

Satisfiability Modulo Theories

Viktor Kunčák

Satisfiability Modulo Theories

SAT = Satisfiability for Propositional Logic

- ▶ formula: $p \wedge (\neg q \vee r) \wedge s$
- ▶ Do there exist truth values p, q, r, s that make formula true?

Satisfiability Modulo Theories

SAT = Satisfiability for Propositional Logic

- ▶ formula: $p \wedge (\neg q \vee r) \wedge s$
- ▶ Do there exist truth values p, q, r, s that make formula true? yes, e.g.,
 $p \mapsto 1, q \mapsto 0, s \mapsto 1$ (r is any)

Satisfiability Modulo Theories

SAT = Satisfiability for Propositional Logic

- ▶ formula: $p \wedge (\neg q \vee r) \wedge s$
- ▶ Do there exist truth values p, q, r, s that make formula true? yes, e.g., $p \mapsto 1, q \mapsto 0, s \mapsto 1$ (r is any)

SMT = Satisfiability Modulo Theories (e.g. Z3 solver)

- ▶ formula: $a = b \wedge (\neg(f(a) = f(b)) \vee b = c) \wedge \neg(f(a) = f(c))$
- ▶ Do there exist values of a, b, c that makes formula true?

Satisfiability Modulo Theories

SAT = Satisfiability for Propositional Logic

- ▶ formula: $p \wedge (\neg q \vee r) \wedge s$
- ▶ Do there exist truth values p, q, r, s that make formula true? yes, e.g., $p \mapsto 1, q \mapsto 0, s \mapsto 1$ (r is any)

SMT = Satisfiability Modulo Theories (e.g. Z3 solver)

- ▶ formula: $a = b \wedge (\neg(f(a) = f(b)) \vee b = c) \wedge \neg(f(a) = f(c))$
- ▶ Do there exist values of a, b, c that makes formula true? No. We have: $a = b, f(a) = f(b), b = c, a = c, f(a) = f(c), \neg(f(a) = f(c))$ - unsat

Satisfiability Modulo Theories

SAT = Satisfiability for Propositional Logic

- ▶ formula: $p \wedge (\neg q \vee r) \wedge s$
- ▶ Do there exist truth values p, q, r, s that make formula true? yes, e.g., $p \mapsto 1, q \mapsto 0, s \mapsto 1$ (r is any)

SMT = Satisfiability Modulo Theories (e.g. Z3 solver)

- ▶ formula: $a = b \wedge (\neg(f(a) = f(b)) \vee b = c) \wedge \neg(f(a) = f(c))$
- ▶ Do there exist values of a, b, c that makes formula true? No. We have: $a = b, f(a) = f(b), b = c, a = c, f(a) = f(c), \neg(f(a) = f(c))$ - unsat
- ▶ Another example: $x = y \wedge f(x) < f(y)$

Satisfiability Modulo Theories

SAT = Satisfiability for Propositional Logic

- ▶ formula: $p \wedge (\neg q \vee r) \wedge s$
- ▶ Do there exist truth values p, q, r, s that make formula true? yes, e.g., $p \mapsto 1, q \mapsto 0, s \mapsto 1$ (r is any)

SMT = Satisfiability Modulo Theories (e.g. Z3 solver)

- ▶ formula: $a = b \wedge (\neg(f(a) = f(b)) \vee b = c) \wedge \neg(f(a) = f(c))$
- ▶ Do there exist values of a, b, c that makes formula true? No. We have: $a = b, f(a) = f(b), b = c, a = c, f(a) = f(c), \neg(f(a) = f(c))$ - unsat
- ▶ Another example: $x = y \wedge f(x) < f(y)$ - unsat

Satisfiability Modulo Theories

SAT = Satisfiability for Propositional Logic

- ▶ formula: $p \wedge (\neg q \vee r) \wedge s$
- ▶ Do there exist truth values p, q, r, s that make formula true? yes, e.g., $p \mapsto 1, q \mapsto 0, s \mapsto 1$ (r is any)

SMT = Satisfiability Modulo Theories (e.g. Z3 solver)

- ▶ formula: $a = b \wedge (\neg(f(a) = f(b)) \vee b = c) \wedge \neg(f(a) = f(c))$
- ▶ Do there exist values of a, b, c that makes formula true? No. We have: $a = b, f(a) = f(b), b = c, a = c, f(a) = f(c), \neg(f(a) = f(c))$ - unsat
- ▶ Another example: $x = y \wedge f(x) < f(y)$ - unsat
- ▶ Another example: $1 \leq x \wedge x \leq 2 \wedge f(x) \neq f(1)$

