SMT in practice. veriT

David Déharbe, *Pascal Fontaine* & Christophe Ringeissen

Procédures de décision et vérification de programmes: Lecture 8

SMT = SAT + expressiveness

- SAT solvers

$$\neg\big[\,(p \Rightarrow q) \Rightarrow \big[\,(\neg p \Rightarrow q) \Rightarrow q\big]\big]$$

- Congruence closure (uninterpreted symbols + equality)

$$a = b \wedge \big[f(a) \neq f(b) \vee (p(a) \wedge \neg p(b))\big]$$

- adding arithmetic

$$a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \big[f(a) \neq f(b) \vee (p(a) \wedge \neg p(b + x))\big]$$

- ...

Some examples: Alt-Ergo, Barcelogic, CVC4 (SVC, CVC, CVC-lite, CVC3), MathSAT, OpenSMT, Yices, Z3 ...

The veriT solver

Standard input language: SMT-LIB 2.0

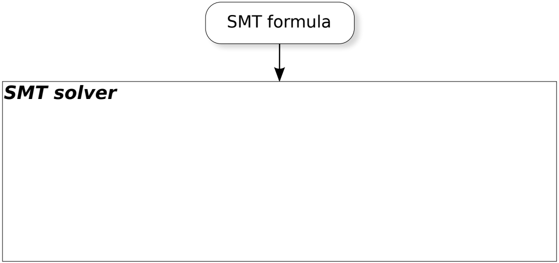
$$a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \left[f(a) \neq f(b) \vee (q(a) \wedge \neg q(b + x)) \right]$$

In SMT-LIB 2.0 format:

```
(set-logic QF_UFLRA)
(set-info :source | Example formula in SMT-LIB 2.0 |)
(set-info :smt-lib-version 2.0)
(declare-fun f (Real) Real)
(declare-fun q (Real) Bool)
(declare-fun a () Real)
(declare-fun b () Real)
(declare-fun x () Real)
(assert (and (<= a b) (<= b (+ a x)) (= x 0)
             (or (not (= (f a) (f b)))
                 (and (q a) (not (q (+ b x)))))))
(check-sat)
(exit)
```

From propositional SAT to SMT

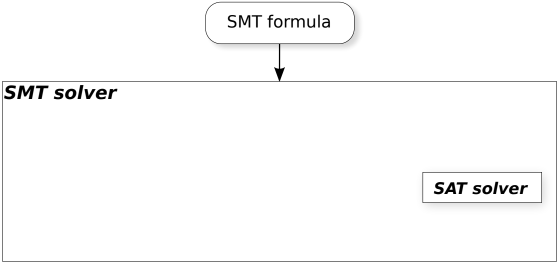
Reducing arbitrary boolean combinations to conjunctions



Input: $a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \left[f(a) \neq f(b) \vee (q(a) \wedge \neg q(b + x)) \right]$

From propositional SAT to SMT

Reducing arbitrary boolean combinations to conjunctions

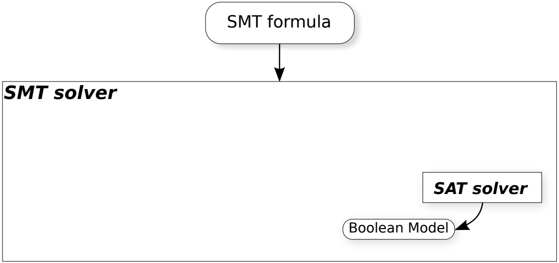


Input: $a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \big[f(a) \neq f(b) \vee (q(a) \wedge \neg q(b+x))\big]$

To SAT solver: $p_{a \leq b} \wedge p_{b \leq a+x} \wedge p_{x=0} \wedge \big[\neg p_{f(a)=f(b)} \vee (p_{q(a)} \wedge \neg p_{q(b+x)})\big]$

