Computer Algebra and Computer Science

It's complicated...

Gereon Kremer

June 21st, 2018

ACA'18 – Santiago de Compostela

RWTH AACHEN UNIVERSITY

Context: SC^2

EU H2020-FETOPEN-2015-CSA 712689

Satisfiability Checking and Symbolic Computation

EU project to stimulate cooperations
More than 50 partners and associates

Industry: Altran, BTC, ClearSy, Imandra, L4B, Maplesoft, Microsoft, MJC2, NAG, SRI, Systerel, Wolfram

Also present at ACA'18:
Anna Bigatti, Francisco Botana, James Davenport, Vijay Ganesh, Martin Kreuzer, Antonio Montes, Lorenzo Robbiano, Werner Seiler

Computer Science: SMT solving

Satisfiability Modulo Theories (SMT)

Is an existentially quantified first-order formula φ satisfiable?

$$\exists x.\varphi(x) \equiv true$$

Computer Science: SMT solving

Satisfiability Modulo Theories (SMT)

Is an existentially quantified first-order formula φ satisfiable?

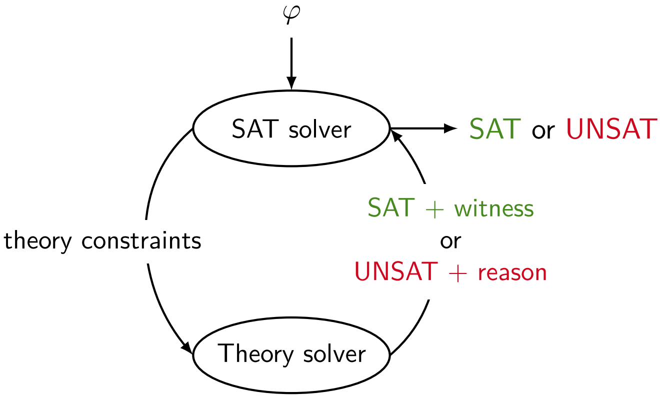
$$\exists x.\varphi(x) \equiv true$$

Applications:

- Software verification, test-case generation
- Termination proving
- Controller synthesis
- Scheduling and planning
- Product design automation
- And growing ...

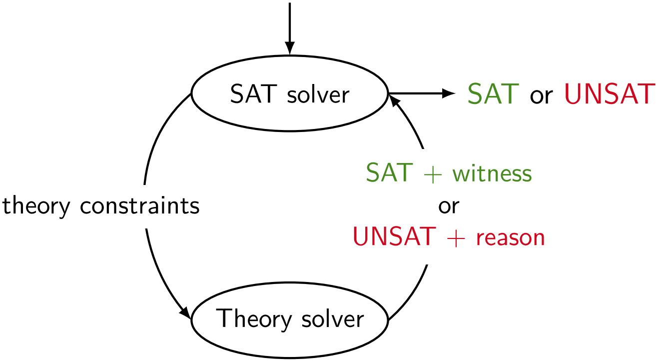
RWTH AACHEN UNIVERSITY

SMT solving



SMT solving

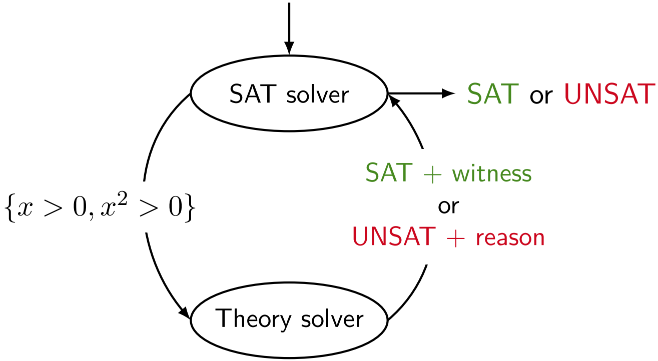
$$x > 0 \wedge (x^2 > 0 \vee x < 0) \wedge (x^3 < 0 \vee x = 3)$$