Satisfiability Modulo Theories

SAT = Satisfiability for Propositional Logic

- ▶ formula: $p \wedge (\neg q \vee r) \wedge s$
- ▶ Do there exist truth values p, q, r, s that make formula true? yes, e.g., $p \mapsto 1, q \mapsto 0, s \mapsto 1$ (r is any)

SMT = Satisfiability Modulo Theories (e.g. Z3 solver)

- ▶ formula: $a = b \wedge (\neg(f(a) = f(b)) \vee b = c) \wedge \neg(f(a) = f(c))$
- ▶ Do there exist values of a, b, c that makes formula true? No. We have: $a = b, f(a) = f(b), b = c, a = c, f(a) = f(c), \neg(f(a) = f(c))$ - unsat
- ▶ Another example: $x = y \wedge f(x) < f(y)$ - unsat
- ▶ Another example: $1 \leq x \wedge x \leq 2 \wedge f(x) \neq f(1)$: $x \mapsto 2$

$$1 \leq x \wedge x \leq 2 \wedge f(x) \neq f(1) \wedge f(x) \neq f(2)$$

x - integer

Satisfiability Modulo Theories

SAT = Satisfiability for Propositional Logic

- ▶ formula: $p \wedge (\neg q \vee r) \wedge s$
- ▶ Do there exist truth values p, q, r, s that make formula true? yes, e.g., $p \mapsto 1, q \mapsto 0, s \mapsto 1$ (r is any)

SMT = Satisfiability Modulo Theories (e.g. Z3 solver)

- ▶ formula: $a = b \wedge (\neg(f(a) = f(b)) \vee b = c) \wedge \neg(f(a) = f(c))$
- ▶ Do there exist values of a, b, c that makes formula true? No. We have: $a = b, f(a) = f(b), b = c, a = c, f(a) = f(c), \neg(f(a) = f(c))$ - unsat
- ▶ Another example: $x = y \wedge f(x) < f(y)$ - unsat
- ▶ Another example: $1 \leq x \wedge x \leq 2 \wedge f(x) \neq f(1) : x \mapsto 2$

↙ E, SPASS, Vampire

Large formulas with few no or few quantifiers (unlike pure FOL provers)

- ▶ propositional structure explored using SAT solver
- ▶ function and relation symbols come from decidable theories (quantifier-free linear arithmetic, algebraic data types)
- ▶ atomic formulas solved using decision procedures (theory solvers)
- ▶ quantifiers handled mostly by instantiation

Flattening and Extracting Propositional Structure

$$a = b \wedge (f(a) \neq f(b) \vee b = c) \wedge f(a) \neq f(c)$$

for each atomic formula introduce propositional variable:

Flattening and Extracting Propositional Structure

$$a = b \wedge (f(a) \neq f(b) \vee b = c) \wedge f(a) \neq f(c)$$

for each atomic formula introduce propositional variable:

$$p \wedge (\neg q \vee r) \wedge \neg s$$

$$p \Leftrightarrow a = b$$

$$q \Leftrightarrow f(a) = f(b)$$

$$r \Leftrightarrow b = c$$

$$s \Leftrightarrow f(a) = f(c)$$

flatten: give name to each subterm, e.g. f_a denotes $f(a)$:

Flattening and Extracting Propositional Structure

$$a = b \wedge (f(a) \neq f(b) \vee b = c) \wedge f(a) \neq f(c)$$

for each atomic formula introduce propositional variable:

$$p \wedge (\neg q \vee r) \wedge \neg s$$

$$p \Leftrightarrow a = b$$

$$q \Leftrightarrow f(a) = f(b)$$

$$r \Leftrightarrow b = c$$

$$s \Leftrightarrow f(a) = f(c)$$

flatten: give name to each subterm, e.g. f_a denotes $f(a)$:

$$p \wedge (\neg q \vee r) \wedge \neg s \} \text{ give to SAT solver, who returns e.g. } p \wedge \neg q \wedge s$$

Flattening and Extracting Propositional Structure

$$a = b \wedge (f(a) \neq f(b) \vee b = c) \wedge f(a) \neq f(c)$$

for each atomic formula introduce propositional variable:

$$p \wedge (\neg q \vee r) \wedge \neg s$$

$$p \Leftrightarrow a = b$$

$$q \Leftrightarrow f(a) = f(b)$$

$$r \Leftrightarrow b = c$$

$$s \Leftrightarrow f(a) = f(c)$$

flatten: give name to each subterm, e.g. f_a denotes $f(a)$:

