Theory-Specific Proof Steps Witnessing Correctness of SMT Executions

Rodrigo Otoni[1], Martin Blicha[1,2], Patrick Eugster[1,3,4],
Antti E. J. Hyvärinen[1], and Natasha Sharygina[1]

[1] Università della Svizzera italiana, Lugano, Switzerland
[2] Charles University, Prague, Czech Republic
[3] Technische Universität Darmstadt, Darmstadt, Germany
[4] Purdue University, West Lafayette, USA

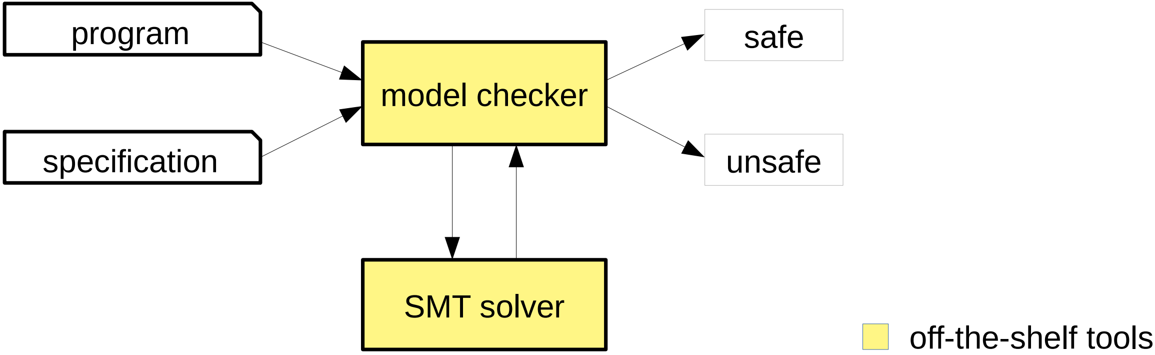
18th July 2021

State of the Art in Verification

- Software verification often relies on SMT solvers

State of the Art in Verification

- Software verification often relies on SMT solvers

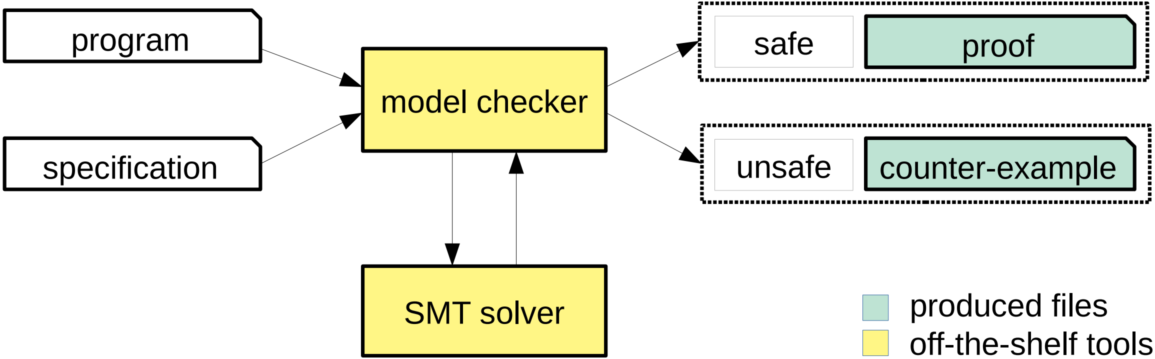


State of the Art in Verification

- Despite use in verification, SMT solvers have bugs
- At SMT-COMP'20, solvers disagreed on 149 instances with unknown status
- Guarantees are needed to trust solvers' results

State of the Art in Verification

- Despite use in verification, SMT solvers have bugs
- At SMT-COMP'20, solvers disagreed on 149 instances with unknown status
- Guarantees are needed to trust solvers' results



program

specification

model checker

SMT solver

safe

proof

unsafe

counter-example

🟩 produced files

🟨 off-the-shelf tools

State of the Art in SMT Proofs

Proofs of unsatisfiability

- The *de facto* standard for SAT proofs is the DRAT proof format
- No standard for SMT proofs, with many competing formats available

State of the Art in SMT Proofs

Proofs of unsatisfiability

- The *de facto* standard for SAT proofs is the DRAT proof format
- No standard for SMT proofs, with many competing formats available

Existing solver support

- Proofs validating SMT solvers' executions vary by implementation
  - Proof-producing solvers include CVC4, veriT, and Z3
- A commonality is the focus on integration with interactive theorem provers
  - SMT proofs can be consumed by Isabelle/HOL and Coq
- Current proof formats unsuitable to automation
  - Integration to automated tools such as model checkers is not available

This Work

Lightweight witnesses validating SMT solvers' executions

- Witnesses checkable by simple automated checkers
- Checkers simple enough to support manual inspection
- A user can write a checker in a few hours/days

