SMT solvers, tools of trade in formal methods

Damien Zufferey

IST Austria

March 17, 2017

1. Introduction
2. Formalism
    - General Concepts
    - First Order Theories
3. Algorithm
    - Propositional Logic
    - Equality with Uninterpreted Function symbols
    - Difference Logic
    - Linear Arithmetic
4. SMT Solver
    - Combining Theories
    - CQFF(T) to QFF(T)

Outline

1. Introduction
2. Formalism
3. Algorithm
4. SMT Solver

What are SMT solvers ?

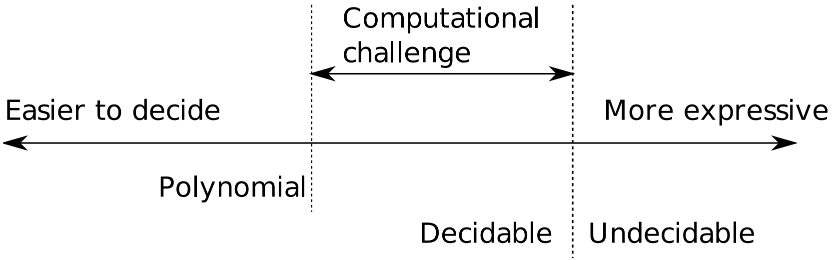
SMT Solver are tools that tell if a given formula has some solution.

For instance:

$$\begin{gathered} p = f(x+a) \ \wedge \ q = f(y+b) \ \wedge \ a = b \ \wedge \\ s = f(p+c) \ \wedge \ t = f(q+d) \ \wedge \ c = d \ \wedge \\ 1 = s - t + z \ \wedge \ x = y \ \wedge \ z = 0 \end{gathered}$$

in unsatisfiable.

Challenges


Computational challenge
Easier to decide
More expressive
Polynomial
Decidable
Undecidable

Outline

1. Introduction
2. Formalism
   - General Concepts
   - First Order Theories
3. Algorithm
4. SMT Solver

Propositional Logic (PL)

Also known as boolean logic.

Syntax

$F :: \quad F \wedge F \ \mid \ F \vee F \ \mid \ \neg F \ \mid \ \top \ \mid \ \bot \ \mid \ \textit{propositional variable}$

Other operators $(\rightarrow, \leftrightarrow, \oplus)$ are syntactic sugar.

Semantics

An interpretation I is an assignment of the propositional variables to either $\top$ or $\bot$, i.e. $I = \{P \mapsto \top, Q \mapsto \bot, \ldots\}$

Propositional Logic: example

You need to schedule 3 talks given by 3 different speakers with their own availability.

Create 9 variables x_{ws} $(w \in 1..3, s \in 1..3)$.
For each speaker s, add $\neg x_{ws}$ where w corresponds to the dates where s in not available.
For each speaker s, add $x_{is} \rightarrow \neg x_{js} \wedge \neg x_{ks}$ with i, j, k all different.
For each week w, add $x_{w1} \vee x_{w2} \vee x_{w3}$.
For each week w, add $x_{wi} \rightarrow \neg x_{wj} \wedge \neg x_{wk}$ with i, j, k all different.

First Order Logic (FOL)

Syntax

$$\begin{array}{ll} T :: & \textit{constants} \mid \textit{variables} \mid \textit{functions} \\ P :: & \textit{predicate} \mid \textit{propositional variables} \mid \top \mid \bot \\ F :: & P \mid F \wedge F \mid F \vee F \mid \neg F \mid \exists x.F[x] \mid \forall x.F[x] \end{array}$$

Example: $\forall x.p(f(x), x) \to (\exists y.p(g(x, y), g(y, x)))$

Semantics

An interpretation $I = \langle D, \alpha \rangle$ is a pair domain, assignment. D is a non-empty set of values. α maps variables and constants to elements of D, n-ary functions to functions over $D^n \to D$, and n-ary predicates to predicates over $D^n \to \{\textit{true}, \textit{false}\}$.

Interpretations are also known as models.

Quantifiers and free variables

Free variables are either universally or existentially quantified, depending on the problem we are solving:

- The universal closure ($\forall$) for the validity problem.
- The existential closure ($\exists$) for the satisfiability problem.