From propositional SAT to SMT

Reducing arbitrary boolean combinations to conjunctions



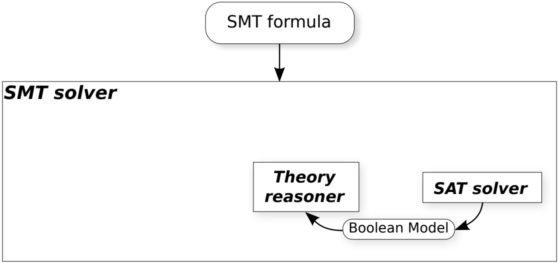
Input: $a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \left[f(a) \neq f(b) \vee (q(a) \wedge \neg q(b+x)) \right]$

To SAT solver: $p_{a \leq b} \wedge p_{b \leq a+x} \wedge p_{x=0} \wedge \left[\neg p_{f(a)=f(b)} \vee (p_{q(a)} \wedge \neg p_{q(b+x)}) \right]$

Boolean model: $p_{a \leq b}, p_{b \leq a+x}, p_{x=0}, \neg p_{f(a)=f(b)}$

From propositional SAT to SMT

Reducing arbitrary boolean combinations to conjunctions



Input: $a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \left[f(a) \neq f(b) \vee (q(a) \wedge \neg q(b+x))\right]$

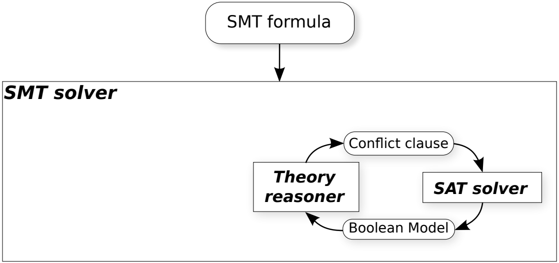
To SAT solver: $p_{a \leq b} \wedge p_{b \leq a+x} \wedge p_{x=0} \wedge \left[\neg p_{f(a)=f(b)} \vee (p_{q(a)} \wedge \neg p_{q(b+x)})\right]$

Boolean model: $p_{a \leq b}, p_{b \leq a+x}, p_{x=0}, \neg p_{f(a)=f(b)}$

Theory reasoner: $a \leq b, b \leq a + x, x = 0, f(a) \neq f(b)$ unsatisfiable

From propositional SAT to SMT

Reducing arbitrary boolean combinations to conjunctions



Input: $a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \left[f(a) \neq f(b) \vee (q(a) \wedge \neg q(b + x)) \right]$

To SAT solver: $p_{a \leq b} \wedge p_{b \leq a+x} \wedge p_{x=0} \wedge \left[\neg p_{f(a)=f(b)} \vee (p_{q(a)} \wedge \neg p_{q(b+x)}) \right]$

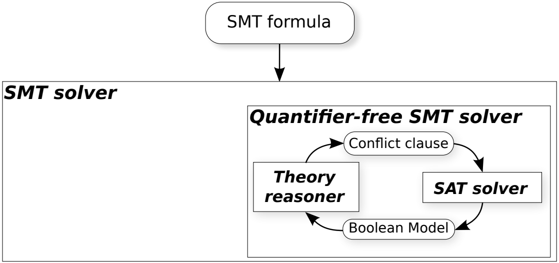
Boolean model: $p_{a \leq b}, p_{b \leq a+x}, p_{x=0}, \neg p_{f(a)=f(b)}$

Theory reasoner: $a \leq b, b \leq a + x, x = 0, f(a) \neq f(b)$ unsatisfiable

New clause: $\neg p_{a \leq b} \vee \neg p_{b \leq a+x} \vee \neg p_{x=0} \vee p_{f(a)=f(b)}$

From propositional SAT to SMT

Reducing arbitrary boolean combinations to conjunctions



Input: $a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \left[f(a) \neq f(b) \vee (q(a) \wedge \neg q(b+x))\right]$