This Work

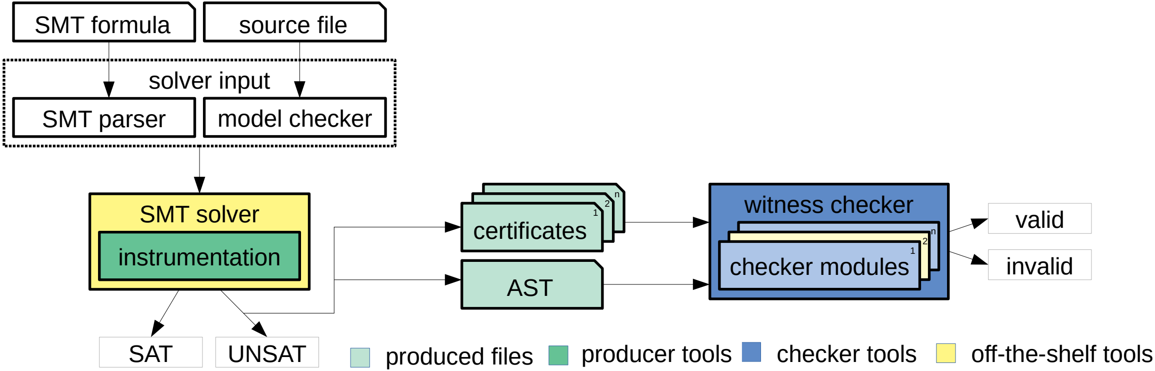
Lightweight witnesses validating SMT solvers' executions

- Witnesses checkable by simple automated checkers
- Checkers simple enough to support manual inspection
- A user can write a checker in a few hours/days

Validation based on the foundations of SMT algorithms

- DRAT proofs for propositional logic
- Theory-specific witnesses for each supported SMT theory