$p \wedge (\neg q \vee r) \wedge \neg s$ } give to SAT solver, who returns e.g. $p \wedge \neg q \wedge s$

$$p \Leftrightarrow a = b$$

$$q \Leftrightarrow f_a = f_b$$

$$r \Leftrightarrow b = c$$

$$s \Leftrightarrow f_a = f_c$$

maps prop. assignment to conjunction of literals

$$a = b \wedge f_a \neq f_b \wedge f_a \neq f_c$$

Flattening and Extracting Propositional Structure

$$a = b \wedge (f(a) \neq f(b) \vee b = c) \wedge f(a) \neq f(c)$$

for each atomic formula introduce propositional variable:

$$p \wedge (\neg q \vee r) \wedge \neg s$$

$$p \Leftrightarrow a = b$$

$$q \Leftrightarrow f(a) = f(b)$$

$$r \Leftrightarrow b = c$$

$$s \Leftrightarrow f(a) = f(c)$$

flatten: give name to each subterm, e.g. f_a denotes $f(a)$:

$p \wedge (\neg q \vee r) \wedge \neg s$ } give to SAT solver, who returns e.g. $p \wedge \neg q \wedge s$

$$p \Leftrightarrow a = b$$

$$q \Leftrightarrow f_a = f_b$$

$$r \Leftrightarrow b = c$$

$$s \Leftrightarrow f_a = f_c$$

} maps prop. assignment to conjunction of literals

$$a = b \wedge f_a \neq f_b \wedge f_a \neq f_c$$

$$f_a = f(a)$$

$$f_b = f(b)$$

$$f_c = f(c)$$

} each theory uses its conjuncts and definitions

$$\dots \wedge a = b \wedge f_a \neq f_b \wedge f_a \neq f_c$$

UNSAT, give to SAT solver: $\neg(p \wedge \neg q \wedge s)$

Formula containing function symbols and arithmetic

- ▶ f is **uninterpreted symbol** (as in FOL)
- ▶ $+, <, \leq, 1, 3, 5$ are as in linear integer arithmetic; x is of type integer

$$\underbrace{1 \leq x}_p \wedge \underbrace{x < 3}_q \wedge (\underbrace{f(1) + 1 \leq f(x)}_r \wedge \underbrace{f(x) < f(2)}_s) \vee \underbrace{4 = 2x}_t$$

Formula containing function symbols and arithmetic

- ▶ f is **uninterpreted symbol** (as in FOL)
- ▶ $+, <, \leq, 1, 3, 5$ are as in linear integer arithmetic; x is of type integer

$$\underbrace{1 \leq x}_p \wedge \underbrace{x < 3}_q \wedge \left(\underbrace{(f(1) + 1 \leq f(x))}_r \wedge \underbrace{f(x) < f(2)}_s \right) \vee \underbrace{4 = 2x}_t$$

$$p \wedge q \wedge ((r \wedge s) \vee t) \wedge$$

$$p \Leftrightarrow 1 \leq x \wedge$$

$$q \Leftrightarrow x < 3 \wedge$$

$$r \Leftrightarrow u_1 \leq u_2 \wedge u_1 = u_3 + 1 \wedge u_3 = f(u_4) \wedge u_2 = f(x) \wedge u_4 = 1$$

$$s \Leftrightarrow u_2 < u_5 \wedge u_5 = f(u_6) \wedge u_6 = 2$$

$$t \Leftrightarrow u_7 = u_8 \wedge u_7 = 4 \wedge u_8 = 2x$$

Formula containing function symbols and arithmetic

- ▶ f is uninterpreted symbol (as in FOL)
- ▶ $+, <, \leq, 1, 3, 5$ are as in linear integer arithmetic; x is of type integer

$$\underbrace{1 \leq x}_p \wedge \underbrace{x < 3}_q \wedge \left(\underbrace{(f(1) + 1 \leq f(x))}_r \wedge \underbrace{f(x) < f(2))}_s \right) \vee \underbrace{4 = 2x}_t$$

$$p \wedge q \wedge ((r \wedge s) \vee t) \wedge$$

$$p \Leftrightarrow 1 \leq x \wedge$$

$$q \Leftrightarrow x < 3 \wedge$$

$$r \Leftrightarrow u_1 \leq u_2 \wedge u_1 = u_3 + 1 \wedge u_3 = f(u_4) \wedge u_2 = f(x) \wedge u_4 = 1$$

$$s \Leftrightarrow u_2 < u_5 \wedge u_5 = f(u_6) \wedge u_6 = 2$$

$$t \Leftrightarrow u_7 = u_8 \wedge u_7 = 4 \wedge u_8 = 2x$$

Who handles which part in this example:

propositional formula	SAT solver
pure equalities ($u_7 = u_8$)	both theory solvers
highlighted formulas	solver for theory of uninterpreted functions
remaining ones	solver for theory of integer linear arithmetic