To SAT solver: $p_{a \leq b} \wedge p_{b \leq a+x} \wedge p_{x=0} \wedge \left[\neg p_{f(a)=f(b)} \vee (p_{q(a)} \wedge \neg p_{q(b+x)})\right]$

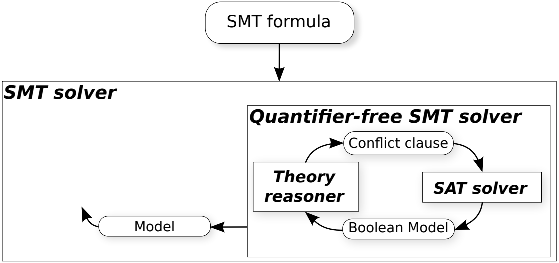
Boolean model: $p_{a \leq b}, p_{b \leq a+x}, p_{x=0}, \neg p_{f(a)=f(b)}$

Theory reasoner: $a \leq b, b \leq a + x, x = 0, f(a) \neq f(b)$ unsatisfiable

New clause: $\neg p_{a \leq b} \vee \neg p_{b \leq a+x} \vee \neg p_{x=0} \vee p_{f(a)=f(b)}$

From propositional SAT to SMT

Reducing arbitrary boolean combinations to conjunctions



Input: $a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \left[f(a) \neq f(b) \vee (q(a) \wedge \neg q(b+x)) \right]$

To SAT solver: $p_{a \leq b} \wedge p_{b \leq a+x} \wedge p_{x=0} \wedge \left[\neg p_{f(a)=f(b)} \vee (p_{q(a)} \wedge \neg p_{q(b+x)}) \right]$

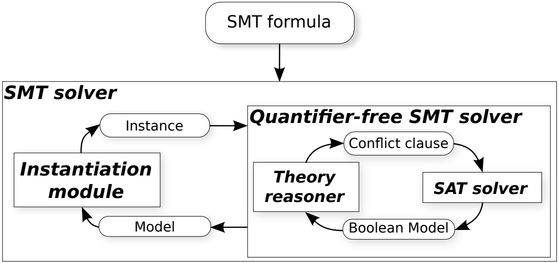
Boolean model: $p_{a \leq b}, p_{b \leq a+x}, p_{x=0}, \neg p_{f(a)=f(b)}$

Theory reasoner: $a \leq b, b \leq a + x, x = 0, f(a) \neq f(b)$ unsatisfiable

New clause: $\neg p_{a \leq b} \vee \neg p_{b \leq a+x} \vee \neg p_{x=0} \vee p_{f(a)=f(b)}$

From propositional SAT to SMT

Reducing arbitrary boolean combinations to conjunctions



Input: $a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \left[f(a) \neq f(b) \vee (q(a) \wedge \neg q(b + x)) \right]$

To SAT solver: $p_{a \leq b} \wedge p_{b \leq a+x} \wedge p_{x=0} \wedge \left[\neg p_{f(a)=f(b)} \vee (p_{q(a)} \wedge \neg p_{q(b+x)}) \right]$

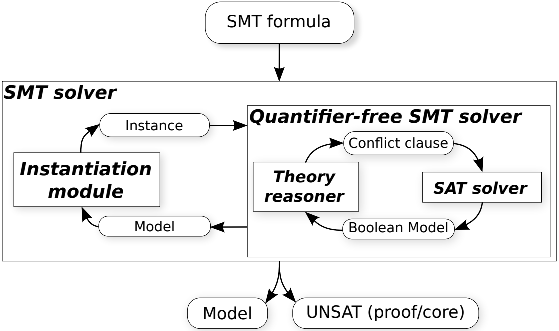
Boolean model: $p_{a \leq b}, p_{b \leq a+x}, p_{x=0}, \neg p_{f(a)=f(b)}$

Theory reasoner: $a \leq b, b \leq a + x, x = 0, f(a) \neq f(b)$ unsatisfiable

New clause: $\neg p_{a \leq b} \vee \neg p_{b \leq a+x} \vee \neg p_{x=0} \vee p_{f(a)=f(b)}$

