Model-Constructing Satisfiability Calculus

A Model-Based Approach to SMT

Dejan Jovanović

SRI International

SAT/SMT Summer School, Stanford, 2015

Outline

Outline

Satisfiability Modulo Theories and DPLL(T)

Problem

Check a given formula for satisfiability modulo the union of background theories.

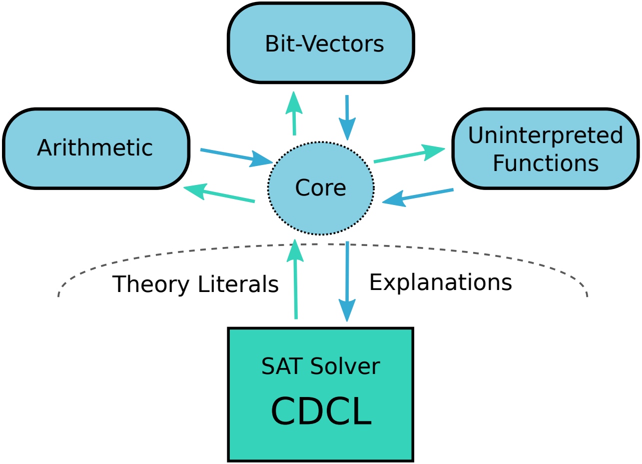
Example (QF_UFLRA)

$$(z = 1 \lor z = 0) \land (x - y + z = 1) \land (f(x) > f(y))$$

Main idea behind DPLL(T):

1. Use a SAT solver to enumerate the Boolean structure.
2. Check Boolean assignments with a decision procedure.

DPLL(T) Architecture



DPLL(T): Pros and Cons

Good

- Simple interface
- Only conjunctions of constraints
- General combination methods

DPLL(T): Pros and Cons

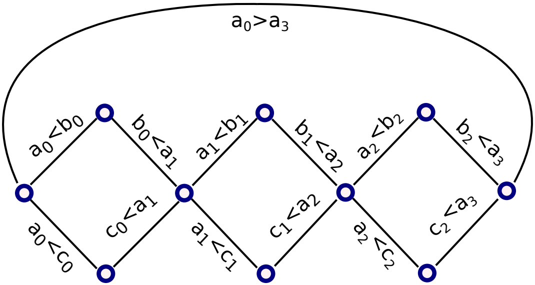
Good

- Simple interface
- Only conjunctions of constraints
- General combination methods

What can be improved?

- Simple interface can be restrictive
- Arbitrary conjunctions of constraints
- General combination methods

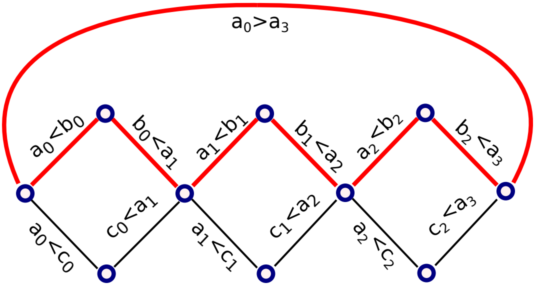
DPLL(T): Simple Interface Issues



Example (Diamonds)

$$a_0 > a_n \wedge \bigwedge_{k=0}^{n-1} ((a_k < b_k \wedge b_k < a_{k+1}) \vee (a_k < c_k \wedge c_k < a_{k+1}))$$

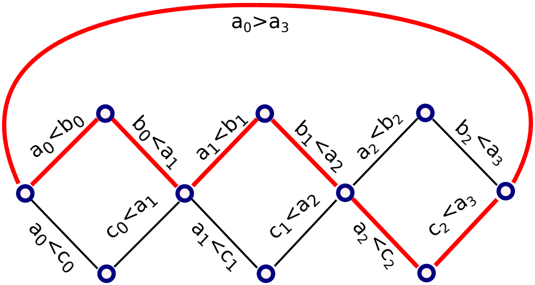
DPLL(T): Simple Interface Issues



Example (Diamonds)

$$a_0 > a_n \wedge \bigwedge_{k=0}^{n-1} ((a_k < b_k \wedge b_k < a_{k+1}) \vee (a_k < c_k \wedge c_k < a_{k+1}))$$

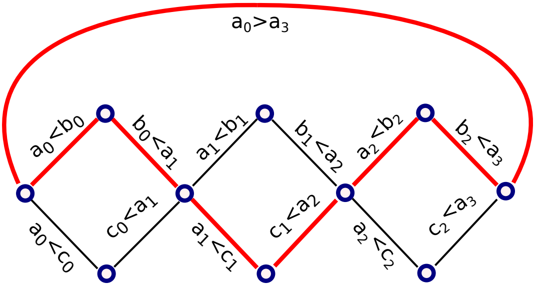
DPLL(T): Simple Interface Issues



Example (Diamonds)

$$a_0 > a_n \wedge \bigwedge_{k=0}^{n-1} \left((a_k < b_k \wedge b_k < a_{k+1}) \vee (a_k < c_k \wedge c_k < a_{k+1}) \right)$$