Certification Framework



SMT formula

source file

solver input

SMT parser

model checker

SMT solver

instrumentation

SAT

UNSAT

certificates

AST

witness checker

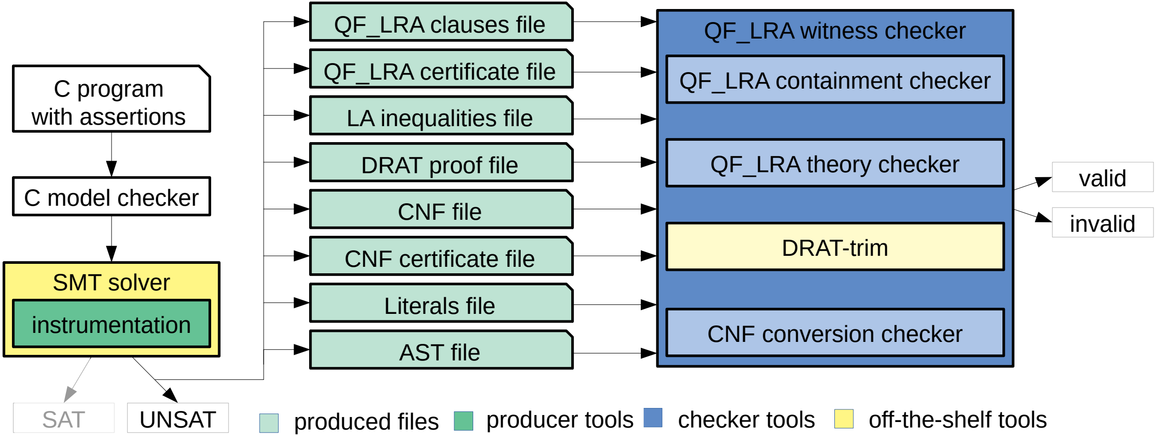
checker modules

valid

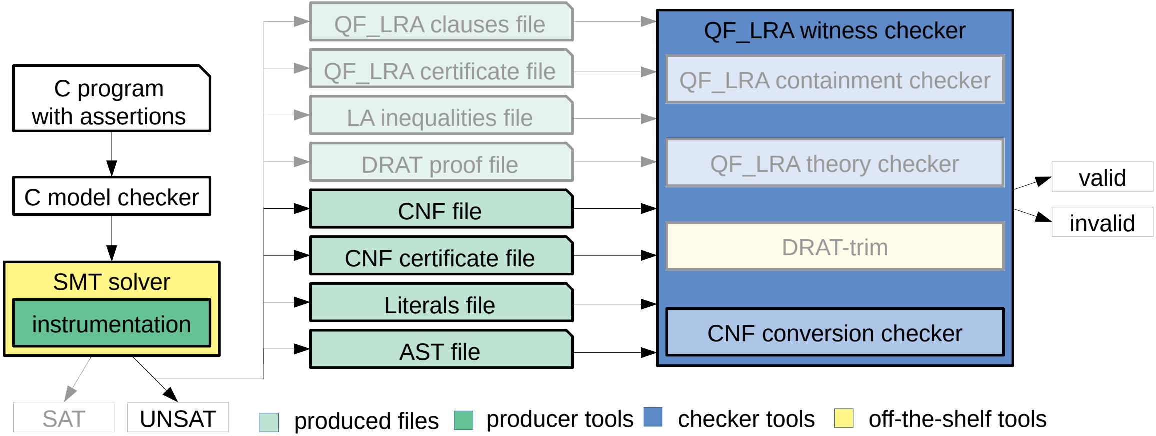
invalid

🟩 produced files 🟩 producer tools 🟦 checker tools 🟨 off-the-shelf tools

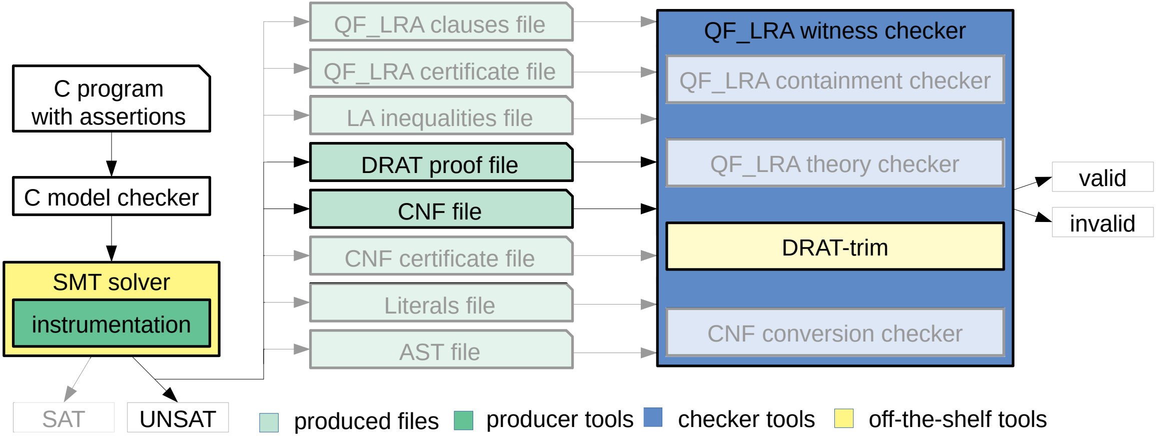
Example for QF_LRA



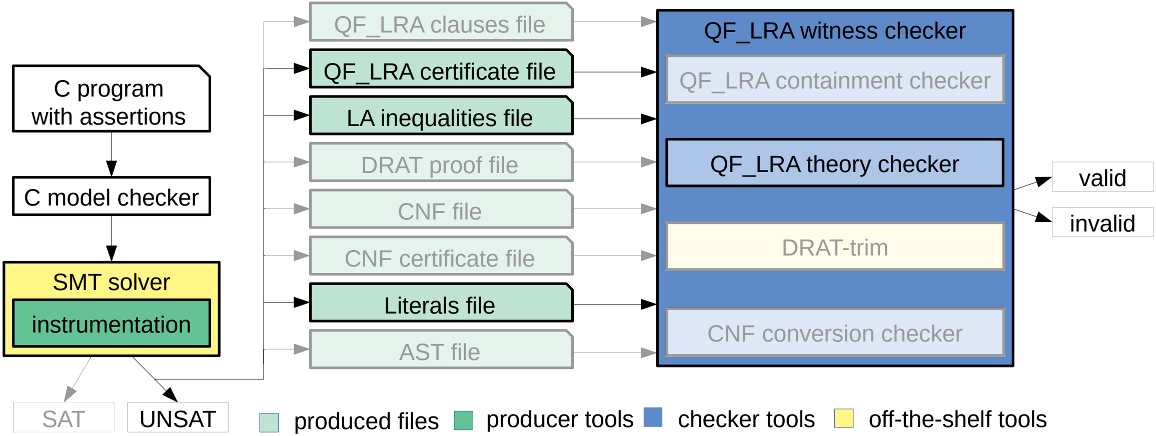
Example for QF_LRA



Example for QF_LRA



Example for QF_LRA



C program with assertions

C model checker

SMT solver

instrumentation

SAT

UNSAT

QF_LRA clauses file

QF_LRA certificate file

LA inequalities file

DRAT proof file

CNF file

CNF certificate file

Literals file

AST file

QF_LRA witness checker

QF_LRA containment checker

QF_LRA theory checker

DRAT-trim

CNF conversion checker

valid

invalid

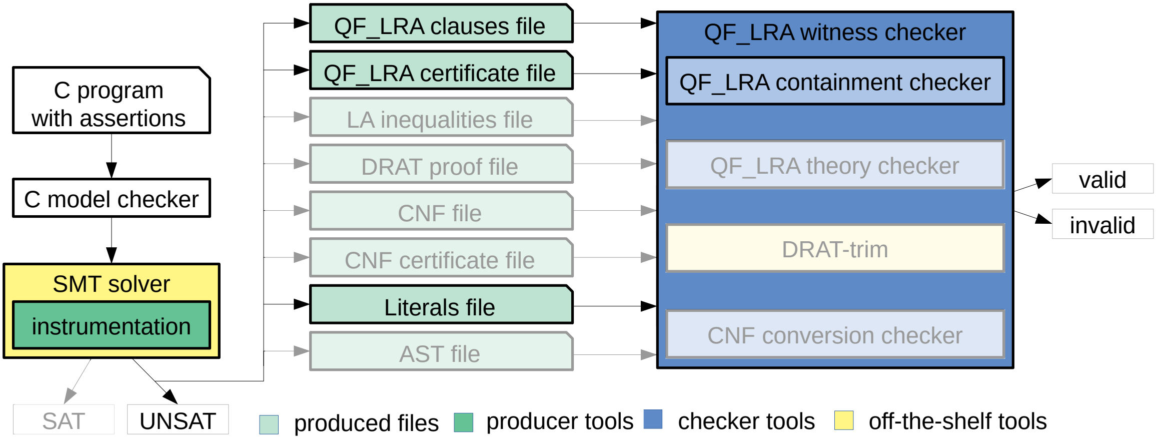
produced files

producer tools

checker tools

off-the-shelf tools

Example for QF_LRA



Example for QF_LRA - Certificate File

```
float x = 1;
if (condition) {
    x = x - 1;
} else {
    x = x + 1;
}
assert (x ≥ 0);
```

$$\begin{aligned} x1 = 1 \land \\ (condition \rightarrow x4 = x2) \land x2 = x1 - 1 \land \\ (\neg condition \rightarrow x4 = x3) \land x3 = x1 + 1 \land \\ \neg(x4 \geq 0) \end{aligned}$$

