo **T**heory
computes Craig **Interpol**ants

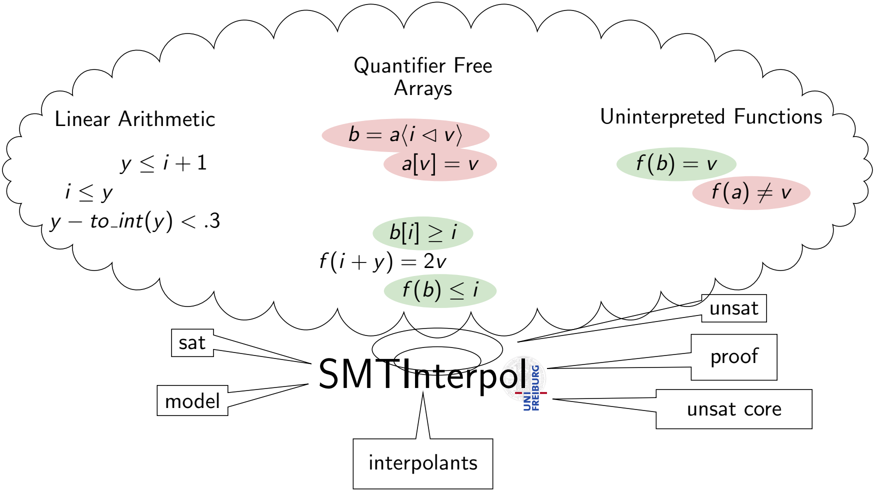
`http://ultimate.informatik.uni-freiburg.de/smtinterpol`



decides **S**atisfiability **M**odulo **T**heory
computes Craig **Interpol**ants

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


Quantifier Free Arrays
Linear Arithmetic
Uninterpreted Functions
$b = a\langle i \lhd v \rangle$
$a[v] = v$
$y \leq i + 1$
$i \leq y$
$y - to\_int(y) < .3$
$f(b) = v$
$f(a) \neq v$
$b[i] \geq i$
$f(i + y) = 2v$
$f(b) \leq i$
unsat
proof
unsat core
SMTInterpol
UNI FREIBURG
interpolants
$b = a\langle i \lhd v \rangle$
$a[v] = v$
$f(a) \neq v$
$\Rightarrow$
$i \neq v \lor f(b) \neq v$
$\Rightarrow \neg$
$f(b) = v$
$b[i] \geq i$
$f(b) \leq i$



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

veriT

http://www.veriT-solver.org

Haniel Barbosa, Pascal Fontaine, Maximilian Jaroschek, Marek Kosta, Thomas Sturm, and Vu Xuan Tung

University of Lorraine, CNRS, Inria, and LORIA (France), MPI Informatics and Saarland University (Germany), and JAIST (Japan)

What is new:

- few improvements for quantifier handling
- fine-grained proofs for formula processing
- veriT+raSAT+Redlog for supporting QF_[UF]NRA
- veriT+Redlog for better handling (N|L)RA

Goals:

- clean, small SMT for UF(N|L)IRA with quantifiers and proofs
- for verification platforms (e.g. B, TLA+) and proof assistants (e.g. Isabelle, Coq)

Yices 2.6

Bruno Dutertre and Dejan Jovanović
SRI International

SMTCOMP 2017
Heidelberg, Germany

Yices 2.6 in SMTCOMP 2017

Status

- No major change since last year
- Supports linear and non-linear arithmetic, arrays, UF, bitvectors
- Limited quantifier reasoning: $\exists\forall$ fragments for bitvector, LRA
- Includes two types of solvers: classic DPPL(T) + MC-SAT

Entered in all the divisions that Yices supports

- Main track: Quantifier-free logics including linear and nonlinear arithmetic, bitvectors, and combination with UF and Arrays.
- Application track: Same logics, except that the MC-SAT solver is not incremental yet.

What's New

Yices2 GPLv3 Free Software GitHub

debian ubuntu

New Licence

- Yices 2 is now GPL

Distributions

- Prebuilt binaries + source tarfile at `http://yices.csl.sri.com`
- Git repository on Github `https://github.com/SRI-CSL/yices2`
- Ubuntu/Debian package
- Homebrew package for MacOS X

What's Next

MC-SAT Extensions

- Add support for incremental solving
- Extends MC-SAT to bitvector problems

CDCL Solver

- New sat solver in progress

Miscellaneous

- Complete support for SMT-LIB 2.6
- Fix some API issues

Teams:

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