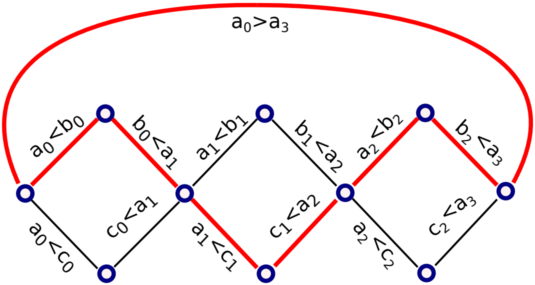
DPLL(T): Simple Interface Issues



Example (Diamonds)

$$a_0 > a_n \wedge \bigwedge_{k=0}^{n-1} ((a_k < b_k \wedge b_k < a_{k+1}) \vee (a_k < c_k \wedge c_k < a_{k+1}))$$

DPLL(T): Simple Interface Issues



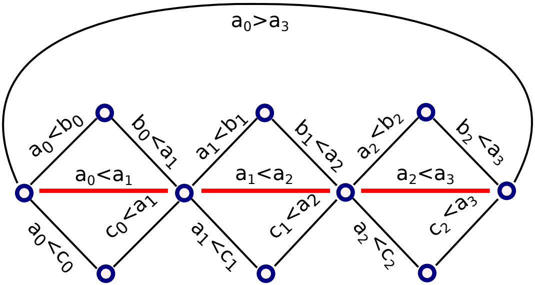
And so on...

Exponential enumeration of paths.

Example (D

$$a_0 > a_n \wedge \bigwedge_{k=0}^{n-1} ((a_k < b_k \wedge b_k < a_{k+1}) \vee (a_k < c_k \wedge c_k < a_{k+1}))$$

DPLL(T): Simple Interface Issues



Example (Diamonds)

$$a_0 > a_n \wedge \bigwedge_{k=0}^{n-1} ((a_k < b_k \wedge b_k < a_{k+1}) \vee (a_k < c_k \wedge c_k < a_{k+1}))$$

DPPL(T): Simple Interface Issues

How to fix this?

- Extensions of DPLL(T) can add new literals [BNOT06].
- Magic needed to discover these literals [BDdM08, HAMM14].
- More pragmatic approach would be desirable.

Rethink the Architecture!

- Why is SAT solver special?
- Why the restriction on the interface?
- Let’s dig deeper into how SAT solvers work.

Boolean Satisfiability

$$\boxed{x_n \vee \cdots \vee x_1 \vee \overline{y_m} \vee \cdots \vee \overline{y_1}}$$

- **Resolution procedure** by Davis, Putnam [DP60]
- **Search procedure** by Davis, Logemann, Loveland [DLL62]

Resolution (DP)

- Find a proof
- Saturation
- Exponential

Search (DLL)

- Find a model
- Search and backtracking
- Exponential

Resolution (DP)

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

Resolution (DP)

$\boxed{x \vee y \vee \bar{z}}$	$\boxed{x \vee \bar{y} \vee \bar{z}}$	$\boxed{\bar{x} \vee y \vee \bar{z}}$	$\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$
$\boxed{x \vee y \vee z}$	$\boxed{x \vee \bar{y} \vee z}$	$\boxed{\bar{x} \vee y \vee z}$	$\boxed{\bar{x} \vee \bar{y} \vee z}$

Boolean Resolution

$$\frac{C \vee x \qquad D \vee \bar{x}}{C \vee D}$$

Resolution (DP)

$x \vee y \vee \bar{z}$	$x \vee \bar{y} \vee \bar{z}$	$\bar{x} \vee y \vee \bar{z}$	$\bar{x} \vee \bar{y} \vee \bar{z}$
$x \vee y \vee z$	$x \vee \bar{y} \vee z$	$\bar{x} \vee y \vee z$	$\bar{x} \vee \bar{y} \vee z$

1. Eliminate z

Resolution (DP)

$\boxed{x \vee y \vee \bar{z}}$	$\boxed{x \vee \bar{y} \vee \bar{z}}$	$\boxed{\bar{x} \vee y \vee \bar{z}}$	$\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$
$\boxed{x \vee y \vee z}$	$\boxed{x \vee \bar{y} \vee z}$	$\boxed{\bar{x} \vee y \vee z}$	$\boxed{\bar{x} \vee \bar{y} \vee z}$
$\boxed{x \vee y}$	$\boxed{x \vee \bar{y}}$	$\boxed{\bar{x} \vee y}$	$\boxed{\bar{x} \vee \bar{y}}$

1. Eliminate z

Resolution (DP)