Example for QF_LRA - Certificate File

```
float x = 1;
if (condition) {
    x = x − 1;
} else {
    x = x + 1;
}
assert(x ≥ 0);
```

$$
\begin{aligned}
& x1 = 1 \wedge \\
& (condition \rightarrow x4 = x2) \wedge x2 = x1 - 1 \wedge \\
& (\neg condition \rightarrow x4 = x3) \wedge x3 = x1 + 1 \wedge \\
& \neg(x4 \geq 0) \wedge \\
& (x4 \geq 0 \vee x4 < 2)
\end{aligned}
$$

Learned clause

$x4 \geq 0 \vee x4 < 2$

Example for QF_LRA - Certificate File

```
float x = 1;
if (condition) {
    x = x − 1;
} else {
    x = x + 1;
}
assert (x ≥ 0);
```

$$\begin{aligned} x1 = 1 \land \\ (condition \rightarrow x4 = x2) \land x2 = x1 - 1 \land \\ (\neg condition \rightarrow x4 = x3) \land x3 = x1 + 1 \land \\ \neg(x4 \geq 0) \land \\ (x4 \geq 0 \lor x4 < 2) \end{aligned}$$

Learned clause

$x4 \geq 0 \lor x4 < 2$

Witness

$x4 < 0 \land x4 \geq 2$

Example for QF_LRA - Certificate File

```
float x = 1;
if (condition) {
      x = x − 1;
} else {
      x = x + 1;
}
assert (x ≥ 0);
```

$$\begin{aligned} x1 = 1 \land \\ (condition \rightarrow x4 = x2) \land x2 = x1 - 1 \land \\ (\neg condition \rightarrow x4 = x3) \land x3 = x1 + 1 \land \\ \neg(x4 \geq 0) \land \\ (x4 \geq 0 \lor x4 < 2) \end{aligned}$$