First Order Theories

Definition

A theory $T = \langle \Sigma, \mathcal{A} \rangle$ is a pair signature, axioms.

- Σ is a set of constants, functions and predicates symbols.
- $\mathcal{A}$ is a set of closed FOL formula over the elements of Σ.

The quantifier-free fragment of a theory (QFF) is a syntactic restriction that prevents using quantifiers in formulas.
The conjunctive QFF (CQFF) is the fragment where formulas are only conjunctions.

Equality with Uninterpreted Function symbols (EUF)

Example: $f(f(f(f(f(a))))) = a \wedge f(f(f(a))) = a \wedge f(a) \neq a$

Signature: $\Sigma_{EUF} = \{=, a, b, c, \ldots, f, g, h, \ldots, p, q, r, \ldots\}$
Axioms:

1. $\forall x. x = x$ (reflexivity)
2. $\forall x, y. x = y \rightarrow y = x$ (symmetry)
3. $\forall x, y, z. x = y \wedge y = z \rightarrow x = z$ (transitivity)
4. for all n-ary function symbol f:
   $\forall \vec{x}, \vec{y}. (\bigwedge_{i=1}^{n} x_i = y_i) \rightarrow f(\vec{x}) = f(\vec{y})$ (function congruence)
5. for all n-ary predicates symbol p:
   $\forall \vec{x}, \vec{y}. (\bigwedge_{i=1}^{n} x_i = y_i) \rightarrow p(\vec{x}) \leftrightarrow p(\vec{y})$ (predicate congruence)

Presburger Arithmetic($\mathbb{N}$), Theory of Integers ($\mathbb{Z}$)

Example: $\forall w, x.\ \exists y, z.\ x + 2y - z - 13 > -3w + 5$

Signature: $\Sigma_{\mathbb{N}} = \{0, 1, +, =\}$
Axioms:

1. $\forall x. \neg(x + 1 = 0)$ (zero)
2. $\forall x, y. x + 1 = y + 1 \rightarrow x = y$ (successor)
3. $F[0] \wedge (\forall x. F[x] \rightarrow F[x + 1]) \rightarrow \forall x. F[x]$ (induction)
4. $\forall x. x + 0 = x$ (plus zero)
5. $\forall x, y. x + (y + 1) = (x + y) + 1$ (plus successor)

Theory of Reals ($\mathbb{R}$), Theory of Rationals ($\mathbb{Q}$)

Signature: $\Sigma_{\mathbb{R}} = \{0, 1, +, \cdot, =, \geq\}$
Axioms: ...

Signature: $\Sigma_{\mathbb{Q}} = \{0, 1, +, -, =, \geq\}$
Axioms: ...

Linear Arithmetic (LA), Difference Logic (DL)

LA and DL are fragments of the theories of $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$.

LA has terms of the form $\sum_i a_i x_i \geq b$.
e.g. $3x + 2y \leq 5z \wedge 2x - 2y = 0$

DL has terms of the form $x - y \geq c$.
e.g. $x < y + 5 \wedge y \leq 4 \wedge x = z - 1$

Outline

1. Introduction
2. Formalism
3. Algorithm
   - Propositional Logic
   - Equality with Uninterpreted Function symbols
   - Difference Logic
   - Linear Arithmetic
4. SMT Solver

DPLL: definition

We are searching a solution for $P \wedge (\neg P \vee Q) \wedge (R \vee \neg Q \vee S)$.

Assumption: formula in conjunctive normal form (CNF): $\bigwedge_i \bigvee_j x_{ij}$.
A literal is a variable or its negation.
A disjunction of literals is a clause.

An unit clause is a clause containing only one literal.
To satisfy the unit clause (P), P has to be assigned to true.

DPLL: algorithm

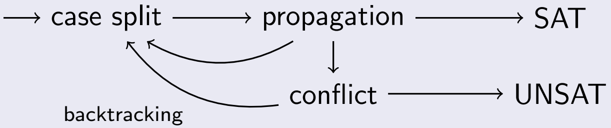
unit resolution (boolean constraint propagation):

$$\frac{l \quad C[\neg l]}{C[\bot]}$$

case splitting:

$$F[x] \leftrightarrow F[\bot] \vee F[\top]$$

Algorithm



DPLL: learning

While backtracking it is possible to learn new clauses by resolution:

$$\frac{P \vee Q \quad \neg P \vee R}{Q \vee R}$$

Example: $(P \vee Q) \wedge (\neg P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee \neg Q)$.