From propositional SAT to SMT

Reducing arbitrary boolean combinations to conjunctions



Input: $a \leq b \wedge b \leq a + x \wedge x = 0 \wedge \left[f(a) \neq f(b) \vee (q(a) \wedge \neg q(b + x)) \right]$

To SAT solver: $p_{a \leq b} \wedge p_{b \leq a+x} \wedge p_{x=0} \wedge \left[\neg p_{f(a)=f(b)} \vee (p_{q(a)} \wedge \neg p_{q(b+x)}) \right]$

Boolean model: $p_{a \leq b}, p_{b \leq a+x}, p_{x=0}, \neg p_{f(a)=f(b)}$

Theory reasoner: $a \leq b, b \leq a + x, x = 0, f(a) \neq f(b)$ unsatisfiable

New clause: $\neg p_{a \leq b} \vee \neg p_{b \leq a+x} \vee \neg p_{x=0} \vee p_{f(a)=f(b)}$

From propositional SAT to SMT: in practice

- online decision procedures
  theory checks propositional assignment on the fly
- small explanations
  unsat core of propositional assignment
  discard classes of propositional assignments (not one by one)
- theory propagation
  instead of guessing propositional variable assignments, SAT solver assigns theory-entailed literals
- ackermannization, simplifications, and other magic

DPLL: abstract view

Rules handle a data-structure $M \mid\mid F$ where M is a partial assignment of Boolean variables, and F is a set of clauses

Propagate	$M \mid\mid F, C \vee \ell$	$\vdash$	$M\ \ell \mid\mid F, C \vee \ell$ if $M \models \neg C, \ell$ undefined in M
Decide	$M \mid\mid F$	$\vdash$	$M\ \ell^d \mid\mid F$ if ℓ or $\overline{\ell}$ in F, ℓ undefined in M
Fail	$M \mid\mid F, C$	$\vdash$	$\perp$ if $M \models \neg C,$ no decision literals in M
Backtrack	$M\ \ell^d\ N \mid\mid F, C$	$\vdash$	$M\ \bar{\ell} \mid\mid F, C$ if $\begin{cases} M\ \ell^d\ N \models \neg C \\ \text{no decision literals in } N \end{cases}$

CDCL: abstract view

Propagate, Decide, Fail as before

$$\text{Learn} \quad M \mid\mid F \quad \vdash \quad M \mid\mid F, C \quad \text{if } \begin{cases} \text{each atom of } C \text{ in } F \text{ or in } M \\ F \models C \end{cases}$$

$$\text{Backjump} \quad M\ \ell^d\ N \mid\mid F, C \quad \vdash \quad M\ \ell' \mid\mid F, C \quad \text{if } \begin{cases} M\ \ell^d\ N \models \neg C \\ \exists C', \ell' : F, C \models C' \vee \ell' \\ M \models \neg C' \\ \ell' \text{ undefined in } M \\ \ell' \text{ or } \overline{\ell'} \text{ in } F \text{ or in } M\ \ell^d\ N \end{cases}$$

$$\text{Forget} \quad M \mid\mid F, C \quad \vdash \quad M \mid\mid F \quad \text{if } F \models C$$

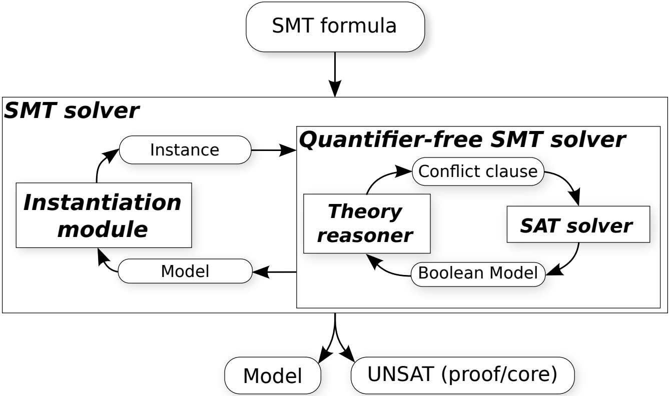
CDCL: SMT level

$$T\text{–Learn} \quad M \mid\mid F \vdash M \mid\mid F, C \quad \text{if } \begin{cases} \text{each atom of } C \text{ in } F \text{ or in } M \\ F \models_T C \end{cases}$$