Learned clause

$x4 \geq 0 \lor x4 < 2$

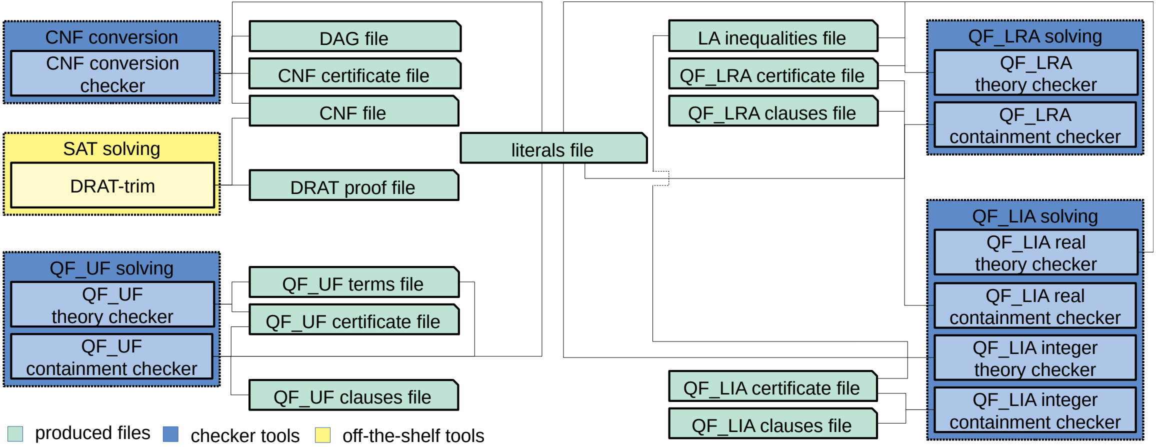
Witness

$x4 < 0 \land x4 \geq 2$

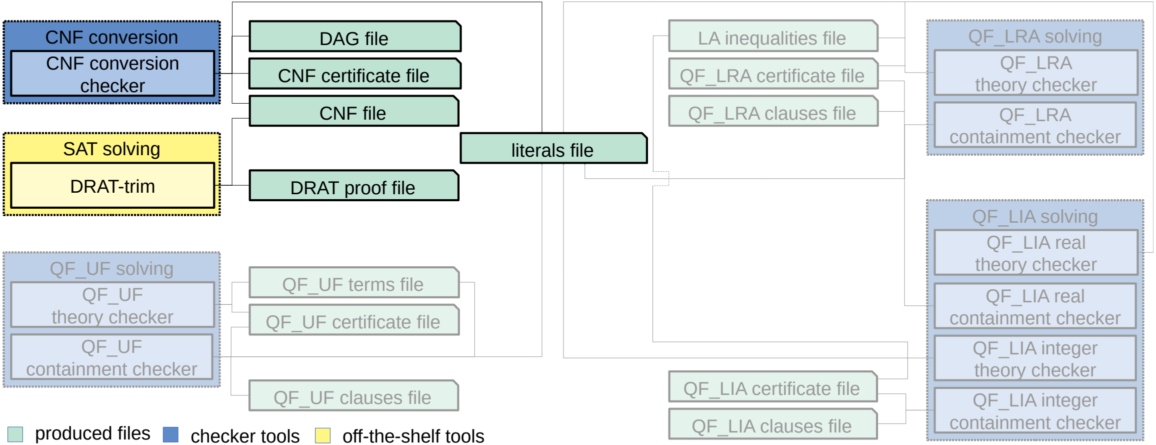
Validation

$$\begin{array}{cccc} (1x) & -x4 & > & 0 \\ (1x) & x4 & \geq & 2 \\ \hline & 0 & > & 2 \end{array}$$