RWTH AACHEN UNIVERSITY

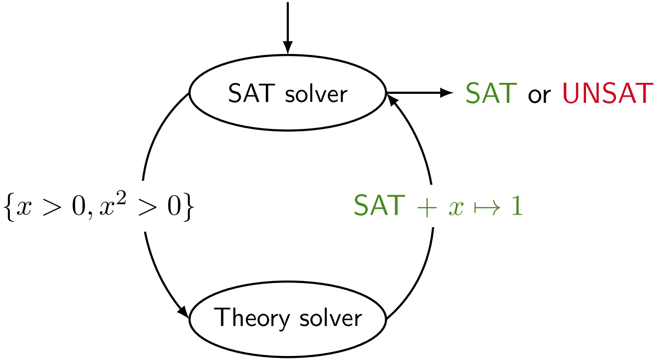
SMT solving

$$x > 0 \wedge (x^2 > 0 \vee x < 0) \wedge (x^3 < 0 \vee x = 3)$$



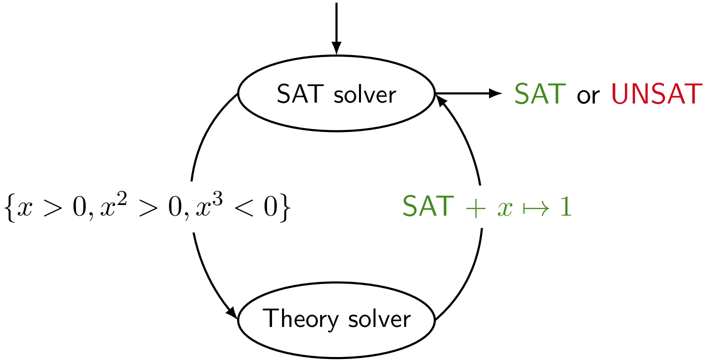
SMT solving

$$x > 0 \land (x^2 > 0 \lor x < 0) \land (x^3 < 0 \lor x = 3)$$

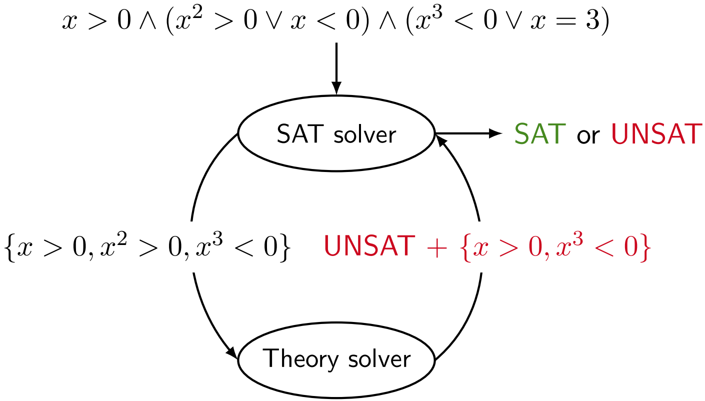


SMT solving

$$x > 0 \land (x^2 > 0 \lor x < 0) \land (x^3 < 0 \lor x = 3)$$

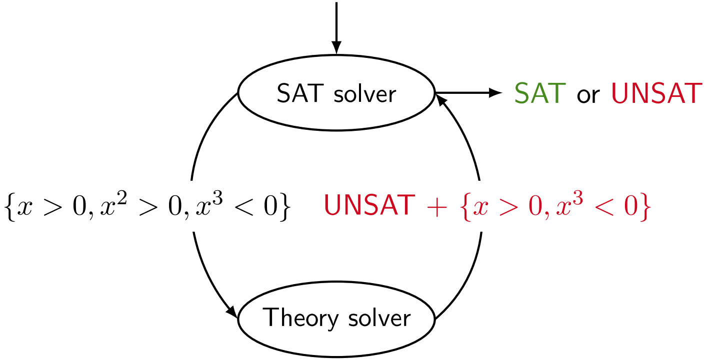


SMT solving


$x > 0 \land (x^2 > 0 \lor x < 0) \land (x^3 < 0 \lor x = 3)$
SAT solver
SAT or UNSAT
$\{x > 0, x^2 > 0, x^3 < 0\}$
UNSAT + $\{x > 0, x^3 < 0\}$
Theory solver

