# Recent trends in SMT solving

## and what to expect from the next generation

Gereon Kremer

RWTH Aachen University, Germany
LuFG Theory of Hybrid Systems

July 17, 2016

# What is this talk about?

## Satisfiability problem

Decide whether an existentially quantified formula $\varphi(x)$ is satisfiable.

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

# What is this talk about?

## Satisfiability problem

Decide whether an existentially quantified formula $\varphi(x)$ is satisfiable.

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

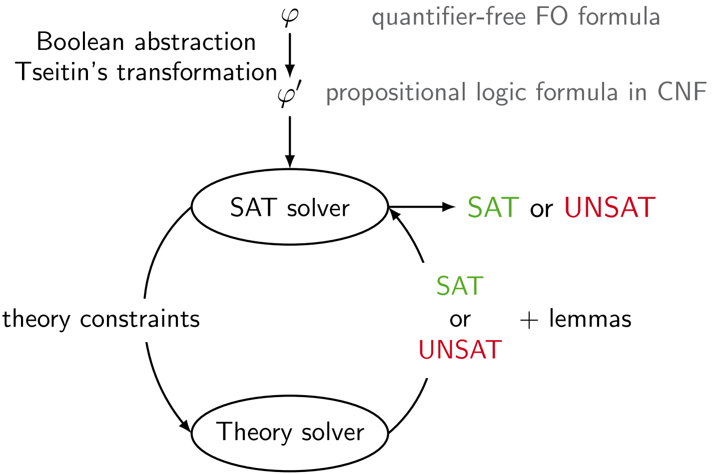
## Satisfiability modulo theories

$\varphi$ is from an existentially quantified first-order logic.

- Fully automated solving
- Common theories: arithmetic (linear / nonlinear, real / integer), arrays, bitvectors, uninterpreted functions, ...
- Combinations of theories

Theory of Hybrid Systems Informatik 2

# Fundamental idea: SAT vs. Theory



# Digression: SAT solving

- $\varphi$ is propositional
- DPLL-style SAT solving
- Combines enumeration, propagation and conflict learning

# Digression: SAT solving

- $\varphi$ is propositional
- DPLL-style SAT solving
- Combines enumeration, propagation and conflict learning
- Solves industrial problems with millions of variables
- Flagship application: digital circuit design and verification

# Digression: SAT solving

- $\varphi$ is propositional
- DPLL-style SAT solving
- Combines enumeration, propagation and conflict learning
- Solves industrial problems with millions of variables
- Flagship application: digital circuit design and verification

Community support:

- Standardized input language, lots of benchmarks available
- Competitions since 2002

  2014 SAT Competition: $3$ categories, $79$ participants with $137$ solvers.

  SAT Live! forum as community platform, dedicated conferences, journals, etc.

# SMT solving

- Modelling is hard if restricted to propositional logic
- Theory constraints express applications more naturally
- Theories must still be solvable

# SMT solving

- Modelling is hard if restricted to propositional logic
- Theory constraints express applications more naturally
- Theories must still be solvable

Applications: verification (model checking, static analysis, termination analysis); test case generation; controller synthesis; predicate abstraction; equivalence checking; scheduling; planning; product design automation and optimisation, ...

# SMT solving

- Modelling is hard if restricted to propositional logic
- Theory constraints express applications more naturally
- Theories must still be solvable

Applications: verification (model checking, static analysis, termination analysis); test case generation; controller synthesis; predicate abstraction; equivalence checking; scheduling; planning; product design automation and optimisation, . . .

Community support:

- SMT-LIB as standard input language since 2004.
- Competitions since 2005.
- SMT-COMP 2016 competition:

  $40$ logical categories, $19$ distinct solvers.

  $154424$ $(+41690)$ benchmark instances.

