Nelson-Open Theory Combination

Aleksandar Zeljić
Materials by Clark Barrett, Stanford University

CS357: October 2019

Acknowledgments: Many thanks to Cesare Tinelli and Albert Oliveras for contributing some of the material used in these slides.

Disclamer: The literature on SMT and its applications is vast. The bibliographic references provided here are just a sample. Apologies to all authors whose work is not cited.

Combining Theories

Need for Combining Theories and Solvers

Recall: Many applications give rise to formulas like:

$$a \approx b + 2 \,\wedge\, A \approx \mathrm{write}(B, a+1, 4) \,\wedge$$
$$(\mathrm{read}(A, b+3) \approx 2 \,\vee\, f(a-1) \neq f(b+1))$$

Solving that formula requires reasoning over

- the theory of linear arithmetic (T_{LA})
- the theory of arrays (T_{A})
- the theory of uninterpreted functions (T_{UF})

Question: Given solvers for each theory, can we combine them modularly into one for $T_{\mathrm{LA}} \cup T_{\mathrm{A}} \cup T_{\mathrm{UF}}$?

Under certain conditions, we can do it with the Nelson-Oppen combination method [NO79, Opp80]

Need for Combining Theories and Solvers

Recall: Many applications give rise to formulas like:

$$\begin{array}{l} a \approx b + 2 \, \wedge \, A \approx \mathrm{write}(B, a + 1, 4) \, \wedge \\ (\mathrm{read}(A, b + 3) \approx 2 \, \vee \, f(a - 1) \neq f(b + 1)) \end{array}$$

Solving that formula requires reasoning over

- the theory of linear arithmetic (T_{LA})
- the theory of arrays (T_{A})
- the theory of uninterpreted functions (T_{UF})

Question: Given solvers for each theory, can we combine them modularly into one for $T_{\mathrm{LA}} \cup T_{\mathrm{A}} \cup T_{\mathrm{UF}}$?

Under certain conditions, we can do it with the Nelson-Oppen combination method [NO79, Opp80]

Need for Combining Theories and Solvers

Recall: Many applications give rise to formulas like:

$$a \approx b + 2 \wedge A \approx \mathrm{write}(B, a+1, 4) \wedge$$
$$(\mathrm{read}(A, b+3) \approx 2 \vee f(a-1) \neq f(b+1))$$

Solving that formula requires reasoning over

- the theory of linear arithmetic (T_{LA})
- the theory of arrays (T_{A})
- the theory of uninterpreted functions (T_{UF})

Question: Given solvers for each theory, can we combine them modularly into one for $T_{\mathrm{LA}} \cup T_{\mathrm{A}} \cup T_{\mathrm{UF}}$?

Under certain conditions, we can do it with the Nelson-Oppen combination method [NO79, Opp80]

Need for Combining Theories and Solvers

Recall: Many applications give rise to formulas like:

$$a \approx b + 2 \wedge A \approx \mathrm{write}(B, a+1, 4) \wedge$$
$$(\mathrm{read}(A, b+3) \approx 2 \vee f(a-1) \neq f(b+1))$$

Solving that formula requires reasoning over

- the theory of linear arithmetic (T_{LA})
- the theory of arrays (T_{A})
- the theory of uninterpreted functions (T_{UF})

Question: Given solvers for each theory, can we combine them modularly into one for $T_{\mathrm{LA}} \cup T_{\mathrm{A}} \cup T_{\mathrm{UF}}$?

Under certain conditions, we can do it with the Nelson-Oppen combination method [NO79, Opp80]

Motivating Example (Convex Case)

Consider the following set of literals over $T_{\mathrm{LRA}} \cup T_{\mathrm{UF}}$ (T_{LRA}, linear real arithmetic):

$$\begin{aligned} f(f(x) - f(y)) &= a \\ f(0) &> a + 2 \\ x &= y \end{aligned}$$

First step: *purify* literals so that each belongs to a single theory

Motivating Example (Convex Case)

Consider the following set of literals over $T_{\mathrm{LRA}} \cup T_{\mathrm{UF}}$ (T_{LRA}, linear real arithmetic):

$$\begin{array}{rcl} f(f(x) - f(y)) & = & a \\ f(0) & > & a + 2 \\ x & = & y \end{array}$$

First step: *purify* literals so that each belongs to a single theory

Motivating Example (Convex Case)

Consider the following set of literals over $T_{\text{LRA}} \cup T_{\text{UF}}$ (T_{LRA}, linear real arithmetic):

$$\begin{array}{rcl} f(f(x) - f(y)) & = & a \\ f(0) & > & a + 2 \\ x & = & y \end{array}$$

First step: *purify* literals so that each belongs to a single theory

$$f(f(x) - f(y)) = a \quad \Longrightarrow \quad \begin{array}{r} f(e_1) = a \\ e_1 = f(x) - f(y) \end{array} \quad \Longrightarrow \quad \begin{array}{r} f(e_1) = a \\ e_1 = e_2 - e_3 \\ e_2 = f(x) \\ e_3 = f(y) \end{array}$$

Motivating Example (Convex Case)

Consider the following set of literals over $T_{\text{LRA}} \cup T_{\text{UF}}$ (T_{LRA}, linear real arithmetic):

$$\begin{array}{rcl} f(f(x) - f(y)) & = & a \\ f(0) & > & a + 2 \\ x & = & y \end{array}$$

First step: *purify* literals so that each belongs to a single theory

$$f(0) > a + 2 \quad \Longrightarrow \quad \begin{array}{c} f(e_4) > a + 2 \\ e_4 = 0 \end{array} \quad \Longrightarrow \quad \begin{array}{c} f(e_4) = e_5 \\ e_4 = 0 \\ e_5 > a + 2 \end{array}$$

Motivating Example (Convex Case)

Second step: exchange entailed *interface equalities*, equalities over shared constants $e_1, e_2, e_3, e_4, e_5, a$

L_1	L_2
$f(e_1) = a$	$e_2 - e_3 = e_1$
$f(x) = e_2$	$e_4 = 0$
$f(y) = e_3$	$e_5 > a + 2$
$f(e_4) = e_5$	$e_2 = e_3$
$x = y$	$a = e_5$
$e_1 = e_4$	

$L_1 \models_{\mathrm{UF}} e_2 = e_3 \qquad L_2 \models_{\mathrm{LRA}} e_1 = e_4$

$L_1 \models_{\mathrm{UF}} a = e_5$

Third step: check for satisfiability locally

Report unsatisfiable

Motivating Example (Convex Case)

Second step: exchange entailed *interface equalities*, equalities over shared constants $e_1, e_2, e_3, e_4, e_5, a$

L_1	L_2
$f(e_1) = a$	$e_2 - e_3 = e_1$
$f(x) = e_2$	$e_4 = 0$
$f(y) = e_3$	$e_5 > a + 2$
$f(e_4) = e_5$	$e_2 = e_3$
$x = y$	$a = e_5$
$e_1 = e_4$	

$L_1 \models_{\mathrm{UF}} e_2 = e_3$ $\quad$ $L_2 \models_{\mathrm{LRA}} e_1 = e_4$

$L_1 \models_{\mathrm{UF}} a = e_5$

Third step: check for satisfiability locally

$L_2 \models_{\mathrm{LRA}} \bot$ $\quad$ Report unsatisfiable

Motivating Example (Convex Case)

Second step: exchange entailed *interface equalities*, equalities over shared constants $e_1, e_2, e_3, e_4, e_5, a$

L_1	L_2
$f(e_1) = a$	$e_2 - e_3 = e_1$
$f(x) = e_2$	$e_4 = 0$
$f(y) = e_3$	$e_5 > a + 2$
$f(e_4) = e_5$	$e_2 = e_3$
$x = y$	$a = e_5$
$e_1 = e_4$	

$L_1 \models_{\mathrm{UF}} e_2 = e_3 \qquad L_2 \models_{\mathrm{LRA}} e_1 = e_4$

$L_1 \models_{\mathrm{UF}} a = e_5$

Third step: check for satisfiability locally

$L_2 \models_{\mathrm{LRA}} \bot$ Report unsatisfiable

Motivating Example (Convex Case)

Second step: exchange entailed *interface equalities*, equalities over shared constants $e_1, e_2, e_3, e_4, e_5, a$

L_1	L_2
$f(e_1) = a$	$e_2 - e_3 = e_1$
$f(x) = e_2$	$e_4 = 0$
$f(y) = e_3$	$e_5 > a + 2$
$f(e_4) = e_5$	$e_2 = e_3$
$x = y$	$a = e_5$
$e_1 = e_4$	

$L_1 \models_{\mathrm{UF}} e_2 = e_3$ $\qquad$ $L_2 \models_{\mathrm{LRA}} e_1 = e_4$

$L_1 \models_{\mathrm{UF}} a = e_5$

Third step: check for satisfiability locally

$L_1 \not\models_{\mathrm{UF}} \bot$
$L_2 \models_{\mathrm{LRA}} \bot$ $\qquad$ Report unsatisfiable

Motivating Example (Convex Case)

Second step: exchange entailed *interface equalities*, equalities over shared constants $e_1, e_2, e_3, e_4, e_5, a$

L_1	L_2
$f(e_1) = a$	$e_2 - e_3 = e_1$
$f(x) = e_2$	$e_4 = 0$
$f(y) = e_3$	$e_5 > a + 2$
$f(e_4) = e_5$	$e_2 = e_3$
$x = y$	$a = e_5$
$e_1 = e_4$	

$L_1 \models_{\mathrm{UF}} e_2 = e_3 \qquad L_2 \models_{\mathrm{LRA}} e_1 = e_4$

$L_1 \models_{\mathrm{UF}} a = e_5$

Third step: check for satisfiability locally

$L_1 \not\models_{\mathrm{UF}} \bot$
$L_2 \models_{\mathrm{LRA}} \bot$

Report unsatisfiable

Motivating Example (Convex Case)

Second step: exchange entailed *interface equalities*, equalities over shared constants $e_1, e_2, e_3, e_4, e_5, a$