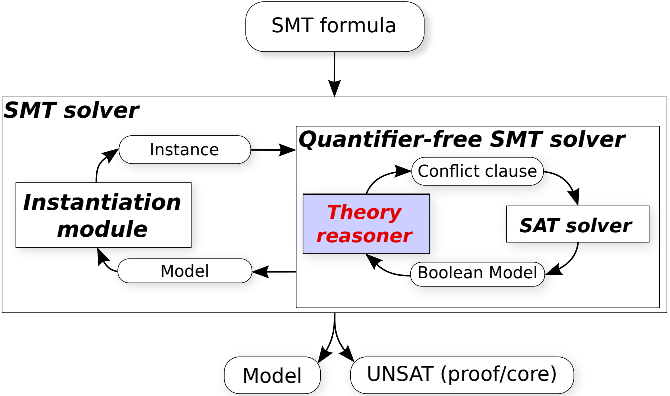
$$T\text{–Forget} \quad M \mid\mid F, C \vdash M \mid\mid F \quad \text{if } F \models_T C$$

$$T\text{–Backjump} \quad M\ l^d\ N \mid\mid F, C \vdash M\ l' \mid\mid F, C \quad \text{if } \begin{cases} M\ l^d\ N \models \neg C \\ \exists C', l' : F, C \models_T C' \vee l' \\ M \models \neg C' \\ l' \text{ undefined in } M \\ l' \text{ or } \neg l' \text{ in } F \text{ or in } M\ l^d\ N \end{cases}$$

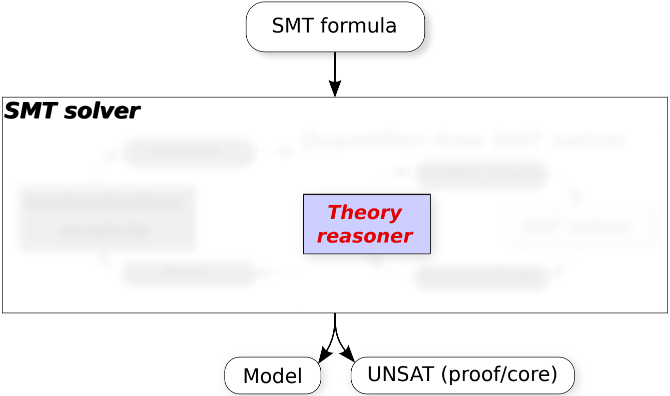
Theory reasoning and combinations in SMT



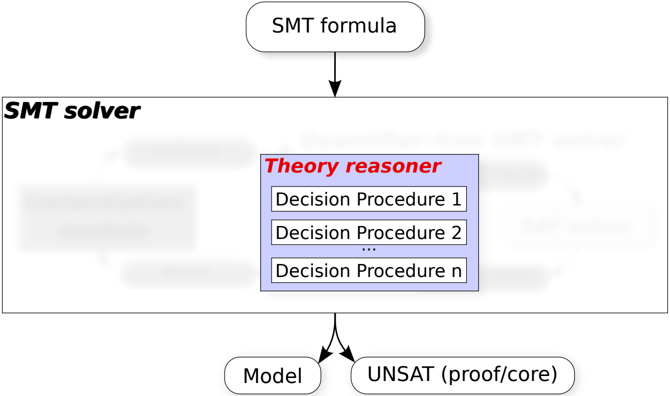
Theory reasoning and combinations in SMT