SMT solving

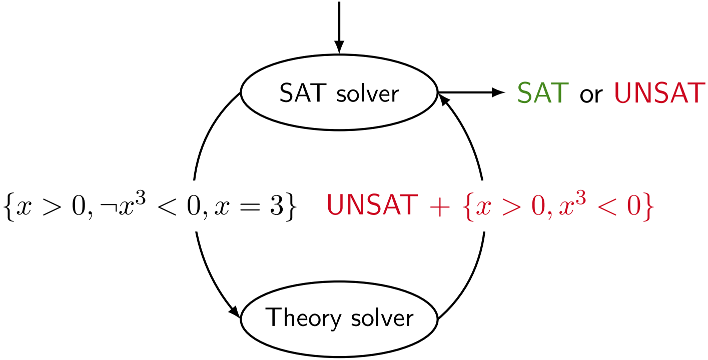
$$x > 0 \wedge (x^2 > 0 \vee x < 0) \wedge (x^3 < 0 \vee x = 3) \wedge (\neg x > 0 \vee \neg x^3 < 0)$$



RWTH AACHEN UNIVERSITY

SMT solving

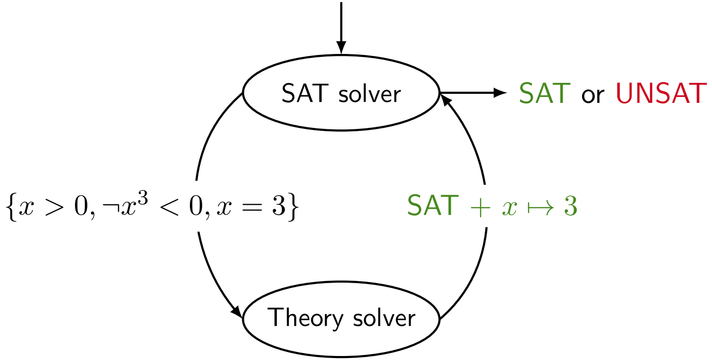
$$x > 0 \wedge (x^2 > 0 \vee x < 0) \wedge (x^3 < 0 \vee x = 3) \wedge (\neg x > 0 \vee \neg x^3 < 0)$$



RWTH AACHEN UNIVERSITY

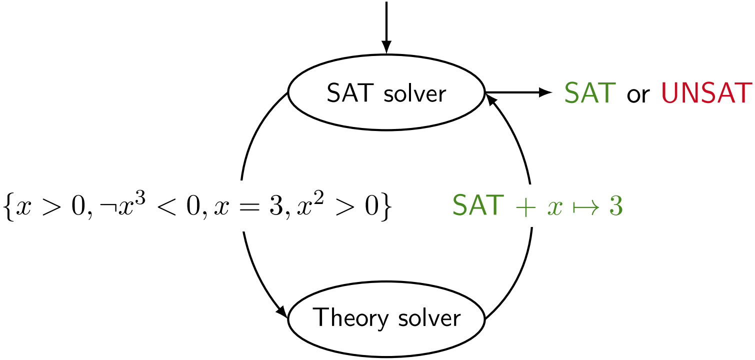
SMT solving

$$x > 0 \wedge (x^2 > 0 \vee x < 0) \wedge (x^3 < 0 \vee x = 3) \wedge (\neg x > 0 \vee \neg x^3 < 0)$$



SMT solving

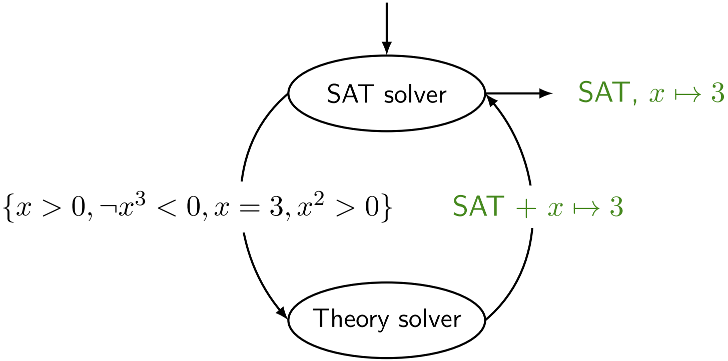
$$x > 0 \wedge (x^2 > 0 \vee x < 0) \wedge (x^3 < 0 \vee x = 3) \wedge (\neg x > 0 \vee \neg x^3 < 0)$$