L_1	L_2
$f(e_1) = a$	$e_2 - e_3 = e_1$
$f(x) = e_2$	$e_4 = 0$
$f(y) = e_3$	$e_5 > a + 2$
$f(e_4) = e_5$	$e_2 = e_3$
$x = y$	$a = e_5$
$e_1 = e_4$	

$L_1 \models_{\text{UF}} e_2 = e_3 \qquad L_2 \models_{\text{LRA}} e_1 = e_4$

$L_1 \models_{\text{UF}} a = e_5$

Third step: check for satisfiability locally

$L_1 \not\models_{\text{UF}} \bot$

$L_2 \models_{\text{LRA}} \bot$

Report Unsatisfiable

Motivating Example (Convex Case)

Second step: exchange entailed *interface equalities*, equalities over shared constants $e_1, e_2, e_3, e_4, e_5, a$

L_1	L_2
$f(e_1) = a$	$e_2 - e_3 = e_1$
$f(x) = e_2$	$e_4 = 0$
$f(y) = e_3$	$e_5 > a + 2$
$f(e_4) = e_5$	$e_2 = e_3$
$x = y$	$a = e_5$
$e_1 = e_4$	

$L_1 \models_{\mathrm{UF}} e_2 = e_3$ $\qquad L_2 \models_{\mathrm{LRA}} e_1 = e_4$

$L_1 \models_{\mathrm{UF}} a = e_5$

Third step: check for satisfiability locally

$L_1 \not\models_{\mathrm{UF}} \bot$

$L_2 \models_{\mathrm{LRA}} \bot$

Report unsatisfiable

Motivating Example (Convex Case)

Second step: exchange entailed *interface equalities*, equalities over shared constants $e_1, e_2, e_3, e_4, e_5, a$

L_1	L_2
$f(e_1) = a$	$e_2 - e_3 = e_1$
$f(x) = e_2$	$e_4 = 0$
$f(y) = e_3$	$e_5 > a + 2$
$f(e_4) = e_5$	$e_2 = e_3$
$x = y$	$a = e_5$
$e_1 = e_4$	

$L_1 \models_{\mathrm{UF}} e_2 = e_3 \qquad L_2 \models_{\mathrm{LRA}} e_1 = e_4$

$L_1 \models_{\mathrm{UF}} a = e_5$

Third step: check for satisfiability locally

$L_1 \not\models_{\mathrm{UF}} \bot$

$L_2 \models_{\mathrm{LRA}} \bot$

Report unsatisfiable

Motivating Example (Convex Case)

Second step: exchange entailed *interface equalities*, equalities over shared constants $e_1, e_2, e_3, e_4, e_5, a$

L_1	L_2
$f(e_1) = a$	$e_2 - e_3 = e_1$
$f(x) = e_2$	$e_4 = 0$
$f(y) = e_3$	$e_5 > a + 2$
$f(e_4) = e_5$	$e_2 = e_3$
$x = y$	$a = e_5$
$e_1 = e_4$	

$L_1 \models_{\mathrm{UF}} e_2 = e_3 \qquad L_2 \models_{\mathrm{LRA}} e_1 = e_4$

$L_1 \models_{\mathrm{UF}} a = e_5$

Third step: check for satisfiability locally

$L_1 \not\models_{\mathrm{UF}} \bot$

$L_2 \models_{\mathrm{LRA}} \bot$

Report unsatisfiable

Motivating Example (Non-convex Case)

Consider the following unsatisfiable set of literals over $T_{\text{LIA}} \cup T_{\text{UF}}$ (T_{LIA}, linear integer arithmetic):

$$\begin{aligned} 1 \leq \quad x & \quad \leq 2 \\ f(1) &= a \\ f(2) &= f(1) + 3 \\ a &= b + 2 \end{aligned}$$

First step: *purify* literals so that each belongs to a single theory

Motivating Example (Non-convex Case)

Consider the following unsatisfiable set of literals over $T_{\mathrm{LIA}} \cup T_{\mathrm{UF}}$ (T_{LIA}, linear integer arithmetic):

$$\begin{aligned} 1 \leq \quad x & \quad \leq 2 \\ f(1) & = a \\ f(2) & = f(1) + 3 \\ a & = b + 2 \end{aligned}$$

First step: *purify* literals so that each belongs to a single theory

Motivating Example (Non-convex Case)

Consider the following unsatisfiable set of literals over $T_{\text{LIA}} \cup T_{\text{UF}}$ (T_{LIA}, linear integer arithmetic):

$$\begin{array}{rcl} 1 \leq & x & \leq 2 \\ f(1) & = & a \\ f(2) & = & f(1) + 3 \\ a & = & b + 2 \end{array}$$

First step: *purify* literals so that each belongs to a single theory

$$f(1) = a \quad \Longrightarrow \quad \begin{array}{r} f(e_1) = a \\ e_1 = 1 \end{array}$$

Motivating Example (Non-convex Case)

Consider the following unsatisfiable set of literals over $T_{\text{LIA}} \cup T_{\text{UF}}$ (T_{LIA}, linear integer arithmetic):

$$\begin{array}{rcl} 1 \leq & x & \leq 2 \\ f(1) & = & a \\ f(2) & = & f(1) + 3 \\ a & = & b + 2 \end{array}$$

First step: *purify* literals so that each belongs to a single theory

$$f(2) = f(1) + 3 \quad \Longrightarrow \quad \begin{aligned} e_2 &= 2 \\ f(e_2) &= e_3 \\ f(e_1) &= e_4 \\ e_3 &= e_4 + 3 \end{aligned}$$

Motivating Example (Non-convex Case)

Second step: exchange entailed *interface equalities* over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_1$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_1$	

Motivating Example (Non-convex Case)

Second step: exchange entailed *interface equalities* over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_1$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_1$	

No more entailed equalities, but $L_1 \models_{\text{LIA}} x = e_1 \lor x = e_2$

Motivating Example (Non-convex Case)

Second step: exchange entailed *interface equalities* over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_1$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_1$	

Consider each case of $x = e_1 \vee x = e_2$ separately

Motivating Example (Non-convex Case)

Second step: exchange entailed *interface equalities* over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_1$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_1$	

Case 1) $x = e_1$

Motivating Example (Non-convex Case)

Second step: exchange entailed *interface equalities* over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_1$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_1$	

Motivating Example (Non-convex Case)

Second step: exchange entailed *interface equalities* over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_1$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_1$	

$L_2 \models_{\mathrm{UF}} a = b$, which entails $\perp$ when sent to L_1

Motivating Example (Non-convex Case)

Second step: exchange entailed *interface equalities* over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_2$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_2$	

Motivating Example (Non-convex Case)

Second step: exchange entailed *interface equalities* over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_2$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_2$	

Case 2) $x = e_2$

Motivating Example (Non-convex Case)

Second step: exchange entailed *interface equalities* over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_2$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_2$	

Motivating Example (Non-convex Case)

Second step: exchange entailed *interface equalities* over shared constants $x, e_1, a, b, e_2, e_3, e_4$

L_1	L_2
$1 \leq x$	$f(e_1) = a$
$x \leq 2$	$f(x) = b$
$e_1 = 1$	$f(e_2) = e_3$
$a = b + 2$	$f(e_1) = e_4$
$e_2 = 2$	$x = e_2$
$e_3 = e_4 + 3$	
$a = e_4$	
$x = e_2$	

$L_2 \models_{\mathrm{UF}} e_3 = b$, which entails $\bot$ when sent to L_1

The Nelson-Oppen Method

- For $i = 1, 2$, let T_i be a first-order theory of *signature* Σ_i (set of function and predicate symbols in T_i other than $=$)
- Let $T = T_1 \cup T_2$
- Let $\mathcal{C}$ be a finite set of *free* constants (i.e., not in $\Sigma_1 \cup \Sigma_2$)

We consider only input problems of the form

$$L_1 \cup L_2$$

where each L_i is a finite set of *ground* (i.e., variable-free) $(\Sigma_i \cup \mathcal{C})$-literals

Note: Because of purification, there is no loss of generality in considering only ground $(\Sigma_i \cup \mathcal{C})$-literals

The Nelson-Oppen Method

- For $i = 1, 2$, let T_i be a first-order theory of *signature* Σ_i (set of function and predicate symbols in T_i other than $=$)
- Let $T = T_1 \cup T_2$
- Let $\mathcal{C}$ be a finite set of *free* constants (i.e., not in $\Sigma_1 \cup \Sigma_2$)

We consider only input problems of the form

$$L_1 \cup L_2$$

where each L_i is a finite set of *ground* (i.e., variable-free) $(\Sigma_i \cup \mathcal{C})$-literals

Note: Because of purification, there is no loss of generality in considering only ground $(\Sigma_i \cup \mathcal{C})$-literals

The Nelson-Oppen Method

- For $i = 1, 2$, let T_i be a first-order theory of *signature* Σ_i (set of function and predicate symbols in T_i other than $=$)
- Let $T = T_1 \cup T_2$
- Let $\mathcal{C}$ be a finite set of *free* constants (i.e., not in $\Sigma_1 \cup \Sigma_2$)

We consider only input problems of the form

$$L_1 \cup L_2$$

where each L_i is a finite set of *ground* (i.e., variable-free) $(\Sigma_i \cup \mathcal{C})$-literals

Note: Because of purification, there is no loss of generality in considering only ground $(\Sigma_i \cup \mathcal{C})$-literals

The Nelson-Oppen Method

Bare-bones, non-deterministic, non-incremental version

[Opp80, Rin96, TH96]:

Input: $L_1 \cup L_2$ with L_i finite set of ground $(\Sigma_i \cup \mathcal{C})$-literals
Output: **sat** or **unsat**

1. Guess an *arrangement* A, i.e., a set of equalities and disequalities over $\mathcal{C}$ such that

$$c = d \in A \text{ or } c \neq d \in A \text{ for all } c, d \in \mathcal{C}$$

2. If $L_i \cup A$ is T_i-unsatisfiable for $i = 1$ or $i = 2$, return unsat
3. Otherwise, return **sat**

Bare-bones, non-deterministic, non-incremental version [Opp80, Rin96, TH96]:

Input: $L_1 \cup L_2$ with L_i finite set of ground $(\Sigma_i \cup \mathcal{C})$-literals
Output: **sat** or **unsat**