Theory of Uninterpreted Function Symbols

$$x=1$$
$$f(x)=f(1)$$

Quantifier-free first-order logic with equality

Assume it is interpreted over an infinite domain

Assume no relation symbols: replace $R(t_1, \dots, t_n)$ with $f_R(t_1, \dots, t_n) = T$ for some fresh constant T

SAT solver handles disjunctions: assume conjunction of equalities and disequalities

Key inference rule, for each function symbol f of n arguments:

$$\frac{s=t \quad t=r}{s=r}$$

$$s=s$$

$$\frac{t_1 = t'_1 \quad \dots \quad t_n = t'_n}{f(t_1, \dots, t_n) = f(t'_1, \dots, t'_n)}$$

$$\frac{s=t}{t=s}$$

$$\frac{t_1 = t'_1}{f(t_1) = f(t'_1)}$$

Also: “=” is equivalence relation and $t \neq t$ is contradictory

Apply these rules only to those terms that occur in the formula

Implementation: E -graph stores congruence relation computed so far.

Applying rules: merging nodes in this graph

constructor:
 $f_c(t_1) = f_c(t'_1)$

$$t_1 = t'_1$$

```
abstract class Nat
  case class Zero ()
  case class Succ(n: Nat) extends Nat
```

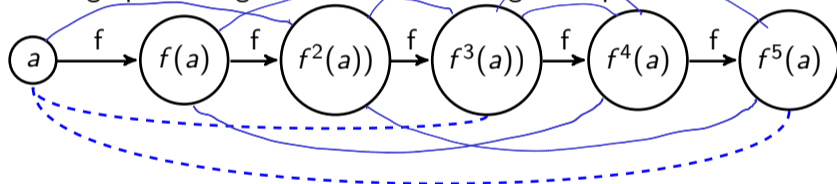
Example of Running the Algorithm

Let $f^k(a)$ denote $f(\dots f(a)\dots)$ with k -fold application of f . Consider

$$f^3(a) = a \wedge f^5(a) = a \wedge f^2(a) \neq a$$

Apply the congruence closure algorithm to check its satisfiability.

Initial graph of all ground terms and the given equalities:



Congruence rule

in this case: $x = y \wedge f(x) = f(y)$

Equivalence maintained using union-find algorithm

Conjunction is satisfiable \Leftrightarrow there is literal $t_1 \neq t_2$ where t_1, t_2 are merged

\Leftarrow): by properties of equality, conclusions are sound

\Rightarrow): computed congruence extends to congruence on the Herbrand model

Decision Procedures are Building Blocks of SMT Solvers

Decision procedures:

- ▶ uninterpreted functions
- ▶ algebraic data types
- ▶ rational linear arithmetic
- ▶ integer linear arithmetic

Many other decidable theories exist (e.g. sets with cardinality bounds)

Tarski has shown that there exists a quantifier elimination algorithm for conjunction of polynomials over reals and over complex numbers.

Later method: Cylindrical Algebraic Decomposition (CAD), used in computer algebra systems

A fork of Z3 has implementation of a complete procedure for reals

Over integers, the problem is undecidable: 10th Hilbert's problem (Y. Matiyasevich)

Over rationals, the decidability of the problem is still open

Quantifier Instantiation During SMT Solving Process

$$\forall x \quad c(x) \Rightarrow \underbrace{s(x) = t(x)}$$

$$G \wedge \forall x.F(x) \rightsquigarrow G \wedge F(t) \wedge \forall x.F(x)$$

where t is a term occurring in G

- ▶ this can go on forever
- ▶ in general this is incomplete: may need to invent terms that do not occur
- ▶ even in the limit it is not complete with respect to the ideal semantics of e.g. integers (theory of quantified integers is not even enumerable)

Controlling the instantiation process using **triggers**

- ▶ for each quantified formula $\forall \bar{x}.F(\bar{x})$ require a pattern $P(\bar{x})$ that contains all free variables in $F(\bar{x})$
- ▶ instantiate $F(\bar{x})$ only if the the pattern $P(x)$ occurs in the ground formula so far
- ▶ introduced in Simplify: a theorem prover for program checking

More information in these papers

- ▶ Solving Quantified Verification Conditions using Satisfiability Modulo Theories
- ▶ Efficient E-matching for SMT solvers

Exercises

QE For Rational Arithmetic

If a_i, b_i, c_j, d_j are linear terms, eliminate x from this formula F :

$$\bigwedge_{i=1}^A a_i < x \wedge \bigwedge_{i=1}^B b_i \leq x \wedge \bigwedge_{j=1}^C x \leq c_j \wedge \bigwedge_{j=1}^D x < d_j$$

That is, describe a quantifier-free formula equivalent to $\exists x.F$.

Prove that the equivalence holds.