Checking Architecture

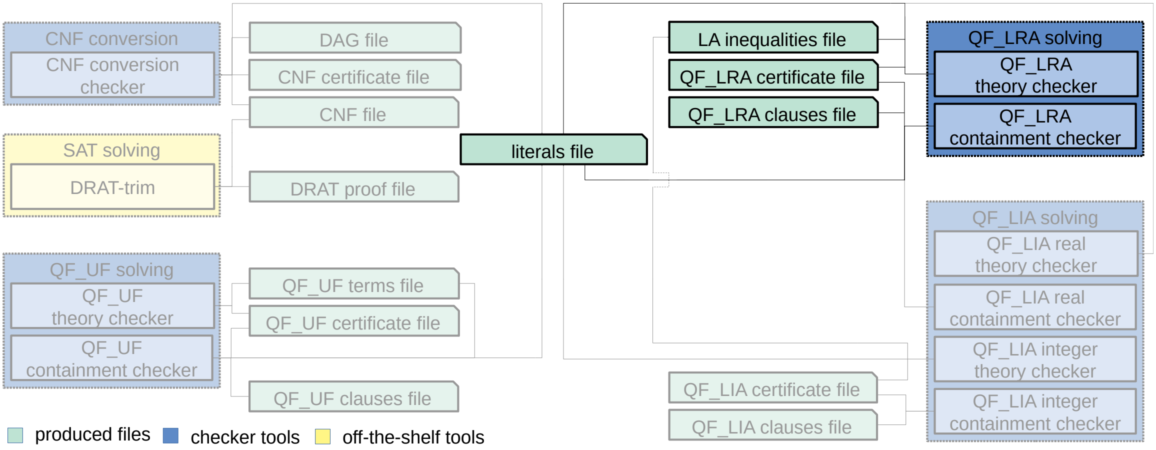


Checking Architecture



🟩 produced files 🟦 checker tools 🟨 off-the-shelf tools

Checking Architecture



CNF conversion
CNF conversion checker

SAT solving
DRAT-trim

QF_UF solving
QF_UF theory checker
QF_UF containment checker

DAG file
CNF certificate file
CNF file
DRAT proof file

QF_UF terms file
QF_UF certificate file
QF_UF clauses file

literals file

LA inequalities file
QF_LRA certificate file
QF_LRA clauses file

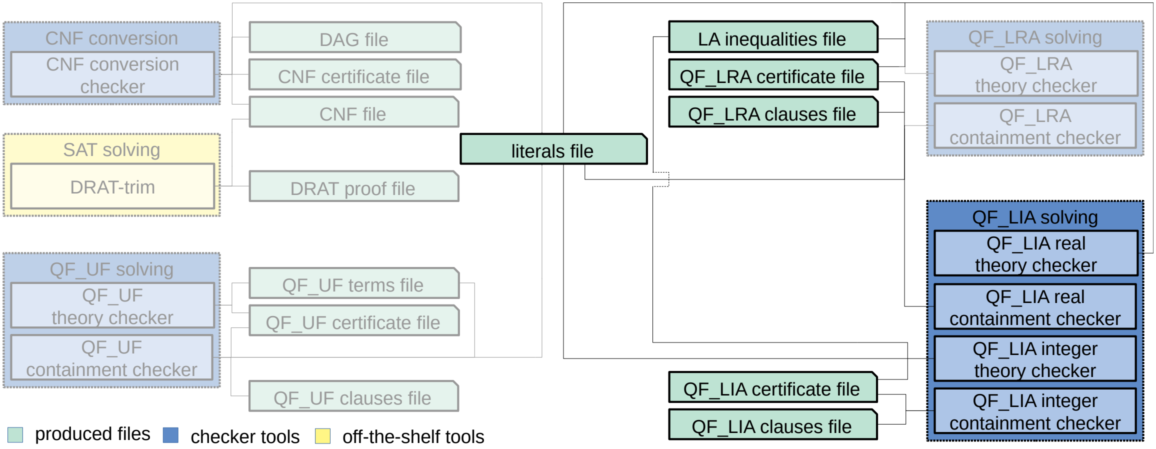
QF_LRA solving
QF_LRA theory checker
QF_LRA containment checker

QF_LIA solving
QF_LIA real theory checker
QF_LIA real containment checker
QF_LIA integer theory checker
QF_LIA integer containment checker

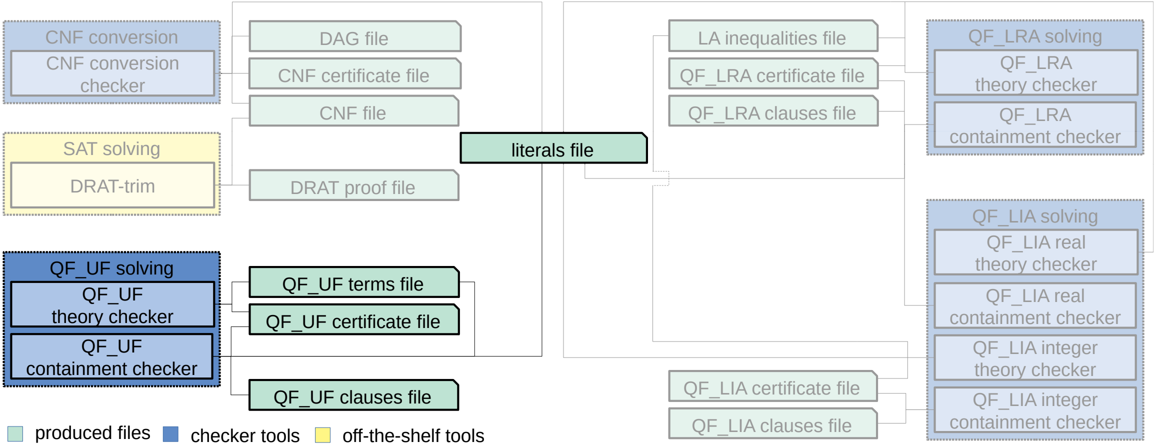
QF_LIA certificate file
QF_LIA clauses file

🟩 produced files 🟦 checker tools 🟨 off-the-shelf tools

Checking Architecture



Checking Architecture



CNF conversion

CNF conversion checker

SAT solving

DRAT-trim

QF_UF solving

QF_UF theory checker

QF_UF containment checker

DAG file

CNF certificate file

CNF file

DRAT proof file

QF_UF terms file

QF_UF certificate file

QF_UF clauses file

literals file

LA inequalities file

QF_LRA certificate file

QF_LRA clauses file

QF_LIA certificate file

QF_LIA clauses file

QF_LRA solving

QF_LRA theory checker

QF_LRA containment checker

QF_LIA solving

QF_LIA real theory checker

QF_LIA real containment checker

QF_LIA integer theory checker

QF_LIA integer containment checker

🟩 produced files 🟦 checker tools 🟨 off-the-shelf tools

Evaluation

Our tools

- Witness production implemented in the OpenSMT solver
- New Theory-Specific Witness Checker (TSWC)