1. Guess an *arrangement* A, i.e., a set of equalities and disequalities over $\mathcal{C}$ such that

$$c = d \in A \text{ or } c \neq d \in A \text{ for all } c, d \in \mathcal{C}$$

2. If $L_i \cup A$ is T_i-unsatisfiable for $i = 1$ or $i = 2$, return unsat
3. Otherwise, return **sat**

The Nelson-Oppen Method

Bare-bones, non-deterministic, non-incremental version

[Opp80, Rin96, TH96]:

Input: $L_1 \cup L_2$ with L_i finite set of ground $(\Sigma_i \cup \mathcal{C})$-literals
Output: **sat** or **unsat**

1. Guess an *arrangement* A, i.e., a set of equalities and disequalities over $\mathcal{C}$ such that

$$c = d \in A \text{ or } c \neq d \in A \text{ for all } c, d \in \mathcal{C}$$

2. If $L_i \cup A$ is T_i-unsatisfiable for $i = 1$ or $i = 2$, return unsat
3. Otherwise, return **sat**

The Nelson-Oppen Method

Bare-bones, non-deterministic, non-incremental version

[Opp80, Rin96, TH96]:

Input: $L_1 \cup L_2$ with L_i finite set of ground $(\Sigma_i \cup \mathcal{C})$-literals
Output: **sat** or **unsat**

1. Guess an *arrangement* A, i.e., a set of equalities and disequalities over $\mathcal{C}$ such that

$$c = d \in A \text{ or } c \neq d \in A \text{ for all } c, d \in \mathcal{C}$$

2. If $L_i \cup A$ is T_i-unsatisfiable for $i = 1$ or $i = 2$, return unsat

3. Otherwise, return **sat**

The Nelson-Oppen Method

Bare-bones, non-deterministic, non-incremental version

[Opp80, Rin96, TH96]:

Input: $L_1 \cup L_2$ with L_i finite set of ground $(\Sigma_i \cup \mathcal{C})$-literals
Output: **sat** or **unsat**

1. Guess an *arrangement* A, i.e., a set of equalities and disequalities over $\mathcal{C}$ such that

$$c = d \in A \;\text{ or }\; c \neq d \in A \;\text{ for all } c, d \in \mathcal{C}$$

2. If $L_i \cup A$ is T_i-unsatisfiable for $i = 1$ or $i = 2$, return unsat

3. Otherwise, return **sat**

Correctness of the NO Method

Proposition (Termination) The method is terminating.

(Trivially, because there is only a finite number of arrangements to guess)

Proposition (Soundness) If the method returns **unsat** for every arrangement, the input is $(T_1 \cup T_2)$-unsatisfiable.

(Because satisfiability in $(T_1 \cup T_2)$ is always preserved)

Proposition (Completeness) If $\Sigma_1 \cap \Sigma_2 = \emptyset$ and T_1 and T_2 are stably infinite, when the method returns **sat** for some arrangement, the input is $(T_1 \cup T_2)$-is satisfiable.

Correctness of the NO Method

Proposition (Termination) The method is terminating.

(Trivially, because there is only a finite number of arrangements to guess)

Proposition (Soundness) If the method returns **unsat** for every arrangement, the input is $(T_1 \cup T_2)$-unsatisfiable.

(Because satisfiability in $(T_1 \cup T_2)$ is always preserved)

Proposition (Completeness) If $\Sigma_1 \cap \Sigma_2 = \emptyset$ and T_1 and T_2 are stably infinite, when the method returns **sat** for some arrangement, the input is $(T_1 \cup T_2)$-is satisfiable.

Correctness of the NO Method

Proposition (Termination) The method is terminating.

(Trivially, because there is only a finite number of arrangements to guess)

Proposition (Soundness) If the method returns **unsat** for every arrangement, the input is $(T_1 \cup T_2)$-unsatisfiable.

(Because satisfiability in $(T_1 \cup T_2)$ is always preserved)

Proposition (Completeness) If $\Sigma_1 \cap \Sigma_2 = \emptyset$ and T_1 and T_2 are stably infinite, when the method returns **sat** for some arrangement, the input is $(T_1 \cup T_2)$-is satisfiable.

Stably Infinite Theories

Def. A theory T is *stably infinite* iff every quantifier-free T-satisfiable formula is satisfiable in an infinite model of T

Many interesting theories are stably infinite:

- Theories of an infinite structure (e.g., integer arithmetic)
- Complete theories with an infinite model (e.g., theory of dense linear orders, theory of lists)
- Convex theories (e.g., EUF, linear real arithmetic)

Def. A theory T is *convex* iff, for any set L of literals
$L \models_T s_1 = t_1 \vee \cdots \vee s_n = t_n \implies L \models_T s_i = t_i$ for some i

Note: With convex theories, arrangements do not need to be guessed—they can be computed by (theory) propagation

Stably Infinite Theories

Def. A theory T is *stably infinite* iff every quantifier-free T-satisfiable formula is satisfiable in an infinite model of T

Many interesting theories are stably infinite:

- Theories of an infinite structure (e.g., integer arithmetic)
- Complete theories with an infinite model (e.g., theory of dense linear orders, theory of lists)
- Convex theories (e.g., EUF, linear real arithmetic)

Def. A theory T is *convex* iff, for any set L of literals
$L \models_T s_1 = t_1 \vee \cdots \vee s_n = t_n \implies L \models_T s_i = t_i$ for some i

Note: With convex theories, arrangements do not need to be guessed—they can be computed by (theory) propagation

Stably Infinite Theories

Def. A theory T is *stably infinite* iff every quantifier-free T-satisfiable formula is satisfiable in an infinite model of T

Many interesting theories are stably infinite:

- Theories of an infinite structure (e.g., integer arithmetic)
- Complete theories with an infinite model (e.g., theory of dense linear orders, theory of lists)
- Convex theories (e.g., EUF, linear real arithmetic)

Def. A theory T is *convex* iff, for any set L of literals
$L \models_T s_1 = t_1 \vee \cdots \vee s_n = t_n \implies L \models_T s_i = t_i$ for some i

Note: With convex theories, arrangements do not need to be guessed—they can be computed by (theory) propagation

Stably Infinite Theories

Def. A theory T is *stably infinite* iff every quantifier-free T-satisfiable formula is satisfiable in an infinite model of T

Other interesting theories are not stably infinite:

- Theories of a finite structure (e.g., theory of bit vectors of finite size, arithmetic modulo n)
- Theories with models of bounded cardinality (e.g., theory of strings of bounded length)
- Some equational/Horn theories

The Nelson-Oppen method has been extended to some classes of non-stably infinite theories [TZ05, RRZ05, JB10]

Stably Infinite Theories

Def. A theory T is *stably infinite* iff every quantifier-free T-satisfiable formula is satisfiable in an infinite model of T

Other interesting theories are not stably infinite:

- Theories of a finite structure (e.g., theory of bit vectors of finite size, arithmetic modulo n)
- Theories with models of bounded cardinality (e.g., theory of strings of bounded length)
- Some equational/Horn theories

The Nelson-Oppen method has been extended to some classes of non-stably infinite theories [TZ05, RRZ05, JB10]

Stably Infinite Theories

Def. A theory T is *stably infinite* iff every quantifier-free T-satisfiable formula is satisfiable in an infinite model of T

Other interesting theories are not stably infinite:

- Theories of a finite structure (e.g., theory of bit vectors of finite size, arithmetic modulo n)
- Theories with models of bounded cardinality (e.g., theory of strings of bounded length)
- Some equational/Horn theories

The Nelson-Oppen method has been extended to some classes of non-stably infinite theories [TZ05, RRZ05, JB10]

SMT Solving with *Multiple* Theories

Let $T_1, \ldots, T_n$ be theories with respective solvers $S_1, \ldots, S_n$

How can we integrate all of them cooperatively into a single SMT solver for $T = T_1 \cup \cdots \cup T_n$?

SMT Solving with *Multiple* Theories

Let $T_1, \ldots, T_n$ be theories with respective solvers $S_1, \ldots, S_n$

How can we integrate all of them cooperatively into a single SMT solver for $T = T_1 \cup \cdots \cup T_n$?

Quick Solution:

1. Combine $S_1, \ldots, S_n$ with Nelson-Oppen into a theory solver for T
2. Build a DPLL(T) solver as usual

SMT Solving with *Multiple* Theories

Let $T_1, \ldots, T_n$ be theories with respective solvers $S_1, \ldots, S_n$

How can we integrate all of them cooperatively into a single SMT solver for $T = T_1 \cup \cdots \cup T_n$?

Better Solution [Bar02, BBC+05b, BNOT06]:

1. Extend DPLL(T) to DPLL($T_1, \ldots, T_n$)
2. Lift Nelson-Oppen to the DPLL($X_1, \ldots, X_n$) level
3. Build a DPLL($T_1, \ldots, T_n$) solver

Modeling DPLL($T_1, \ldots, T_n$) Abstractly

- Let $n = 2$, for simplicity
- Let T_i be of signature Σ_i for $i = 1, 2$, with $\Sigma_1 \cap \Sigma_2 = \emptyset$
- Let $\mathcal{C}$ be a set of free constants
- Assume wlog that each input literal has signature $(\Sigma_1 \cup \mathcal{C})$ or $(\Sigma_2 \cup \mathcal{C})$ (no *mixed* literals)
- Let $\mathsf{M}|_i \stackrel{\text{def}}{=} \{(\Sigma_i \cup \mathcal{C})\text{-literals of } \mathsf{M} \text{ and their complement}\}$
- Let $\mathrm{I}(\mathsf{M}) \stackrel{\text{def}}{=} \{c = d \mid c, d \text{ occur in } \mathcal{C}, \mathsf{M}|_1 \text{ and } \mathsf{M}|_2\} \cup$
  $\{c \neq d \mid c, d \text{ occur in } \mathcal{C}, \mathsf{M}|_1 \text{ and } \mathsf{M}|_2\}$

  (*interface literals*)

Abstract DPLL Modulo Multiple Theories

Propagate, **Conflict**, **Explain**, **Backjump**, **Fail** (unchanged)