DPLL: example

$$(P \vee Q) \wedge (\neg P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee \neg Q)$$

DPLL: example

$$P \mapsto \top$$

$$(\quad Q) \wedge \quad (\quad \neg Q)$$

DPLL: example

$$Q \mapsto \top$$

$$(\qquad \bot)$$

DPLL: example

backtracking

$$(P \vee Q) \wedge (\neg P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee \neg Q)$$

$$\cfrac{\cfrac{(\neg P \vee Q) \quad (\neg P \vee \neg Q)}{\neg P}}{}$$

DPLL: example

$$P \mapsto \bot$$

$$(\quad Q) \wedge \qquad (\quad \neg Q)$$

$$\cfrac{\cfrac{(\neg P \vee Q) \quad (\neg P \vee \neg Q)}{\neg P}}{}$$

DPLL: example

$$Q \mapsto \top$$

$$(\qquad \bot)$$

$$\cfrac{\cfrac{(\neg P \vee Q) \quad (\neg P \vee \neg Q)}{\neg P}}{}$$

DPLL: example

backtracking

$$(P \vee Q) \wedge (\neg P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee \neg Q)$$

$$\dfrac{\dfrac{(P \vee Q) \quad (P \vee \neg Q)}{P} \qquad \dfrac{(\neg P \vee Q) \quad (\neg P \vee \neg Q)}{\neg P}}{\perp}$$

DPLL: Decision policy

Most of the generated sat problems are structured. The goal of a SAT solver is to quickly figure out what is important. The role of the decision policy is to guess which variables are important.

A good decision policy and learning is the key to scaling to problems with thousands of variables.

EUF: Example

$$f(f(f(f(f(a))))) = a \wedge f(f(f(a))) = a \wedge f(a) \neq a$$

Satisfiable or not ?

EUF: Example

$$f(f(f(f(f(a))))) = a \wedge f(f(f(a))) = a \wedge f(a) \neq a$$

Satisfiable or not ?

- $f(f(a)) = a$

EUF: Example

$$f(f(f(f(f(a))))) = a \wedge f(f(f(a))) = a \wedge f(a) \neq a$$

Satisfiable or not ?

- $f(f(a)) = a$
- $f(a) = a$

EUF: Example

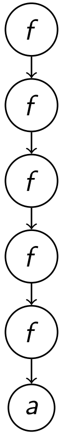
$$f(f(f(f(f(a))))) = a \wedge f(f(f(a))) = a \wedge f(a) \neq a$$

Satisfiable or not ?

- $f(f(a)) = a$
- $f(a) = a$

EUF: Congruence Closure

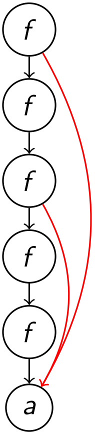
DAG representing the terms:
$\{a, f(a), f(f(a)), f^3(a), f^4(a), f^5(a)\}$



EUF: Congruence Closure

DAG representing the terms:
$\{a, f(a), f(f(a)), f^3(a), f^4(a), f^5(a)\}$

Union-find data structure:
The nodes keep a pointer to the representative of their equivalence class.

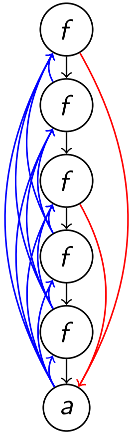


EUF: Congruence Closure

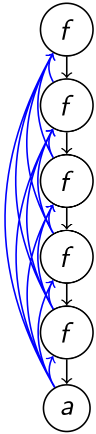
DAG representing the terms:
$\{a, f(a), f(f(a)), f^3(a), f^4(a), f^5(a)\}$

Union-find data structure:
The nodes keep a pointer to the representative of their equivalence class.

The representative of an equivalence class keeps pointers to its congruence closure parents.