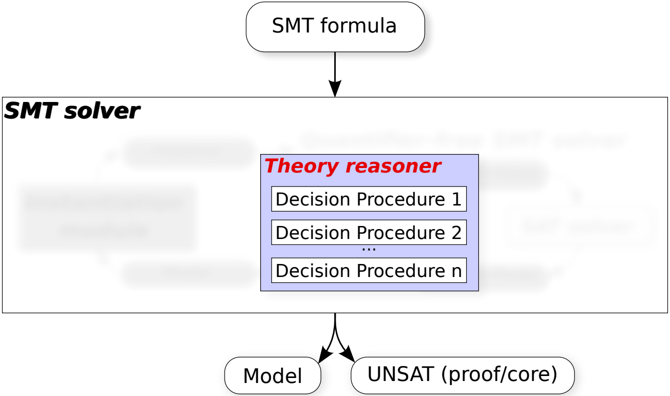
Theory reasoning and combinations in SMT



Theory reasoning and combinations in SMT



Theory reasoning and combinations in SMT



Some examples:

- uninterpreted symbols and equality: congruence closure
- linear arithmetic: mostly simplex
- non-linear arithmetic: CAD, Virtual Substitution, Gröbner Bases, Interval Propagation
- arrays: based on uninterpreted symbols

Combinations of theories

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \leq y,\ y \leq x + f(x),\ P(h(x) - h(y)), \neg P(0),\ f(x) = 0$$

Combinations of theories

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \leq y,\; y \leq x + f(x),\; P(h(x) - h(y)), \neg P(0),\; f(x) = 0$$

Separate into pure literals

Combinations of theories

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \leq y,\; y \leq x + f(x),\; P(h(x) - h(y)), \neg P(0),\; f(x) = 0$$

Separate into pure literals

Arithmetic	Uninterpreted
$x \leq y$	$P(v_2)$
$y \leq x + v_1$	$\neg P(v_5)$
$v_1 = 0$	$v_1 = f(x)$
$v_2 = v_3 - v_4$	$v_3 = h(x)$
$v_5 = 0$	$v_4 = h(y)$

Combinations of theories

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \leq y,\ y \leq x + f(x),\ P(h(x) - h(y)), \neg P(0),\ f(x) = 0$$

Exchange equalities

Arithmetic	Uninterpreted
$x \leq y$	$P(v_2)$
$y \leq x + v_1$	$\neg P(v_5)$
$v_1 = 0$	$v_1 = f(x)$
$v_2 = v_3 - v_4$	$v_3 = h(x)$
$v_5 = 0$	$v_4 = h(y)$

Combinations of theories

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \le y,\; y \le x + f(x),\; P(h(x) - h(y)), \neg P(0),\; f(x) = 0$$

Exchange equalities

Arithmetic	Uninterpreted
$x \le y$	$P(v_2)$
$y \le x + v_1$	$\neg P(v_5)$
$v_1 = 0$	$v_1 = f(x)$
$v_2 = v_3 - v_4$	$v_3 = h(x)$
$v_5 = 0$	$v_4 = h(y)$
	$x = y$

Combinations of theories

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \leq y,\ y \leq x + f(x),\ P(h(x) - h(y)), \neg P(0),\ f(x) = 0$$

Exchange equalities

Arithmetic	Uninterpreted
$x \leq y$	$P(v_2)$
$y \leq x + v_1$	$\neg P(v_5)$
$v_1 = 0$	$v_1 = f(x)$
$v_2 = v_3 - v_4$	$v_3 = h(x)$
$v_5 = 0$	$v_4 = h(y)$
$v_3 = v_4$	$x = y$

Combinations of theories

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \leq y,\ y \leq x + f(x),\ P(h(x) - h(y)), \neg P(0),\ f(x) = 0$$