Decide $\dfrac{l \in \mathrm{Lit}(\mathsf{F}) \cup \mathrm{I}(\mathsf{M}) \quad l, \bar{l} \notin \mathsf{M}}{\mathsf{M} := \mathsf{M} \bullet l}$

Only change: decide on interface equalities as well

T-Propagate $\dfrac{l \in \mathrm{Lit}(\mathsf{F}) \cup \mathrm{I}(\mathsf{M}) \quad i \in \{1, 2\} \quad \mathsf{M} \models_{T_i} l \quad l, \bar{l} \notin \mathsf{M}}{\mathsf{M} := \mathsf{M}\ l}$

Only change: propagate interface equalities as well, but reason locally in each T_i

Abstract DPLL Modulo Multiple Theories

Propagate, **Conflict**, **Explain**, **Backjump**, **Fail** (unchanged)

Decide $\dfrac{l \in \mathrm{Lit}(\mathsf{F}) \cup \mathrm{I}(\mathsf{M}) \quad l, \bar{l} \notin \mathsf{M}}{\mathsf{M} := \mathsf{M} \bullet l}$

Only change: decide on interface equalities as well

T**-Propagate** $\dfrac{l \in \mathrm{Lit}(\mathsf{F}) \cup \mathrm{I}(\mathsf{M}) \quad i \in \{1, 2\} \quad \mathsf{M} \models_{T_i} l \quad l, \bar{l} \notin \mathsf{M}}{\mathsf{M} := \mathsf{M}\, l}$

Only change: propagate interface equalities as well, but reason locally in each T_i

Abstract DPLL Modulo Multiple Theories

Propagate, **Conflict**, **Explain**, **Backjump**, **Fail** (unchanged)

Decide $\dfrac{l \in \mathrm{Lit}(\mathsf{F}) \cup \mathrm{I}(\mathsf{M}) \quad l, \bar{l} \notin \mathsf{M}}{\mathsf{M} := \mathsf{M} \bullet l}$

Only change: decide on interface equalities as well

T**-Propagate** $\dfrac{l \in \mathrm{Lit}(\mathsf{F}) \cup \mathrm{I}(\mathsf{M}) \quad i \in \{1, 2\} \quad \mathsf{M} \models_{T_i} l \quad l, \bar{l} \notin \mathsf{M}}{\mathsf{M} := \mathsf{M}\ l}$

Only change: propagate interface equalities as well, but reason locally in each T_i

Abstract DPLL Modulo Multiple Theories

T-Conflict

$$\frac{\mathsf{C} = \mathsf{no} \quad l_1, \ldots, l_n \in \mathsf{M} \quad l_1, \ldots, l_n \models_{T_i} \bot \quad i \in \{1, 2\}}{\mathsf{C} := \bar{l}_1 \vee \cdots \vee \bar{l}_n}$$

T-Explain

$$\frac{\mathsf{C} = l \vee D \quad \bar{l}_1, \ldots, \bar{l}_n \models_{T_i} \bar{l} \quad i \in \{1, 2\} \quad \bar{l}_1, \ldots, \bar{l}_n \prec_{\mathsf{M}} \bar{l}}{\mathsf{C} := l_1 \vee \cdots \vee l_n \vee D}$$

Only change: reason locally in each T_i

I-Learn

$$\frac{\models_{T_i} l_1 \vee \cdots \vee l_n \quad l_1, \ldots, l_n \in \mathsf{M}|_i \cup \mathrm{I}(\mathsf{M}) \quad i \in \{1, 2\}}{\mathsf{F} := \mathsf{F} \cup \{l_1 \vee \cdots \vee l_n\}}$$

New rule: for entailed disjunctions of interface literals

Abstract DPLL Modulo Multiple Theories

T-Conflict

$$\frac{\mathsf{C} = \mathsf{no} \quad l_1, \ldots, l_n \in \mathsf{M} \quad l_1, \ldots, l_n \models_{T_i} \bot \quad i \in \{1, 2\}}{\mathsf{C} := \bar{l}_1 \vee \cdots \vee \bar{l}_n}$$

T-Explain

$$\frac{\mathsf{C} = l \vee D \quad \bar{l}_1, \ldots, \bar{l}_n \models_{T_i} \bar{l} \quad i \in \{1, 2\} \quad \bar{l}_1, \ldots, \bar{l}_n \prec_{\mathsf{M}} \bar{l}}{\mathsf{C} := l_1 \vee \cdots \vee l_n \vee D}$$

Only change: reason locally in each T_i

I-Learn

$$\frac{\models_{T_i} l_1 \vee \cdots \vee l_n \quad l_1, \ldots, l_n \in \mathsf{M}|_i \cup \mathrm{I}(\mathsf{M}) \quad i \in \{1, 2\}}{\mathsf{F} := \mathsf{F} \cup \{l_1 \vee \cdots \vee l_n\}}$$

New rule: for entailed disjunctions of interface literals

Example — Convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = e_2}^{1} \wedge \overbrace{f(y) = e_3}^{2} \wedge \overbrace{f(e_4) = e_5}^{3} \wedge \overbrace{x = y}^{4} \wedge$$

$$\underbrace{e_2 - e_3 = e_1}_{5} \wedge \underbrace{e_4 = 0}_{6} \wedge \underbrace{e_5 > a + 2}_{7}$$

$$\underbrace{e_2 = e_3}_{8} \quad \underbrace{e_1 = e_4}_{9} \quad \underbrace{a = e_5}_{10}$$

M	F	C	rule
	F	no	
0 1 2 3 4 5 6 7	F	no	by **Propagate**$^+$
0 1 2 3 4 5 6 7 8	F	no	by T-**Propagate** $(1, 2, 4 \models_{\text{UF}} 8)$
0 1 2 3 4 5 6 7 8 9	F	no	by T-**Propagate** $(5, 6, 8 \models_{\text{LRA}} 9)$
0 1 2 3 4 5 6 7 8 9 10	F	no	by T-**Propagate** $(0, 3, 9 \models_{\text{UF}} 10)$
0 1 2 3 4 5 6 7 8 9 10	F	$\overline{7} \vee \overline{10}$	by T-**Conflict** $(7, 10 \models_{\text{LRA}} \bot)$
fail			by **Fail**

Example — Convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = e_2}^{1} \wedge \overbrace{f(y) = e_3}^{2} \wedge \overbrace{f(e_4) = e_5}^{3} \wedge \overbrace{x = y}^{4} \wedge$$
$$\underbrace{e_2 - e_3 = e_1}_{5} \wedge \underbrace{e_4 = 0}_{6} \wedge \underbrace{e_5 > a + 2}_{7}$$

$$\underbrace{e_2 = e_3}_{8} \quad \underbrace{e_1 = e_4}_{9} \quad \underbrace{a = e_5}_{10}$$

M	F	C	rule
	F	no	
0 1 2 3 4 5 6 7	F	no	by **Propagate**$^+$
0 1 2 3 4 5 6 7 8	F	no	by T-**Propagate** $(1,\ 2,\ 4 \models_{\mathrm{UF}} 8)$
0 1 2 3 4 5 6 7 8 9	F	no	by T-**Propagate** $(5,\ 6,\ 8 \models_{\mathrm{LRA}} 9)$
0 1 2 3 4 5 6 7 8 9 10	F	no	by T-**Propagate** $(0,\ 3,\ 9 \models_{\mathrm{UF}} 10)$
0 1 2 3 4 5 6 7 8 9 10	F	$\overline{7} \vee \overline{10}$	by T-**Conflict** $(7,\ 10 \models_{\mathrm{LRA}} \perp)$
fail			by **Fail**

Example — Convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = e_2}^{1} \wedge \overbrace{f(y) = e_3}^{2} \wedge \overbrace{f(e_4) = e_5}^{3} \wedge \overbrace{x = y}^{4} \wedge$$
$$\underbrace{e_2 - e_3 = e_1}_{5} \wedge \underbrace{e_4 = 0}_{6} \wedge \underbrace{e_5 > a + 2}_{7}$$

$$\underbrace{e_2 = e_3}_{8} \quad \underbrace{e_1 = e_4}_{9} \quad \underbrace{a = e_5}_{10}$$

M	F	C	rule
	F	no	
0 1 2 3 4 5 6 7	F	no	by **Propagate**$^+$
0 1 2 3 4 5 6 7 8	F	no	by T-**Propagate** $(1, 2, 4 \models_{\mathrm{UF}} 8)$
0 1 2 3 4 5 6 7 8 9	F	no	by T-**Propagate** $(5, 6, 8 \models_{\mathrm{LRA}} 9)$
0 1 2 3 4 5 6 7 8 9 10	F	no	by T-**Propagate** $(0, 3, 9 \models_{\mathrm{UF}} 10)$
0 1 2 3 4 5 6 7 8 9 10	F	$\overline{7} \vee \overline{10}$	by T-**Conflict** $(7, 10 \models_{\mathrm{LRA}} \perp)$
fail			by **Fail**

Example — Convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = e_2}^{1} \wedge \overbrace{f(y) = e_3}^{2} \wedge \overbrace{f(e_4) = e_5}^{3} \wedge \overbrace{x = y}^{4} \wedge$$
$$\underbrace{e_2 - e_3 = e_1}_{5} \wedge \underbrace{e_4 = 0}_{6} \wedge \underbrace{e_5 > a + 2}_{7}$$

$$\underbrace{e_2 = e_3}_{8} \quad \underbrace{e_1 = e_4}_{9} \quad \underbrace{a = e_5}_{10}$$

M	F	C	rule
	F	no	
0 1 2 3 4 5 6 7	F	no	by **Propagate**$^{+}$
0 1 2 3 4 5 6 7 8	F	no	by **T-Propagate** $(1, 2, 4 \models_{\mathrm{UF}} 8)$
0 1 2 3 4 5 6 7 8 9	F	no	by **T-Propagate** $(5, 6, 8 \models_{\mathrm{LRA}} 9)$
0 1 2 3 4 5 6 7 8 9 10	F	no	by **T-Propagate** $(0, 3, 9 \models_{\mathrm{UF}} 10)$
0 1 2 3 4 5 6 7 8 9 10	F	$\overline{7} \vee \overline{10}$	by **T-Conflict** $(7, 10 \models_{\mathrm{LRA}} \perp)$
fail			by **Fail**