$\boxed{x \vee y \vee \bar{z}}$	$\boxed{x \vee \bar{y} \vee \bar{z}}$	$\boxed{\bar{x} \vee y \vee \bar{z}}$	$\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$
$\boxed{x \vee y \vee z}$	$\boxed{x \vee \bar{y} \vee z}$	$\boxed{\bar{x} \vee y \vee z}$	$\boxed{\bar{x} \vee \bar{y} \vee z}$
$\boxed{x \vee y}$	$\boxed{x \vee \bar{y}}$	$\boxed{\bar{x} \vee y}$	$\boxed{\bar{x} \vee \bar{y}}$

1. Eliminate z
2. Eliminate y

Resolution (DP)

$$\boxed{x \vee y \vee \bar{z}} \quad \boxed{x \vee \bar{y} \vee \bar{z}} \quad \boxed{\bar{x} \vee y \vee \bar{z}} \quad \boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$$

$$\boxed{x \vee y \vee z} \quad \boxed{x \vee \bar{y} \vee z} \quad \boxed{\bar{x} \vee y \vee z} \quad \boxed{\bar{x} \vee \bar{y} \vee z}$$

$$\boxed{x \vee y} \quad \boxed{x \vee \bar{y}} \quad \boxed{\bar{x} \vee y} \quad \boxed{\bar{x} \vee \bar{y}}$$

$$\boxed{x} \quad \boxed{\bar{x}}$$

1. Eliminate z
2. Eliminate y

Resolution (DP)

$x \vee y \vee \bar{z}$	$x \vee \bar{y} \vee \bar{z}$	$\bar{x} \vee y \vee \bar{z}$	$\bar{x} \vee \bar{y} \vee \bar{z}$
$x \vee y \vee z$	$x \vee \bar{y} \vee z$	$\bar{x} \vee y \vee z$	$\bar{x} \vee \bar{y} \vee z$
$x \vee y$	$x \vee \bar{y}$	$\bar{x} \vee y$	$\bar{x} \vee \bar{y}$
	x	$\bar{x}$	

1. Eliminate z
2. Eliminate y
3. Eliminate x

Resolution (DP)

$\boxed{x \vee y \vee \bar{z}}$	$\boxed{x \vee \bar{y} \vee \bar{z}}$	$\boxed{\bar{x} \vee y \vee \bar{z}}$	$\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$
$\boxed{x \vee y \vee z}$	$\boxed{x \vee \bar{y} \vee z}$	$\boxed{\bar{x} \vee y \vee z}$	$\boxed{\bar{x} \vee \bar{y} \vee z}$
$\boxed{x \vee y}$	$\boxed{x \vee \bar{y}}$	$\boxed{\bar{x} \vee y}$	$\boxed{\bar{x} \vee \bar{y}}$
	$\boxed{x}$	$\boxed{\bar{x}}$	

1. Eliminate z
2. Eliminate y
3. Eliminate x
4. **unsat**

Search (DLL)

$x \vee y \vee \bar{z}$	$x \vee \bar{y} \vee \bar{z}$	$\bar{x} \vee y \vee \bar{z}$	$\bar{x} \vee \bar{y} \vee \bar{z}$
$x \vee y \vee z$	$x \vee \bar{y} \vee z$	$\bar{x} \vee y \vee z$	$\bar{x} \vee \bar{y} \vee z$

Search (DLL)

$x \vee y \vee \bar{z}$	$x \vee \bar{y} \vee \bar{z}$	$\bar{x} \vee y \vee \bar{z}$	$\bar{x} \vee \bar{y} \vee \bar{z}$
$x \vee y \vee z$	$x \vee \bar{y} \vee z$	$\bar{x} \vee y \vee z$	$\bar{x} \vee \bar{y} \vee z$

1. try $x \mapsto \top$

Search (DLL)

$y \vee \bar{z}$	$\bar{y} \vee \bar{z}$
$y \vee z$	$\bar{y} \vee z$

1. try $x \mapsto \top$
   1. try $y \mapsto \top$

Search (DLL)

$\boxed{\quad\quad \bar{z}}$

$\boxed{\quad\quad z}$

1. try $x \mapsto \top$
   1. try $y \mapsto \top$

Search (DLL)

$\boxed{\bar{z}}$

$\boxed{z}$

1. try $x \mapsto \top$
    1. try $y \mapsto \top$, unsat

Search (DLL)

$y \vee \bar{z}$	$\bar{y} \vee \bar{z}$
$y \vee z$	$\bar{y} \vee z$

1. try $x \mapsto \top$
   1. try $y \mapsto \top$, unsat
   2. try $y \mapsto \bot$

Search (DLL)

$\boxed{\bar{z}}$

$\boxed{z}$

1. try $x \mapsto \top$
   1. try $y \mapsto \top$, unsat
   2. try $y \mapsto \bot$

Search (DLL)

$\bar{z}$

z

1. try $x \mapsto \top$
    1. try $y \mapsto \top$, unsat
    2. try $y \mapsto \bot$, unsat

Search (DLL)