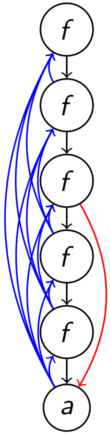


Congruence Closure: example



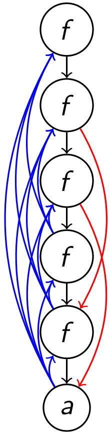
Congruence Closure: example

- adding $f^3(a) = a$



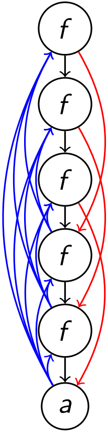
Congruence Closure: example

- adding $f^3(a) = a$
- congruence $f^4(a) = f(a)$



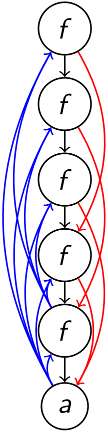
Congruence Closure: example

- adding $f^3(a) = a$
- congruence $f^4(a) = f(a)$
- congruence $f^5(a) = f^2(a)$



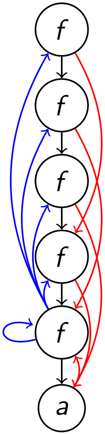
Congruence Closure: example

- adding $f^3(a) = a$
- congruence $f^4(a) = f(a)$
- congruence $f^5(a) = f^2(a)$
- adding $f^5(a) = a$



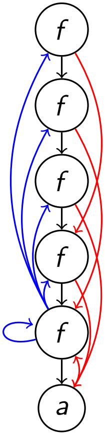
Congruence Closure: example

- adding $f^3(a) = a$
- congruence $f^4(a) = f(a)$
- congruence $f^5(a) = f^2(a)$
- adding $f^5(a) = a$
- congruence $f^3(a) = f(a)$



Congruence Closure: example

- adding $f^3(a) = a$
- congruence $f^4(a) = f(a)$
- congruence $f^5(a) = f^2(a)$
- adding $f^5(a) = a$
- congruence $f^3(a) = f(a)$
- conflict with $f(a) \neq a$



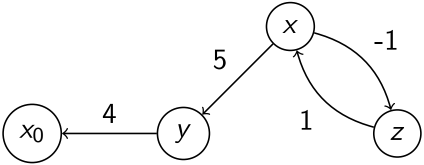
Difference Bound Matrices (1)

$x \leq y + 5 \ \wedge \ y \leq 4 \ \wedge \ x = z - 1$

rewritten as a DL formula:

$x - y \leq 5 \ \wedge \ y - x_0 \leq 4 \ \wedge \ x - z \leq -1 \ \wedge \ z - x \leq 1$

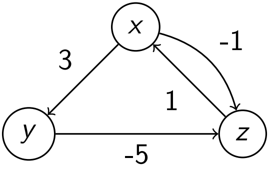
as a graph:



Difference Bound Matrices (2)

$$x - y \leq 3 \ \wedge \ y - z \leq -5 \ \wedge \ x - z \leq -1 \ \wedge \ z - x \leq 1$$

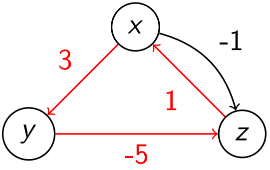
as a graph:



Difference Bound Matrices (2)

$x - y \leq 3 \ \wedge \ y - z \leq -5 \ \wedge \ x - z \leq -1 \ \wedge \ z - x \leq 1$

as a graph:

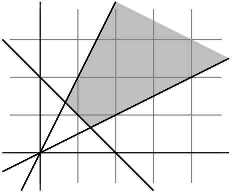


$x - y + y - z + z - x \leq 3 - 5 + 1 \quad \leftrightarrow \quad 0 \leq -1$

The formula is satisfiable iff there is no negative cycle.

Simplex ($\mathbb{Q}, \mathbb{R}$)

$$2x - y \geq 0 \ \wedge \ -x + 2y \geq 0 \ \wedge \ x + y \geq 2$$



Wlog such a problem can be written as $A\vec{x} \geq \vec{b}$.