Example — Convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = e_2}^{1} \wedge \overbrace{f(y) = e_3}^{2} \wedge \overbrace{f(e_4) = e_5}^{3} \wedge \overbrace{x = y}^{4} \wedge$$
$$\underbrace{e_2 - e_3 = e_1}_{5} \wedge \underbrace{e_4 = 0}_{6} \wedge \underbrace{e_5 > a + 2}_{7}$$

$$\underbrace{e_2 = e_3}_{8} \quad \underbrace{e_1 = e_4}_{9} \quad \underbrace{a = e_5}_{10}$$

M	F	C	rule
	F	no	
0 1 2 3 4 5 6 7	F	no	by **Propagate**$^+$
0 1 2 3 4 5 6 7 8	F	no	by T-**Propagate** $(1, 2, 4 \models_{\mathrm{UF}} 8)$
0 1 2 3 4 5 6 7 8 9	F	no	by T-**Propagate** $(5, 6, 8 \models_{\mathrm{LRA}} 9)$
0 1 2 3 4 5 6 7 8 9 10	F	no	by T-**Propagate** $(0, 3, 9 \models_{\mathrm{UF}} 10)$
0 1 2 3 4 5 6 7 8 9 10	F	$\overline{7} \vee \overline{10}$	by T-**Conflict** $(7, 10 \models_{\mathrm{LRA}} \bot)$
fail			by **Fail**

Example — Convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = e_2}^{1} \wedge \overbrace{f(y) = e_3}^{2} \wedge \overbrace{f(e_4) = e_5}^{3} \wedge \overbrace{x = y}^{4} \wedge$$

$$\underbrace{e_2 - e_3 = e_1}_{5} \wedge \underbrace{e_4 = 0}_{6} \wedge \underbrace{e_5 > a + 2}_{7}$$

$$\underbrace{e_2 = e_3}_{8} \quad \underbrace{e_1 = e_4}_{9} \quad \underbrace{a = e_5}_{10}$$

M	F	C	rule
	F	no	
0 1 2 3 4 5 6 7	F	no	by **Propagate**$^+$
0 1 2 3 4 5 6 7 8	F	no	by T-**Propagate** $(1, 2, 4 \models_{\mathrm{UF}} 8)$
0 1 2 3 4 5 6 7 8 9	F	no	by T-**Propagate** $(5, 6, 8 \models_{\mathrm{LRA}} 9)$
0 1 2 3 4 5 6 7 8 9 10	F	no	by T-**Propagate** $(0, 3, 9 \models_{\mathrm{UF}} 10)$
0 1 2 3 4 5 6 7 8 9 10	F	$\overline{7} \vee \overline{10}$	by T-**Conflict** $(7, 10 \models_{\mathrm{LRA}} \bot)$
fail			by **Fail**

Example — Convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = e_2}^{1} \wedge \overbrace{f(y) = e_3}^{2} \wedge \overbrace{f(e_4) = e_5}^{3} \wedge \overbrace{x = y}^{4} \wedge$$
$$\underbrace{e_2 - e_3 = e_1}_{5} \wedge \underbrace{e_4 = 0}_{6} \wedge \underbrace{e_5 > a + 2}_{7}$$

$$\underbrace{e_2 = e_3}_{8} \quad \underbrace{e_1 = e_4}_{9} \quad \underbrace{a = e_5}_{10}$$

M	F	C	rule
	F	no	
0 1 2 3 4 5 6 7	F	no	by **Propagate**$^+$
0 1 2 3 4 5 6 7 8	F	no	by T-**Propagate** $(1, 2, 4 \models_{\text{UF}} 8)$
0 1 2 3 4 5 6 7 8 9	F	no	by T-**Propagate** $(5, 6, 8 \models_{\text{LRA}} 9)$
0 1 2 3 4 5 6 7 8 9 10	F	no	by T-**Propagate** $(0, 3, 9 \models_{\text{UF}} 10)$
0 1 2 3 4 5 6 7 8 9 10	F	$\overline{7} \vee \overline{10}$	by T-**Conflict** $(7, 10 \models_{\text{LRA}} \bot)$
fail			by **Fail**

Example — Convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = e_2}^{1} \wedge \overbrace{f(y) = e_3}^{2} \wedge \overbrace{f(e_4) = e_5}^{3} \wedge \overbrace{x = y}^{4} \wedge$$
$$\underbrace{e_2 - e_3 = e_1}_{5} \wedge \underbrace{e_4 = 0}_{6} \wedge \underbrace{e_5 > a + 2}_{7}$$

$$\underbrace{e_2 = e_3}_{8} \quad \underbrace{e_1 = e_4}_{9} \quad \underbrace{a = e_5}_{10}$$

M	F	C	rule
	F	no	
0 1 2 3 4 5 6 7	F	no	by **Propagate**$^+$
0 1 2 3 4 5 6 7 8	F	no	by T-**Propagate** $(1, 2, 4 \models_{\mathrm{UF}} 8)$
0 1 2 3 4 5 6 7 8 9	F	no	by T-**Propagate** $(5, 6, 8 \models_{\mathrm{LRA}} 9)$
0 1 2 3 4 5 6 7 8 9 10	F	no	by T-**Propagate** $(0, 3, 9 \models_{\mathrm{UF}} 10)$
0 1 2 3 4 5 6 7 8 9 10	F	$\overline{7} \vee \overline{10}$	by T-**Conflict** $(7, 10 \models_{\mathrm{LRA}} \perp)$
fail			by **Fail**

Example — Non-convex Theories

$$
F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge
$$

$$
\underbrace{1 \leq x}_{4} \wedge \underbrace{x \leq 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9}
$$

$$
\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}
$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by T-**Propagate** $(0,\ 3 \models_{\mathrm{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{\mathrm{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T-**Propagate** $(0,\ 1,\ 11 \models_{\mathrm{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T-**Conflict** $(7,\ 13 \models_{\mathrm{UF}} \perp)$
$0 \cdots 9\ 10\ \overline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T-**Propagate** $(0,\ 1,\ \overline{13} \models_{\mathrm{UF}} \overline{11})$
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge$$
$$\underbrace{1 \leq x}_{4} \wedge \underbrace{x \leq 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by T-**Propagate** $(0,\ 3 \models_{\text{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{\text{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T-**Propagate** $(0,\ 1,\ 11 \models_{\text{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ \underline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T-**Conflict** $(7,\ 13 \models_{\text{UF}} \perp)$
$0 \cdots 9\ \underline{10}\ \underline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ \underline{10}\ \underline{13}\ 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T-**Propagate** $(0,\ 1,\ \overline{13} \models_{\text{UF}} \overline{11})$
$0 \cdots 9\ 10\ 13\ 11\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge$$
$$\underbrace{1 \leq x}_{4} \wedge \underbrace{x \leq 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by T**-Propagate** $(0,\ 3 \models_{\mathrm{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{\mathrm{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ 11 \models_{\mathrm{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T**-Conflict** $(7,\ 13 \models_{\mathrm{UF}} \bot)$
$0 \cdots 9\ 10\ \overline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ \overline{13} \models_{\mathrm{UF}} \overline{11})$
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge$$
$$\underbrace{1 \le x}_{4} \wedge \underbrace{x \le 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by T**-Propagate** $(0,\ 3 \models_{\mathrm{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{\mathrm{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ 11 \models_{\mathrm{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ \underline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T**-Conflict** $(7,\ 13 \models_{\mathrm{UF}} \perp)$
$0 \cdots 9\ \underline{10}\ \underline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ \underline{10}\ \underline{13}\ 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ \overline{13} \models_{\mathrm{UF}} \overline{11})$
$0 \cdots 9\ 10\ 13\ 11\ 12$	$F,\ 4 \vee 5 \vee 11 \vee 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge$$
$$\underbrace{1 \leq x}_{4} \wedge \underbrace{x \leq 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by T**-Propagate** $(0,\ \underline{3} \models_{UF} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{LIA} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ 11 \models_{UF} 13)$
$0 \cdots 9\ 10 \bullet 11\ \underline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T**-Conflict** $(7,\ 13 \models_{UF} \bot)$
$0 \cdots 9\ \underline{10}\ \underline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ \underline{10}\ \underline{13}\ 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ \overline{13} \models_{UF} \overline{11})$
$0 \cdots 9\ 10\ 13\ 11\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \begin{array}{l} \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge \\ \underbrace{1 \leq x}_{4} \wedge \underbrace{x \leq 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9} \end{array}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by T**-Propagate** $(0,\ \overline{3} \models_{\text{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{\text{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ 11 \models_{\text{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T**-Conflict** $(7,\ 13 \models_{\text{UF}} \bot)$
$0 \cdots 9\ 10\ \overline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ \overline{13} \models_{\text{UF}} \overline{11})$
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge$$
$$\underbrace{1 \le x}_{4} \wedge \underbrace{x \le 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^{+}$
$0 \cdots 9\ 10$	F	no	by **T-Propagate** $(0,\ 3 \models_{\mathrm{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{\mathrm{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **T-Propagate** $(0,\ 1,\ 11 \models_{\mathrm{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ \underline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by **T-Conflict** $(7,\ 13 \models_{\mathrm{UF}} \perp)$
$0 \cdots 9\ \underline{10}\ \underline{\overline{13}}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ \overline{\underline{10}}\ \overline{\underline{13}}\ 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **T-Propagate** $(0,\ 1,\ \overline{13} \models_{\mathrm{UF}} \overline{11})$
$0 \cdots 9\ 10\ 13\ 11\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \land \overbrace{f(x) = b}^{1} \land \overbrace{f(e_2) = e_3}^{2} \land \overbrace{f(e_1) = e_4}^{3} \land$$
$$\underbrace{1 \le x}_{4} \land \underbrace{x \le 2}_{5} \land \underbrace{e_1 = 1}_{6} \land \underbrace{a = b + 2}_{7} \land \underbrace{e_2 = 2}_{8} \land \underbrace{e_3 = e_4 + 3}_{9}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by **T-Propagate** $(0,\ 3 \models_{UF} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \lor \overline{5} \lor 11 \lor 12$	no	by **I-Learn** $(\models_{LIA} \overline{4} \lor \overline{5} \lor 11 \lor 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \lor \overline{5} \lor 11 \lor 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \lor \overline{5} \lor 11 \lor 12$	no	by **T-Propagate** $(0,\ 1,\ 11 \models_{UF} 13)$
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \lor \overline{5} \lor 11 \lor 12$	$\overline{7} \lor \overline{13}$	by **T-Conflict** $(7,\ 13 \models_{UF} \bot)$
$0 \cdots 9\ 10\ \overline{13}$	$F,\ \overline{4} \lor \overline{5} \lor 11 \lor 12$	no	by **Backjump**
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}$	$F,\ \overline{4} \lor \overline{5} \lor 11 \lor 12$	no	by **T-Propagate** $(0,\ 1,\ \overline{13} \models_{UF} \overline{11})$
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}\ 12$	$F,\ \overline{4} \lor \overline{5} \lor 11 \lor 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge$$
$$\underbrace{1 \le x}_{4} \wedge \underbrace{x \le 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by T**-Propagate** $(0,\ 3 \models_{\mathrm{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{\mathrm{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ 11 \models_{\mathrm{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T**-Conflict** $(7,\ 13 \models_{\mathrm{UF}} \perp)$
$0 \cdots 9\ 10\ \overline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ \overline{13} \models_{\mathrm{UF}} \overline{11})$
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge \underbrace{1 \le x}_{4} \wedge \underbrace{x \le 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^{+}$
$0 \cdots 9\ 10$	F	no	by T**-Propagate** $(0,\ 3 \models_{\mathrm{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{\mathrm{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ 11 \models_{\mathrm{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T**-Conflict** $(7,\ 13 \models_{\mathrm{UF}} \perp)$
$0 \cdots 9\ 10\ \overline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ \overline{13} \models_{\mathrm{UF}} \overline{11})$
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \begin{array}{l} \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge \\ \underbrace{1 \leq x}_{4} \wedge \underbrace{x \leq 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9} \end{array}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by T**-Propagate** $(0,\ 3 \models_{\mathrm{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \underline{\overline{4}} \vee \underline{\overline{5}} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{\mathrm{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \underline{\overline{4}} \vee \underline{\overline{5}} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \underline{\overline{4}} \vee \underline{\overline{5}} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ 11 \models_{\mathrm{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ \underline{13}$	$F,\ \underline{\overline{4}} \vee \underline{\overline{5}} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T**-Conflict** $(7,\ 13 \models_{\mathrm{UF}} \bot)$
$0 \cdots 9\ \underline{10}\ \underline{\overline{13}}$	$F,\ \underline{\overline{4}} \vee \underline{\overline{5}} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ \underline{10}\ \underline{\overline{13}}\ \overline{11}$	$F,\ \underline{\overline{4}} \vee \underline{\overline{5}} \vee 11 \vee 12$	no	by T**-Propagate** $(0,\ 1,\ \overline{13} \models_{\mathrm{UF}} \overline{11})$
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\cdots$			(exercise)
fail	$\cdots$	$\cdots$	by **Fail**