$\boxed{x \vee y \vee \bar{z}}$	$\boxed{x \vee \bar{y} \vee \bar{z}}$	$\boxed{\bar{x} \vee y \vee \bar{z}}$	$\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$
$\boxed{x \vee y \vee z}$	$\boxed{x \vee \bar{y} \vee z}$	$\boxed{\bar{x} \vee y \vee z}$	$\boxed{\bar{x} \vee \bar{y} \vee z}$

1. try $x \mapsto \top$
   1. try $y \mapsto \top$, unsat
   2. try $y \mapsto \bot$, unsat
2. try $x \mapsto \bot \ldots$

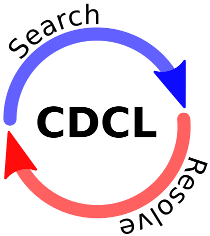
Boolean Satisfiability: CDCL

Marques-Silva, Sakallah
[SS97] GRASP: A new search algorithm for satisfiabiliy

Moskewicz, Madigan, Zhao, Zhang, Malik
[MMZ+01] CHAFF: Engineering an efficient SAT solver

Conflict-Directed Clause Learning

- Use the search to guide resolution
- Use resolution to guide the search



Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$	$\boxed{x \vee \bar{y} \vee \bar{z}}$	$\boxed{\bar{x} \vee y \vee \bar{z}}$	$\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$
$\boxed{x \vee y \vee z}$	$\boxed{x \vee \bar{y} \vee z}$	$\boxed{\bar{x} \vee y \vee z}$	$\boxed{\bar{x} \vee \bar{y} \vee z}$

$[\![\,]\!]$

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$	$\boxed{x \vee \bar{y} \vee \bar{z}}$	$\boxed{\bar{x} \vee y \vee \bar{z}}$	$\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$
$\boxed{x \vee y \vee z}$	$\boxed{x \vee \bar{y} \vee z}$	$\boxed{\bar{x} \vee y \vee z}$	$\boxed{\bar{x} \vee \bar{y} \vee z}$

$[\![x]\!]$

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

$$[\![x, y]\!]$$

Boolean Satisfiability: CDCL

$x \vee y \vee \bar{z}$	$x \vee \bar{y} \vee \bar{z}$	$\bar{x} \vee y \vee \bar{z}$	$\bar{x} \vee \bar{y} \vee \bar{z}$
$x \vee y \vee z$	$x \vee \bar{y} \vee z$	$\bar{x} \vee y \vee z$	$\bar{x} \vee \bar{y} \vee z$

$[\![x, y]\!]$

Unit Propagation

$(\bar{x} \vee \bar{y} \vee \bar{z})$ is unit, propagate $\bar{z}$.

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

$$[\![x, y, \bar{z}]\!]$$

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

$[\![x, y, \bar{z}]\!]$

Boolean Satisfiability: CDCL

$x\vee y\vee \bar{z}$ $\quad$ $x\vee \bar{y}\vee \bar{z}$ $\quad$ $\bar{x}\vee y\vee \bar{z}$ $\quad$ $\bar{x}\vee \bar{y}\vee \bar{z}$

$x\vee y\vee z$ $\quad$ $x\vee \bar{y}\vee z$ $\quad$ $\bar{x}\vee y\vee z$ $\quad$ $\bar{x}\vee \bar{y}\vee z$

$[\![x, y, \bar{z}]\!]$

Resolve Conflict

$$\frac{\bar{x}\vee\bar{y}\vee z \qquad \bar{x}\vee\bar{y}\vee\bar{z}}{\bar{x}\vee\bar{y}}$$

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

$\boxed{\bar{x} \vee \bar{y}}$

$[\![x, y, \bar{z}]\!]$

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

$\boxed{\bar{x} \vee \bar{y}}$

$[\![x]\!]$

Boolean Satisfiability: CDCL

$x \vee y \vee \bar{z}$	$x \vee \bar{y} \vee \bar{z}$	$\bar{x} \vee y \vee \bar{z}$	$\bar{x} \vee \bar{y} \vee \bar{z}$
$x \vee y \vee z$	$x \vee \bar{y} \vee z$	$\bar{x} \vee y \vee z$	$\bar{x} \vee \bar{y} \vee z$
$\bar{x} \vee \bar{y}$			

$$[\![x, \bar{y}]\!]$$

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

$\boxed{\bar{x} \vee \bar{y}}$

$$[\![x, \bar{y}, \bar{z}]\!]$$

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

$\boxed{\bar{x} \vee \bar{y}}$

$$[\![x, \bar{y}, \bar{z}]\!]$$

Boolean Satisfiability: CDCL

$x \vee y \vee \bar{z}$ $\quad$ $x \vee \bar{y} \vee \bar{z}$ $\quad$ $\bar{x} \vee y \vee \bar{z}$ $\quad$ $\bar{x} \vee \bar{y} \vee \bar{z}$