RWTH AACHEN UNIVERSITY

SMT solving

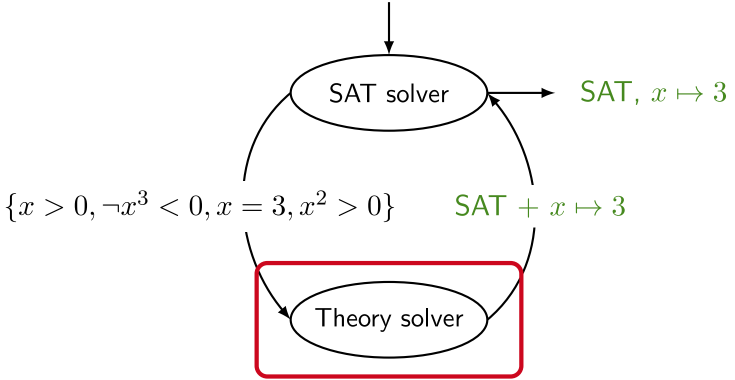
$$x > 0 \land (x^2 > 0 \lor x < 0) \land (x^3 < 0 \lor x = 3) \land (\neg x > 0 \lor \neg x^3 < 0)$$



RWTH AACHEN UNIVERSITY

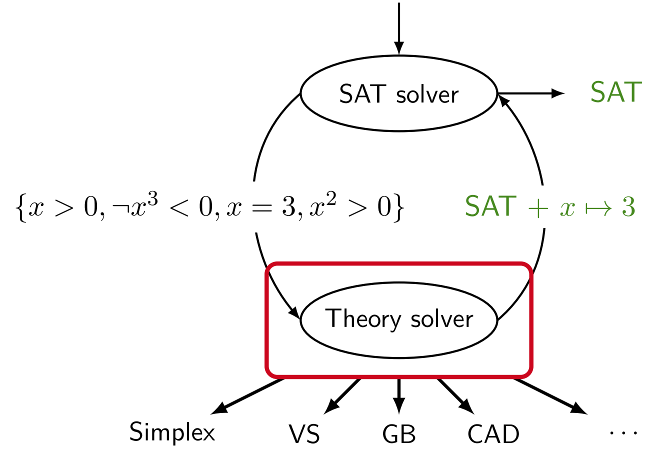
SMT solving

$$x > 0 \land (x^2 > 0 \lor x < 0) \land (x^3 < 0 \lor x = 3) \land (\neg x > 0 \lor \neg x^3 < 0)$$



SMT solving

$$x > 0 \wedge (x^2 > 0 \vee x < 0) \wedge (x^3 < 0 \vee x = 3) \wedge (\neg x > 0 \vee \neg x^3 < 0)$$



Our solver: SMT-RAT [CKJ+15]

Toolbox for SMT solving

- Modular framework to combine solving techniques
- Various solving modules: SAT, Simplex, ICP, GB, VS, CAD, ...
- Strategic combination to build an SMT solver
- Low-threshold platform for experiments

Aimed at: `QF_NRA`, `QF_NIA`, `QF_PB`
Also supported: `QF_LRA`, `QF_LIA`, `QF_RDL`, `QF_IDL`, `QF_BV`

See `https://github.com/smtrat/smtrat`

Theory solvers

Nonlinear problems are difficult, but you know how to tackle them.

Theory solvers

Nonlinear problems are difficult, but you know how to tackle them.

Properties we like (SMT compliancy)

- Automatable (push-button solution)
- Preferably complete, at least fail verbosely
- Satisfying witness
- Reason for unsatisfiability (infeasible subset)
- Input can be extended (incrementality)
- Input can be reduced (backtracking)

RWTH AACHEN UNIVERSITY

SMT compliancy – what we can do

- Automation
- Early abort
- Adapt method to our application
  Effective heuristics, low-end modifications, preprocessing, ...
- Provide (reasonably) efficient implementations
- Apply our solutions to industrial problems

SMT compliancy – what we can do

- Automation
- Early abort
- Adapt method to our application
  Effective heuristics, low-end modifications, preprocessing, ...
- Provide (reasonably) efficient implementations
- Apply our solutions to industrial problems
- Incorporate incrementality and backtracking
  Gröbner Bases [JLCA13], CAD [CKJ+15, Hae17]
- Reasons for unsatisfiability [JLCA13, Hen17]
- Combine solving techniques [CKJ+15]

RWTH AACHEN UNIVERSITY

Success stories

- Virtual Substitution as theory solver [CA11, KCA16]
  Incrementality and backtracking, reasons for unsatisfiability, support for integer problems