# Common theories – Arithmetic

## Linear arithmetic

$3x - 7y \leq 8$

Simplex, Fourier-Motzkin, B&B, Bit-blasting, Gomory Cuts

CVC4, MathSAT 5, OpenSMT2, SMT-RAT, SMTInterpol, toysmt, veriT, Yices, Z3

Theory of Hybrid Systems Informatik 2

# Common theories – Arithmetic

## Linear arithmetic

$3x - 7y \leq 8$

Simplex, Fourier-Motzkin, B&B, Bit-blasting, Gomory Cuts

CVC4, MathSAT 5, OpenSMT2, SMT-RAT, SMTInterpol, toysmt, veriT, Yices, Z3

## Nonlinear arithmetic

$3x^2 - 7xy \leq 8$

CAD, VS, Gröbner Bases, ICP, Bit-blasting, B&B

Real: CVC4, raSAT, SMT-RAT, Yices, Z3

Integer: AProVE, CVC4, ProB, raSAT, SMT-RAT, Yices, Z3

# Common theories – Arithmetic

## Linear arithmetic

$3x - 7y \leq 8$

Simplex, Fourier-Motzkin, B&B, Bit-blasting, Gomory Cuts

CVC4, MathSAT 5, OpenSMT2, SMT-RAT, SMTInterpol, toysmt, veriT, Yices, Z3

## Nonlinear arithmetic

$3x^2 - 7xy \leq 8$

CAD, VS, Gröbner Bases, ICP, Bit-blasting, B&B

Real: CVC4, raSAT, SMT-RAT, Yices, Z3

Integer: AProVE, CVC4, ProB, raSAT, SMT-RAT, Yices, Z3

## Uninterpreted functions

$a = b \land \neg(f(b) = f(a))$

Congruence closure

CVC4, MathSAT 5, OpenSMT2, SMTInterpol, toysmt, veriT, Yices, Z3

# Common theories – Other

## Arrays

$i = j \rightarrow read(write(a, i, v), j) = v$

On-demand lemma generation / lazy atom instantiation

CVC4, MathSAT 5, SMTInterpol, Yices, Z3

# Common theories – Other

## Arrays

$i = j \to read(write(a, i, v), j) = v$

On-demand lemma generation / lazy atom instantiation

CVC4, MathSAT 5, SMTInterpol, Yices, Z3

## Bitvectors

$a \mid b \leq a\&b$

Bit-blasting

ABC, Boolector, CVC4, MapleSTP, MathSAT 5, Minkeyrink, Q3B, STP, Yices, Z3

Theory
of Hybrid
Systems
Informatik 2

# Common theories – Other

## Arrays

$i = j \rightarrow read(write(a, i, v), j) = v$

On-demand lemma generation / lazy atom instantiation

CVC4, MathSAT 5, SMTInterpol, Yices, Z3

## Bitvectors

$a \mid b \leq a\&b$

Bit-blasting

ABC, Boolector, CVC4, MapleSTP, MathSAT 5, Minkeyrink, Q3B, STP, Yices, Z3

## Floating point

$sub_{RNE}(x, y) = z$

Bit-blasting

MathSAT 5

# What about existing tools?

We have

- SAT solvers (MiniSAT, Glucose, Sat4j),
- LP solvers (CPlex, Gurobi, SCIP),
- CAS (Maple, Mathematica, Matlab) and many more.

Just plug them together! What is the problem?

# What about existing tools?

We have

- SAT solvers (MiniSAT, Glucose, Sat4j),
- LP solvers (CPlex, Gurobi, SCIP),
- CAS (Maple, Mathematica, Matlab) and many more.

Just plug them together! What is the problem?

- Many theory calls that only differ slightly
- Explanations for unsatisfiability
- Removal of constraints

# What about existing tools?

We have

- SAT solvers (MiniSAT, Glucose, Sat4j),
- LP solvers (CPlex, Gurobi, SCIP),
- CAS (Maple, Mathematica, Matlab) and many more.

Just plug them together! What is the problem?

- Many theory calls that only differ slightly
- Explanations for unsatisfiability
- Removal of constraints

Incrementality, lemma generation, backtracking!

hybrid Theory of Hybrid Systems Informatik 2

# At Aachen: SMT-RAT

- Toolbox for SMT solving [SAT'12, SAT'15]
- SAT solver, many theory modules, preprocessing
- Basic datastructures: formulas, constraints, polynomials, ...

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

# Extension: Optimization

## SMT with optimization

$$\min f(x) \text{ w.r.t. } \varphi(x)$$

Theory
of Hybrid
Systems
Informatik 2

# Extension: Optimization

## SMT with optimization

$$\min f(x) \text{ w.r.t. } \varphi(x)$$

- $f$ and $\varphi$ use same theory
- Multiobjective: lexicographic ordering
- Straightforward for linear arithmetic ($\nu Z$, SMT-RAT)
- More difficult for nonlinear arithmetic
- How would $f$ look like for uninterpreted functions?

# Extension: Quantification

## SMT with quantifiers

$$\exists x_1 . \forall x_2 . ... \exists x_n . \varphi(x_1, ..., x_n)$$

# Extension: Quantification

## SMT with quantifiers

$$\exists x_1 . \forall x_2 .... \exists x_n . \varphi(x_1, ..., x_n)$$

- More expressive
- Easy case: small domain for $\forall$ variables
- Most decision procedure are designed for $\exists$ only

# Extension: Parallelization

- Portfolio
  - Run multiple solvers / configurations in parallel
  - As many cores as there are solvers

# Extension: Parallelization

- Portfolio
  - Run multiple solvers / configurations in parallel
  - As many cores as there are solvers
- Strategic parallelization
  - If solver is modularized, run modules in parallel
  - In SMT-RAT: multiple theory modules in parallel

Theory of Hybrid Systems Informatik 2

# Extension: Parallelization

- Portfolio
  - Run multiple solvers / configurations in parallel
  - As many cores as there are solvers
- Strategic parallelization
  - If solver is modularized, run modules in parallel
  - In SMT-RAT: multiple theory modules in parallel
- Parallel algorithms
  - Parallelization in individual decision procedures

# Extension: Unsat cores / Proofs

## Unsat core

$$\varphi'(x) \equiv False \text{ where } \varphi' \subseteq \varphi$$

Give proof that $\neg\exists x.\varphi(x)$.

# Extension: Unsat cores / Proofs

## Unsat core

$$\varphi'(x) \equiv False \text{ where } \varphi' \subseteq \varphi$$

Give proof that $\neg\exists x.\varphi(x)$.

- Minimal vs. minimum
- Meaningful measure: size? complexity?
- Proofs for humans or theorem provers?
- Proofs without encoding the whole algorithm?

Theory of Hybrid Systems Informatik 2

# Extension: Additional theories

- Floating point

  Philipp Rümmer and Thomas Wahl. An SMT-LIB theory of binary floating-point arithmetic. In *SMT'10*, 2010

- Recursive functions

  Clark Barrett, Pascal Fontaine, and Cesare Tinelli. The SMT-LIB Standard Version 2.6. 2015

- Infinitesimals

  Leonardo De Moura and Grant Olney Passmore. Computation in real closed infinitesimal and transcendental extensions of the rationals. In *CADE-24*. 2013

- Trigonometric and exponential functions

  Sicun Gao, Soonho Kong, and Edmund M Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *CADE-24*. 2013

- ... ?