Example — Non-convex Theories

$$F := \begin{array}{l} \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge \\ \underbrace{1 \le x}_{4} \wedge \underbrace{x \le 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9} \end{array}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^+$
$0 \cdots 9\ 10$	F	no	by T-**Propagate** $(0,\ 3 \models_{\mathrm{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **I-Learn** $(\models_{\mathrm{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T-**Propagate** $(0,\ 1,\ 11 \models_{\mathrm{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T-**Conflict** $(7,\ 13 \models_{\mathrm{UF}} \bot)$
$0 \cdots 9\ 10\ \overline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T-**Propagate** $(0,\ 1,\ \overline{13} \models_{\mathrm{UF}} \overline{11})$
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\ldots$			(exercise)
fail	$\ldots$	$\ldots$	by **Fail**

Example — Non-convex Theories

$$F := \overbrace{f(e_1) = a}^{0} \wedge \overbrace{f(x) = b}^{1} \wedge \overbrace{f(e_2) = e_3}^{2} \wedge \overbrace{f(e_1) = e_4}^{3} \wedge$$
$$\underbrace{1 \leq x}_{4} \wedge \underbrace{x \leq 2}_{5} \wedge \underbrace{e_1 = 1}_{6} \wedge \underbrace{a = b + 2}_{7} \wedge \underbrace{e_2 = 2}_{8} \wedge \underbrace{e_3 = e_4 + 3}_{9}$$

$$\underbrace{a = e_4}_{10} \quad \underbrace{x = e_1}_{11} \quad \underbrace{x = e_2}_{12} \quad \underbrace{a = b}_{13}$$

M	F	C	rule
	F	no	
$0 \cdots 9$	F	no	by **Propagate**$^{+}$
$0 \cdots 9\ 10$	F	no	by T-**Propagate** $(0,\ 3 \models_{\mathrm{UF}} 10)$
$0 \cdots 9\ 10$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by I-**Learn** $(\models_{\mathrm{LIA}} \overline{4} \vee \overline{5} \vee 11 \vee 12)$
$0 \cdots 9\ 10 \bullet 11$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Decide**
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T-**Propagate** $(0,\ 1,\ 11 \models_{\mathrm{UF}} 13)$
$0 \cdots 9\ 10 \bullet 11\ 13$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	$\overline{7} \vee \overline{13}$	by T-**Conflict** $(7,\ 13 \models_{\mathrm{UF}} \bot)$
$0 \cdots 9\ 10\ \overline{13}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Backjump**
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by T-**Propagate** $(0,\ 1,\ \overline{13} \models_{\mathrm{UF}} \overline{11})$
$0 \cdots 9\ 10\ \overline{13}\ \overline{11}\ 12$	$F,\ \overline{4} \vee \overline{5} \vee 11 \vee 12$	no	by **Propagate**
$\dots$			(exercise)
fail	$\dots$	$\dots$	by **Fail**

Theory Solvers

Theory Solvers

Given a theory T, a *Theory Solver* for T takes as input a set Φ of literals and determines whether Φ is T-satisfiable.

Φ is T-satisfiable iff there is some model M of T such that each formula in Φ holds in M.

Theories of Interest: UF

Equality ($=$) with Uninterpreted Functions [NO80, BD94, NO07]

Typically used to abstract unsupported constructs, e.g.:

- non-linear multiplication in arithmetic
- ALUs in circuits

Example: The formula

$$a * (|b| + c) = d \quad \wedge \quad b * (|a| + c) \neq d \quad \wedge \quad a = b$$

is unsatisfiable, but no arithmetic reasoning is needed

if we abstract it to

$$mul(a, add(abs(b), c)) = d \quad \wedge \quad mul(b, add(abs(a), c)) \neq d \quad \wedge \quad a = b$$

Theories of Interest: Arithmetic

Very useful, for obvious reasons

Restricted fragments (over the reals or the integers) support more efficient methods:

- Bounds: $x \bowtie k$ with $\bowtie \in \{<, >, \leq, \geq, =\}$ [BBC+05a]
- Difference logic: $x - y \bowtie k$, with $\bowtie \in \{<, >, \leq, \geq, =\}$ [NO05, WIGG05, CM06]
- UTVPI: $\pm x \pm y \bowtie k$, with $\bowtie \in \{<, >, \leq, \geq, =\}$ [LM05]
- Linear arithmetic, e.g: $2x - 3y + 4z \leq 5$ [DdM06]
- Non-linear arithmetic, e.g: $2xy + 4xz^2 - 5y \leq 10$ [BLNM+09, ZM10, JdM12]

Theories of Interest: Arrays

Used in software verification and hardware verification (for memories) [SBDL01, BNO+08a, dMB09]

Two interpreted function symbols read and write

Axiomatized by:

- $\forall a\, \forall i\, \forall v\ \mathrm{read}(\mathrm{write}(a, i, v), i) = v$
- $\forall a\, \forall i\, \forall j\, \forall v\ i \neq j \rightarrow \mathrm{read}(\mathrm{write}(a, i, v), j) = \mathrm{read}(a, j)$

Sometimes also with *extensionality*:

- $\forall a\, \forall b\ (\forall i\, \mathrm{read}(a, i) = \mathrm{read}(b, i)\ \rightarrow a = b)$

Is the following set of literals satisfiable in this theory?

$$\mathrm{write}(a, i, x) \neq b,\ \mathrm{read}(b, i) = y,\ \mathrm{read}(\mathrm{write}(b, i, x), j) = y,\ a = b,\ i = j$$

Theories of Interest: Bit vectors

Useful both in hardware and software verification [BCF+07, BB09, HBJ+14]

Universe consists of (fixed-sized) vectors of bits

Different types of operations:

- *String-like*: concat, extract, ...
- *Logical*: bit-wise not, or, and, ...
- *Arithmetic*: add, subtract, multiply, ...
- *Comparison*: $<, >, \ldots$

Is this formula satisfiable over bit vectors of size 3?

$$a[1:0] \neq b[1:0] \;\wedge\; (a \mid b) = c \;\wedge\; c[0] = 0 \;\wedge\; a[1] + b[1] = 0$$

Implementing a Theory Solver: Difference Logic

We consider a simple example: difference logic.

In *difference logic*, we are interested in the satisfiability of a conjunction of arithmetic atoms.

Each atom is of the form $x - y \bowtie c$, where x and y are variables, c is a numeric constant, and $\bowtie \in \{=, <, \leq, >, \geq\}$.

The variables can range over either the *integers* (QF_IDL) or the *reals* (QF_RDL).