Introducing one slack variable per constraint we get:
$A'\vec{x}' = 0 \ \bigwedge_{i=1}^{m} l_i \leq s_i \leq u_i$ where $A' = [AI_m], \vec{x}' = [\vec{x}\vec{s}]$

Simplex $(\mathbb{Q}, \mathbb{R})$

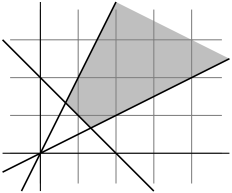
$$\begin{array}{rcll} 2x - y & \geq & 0 & \wedge \\ -x + 2y & \geq & 0 & \wedge \\ x + y & \geq & 2 & \end{array} \quad \Longrightarrow \quad \begin{array}{rcll} 2x - y - s_1 & = & 0 & \wedge \\ -x + 2y - s_2 & = & 0 & \wedge \\ x + y - s_3 & = & 0 & \wedge \\ s_1 & \geq & 0 & \wedge \\ s_2 & \geq & 0 & \wedge \\ s_3 & \geq & 2 & \end{array}$$

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \\ 1 & 1 \end{pmatrix} \quad \Longrightarrow \quad A' = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 & -1 \end{pmatrix}$$

Simplex $(\mathbb{Q}, \mathbb{R})$

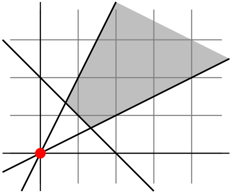
$$\begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{aligned} s_1 &\geq 0 \\ s_2 &\geq 0 \\ s_3 &\geq 2 \end{aligned}$$



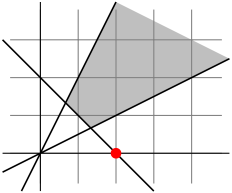
Simplex ($\mathbb{Q}, \mathbb{R}$)

$$\begin{pmatrix} & -1 & 0 & 0 \\ & 0 & -1 & 0 \\ & 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad \begin{array}{lcl} s_1 & \geq & 0 \\ s_2 & \geq & 0 \\ s_3 & \geq & 2 \end{array}$$



Simplex ($\mathbb{Q}$, $\mathbb{R}$)

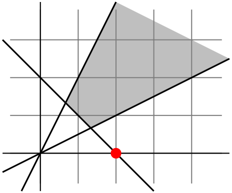
$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} \qquad \begin{array}{rcl} s_1 & \geq & 0 \\ s_2 & \geq & 0 \\ s_3 & = & 2 \end{array}$$



Simplex $(\mathbb{Q}, \mathbb{R})$

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$$

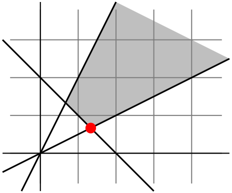
$$\begin{array}{ccc} s_1 & \geq & 0 \\ s_2 & \geq & 0 \\ s_3 & = & 2 \end{array}$$



Simplex $(\mathbb{Q}, \mathbb{R})$

$$\begin{pmatrix} 2 & -1 & -1 & & & \\ -1 & 2 & 0 & & & \\ 1 & 1 & 0 & & & \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$$

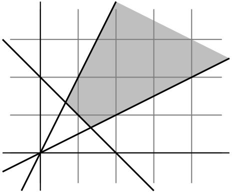
$$\begin{array}{lcl} s_1 & \geq & 0 \\ s_2 & = & 0 \\ s_3 & = & 2 \end{array}$$



Simplex ($\mathbb{N}$)

Branch-and-bound method:

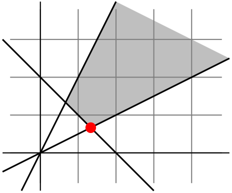
- solve the *relaxed* linear problem (solution in $\mathbb{R}^n$)
- *branch* on non-integral variables ($\leq \lfloor x \rfloor \;\vee\; \lceil x \rceil \leq$)



Simplex ($\mathbb{N}$)

Branch-and-bound method:

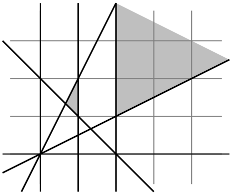
- solve the *relaxed* linear problem (solution in $\mathbb{R}^n$)
- *branch* on non-integral variables ($\leq \lfloor x \rfloor \ \vee \ \lceil x \rceil \leq$)



Simplex ($\mathbb{N}$)

Branch-and-bound method:

- solve the *relaxed* linear problem (solution in $\mathbb{R}^n$)
- *branch* on non-integral variables ($\leq \lfloor x \rfloor \ \vee \ \lceil x \rceil \leq$)



Outline

1 Introduction

2 Formalism

3 Algorithm

4 SMT Solver
- Combining Theories
- CQFF(T) to QFF(T)

Nelson-Oppen ($T_1 + T_2$): requirements

Idea

T_1, T_2 share the '$=$' symbol. Propagating equality constraints across the theories is sufficient to derive contradictions.