$x \vee y \vee z$ $\quad$ $x \vee \bar{y} \vee z$ $\quad$ $\bar{x} \vee y \vee z$ $\quad$ $\bar{x} \vee \bar{y} \vee z$

$\bar{x} \vee \bar{y}$

$$[\![x, \bar{y}, \bar{z}]\!]$$

Resolve Conflict

$$\cfrac{\cfrac{\bar{x} \vee y \vee z \qquad \bar{x} \vee y \vee \bar{z}}{\bar{x} \vee y} \qquad \bar{x} \vee \bar{y}}{\bar{x}}$$

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

$\boxed{\bar{x} \vee \bar{y}}$ $\boxed{\bar{x}}$

$$[\![{\color{red}x}, \bar{y}, \bar{z}]\!]$$

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

$\boxed{\bar{x} \vee \bar{y}}$ $\boxed{\bar{x}}$

$[\![\,]\!]$

Boolean Satisfiability: CDCL

$\boxed{x \vee y \vee \bar{z}}$ $\boxed{x \vee \bar{y} \vee \bar{z}}$ $\boxed{\bar{x} \vee y \vee \bar{z}}$ $\boxed{\bar{x} \vee \bar{y} \vee \bar{z}}$

$\boxed{x \vee y \vee z}$ $\boxed{x \vee \bar{y} \vee z}$ $\boxed{\bar{x} \vee y \vee z}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$

$\boxed{\bar{x} \vee \bar{y}}$ $\boxed{\bar{x}}$

$[\![\bar{x}]\!]$

Boolean Satisfiability: CDCL

Model Construction

Build partial model by assigning variables to values

$$[\![\ldots, x, \ldots, \bar{y}, \ldots, z, \ldots]\!] \ .$$

Unit Reasoning

Reason about unit constraints

$$(\bar{x} \vee y \vee \bar{z} \vee w) \ .$$

Explain Conflicts

Explain conflicts using clausal reasons

$$(\bar{x} \vee y \vee \bar{z}) \ .$$

Outline

Linear Real Arithmetic

Linear Arithmetic

$$a_1x_1 + \cdots + a_nx_n \geq b \qquad a_1x_1 + \cdots + a_nx_n = b$$

DPLL(T): Simplex

A model builder for a conjunction of linear constraints.

- Search for a model
- Escape conflicts through pivoting
- Built for the DPLL(T) framework

[DDM06] A fast linear-arithmetic solver for DPLL(T)

Linear Real Arithmetic

Linear Arithmetic

$$a_1x_1 + \cdots + a_nx_n \geq b \qquad a_1x_1 + \cdots + a_nx_n = b$$

Fourier-Motzkin Resolution

$$\frac{\dfrac{2x + 3y - z \geq -1}{6x + 9y - 3z \geq -3} \qquad \dfrac{-3x - 2y + 4z \geq 2}{-6x - 4y + 8z \geq 4}}{5y + 5z \geq 1}$$

- Feels like Boolean resolution (elimination).
- Behaves like Boolean resolution (exponential).

Linear Real Arithmetic

Model Construction

Build partial model by assigning variables to values

$$[\![\ldots, C_1, C_2, \ldots, x \mapsto 1/2, \ldots, y \mapsto 1/2, \ldots, z \mapsto -1, \ldots]\!] \ .$$

Unit Reasoning

Reason about unit constraints

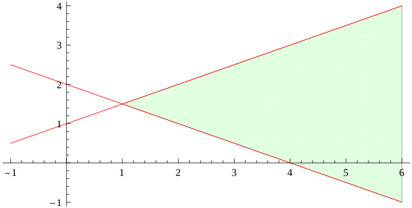
$$C_1 \equiv (x + y + z + w \geq 0) \qquad C_2 \equiv (x + y + z - w > 0) \ .$$

Explain Conflicts

Explain conflicts using valid clausal reasons

$$(\overline{C_1} \vee \overline{C_2} \vee x + y + z > 0) \ .$$

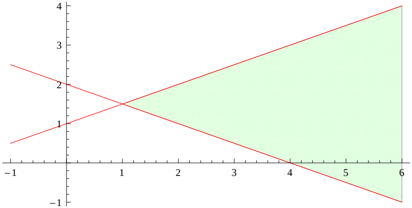
Linear Real Arithmetic



$$\overbrace{2y - x - 2 < 0}^{c_1} \;\wedge\; \overbrace{-2y - x + 4 < 0}^{c_2}$$

$$[\![\,]\!]$$

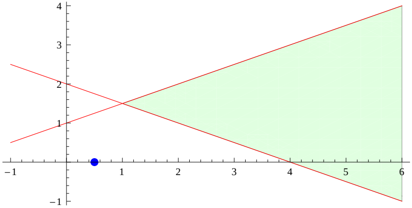
Linear Real Arithmetic



$$\overbrace{2y - x - 2 < 0}^{c_1} \;\wedge\; \overbrace{-2y - x + 4 < 0}^{c_2}$$