Exchange equalities

Arithmetic	Uninterpreted
$x \leq y$	$P(v_2)$
$y \leq x + v_1$	$\neg P(v_5)$
$v_1 = 0$	$v_1 = f(x)$
$v_2 = v_3 - v_4$	$v_3 = h(x)$
$v_5 = 0$	$v_4 = h(y)$
$v_3 = v_4$	$x = y$
	$v_2 = v_5$

Combinations of theories

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \leq y,\ y \leq x + f(x),\ P(h(x) - h(y)), \neg P(0),\ f(x) = 0$$

Exchange equalities until unsatisfiability is deduced

Arithmetic	Uninterpreted
$x \leq y$	$P(v_2)$
$y \leq x + v_1$	$\neg P(v_5)$
$v_1 = 0$	$v_1 = f(x)$
$v_2 = v_3 - v_4$	$v_3 = h(x)$
$v_5 = 0$	$v_4 = h(y)$
$v_3 = v_4$	$x = y$
	$v_2 = v_5$

UNSAT

Combinations of theories

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \le y,\ y \le x + f(x),\ P(h(x) - h(y)), \neg P(0),\ f(x) = 0$$

Exchange equalities until unsatisfiability is deduced

Arithmetic	Uninterpreted
$x \le y$	$P(v_2)$
$y \le x + v_1$	$\neg P(v_5)$
$v_1 = 0$	$v_1 = f(x)$
$v_2 = v_3 - v_4$	$v_3 = h(x)$
$v_5 = 0$	$v_4 = h(y)$
$v_3 = v_4$	$x = y$
	$v_2 = v_5$

UNSAT

Sound: deduce only logical consequences

Combinations of theories

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \leq y,\ y \leq x + f(x),\ P(h(x) - h(y)), \neg P(0),\ f(x) = 0$$

Exchange equalities until unsatisfiability is deduced

Arithmetic	Uninterpreted
$x \leq y$	$P(v_2)$
$y \leq x + v_1$	$\neg P(v_5)$
$v_1 = 0$	$v_1 = f(x)$
$v_2 = v_3 - v_4$	$v_3 = h(x)$
$v_5 = 0$	$v_4 = h(y)$
$v_3 = v_4$	$x = y$
	$v_2 = v_5$

UNSAT

Sound: deduce only logical consequences

Complete: decidable theories with cardinality restrictions

Combinations of theories (1/2)

Nelson-Oppen

Combining theories: uninterpreted symbols and arithmetic.

$$x \leq y,\ y \leq x + f(x),\ P(h(x) - h(y)), \neg P(0),\ f(x) = 0$$

Exchange equalities until unsatisfiability is deduced

Arithmetic	Uninterpreted
$x \leq y$	$P(v_2)$
$y \leq x + v_1$	$\neg P(v_5)$
$v_1 = 0$	$v_1 = f(x)$
$v_2 = v_3 - v_4$	$v_3 = h(x)$
$v_5 = 0$	$v_4 = h(y)$
$v_3 = v_4$	$x = y$
	$v_2 = v_5$

UNSAT

Sound: deduce only logical consequences

Complete: decidable theories with cardinality restrictions

Combinations of theories (2/2)

Nelson-Oppen

Non linear arithmetic is also stably infinite.

Uninterpreted symbols and non linear arithmetic:

$$x^2 = 1, P(x), \neg P(-1), \neg P(1)$$

Combinations of theories (2/2)

Nelson-Oppen

Non linear arithmetic is also stably infinite.
Uninterpreted symbols and non linear arithmetic:

$$x^2 = 1, P(x), \neg P(-1), \neg P(1)$$

Separate into pure literals

Combinations of theories (2/2)

Nelson-Oppen

Non linear arithmetic is also stably infinite.
Uninterpreted symbols and non linear arithmetic:

$$x^2 = 1, P(x), \neg P(-1), \neg P(1)$$

Separate into pure literals

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$

Combinations of theories (2/2)