Requirements:

- T_1, T_2 are quantifier-free first-order theories with equality.
- $\Sigma_1 \cap \Sigma_2 = \{=\}$
- There are decision procedure for T_1 and T_2.
- T_1, T_2 are interpreted over an infinite domain (stably infinite).
- *optionally* T_1, T_2 are convex theories.

Nelson-Oppen ($T_1 + T_2$): convex theory

Consider a CQF formula F and a disjunction $\bigvee_{i=1}^{n} u_i = v_i$.
The theory T is convex if

$$\left(F \rightarrow \bigvee_{i=1}^{n} u_i = v_i \right) \rightarrow (F \rightarrow u_k = v_k) \text{ for some } k \in \{1..n\}$$

Nelson-Oppen ($T_1 + T_2$): purification

$$f(x_1, 0) \geq x_3 \ \wedge \ f(x_2, 0) \leq x_3 \ \wedge$$
$$x_1 \geq x_2 \ \wedge \ x_2 \geq x_1 \ \wedge \ x_3 - f(x_1, 0) \geq 1$$

F_1 (LA($\mathbb{Q}$))	F_2 (EUF)
$a_1 \geq x_3$	$a_1 = f(x_1, a_0)$
$a_2 \leq x_3$	$a_2 = f(x_2, a_0)$
$x_1 \geq x_2$	
$x_2 \geq x_1$	
$x_3 - a_1 \geq 1$	
$a_0 = 0$	

Nelson-Oppen ($T_1 + T_2$): equality propagation

F_1 (LA($\mathbb{Q}$))	F_2 (EUF)
$a_1 \geq x_3$	$a_1 = f(x_1, a_0)$
$a_2 \leq x_3$	$a_2 = f(x_2, a_0)$
$x_1 \geq x_2$	
$x_2 \geq x_1$	
$x_3 - a_1 \geq 1$	
$a_0 = 0$	

Nelson-Oppen ($T_1 + T_2$): equality propagation

F_1 (LA($\mathbb{Q}$))		F_2 (EUF)
$a_1 \geq x_3$		$a_1 = f(x_1, a_0)$
$a_2 \leq x_3$		$a_2 = f(x_2, a_0)$
$x_1 \geq x_2$		
$x_2 \geq x_1$		
$x_3 - a_1 \geq 1$		
$a_0 = 0$		
$x_1 = x_2$	$\Rightarrow$	$x_1 = x_2$

Nelson-Oppen ($T_1 + T_2$): equality propagation

F_1 (LA($\mathbb{Q}$))		F_2 (EUF)
$a_1 \geq x_3$		$a_1 = f(x_1, a_0)$
$a_2 \leq x_3$		$a_2 = f(x_2, a_0)$
$x_1 \geq x_2$		
$x_2 \geq x_1$		
$x_3 - a_1 \geq 1$		
$a_0 = 0$		
$x_1 = x_2$		$x_1 = x_2$
$a_1 = a_2$	$\Leftarrow$	$a_1 = a_2$

Nelson-Oppen ($T_1 + T_2$): equality propagation

F_1 (LA($\mathbb{Q}$))	F_2 (EUF)
$a_1 \geq x_3$	$a_1 = f(x_1, a_0)$
$a_2 \leq x_3$	$a_2 = f(x_2, a_0)$
$x_1 \geq x_2$	
$x_2 \geq x_1$	
$x_3 - a_1 \geq 1$	
$a_0 = 0$	
$x_1 = x_2$	$x_1 = x_2$
$a_1 = a_2$	$a_1 = a_2$
$a_1 = x_3$	