Evaluation

Our tools

- Witness production implemented in the OpenSMT solver
- New Theory-Specific Witness Checker (TSWC)

Evaluation setup

- All non-incremental SMT-LIB instances for the QF_LRA, QF_LIA, and QF_UF theories
- Witness-production comparison against CVC4, veriT, Z3
- Witness-checking comparison against CVC4's LFSC checker

Witness Production Results

		UNSAT retention (%)	Avg. runtime (s)	Avg. witness size (MB)
QF_LRA (1648 instances)	OpenSMT	**99.5%**	**3.91**/**4.31**	3.9
	CVC4	97.7%	5.73/6.61	13.6
	veriT	96.7%	4.12/5.22	69.4
	Z3	97.8%	5.17/5.30	**3.5**
QF_LIA (6947 instances)	OpenSMT	75%	10.07/9.99	**13.0**
	CVC4	66.3%	6.30/**1.30**	18.8
	veriT	95.1%	**1.82**/4.05	165.9
	Z3	**104.5%**	5.17/6.62	24.8
QF_UF (7457 instances)	OpenSMT	**99.6%**	0.95/1.14	**6.7**
	CVC4	97.6%	0.39/1.83	6.8
	veriT	96.0%	**0.10**/**0.79**	20.1
	Z3	97.5%	0.22/1.26	12.6

Witness Production Results

		UNSAT retention (%)	Avg. runtime (s)	Avg. witness size (MB)
QF_LRA (1648 instances)	OpenSMT	**99.5%**	**3.91**/**4.31**	3.9
	CVC4	97.7%	5.73/6.61	13.6
	veriT	96.7%	4.12/5.22	69.4
	Z3	97.8%	5.17/5.30	**3.5**
QF_LIA (6947 instances)	OpenSMT	75%	10.07/9.99	**13.0**
	CVC4	66.3%	6.30/**1.30**	18.8
	veriT	95.1%	**1.82**/4.05	165.9
	Z3	**104.5%**	5.17/6.62	24.8
QF_UF (7457 instances)	OpenSMT	**99.6%**	0.95/1.14	**6.7**
	CVC4	97.6%	0.39/1.83	6.8
	veriT	96.0%	**0.10**/**0.79**	20.1
	Z3	97.5%	0.22/1.26	12.6

Witness Production Results

		UNSAT retention (%)	Avg. runtime (s)	Avg. witness size (MB)
QF_LRA (1648 instances)	OpenSMT	**99.5%**	**3.91**/**4.31**	3.9
	CVC4	97.7%	5.73/6.61	13.6
	veriT	96.7%	4.12/5.22	69.4
	Z3	97.8%	5.17/5.30	**3.5**
QF_LIA (6947 instances)	OpenSMT	75%	10.07/9.99	**13.0**
	CVC4	66.3%	6.30/**1.30**	18.8
	veriT	95.1%	**1.82**/4.05	165.9
	Z3	**104.5%**	5.17/6.62	24.8
QF_UF (7457 instances)	OpenSMT	**99.6%**	0.95/1.14	**6.7**
	CVC4	97.6%	0.39/1.83	6.8
	veriT	96.0%	**0.10**/**0.79**	20.1
	Z3	97.5%	0.22/1.26	12.6

Witness Production Results

		UNSAT retention (%)	Avg. runtime (s)	Avg. witness size (MB)
QF_LRA (1648 instances)	OpenSMT	**99.5%**	**3.91**/**4.31**	3.9
	CVC4	97.7%	5.73/6.61	13.6
	veriT	96.7%	4.12/5.22	69.4
	Z3	97.8%	5.17/5.30	**3.5**
QF_LIA (6947 instances)	OpenSMT	75%	10.07/9.99	**13.0**
	CVC4	66.3%	6.30/**1.30**	18.8
	veriT	95.1%	**1.82**/4.05	165.9
	Z3	**104.5%**	5.17/6.62	24.8
QF_UF (7457 instances)	OpenSMT	**99.6%**	0.95/1.14	**6.7**
	CVC4	97.6%	0.39/1.83	6.8
	veriT	96.0%	**0.10**/**0.79**	20.1
	Z3	97.5%	0.22/1.26	12.6

Witness Checking Results

		Verified	Timeout	Error	Avg. runtime (s)
QF_LRA (567 instances)	OpenSMT + TSWC	**564**	**3**	**0**	3.15
	CVC4 + LFSC	471	8	88	**2.85**
QF_LIA (913 instances)	OpenSMT + TSWC	**903**	10	**0**	1.37
	CVC4 + LFSC	128	**0**	785	**0.12**
QF_UF (4218 instances)	OpenSMT + TSWC	**4217**	**1**	**0**	**1.44**
	CVC4 + LFSC	4157	50	11	4.01