Difference Logic

The first step is to rewrite everything in terms of $\leq$:

Difference Logic

The first step is to rewrite everything in terms of $\leq$:

- $x - y = c \quad \Longrightarrow \quad x - y \leq c \land x - y \geq c$

Difference Logic

The first step is to rewrite everything in terms of $\leq$:

- $x - y = c \quad \implies \quad x - y \leq c \wedge x - y \geq c$
- $x - y \geq c \quad \implies \quad y - x \leq -c$

Difference Logic

The first step is to rewrite everything in terms of $\leq$:

- $x - y = c \quad \implies \quad x - y \leq c \land x - y \geq c$
- $x - y \geq c \quad \implies \quad y - x \leq -c$
- $x - y > c \quad \implies \quad y - x < -c$

Difference Logic

The first step is to rewrite everything in terms of $\leq$:

- $x - y = c \quad \implies \quad x - y \leq c \wedge x - y \geq c$
- $x - y \geq c \quad \implies \quad y - x \leq -c$
- $x - y > c \quad \implies \quad y - x < -c$
- $x - y < c \quad \implies \quad x - y \leq c - 1$ (integers)

Difference Logic

The first step is to rewrite everything in terms of $\leq$:

- $x - y = c \quad \implies \quad x - y \leq c \wedge x - y \geq c$
- $x - y \geq c \quad \implies \quad y - x \leq -c$
- $x - y > c \quad \implies \quad y - x < -c$
- $x - y < c \quad \implies \quad x - y \leq c - 1$ (integers)
- $x - y < c \quad \implies \quad x - y \leq c - \delta$ (reals)

Difference Logic

Now we have a conjunction of literals, all of the form $x - y \leq c$.

From these literals, we form a weighted directed graph with a vertex for each variable.

For each literal $x - y \leq c$, there is an edge $x \xrightarrow{c} y$.

The set of literals is satisfiable iff there is no cycle for which the sum of the weights on the edges is negative.

There are a number of efficient algorithms for detecting negative cycles in graphs.

Difference Logic Example

$$x - y = 5 \ \wedge \ z - y \geq 2 \ \wedge \ z - x > 2 \ \wedge \ w - x = 2 \ \wedge \ z - w < 0$$

Difference Logic Example

$$x - y = 5 \;\wedge\; z - y \geq 2 \;\wedge\; z - x > 2 \;\wedge\; w - x = 2 \;\wedge\; z - w < 0$$

$$\begin{aligned}
&x - y = 5 \\
&z - y \geq 2 \\
&z - x > 2 \\
&w - x = 2 \\
&z - w < 0
\end{aligned}$$

Difference Logic Example

$$x - y = 5 \ \wedge \ z - y \geq 2 \ \wedge \ z - x > 2 \ \wedge \ w - x = 2 \ \wedge \ z - w < 0$$

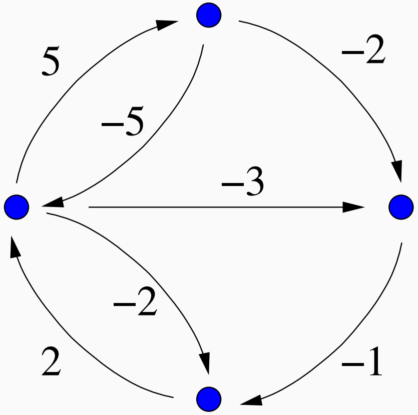
$$\begin{aligned} x - y &= 5 \\ z - y &\geq 2 \\ z - x &> 2 \qquad \Rightarrow \\ w - x &= 2 \\ z - w &< 0 \end{aligned}$$

Difference Logic Example

$$x - y = 5 \ \wedge \ z - y \geq 2 \ \wedge \ z - x > 2 \ \wedge \ w - x = 2 \ \wedge \ z - w < 0$$

$$\begin{array}{lcl} x - y = 5 & & x - y \leq 5 \wedge y - x \leq -5 \\ z - y \geq 2 & & y - z \leq -2 \\ z - x > 2 & \Rightarrow & x - z \leq -3 \\ w - x = 2 & & w - x \leq 2 \wedge x - w \leq -2 \\ z - w < 0 & & z - w \leq -1 \end{array}$$

Difference Logic Example



Suggested Readings

1. R. Nieuwenhuis, A. Oliveras, and C. Tinelli. **Solving SAT and SAT Modulo Theories: From an abstract Davis-Putnam-Logemann- Loveland procedure to DPLL(T)**. Journal of the ACM, 53(6):937-977, 2006.

2. R. Sebastiani. **Lazy Satisfiability Modulo Theories**. Journal on Satisfiability, Boolean Modeling and Computation 3:141-224, 2007.

3. S. Krstić and A. Goel. **Architecting Solvers for SAT Modulo Theories: Nelson-Oppen with DPLL**. In Proceeding of the Symposium on Frontiers of Combining Systems (FroCoS'07). Volume 4720 of LNCS. Springer, 2007.

4. C. Barrett, R. Sebastiani, S. Seshia, and C. Tinelli. **Satisfiability Modulo Theories**. In Handbook of Satisfiability. IOS Press, 2009.

References

[ABC+02] Gilles Audemard, Piergiorgio Bertoli, Alessandro Cimatti, Artur Korniłowicz, and Roberto Sebastiani.
A SAT-based approach for solving formulas over boolean and linear mathematical propositions.
In Andrei Voronkov, editor, *Proceedings of the 18th International Conference on Automated Deduction*, volume 2392 of *Lecture Notes in Artificial Intelligence*, pages 195–210. Springer, 2002

[ACG00] Alessandro Armando, Claudio Castellini, and Enrico Giunchiglia. SAT-based procedures for temporal reasoning.
In S. Biundo and M. Fox, editors, *Proceedings of the 5th European Conference on Planning (Durham, UK)*, volume 1809 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 2000

[AMP06] Alessandro Armando, Jacopo Mantovani, and Lorenzo Platania. Bounded model checking of software using SMT solvers instead of SAT solvers.
In *Proceedings of the 13th International SPIN Workshop on Model Checking of Software (SPIN'06)*, volume 3925 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2006

[Bar02] Clark W. Barrett. *Checking Validity of Quantifier-Free Formulas in Combinations of First-Order Theories*.
PhD dissertation, Department of Computer Science, Stanford University, Stanford, CA, Sep 2002

References

[BB09] R. Brummayer and A. Biere. Boolector: An Efficient SMT Solver for Bit-Vectors and Arrays. In S. Kowalewski and A. Philippou, editors, *15th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'05*, volume 5505 of *Lecture Notes in Computer Science*, pages 174–177. Springer, 2009

[BBC+05a] M. Bozzano, R. Bruttomesso, A. Cimatti, T. Junttila, P. van Rossum, S. Schulz, and R. Sebastiani. An incremental and layered procedure for the satisfiability of linear arithmetic logic. In *Tools and Algorithms for the Construction and Analysis of Systems, 11th Int. Conf., (TACAS)*, volume 3440 of *Lecture Notes in Computer Science*, pages 317–333, 2005

[BBC+05b] Marco Bozzano, Roberto Bruttomesso, Alessandro Cimatti, Tommi Junttila, Silvio Ranise, Roberto Sebastiani, and Peter van Rossu. Efficient satisfiability modulo theories via delayed theory combination. In K.Etessami and S. Rajamani, editors, *Proceedings of the 17th International Conference on Computer Aided Verification*, volume 3576 of *Lecture Notes in Computer Science*, pages 335–349. Springer, 2005

References

[BCF+07] Roberto Bruttomesso, Alessandro Cimatti, Anders Franzén, Alberto Griggio, Ziyad Hanna, Alexander Nadel, Amit Palti, and Roberto Sebastiani. A lazy and layered SMT(BV) solver for hard industrial verification problems.
In Werner Damm and Holger Hermanns, editors, *Proceedings of the* 19^{th} *International Conference on Computer Aided Verification*, volume 4590 of *Lecture Notes in Computer Science*, pages 547–560. Springer-Verlag, July 2007

[BCLZ04] Thomas Ball, Byron Cook, Shuvendu K. Lahiri, and Lintao Zhang. Zapato: Automatic theorem proving for predicate abstraction refinement.
In R. Alur and D. Peled, editors, *Proceedings of the 16th International Conference on Computer Aided Verification*, volume 3114 of *Lecture Notes in Computer Science*, pages 457–461. Springer, 2004

[BD94] J. R. Burch and D. L. Dill. Automatic verification of pipelined microprocessor control.
In *Procs. 6th Int. Conf. Computer Aided Verification (CAV)*, LNCS 818, pages 68–80, 1994

[BDS02] Clark W. Barrett, David L. Dill, and Aaron Stump. Checking satisfiability of first-order formulas by incremental translation to SAT.
In J. C. Godskesen, editor, *Proceedings of the International Conference on Computer-Aided Verification*, Lecture Notes in Computer Science, 2002

References

[BGV01] R. E. Bryant, S. M. German, and M. N. Velev. Processor Verification Using Efficient Reductions of the Logic of Uninterpreted Functions to Propositional Logic.
ACM Transactions on Computational Logic, TOCL, 2(1):93–134, 2001

[BLNM+09] C. Borralleras, S. Lucas, R. Navarro-Marset, E. Rodríguez-Carbonell, and A. Rubio. Solving Non-linear Polynomial Arithmetic via SAT Modulo Linear Arithmetic.
In R. A. Schmidt, editor, *22nd International Conference on Automated Deduction , CADE-22*, volume 5663 of *Lecture Notes in Computer Science*, pages 294–305. Springer, 2009

[BLS02] Randal E. Bryant, Shuvendu K. Lahiri, and Sanjit A. Seshia. Deciding CLU logic formulas via boolean and pseudo-boolean encodings.
In *Proc. Intl. Workshop on Constraints in Formal Verification*, 2002

[BNO+08a] M. Bofill, R. Nieuwenhuis, A. Oliveras, E. Rodríguez-Carbonell, and A. Rubio. A Write-Based Solver for SAT Modulo the Theory of Arrays.
In *Formal Methods in Computer-Aided Design, FMCAD*, pages 1–8, 2008

[BNO+08b] Miquel Bofill, Robert Nieuwenhuis, Albert Oliveras, Enric Rodríguez-Carbonell, and Albert Rubio.
The Barcelogic SMT solver.
In *Computer-aided Verification (CAV)*, volume 5123 of *Lecture Notes in Computer Science*, pages 294–298. Springer, 2008