$$[\![c_1, c_2]\!]$$

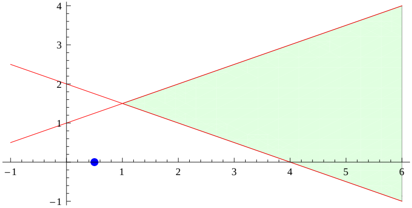
Linear Real Arithmetic



$$\overbrace{2y - x - 2 < 0}^{c_1} \;\wedge\; \overbrace{-2y - x + 4 < 0}^{c_2}$$

$$[\![c_1, c_2, x \mapsto 0.5]\!]$$

Linear Real Arithmetic

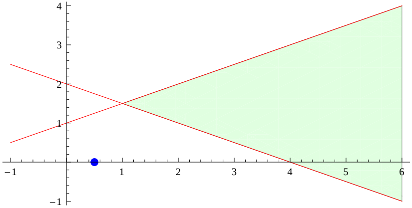


Unit Constraint Reasoning

$$2y - x - 2 < 0 \Rightarrow (y < 1.25)$$
$$-2y - x + 4 < 0 \Rightarrow (y > 1.75)$$

$[\![C_1, C_2, x \mapsto 0.5]\!]$

Linear Real Arithmetic

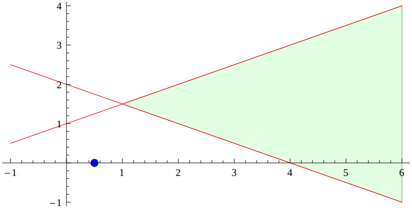


$$\overbrace{2y - x - 2 < 0}^{C_1} \;\wedge\; \overbrace{-2y - x + 4 < 0}^{C_2}$$

$$[\![C_1, C_2, x \mapsto 0.5]\!]$$

Explanation $C_1 \wedge C_2 \Rightarrow x \neq 0.5$

Linear Real Arithmetic

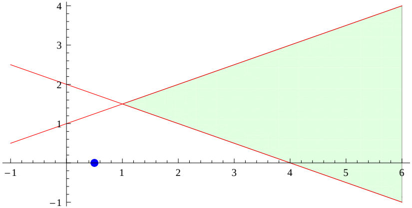


$$\overbrace{2y - x - 2 < 0}^{C_1} \;\wedge\; \overbrace{-2y - x + 4 < 0}^{C_2}$$

$$[\![C_1, C_2, x \mapsto 0.5]\!]$$

Explanation $C_1 \wedge C_2 \Rightarrow$

Linear Real Arithmetic



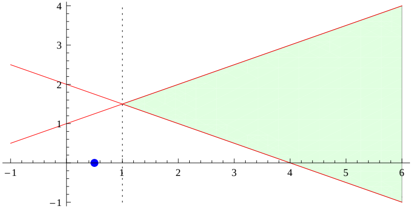
Fourier-Motzkin

$$\frac{2y - x - 2 < 0 \qquad -2y - x + 4 < 0}{-2x + 2 < 0}$$

$$[\![C_1, C_2, x \mapsto 0.5]\!]$$

Explanation $C_1 \land C_2 \Rightarrow$

Linear Real Arithmetic



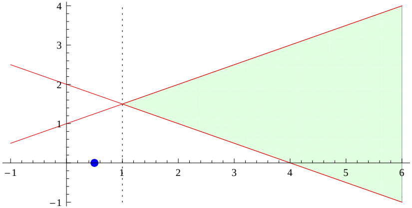
Fourier-Motzkin

$$\frac{2y - x - 2 < 0 \qquad -2y - x + 4 < 0}{-2x + 2 < 0}$$

$$[\![C_1, C_2, x \mapsto 0.5]\!]$$

Explanation $C_1 \wedge C_2 \Rightarrow x > 1$

Linear Real Arithmetic

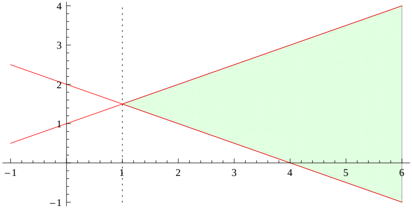


$$\overbrace{2y - x - 2 < 0}^{C_1} \;\wedge\; \overbrace{-2y - x + 4 < 0}^{C_2}$$

$$[\![C_1, C_2, {\color{red}x \mapsto 0.5}]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee (x > 1)$

Linear Real Arithmetic

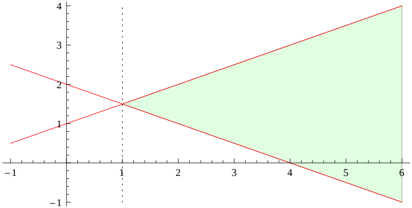


$$\overbrace{2y - x - 2 < 0}^{C_1} \wedge \overbrace{-2y - x + 4 < 0}^{C_2}$$

$$[\![C_1, C_2]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee (x > 1)$

Linear Real Arithmetic

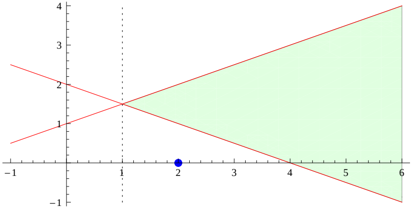


$$\overbrace{2y - x - 2 < 0}^{C_1} \;\wedge\; \overbrace{-2y - x + 4 < 0}^{C_2}$$

$$[\![C_1, C_2, x > 1]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee (x > 1)$