Witness Checking Results

		Verified	Timeout	Error	Avg. runtime (s)
QF_LRA (567 instances)	OpenSMT + TSWC	**564**	**3**	**0**	3.15
	CVC4 + LFSC	471	8	88	**2.85**
QF_LIA (913 instances)	OpenSMT + TSWC	**903**	10	**0**	1.37
	CVC4 + LFSC	128	**0**	785	**0.12**
QF_UF (4218 instances)	OpenSMT + TSWC	**4217**	**1**	**0**	**1.44**
	CVC4 + LFSC	4157	50	11	4.01

Witness Checking Results

		Verified	Timeout	Error	Avg. runtime (s)
QF_LRA (567 instances)	OpenSMT + TSWC	**564**	**3**	**0**	3.15
	CVC4 + LFSC	471	8	88	**2.85**
QF_LIA (913 instances)	OpenSMT + TSWC	**903**	10	**0**	1.37
	CVC4 + LFSC	128	**0**	785	**0.12**
QF_UF (4218 instances)	OpenSMT + TSWC	**4217**	**1**	**0**	**1.44**
	CVC4 + LFSC	4157	50	11	4.01

Witness Checking Results

		Verified	Timeout	Error	Avg. runtime (s)
QF_LRA (567 instances)	OpenSMT + TSWC	**564**	**3**	**0**	3.15
	CVC4 + LFSC	471	8	88	**2.85**
QF_LIA (913 instances)	OpenSMT + TSWC	**903**	10	**0**	1.37
	CVC4 + LFSC	128	**0**	785	**0.12**
QF_UF (4218 instances)	OpenSMT + TSWC	**4217**	**1**	**0**	**1.44**
	CVC4 + LFSC	4157	50	11	4.01

Witness Checking Results

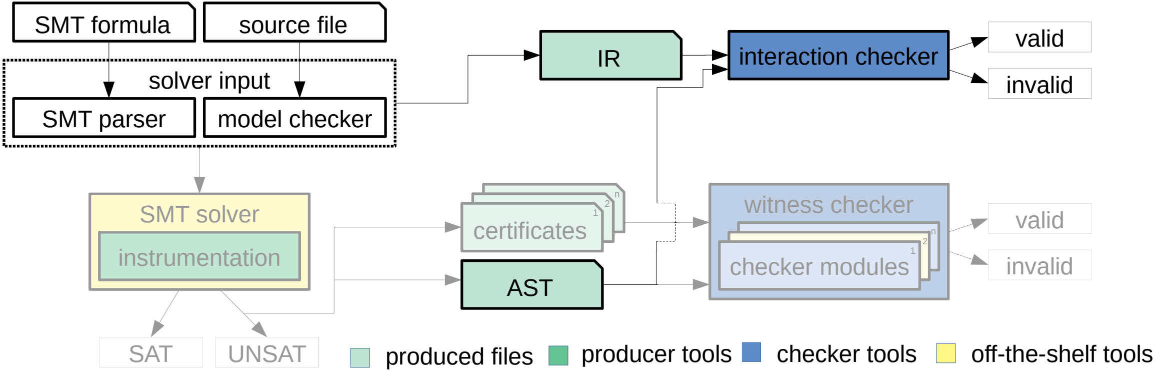
		Verified	Timeout	Error	Avg. runtime (s)
QF_LRA (567 instances)	OpenSMT + TSWC	**564**	**3**	**0**	3.15
	CVC4 + LFSC	471	8	88	**2.85**
QF_LIA (913 instances)	OpenSMT + TSWC	**903**	10	**0**	1.37
	CVC4 + LFSC	128	**0**	785	**0.12**
QF_UF (4218 instances)	OpenSMT + TSWC	**4217**	**1**	**0**	**1.44**
	CVC4 + LFSC	4157	50	11	4.01

Future Work

- Witnesses for additional theories and theory combinations
- Witnesses for parallel SMT solving
- Certify interaction between solvers and verification tools

Future Work

- Witnesses for additional theories and theory combinations
- Witnesses for parallel SMT solving
- Certify interaction between solvers and verification tools



Summary

- Lightweight witnesses that validate unsatisfiable results of SMT solvers
- Implementation of production and checking for three SMT theories
- Evaluation results indicate the compactness and easy checking of our format

Summary

- Lightweight witnesses that validate unsatisfiable results of SMT solvers
- Implementation of production and checking for three SMT theories
- Evaluation results indicate the compactness and easy checking of our format

- Details about this work can be found on our DAC'21 paper
- Check our group's website for future updates

`verify.inf.usi.ch/research/cevt`