Nelson-Oppen ($T_1 + T_2$): equality propagation

F_1 (LA($\mathbb{Q}$))	F_2 (EUF)
$a_1 \geq x_3$	$a_1 = f(x_1, a_0)$
$a_2 \leq x_3$	$a_2 = f(x_2, a_0)$
$x_1 \geq x_2$	
$x_2 \geq x_1$	
$x_3 - a_1 \geq 1$	
$a_0 = 0$	
$x_1 = x_2$	$x_1 = x_2$
$a_1 = a_2$	$a_1 = a_2$
$a_1 = x_3$	

DPLL + T: Propositional skeleton of a formula

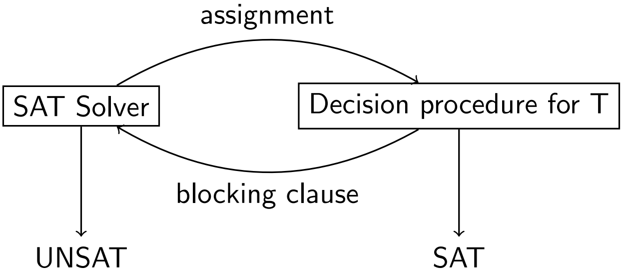
$$x = y \wedge (x = z \vee (y = z \wedge x \neq z))$$

$$\Downarrow$$

$$a \wedge (b \vee (c \wedge \neg b))$$

where $a \mapsto (x = y), b \mapsto (x = z), c \mapsto (y = z)$

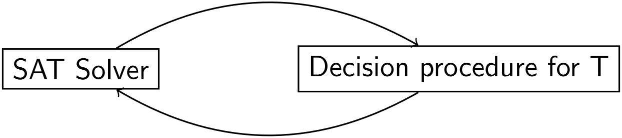
DPLL + T: Idea


assignment
SAT Solver
Decision procedure for T
blocking clause
UNSAT
SAT

DPLL + T: example

$$a \wedge (b \vee (c \wedge \neg b))$$

where $a \mapsto (x = y), b \mapsto (x = z), c \mapsto (y = z)$

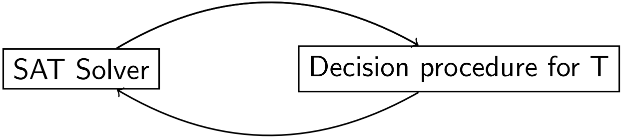


DPLL + T: example

$$a \wedge (b \vee (c \wedge \neg b))$$

where $a \mapsto (x = y), b \mapsto (x = z), c \mapsto (y = z)$

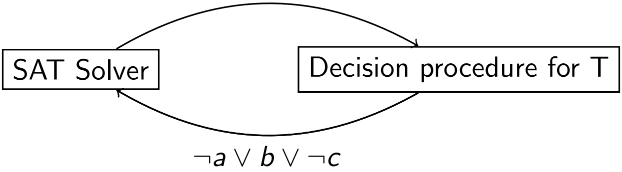
$$a \wedge \neg b \wedge c \quad \mapsto \quad x = y \wedge y = z \wedge x \neq z$$



DPLL + T: example

$$a \wedge (b \vee (c \wedge \neg b))$$

where $a \mapsto (x = y), b \mapsto (x = z), c \mapsto (y = z)$

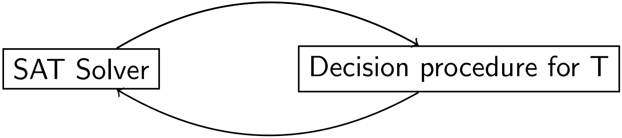


DPLL + T: example

$$a \wedge (b \vee (c \wedge \neg b))$$

where $a \mapsto (x = y), b \mapsto (x = z), c \mapsto (y = z)$

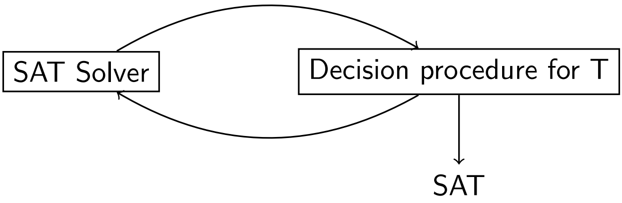
$$a \wedge b \quad \mapsto \quad x = y \wedge x = z$$



DPLL + T: example

$$a \land (b \lor (c \land \neg b))$$

where $a \mapsto (x = y), b \mapsto (x = z), c \mapsto (y = z)$



Questions ?