Linear Real Arithmetic

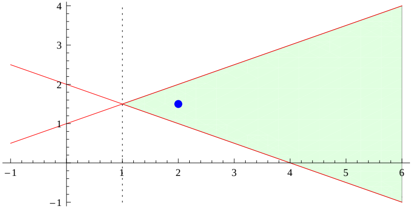


$$\overbrace{2y - x - 2 < 0}^{C_1} \;\wedge\; \overbrace{-2y - x + 4 < 0}^{C_2}$$

$$[\![C_1, C_2, x > 1, \color{red}{x \mapsto 2}]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee (x > 1)$

Linear Real Arithmetic

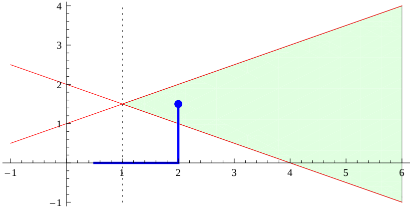


$$\overbrace{2y - x - 2 < 0}^{C_1} \;\wedge\; \overbrace{-2y - x + 4 < 0}^{C_2}$$

$$[\![C_1, C_2, x > 1, {\color{red} x \mapsto 2, y \mapsto 1.5}]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee (x > 1)$

Linear Real Arithmetic

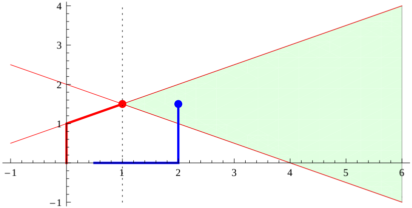


$$\overbrace{2y - x - 2 < 0}^{C_1} \quad \wedge \quad \overbrace{-2y - x + 4 < 0}^{C_2}$$

$$[\![C_1, C_2, x > 1, {\color{red} x \mapsto 2}, {\color{red} y \mapsto 1.5}]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee (x > 1)$

Linear Real Arithmetic

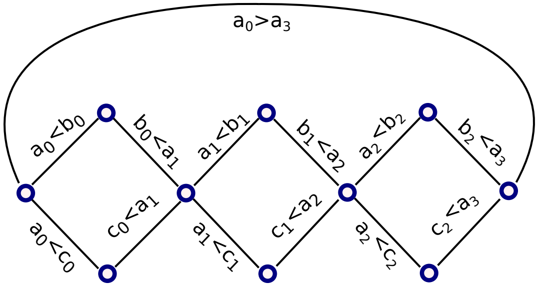


$$\overbrace{2y - x - 2 < 0}^{C_1} \;\wedge\; \overbrace{-2y - x + 4 < 0}^{C_2}$$

$$[\![C_1, C_2, x > 1, x \mapsto 2, y \mapsto 1.5]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee (x > 1)$

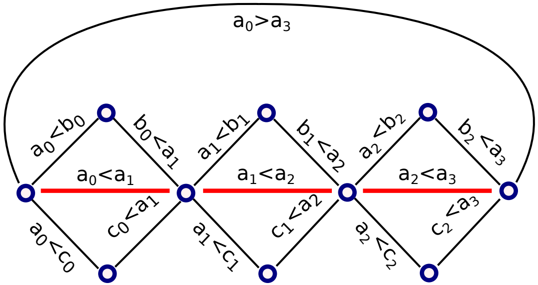
Linear Real Arithmetic: Comparison to DPLL(T)



Example (Diamonds)

$$a_0 > a_n \wedge \bigwedge_{k=0}^{n-1} ((a_k < b_k \wedge b_k < a_{k+1}) \vee (a_k < c_k \wedge c_k < a_{k+1}))$$

Linear Real Arithmetic: Comparison to DPLL(T)



Example (Diamonds)

$$a_0 > a_n \wedge \bigwedge_{k=0}^{n-1} ((a_k < b_k \wedge b_k < a_{k+1}) \vee (a_k < c_k \wedge c_k < a_{k+1}))$$