Success stories

- Virtual Substitution as theory solver [CA11, KCA16]
  Incrementality and backtracking, reasons for unsatisfiability, support for integer problems
- Gröbner Bases as theory solver [JLCA13]
  Approximates real radical, tries to construct satisfying witness, reasons for unsatisfiability, handles inequalities

RWTH AACHEN UNIVERSITY

Success stories

- Virtual Substitution as theory solver [CA11, KCA16]
  Incrementality and backtracking, reasons for unsatisfiability, support for integer problems
- Gröbner Bases as theory solver [JLCA13]
  Approximates real radical, tries to construct satisfying witness, reasons for unsatisfiability, handles inequalities
- Cylindrical Algebraic Decomposition as theory solver [CKJ+15, KCA16]
  Incrementality and backtracking in projection and lifting, reasons for unsatisfiability, support for integer problems

RWTH AACHEN UNIVERSITY

Success stories

- Virtual Substitution as theory solver [CA11, KCA16]
  Incrementality and backtracking, reasons for unsatisfiability, support for integer problems
- Gröbner Bases as theory solver [JLCA13]
  Approximates real radical, tries to construct satisfying witness, reasons for unsatisfiability, handles inequalities
- Cylindrical Algebraic Decomposition as theory solver [CKJ+15, KCA16]
  Incrementality and backtracking in projection and lifting, reasons for unsatisfiability, support for integer problems
- NLSAT: novel CAD-based solving scheme [JDM12]
  Uses CAD to construct single cells

Wait a second...

Wait a second...

Custom implementations for all of this?

Wait a second...

Custom implementations for all of this?

Seriously?

RWTH AACHEN UNIVERSITY

Wait a second...

Custom implementations for all of this?

Seriously?

Yes.

Using other software

What works well (for us):

- `GMP`, `Eigen`
- Originally used `GiNaC` and `CLN`, not anymore
- Some functions from `CoCoALib`
  `gcd()`, `factor()`, `squareFreePart()`
- Finding symmetries using `bliss`

Using other software

What works well (for us):

- `GMP`, `Eigen`
- Originally used `GiNaC` and `CLN`, not anymore
- Some functions from `CoCoALib`
  `gcd()`, `factor()`, `squareFreePart()`
- Finding symmetries using `bliss`

Common Problems:

- Usable `C` / `C++` interface
- Performance
- Conversion overhead
- SMT compliancy

RWTH AACHEN UNIVERSITY

Gröbner Bases from `CoCoALib` [AB]

`CoCoALib` is dedicated to computing Gröbner Bases.

Open problems:

- Approximate real radical (work in progress, quality vs. speed)
- Backtracking (snapshots?)
- Satisfying witness
- Reason for unsatisfiability ($\rightarrow$ `GenRepr`, expensive?)

Gröbner Bases from `CoCoALib` [AB]

`CoCoALib` is dedicated to computing Gröbner Bases.

Open problems:

- Approximate real radical (work in progress, quality vs. speed)
- Backtracking (snapshots?)
- Satisfying witness
- Reason for unsatisfiability ($\rightarrow$ `GenRepr`, expensive?)

We do not need a Gröbner Basis.
We need an answer to a theory query.

And we guess a Gröbner Basis could provide this answer...

Maple as a theory solver

Maple is better at everything...

```
solve()
```

```
RootFinding:-WitnessPoints()
```

```
RegularChains:-CylindricalAlgebraicDecompose()
```

```
RegularChains:-LazyRealTriangularize()
```

Maple as a theory solver

Maple is better at everything...

`solve()`

The standard solution, unfortunately not suitable here:

- No satisfying witness, „just" a simplified set of constraints
- No information if no solution exists (`NULL`)
- May be incomplete (`_SolutionsMayBeLost`)
- Result may leave theory ($x < 3/y$, $x = \sqrt{2}$, ...)

`RootFinding:-WitnessPoints()`

`RegularChains:-CylindricalAlgebraicDecompose()`

`RegularChains:-LazyRealTriangularize()`

RWTH AACHEN UNIVERSITY

Maple as a theory solver

Maple is better at everything...

`solve()`

`RootFinding:-WitnessPoints()`

Numeric approach to find solutions of equalities or inequalities

- No way to combine equalities and inequalities
- No support for weak inequalities
- Rounding errors? Reasons for unsatisfiability?