References

[BNOT06] Clark Barrett, Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. Splitting on demand in sat modulo theories.
In M. Hermann and A. Voronkov, editors, *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR'06), Phnom Penh, Cambodia*, volume 4246 of *Lecture Notes in Computer Science*, pages 512–526. Springer, 2006

[BV02] R. E. Bryant and M. N. Velev. Boolean Satisfiability with Transitivity Constraints.
ACM Transactions on Computational Logic, TOCL, 3(4):604–627, 2002

[CKSY04] Edmund Clarke, Daniel Kroening, Natasha Sharygina, and Karen Yorav. Predicate abstraction of ANSI–C programs using SAT.
Formal Methods in System Design (FMSD), 25:105–127, September–November 2004

[CM06] S. Cotton and O. Maler. Fast and Flexible Difference Constraint Propagation for DPLL(T).
In A. Biere and C. P. Gomes, editors, *9th International Conference on Theory and Applications of Satisfiability Testing, SAT'06*, volume 4121 of *Lecture Notes in Computer Science*, pages 170–183. Springer, 2006

References

[DdM06] Bruno Dutertre and Leonardo de Moura. A Fast Linear-Arithmetic Solver for DPLL(T). In T. Ball and R. B. Jones, editors, *18th International Conference on Computer Aided Verification, CAV'06*, volume 4144 of *Lecture Notes in Computer Science*, pages 81–94. Springer, 2006

[DLL62] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, July 1962

[dMB09] L. de Moura and N. Bjørner. Generalized, efficient array decision procedures. In *9th International Conference on Formal Methods in Computer-Aided Design, FMCAD 2009*, pages 45–52. IEEE, 2009

[dMR02] L. de Moura and H. Rueß. Lemmas on Demand for Satisfiability Solvers. In *5th International Conference on Theory and Applications of Satisfiability Testing, SAT'02*, pages 244–251, 2002

[DP60] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, July 1960

[FLL+02] C. Flanagan, K. R. M Leino, M. Lillibridge, G. Nelson, and J. B. Saxe. Extended static checking for Java. In *Proc. ACM Conference on Programming Language Design and Implementation*, pages 234–245, June 2002

References

[GHN⁺04] Harald Ganzinger, George Hagen, Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli.

DPLL(T): Fast decision procedures.
In R. Alur and D. Peled, editors, *Proceedings of the 16th International Conference on Computer Aided Verification, CAV'04 (Boston, Massachusetts)*, volume 3114 of *Lecture Notes in Computer Science*, pages 175–188. Springer, 2004

[HBJ⁺14] Liana Hadarean, Clark Barrett, Dejan Jovanović, Cesare Tinelli, and Kshitij Bansal. A tale of two solvers: Eager and lazy approaches to bit-vectors.
In Armin Biere and Roderick Bloem, editors, *Proceedings of the* 26^{th} *International Conference on Computer Aided Verification (CAV '14)*, volume 8559 of *Lecture Notes in Computer Science*, pages 680–695. Springer, July 2014

[HT08] George Hagen and Cesare Tinelli. Scaling up the formal verification of Lustre programs with SMT-based techniques.
In A. Cimatti and R. Jones, editors, *Proceedings of the 8th International Conference on Formal Methods in Computer-Aided Design (FMCAV'08), Portland, Oregon*, pages 109–117. IEEE, 2008

[JdM12] Dejan Jovanović and Leonardo de Moura. Solving Non-linear Arithmetic.
In Bernhard Gramlich, Dale Miller, and Uli Sattler, editors, *6th International Joint Conference on Automated Reasoning (IJCAR '12)*, volume 7364 of *Lecture Notes in Computer Science*, pages 339–354. Springer, 2012

[JB10] Dejan Jovanović and Clark Barrett. Polite theories revisited.
In Chris Fermüller and Andrei Voronkov, editors, *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 6397 of *Lecture Notes in Computer Science*, pages 402–416. Springer-Verlag, 2010

[KG07] Sava Krstić and Amit Goel. Architecting solvers for SAT modulo theories: Nelson-Oppen with DPLL.

In B. Konev and F. Wolter, editors, *Proceeding of the Symposium on Frontiers of Combining Systems (Liverpool, England)*, volume 4720 of *Lecture Notes in Computer Science*, pages 1–27. Springer, 2007

References

[LM05] Shuvendu K. Lahiri and Madanlal Musuvathi. An Efficient Decision Procedure for UTVPI Constraints.
In B. Gramlich, editor, *5th International Workshop on Frontiers of Combining Systems, FroCos'05*, volume 3717 of *Lecture Notes in Computer Science*, pages 168–183. Springer, 2005

[LNO06] S. K. Lahiri, R. Nieuwenhuis, and A. Oliveras. SMT Techniques for Fast Predicate Abstraction.
In T. Ball and R. B. Jones, editors, *18th International Conference on Computer Aided Verification, CAV'06*, volume 4144 of *Lecture Notes in Computer Science*, pages 413–426. Springer, 2006

[NO79] Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures.
ACM Trans. on Programming Languages and Systems, 1(2):245–257, October 1979

[NO80] Greg Nelson and Derek C. Oppen. Fast decision procedures based on congruence closure.
Journal of the ACM, 27(2):356–364, 1980

[NO05] Robert Nieuwenhuis and Albert Oliveras. DPLL(T) with Exhaustive Theory Propagation and its Application to Difference Logic.
In Kousha Etessami and Sriram K. Rajamani, editors, *Proceedings of the 17th International Conference on Computer Aided Verification, CAV'05 (Edimburgh, Scotland)*, volume 3576 of *Lecture Notes in Computer Science*, pages 321–334. Springer, July 2005

References

[NO07] R. Nieuwenhuis and A. Oliveras. Fast Congruence Closure and Extensions. *Information and Computation, IC*, 2005(4):557–580, 2007

[NOT06] Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. Solving SAT and SAT Modulo Theories: from an Abstract Davis-Putnam-Logemann-Loveland Procedure to DPLL(T). *Journal of the ACM*, 53(6):937–977, November 2006

[Opp80] Derek C. Oppen. Complexity, convexity and combinations of theories. *Theoretical Computer Science*, 12:291–302, 1980

[PRSS99] A. Pnueli, Y. Rodeh, O. Shtrichman, and M. Siegel. Deciding Equality Formulas by Small Domains Instantiations. In N. Halbwachs and D. Peled, editors, *11th International Conference on Computer Aided Verification, CAV'99*, volume 1633 of *Lecture Notes in Computer Science*, pages 455–469. Springer, 1999

[Rin96] Christophe Ringeissen. Cooperation of decision procedures for the satisfiability problem. In F. Baader and K.U. Schulz, editors, *Frontiers of Combining Systems: Proceedings of the 1st International Workshop, Munich (Germany)*, Applied Logic, pages 121–140. Kluwer Academic Publishers, March 1996

References

[RRZ05] Silvio Ranise, Christophe Ringeissen, and Calogero G. Zarba. Combining data structures with nonstably infinite theories using many-sorted logic.
In B. Gramlich, editor, *Proceedings of the Workshop on Frontiers of Combining Systems*, volume 3717 of *Lecture Notes in Computer Science*, pages 48–64. Springer, 2005

[SBDL01] A. Stump, C. W. Barrett, D. L. Dill, and J. R. Levitt. A Decision Procedure for an Extensional Theory of Arrays.
In *16th Annual IEEE Symposium on Logic in Computer Science, LICS'01*, pages 29–37. IEEE Computer Society, 2001

[Sha02] Natarajan Shankar. Little engines of proof.
In Lars-Henrik Eriksson and Peter A. Lindsay, editors, *FME 2002: Formal Methods - Getting IT Right, Proceedings of the International Symposium of Formal Methods Europe (Copenhagen, Denmark)*, volume 2391 of *Lecture Notes in Computer Science*, pages 1–20. Springer, July 2002

[SLB03] Sanjit A. Seshia, Shuvendu K. Lahiri, and Randal E. Bryant. A hybrid SAT-based decision procedure for separation logic with uninterpreted functions.
In *Proc. 40th Design Automation Conference*, pages 425–430. ACM Press, 2003

[SSB02] O. Strichman, S. A. Seshia, and R. E. Bryant. Deciding Separation Formulas with SAT.
In E. Brinksma and K. G. Larsen, editors, *14th International Conference on Computer Aided Verification, CAV'02*, volume 2404 of *Lecture Notes in Computer Science*, pages 209–222. Springer, 2002

References

[TdH08] N. Tillmann and J. de Halleux. Pex-White Box Test Generation for .NET. In B. Beckert and R. Hähnle, editors, *2nd International Conference on Tests and Proofs, TAP'08*, volume 4966 of *Lecture Notes in Computer Science*, pages 134–153. Springer, 2008

[TH96] Cesare Tinelli and Mehdi T. Harandi. A new correctness proof of the Nelson–Oppen combination procedure. In F. Baader and K. U. Schulz, editors, *Frontiers of Combining Systems: Proceedings of the 1st International Workshop (Munich, Germany)*, Applied Logic, pages 103–120. Kluwer Academic Publishers, March 1996

[Tin02] C. Tinelli. A DPLL-based calculus for ground satisfiability modulo theories. In G. Ianni and S. Flesca, editors, *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (Cosenza, Italy)*, volume 2424 of *Lecture Notes in Artificial Intelligence*. Springer, 2002

[TZ05] Cesare Tinelli and Calogero Zarba. Combining nonstably infinite theories. *Journal of Automated Reasoning*, 34(3):209–238, April 2005

References

[WIGG05] C. Wang, F. Ivancic, M. K. Ganai, and A. Gupta. Deciding Separation Logic Formulae by SAT and Incremental Negative Cycle Elimination.
In G. Sutcliffe and A. Voronkov, editors, *12h International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR'05*, volume 3835 of *Lecture Notes in Computer Science*, pages 322–336. Springer, 2005

[ZM10] Harald Zankl and Aart Middeldorp. Satisfiability of Non-linear (Ir)rational Arithmetic.
In Edmund M. Clarke and Andrei Voronkov, editors, *16th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR'10*, volume 6355 of *Lecture Notes in Computer Science*, pages 481–500. Springer, 2010