Nelson-Oppen

Non linear arithmetic is also stably infinite.
Uninterpreted symbols and non linear arithmetic:

$$x^2 = 1, P(x), \neg P(-1), \neg P(1)$$

Exchange equalities

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$

Combinations of theories (2/2)

Nelson-Oppen

Non linear arithmetic is also stably infinite.
Uninterpreted symbols and non linear arithmetic:

$$x^2 = 1, P(x), \neg P(-1), \neg P(1)$$

Exchange disjunctions of equalities

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$

Combinations of theories (2/2)

Nelson-Oppen

Non linear arithmetic is also stably infinite.
Uninterpreted symbols and non linear arithmetic:

$$x^2 = 1, P(x), \neg P(-1), \neg P(1)$$

Exchange disjunctions of equalities: unpractical

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$

Combinations of theories (2/2)

Nelson-Oppen

Non linear arithmetic is also stably infinite.
Uninterpreted symbols and non linear arithmetic:

$$x^2 = 1, P(x), \neg P(-1), \neg P(1)$$

Exchange disjunctions of equalities: unpractical

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$

For non-convex theories, disjunctions have to be exchanged
Even deducing equalities is unpractical with a black box

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get "model" if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$

On SAT, get a model from NLRA

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x = 1, v_1 = 1, v_2 = -1$	

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get "model" if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$

Pretend equalities in the model were in the input

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x = 1, v_1 = 1, v_2 = -1$	$x = v_1$

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get "model" if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$

Compute conflict clause

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x = 1, v_1 = 1, v_2 = -1$	$x = v_1$

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get "model" if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$
$\neg P(x) \vee x \neq 1 \vee P(1)$

Add conflict clause to underlying SAT solver

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x = 1, v_1 = 1, v_2 = -1$	$x = v_1$

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get "model" if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$
$\neg P(x) \vee x \neq 1 \vee P(1)$

Update literals

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x \neq 1$	

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get "model" if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$
$\neg P(x) \vee x \neq 1 \vee P(1)$

Get a model from NLRA (again)

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x \neq 1$	
$x = -1, v_1 = 1, v_2 = -1$	

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get "model" if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$
$\neg P(x) \vee x \neq 1 \vee P(1)$

Pretend equalities in the model were in the input (again)

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x \neq 1$	$x = v_2$
$x = -1, v_1 = 1, v_2 = -1$	

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get "model" if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$
$\neg P(x) \vee x \neq 1 \vee P(1)$

Compute conflict clause (again)

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x \neq 1$	$x = v_2$
$x = -1, v_1 = 1, v_2 = -1$	

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get “model” if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$
$\neg P(x) \vee x \neq 1 \vee P(1)$
$\neg P(x) \vee x \neq -1 \vee P(-1)$

Add conflict clause to underlying SAT solver (again)

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x \neq 1$	$x = v_2$
$x = -1, v_1 = 1, v_2 = -1$	

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get "model" if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$
$\neg P(x) \vee x \neq 1 \vee P(1)$
$\neg P(x) \vee x \neq -1 \vee P(-1)$

Update literals (again)

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x \neq 1$	$x = v_2$
$x \neq -1$	

Combining theories: using model equalities

Open the box a bit: besides (un)sat, get "model" if sat

$x^2 = 1, P(x), \neg P(-1), \neg P(1)$
$\neg P(x) \vee x \neq 1 \vee P(1)$
$\neg P(x) \vee x \neq -1 \vee P(-1)$
$x = 1 \vee x = -1 \vee x^2 = 1$

Conclude unsatisfiability (finally)

Arithmetic	Uninterpreted
$x^2 = 1$	$P(x)$
$v_1 = 1$	$\neg P(v_1)$
$v_2 = -1$	$\neg P(v_2)$
$x \neq 1$	$x = v_2$
$x \neq -1$	

Perspectives

- Quantifiers: better instantiations, superposition+SMT
- Higher-order
- More theories: data-structures, floating points. . .
- Higher efficiency
- Parallelism