`RegularChains:-CylindricalAlgebraicDecompose()`

`RegularChains:-LazyRealTriangularize()`

RWTH AACHEN UNIVERSITY

Maple as a theory solver

Maple is better at everything...

`solve()`

`RootFinding:-WitnessPoints()`

`RegularChains:-CylindricalAlgebraicDecompose()`

Essentially the same approach as our own implementation
No early abort, incrementality or backtracking
$\rightarrow$ comparably slow

`RegularChains:-LazyRealTriangularize()`

RWTH AACHEN UNIVERSITY

Maple as a theory solver

Maple is better at everything...

`solve()`

`RootFinding:-WitnessPoints()`

`RegularChains:-CylindricalAlgebraicDecompose()`

`RegularChains:-LazyRealTriangularize()`

Some early abort compared to `CylindricalAlgebraicDecompose`
Still no incrementality or backtracking
Subject of future investigation

Maple as a theory solver

Maple is better at everything...

```
solve()
```

```
RootFinding:-WitnessPoints()
```

```
RegularChains:-CylindricalAlgebraicDecompose()
```

```
RegularChains:-LazyRealTriangularize()
```

This is alright for an interactive system.
It must be taken care of in a fully automated one.

What we need help with

- Gröbner Bases for problems on $\mathbb{R}$?
- Satisfying witnesses from Gröbner bases?
- Stability of numerical approaches?
- Guarantees on rounding errors?
- Factorization?
- Multivariate GCD?

Conclusions

You create amazing mathematics.

RWTH AACHEN UNIVERSITY

Conclusions

You create amazing mathematics.

We use mathematics as a tool, not for its own sake.

Conclusions

You create amazing mathematics.
We use mathematics as a tool, not for its own sake.

- We (want to) use your methods...
- ... but have somewhat peculiar requirements ...
- ... and end up re-implementing a lot.

RWTH AACHEN UNIVERSITY

Conclusions

You create amazing mathematics.
We use mathematics as a tool, not for its own sake.

- We (want to) use your methods...
- ... but have somewhat peculiar requirements ...
- ... and end up re-implementing a lot.

Maybe we can improve by collaborating?

Conclusions

You create amazing mathematics.
We use mathematics as a tool, not for its own sake.

- We (want to) use your methods...
- ... but have somewhat peculiar requirements ...
- ... and end up re-implementing a lot.

Maybe we can improve by collaborating?

Also:

- You can use our software (Maple does)
- We can provide benchmarks

RWTH AACHEN UNIVERSITY

References

[AB] J. Abbott and A. M. Bigatti. CoCoALib: a c++ library for doing Computations in Commutative Algebra. Available at `http://cocoa.dima.unige.it/cocoalib`.

[CA11] Florian Corzilius and Erika Abraham. Virtual Substitution for SMT Solving. In *FCT'11*, volume 6914 of *LNCS*, pages 360–371. Springer, 2011.

[CKJ+15] Florian Corzilius, Gereon Kremer, Sebastian Junges, Stefan Schupp, and Erika Abraham. SMT-RAT: An Open Source C++ Toolbox for Strategic and Parallel SMT Solving. In *SAT'15*, volume 9340 of *LNCS*, pages 360–368. Springer, 2015.

[Hae17] Rebecca Haehn. Using equational constraints in an incremental CAD projection. Master's thesis, RWTH Aachen University, 2017.

[Hen17] Wanja Hentze. Computing Minimal Infeasible Subsets for the Cylindrical Algebraic Cecomposition, 2017.

[JDM12] Dejan Jovanović and Leonardo De Moura. Solving non-linear arithmetic. In *ICJAR'12*, pages 339–354. Springer, 2012.

[JLCA13] Sebastian Junges, Ulrich Loup, Florian Corzilius, and Erika Abraham. On Gröbner Bases in the Context of Satisfiability-Modulo-Theories Solving over the Real Numbers. In *CAI'13*, volume 8080 of *LNCS*, pages 186–198. Springer, 2013.

[KCA16] Gereon Kremer, Florian Corzilius, and Erika Abraham. A Generalised Branch-and-Bound Approach and its Application in SAT Modulo Nonlinear Integer Arithmetic. In *CASC'16*, volume 9890 of *LNCS*, pages 315–335. Springer, 2016.