Linear Real Arithmetic: Comparison to DPLL(T)

	mcsat		cvc4		z3		mathsat5		yices	
set	solved	time (s)	solved	time (s)	solved	time (s)	solved	time (s)	solved	time (s)
clocksynchro (36)	**36**	**123.11**	36	1166.55	36	1828.74	36	1732.59	36	1093.80
DTPScheduling (91)	**91**	**31.33**	91	72.92	91	100.55	89	1980.96	91	926.22
miplib (42)	8	97.16	**27**	**3359.40**	23	3307.92	19	5447.46	23	466.44
sal (107)	107	12.68	107	13.46	107	6.37	107	7.99	**107**	**2.45**
sc (144)	144	1655.06	144	1389.72	144	954.42	144	880.27	**144**	**401.64**
spiderbenchmarks (42)	42	2.38	42	2.47	42	1.66	42	1.22	**42**	**0.44**
TM (25)	25	1125.21	25	82.12	**25**	**51.64**	25	1142.98	25	55.32
ttastartup (72)	70	4443.72	72	1305.93	72	1647.94	72	2607.49	**72**	**1218.68**
uart (73)	73	5244.70	73	1439.89	73	1379.90	73	1481.86	**73**	**679.54**
	596	12735.35	**617**	**8832.46**	613	9279.14	607	15282.82	613	4844.53

Linear Real Arithmetic: Comparison to DPLL(T)

DPLL(T) Simplex (CVC4)

```
Total Physical Source Lines of Code (SLOC)                = 22,597
Development Effort Estimate, Person-Years (Person-Months) = 5.28 (63.38)
 (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                         = 1.01 (12.10)
 (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 5.24
Total Estimated Cost to Develop                           = $ 713,502
 (average salary = $56,286/year, overhead = 2.40).
```

MCSAT Fourier-Motzkin (CVC4)

```
Total Physical Source Lines of Code (SLOC)                = 1,966
Development Effort Estimate, Person-Years (Person-Months) = 0.41 (4.88)
 (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                         = 0.38 (4.57)
 (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 1.07
Total Estimated Cost to Develop                           = $ 54,942
 (average salary = $56,286/year, overhead = 2.40).
```

Generated using David A. Wheeler's 'SLOCCount'.

Non-Linear Arithmetic

$$f(\vec{y}, x) = a_m \cdot x^{d_m} + a_{m-1} \cdot x^{d_{m-1}} + \cdots + a_1 \cdot x^{d_1} + a_0$$

f is in $\mathbb{Z}[\vec{y}, x]$, a_i are in $\mathbb{Z}[\vec{y}]$

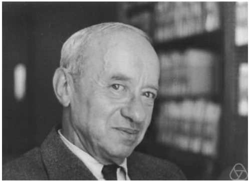
Examples

$$f(x, y) = (x^2 - 1)y^2 + (x + 1)y - 1 \in \mathbb{Z}[x, y]$$
$$g(x) = 16x^3 - 8x^2 + x + 16 \in \mathbb{Z}[x]$$

Polynomial Constraints

$$f(x, y) > 0 \land g(x) < 0$$

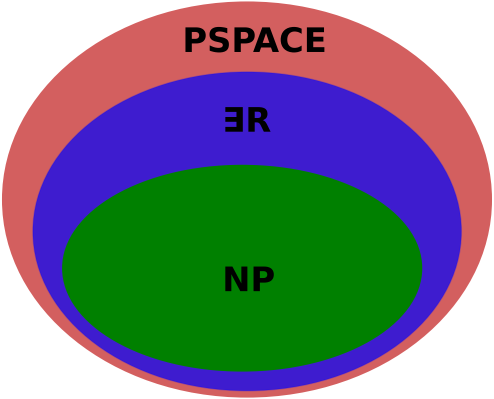
Non-Linear Arithmetic



Tarski [Tar48]
Quantifier elimination
Decidable, non-elementary

Collins [Col75]
Cylindrical Algebraic Decomposition
Doubly-exponential

Non-Linear Real Arithmetic



Canny [Can88], Grigor'ev [Gri88]

Non-Linear Real Arithmetic: CAD

$$p_1 > 0 \vee (p_2 = 0 \wedge p_3 < 0) \qquad p_1, p_2, p_3 \in \mathbb{Z}[x_1, \ldots, x_n]$$

Projection (Saturation)

Project polynomials using a projection P

$$\{p_1, p_2, p_3\} \mapsto \{p_1, p_2, p_3, p_4, \ldots, p_n\} \ .$$

Lifting (Model construction)

For each variable x_k

1. Isolate roots of $p_i(\alpha, x_k)$.
2. Choose a cell C and assign $x_k \mapsto \alpha_k \in C$, continue.
3. If no more cells, backtrack.

Non-Linear Real Arithmetic

Model Construction

Build partial model by assigning variables to values

$$[\![\ldots, C_1, C_2, \ldots, x \mapsto \sqrt{2}/2, \ldots]\!] \ .$$

Unit Reasoning

Reason about unit constraints

$$C_1 \equiv (x^2 + y^2 < 1) \qquad\qquad C_2 \equiv (xy > 1) \ .$$

Explain Conflicts

Explain conflicts using valid clausal reasons

$$(\overline{C_1} \vee \overline{C_2} \vee x \leq 0 \vee x \geq 1) \ .$$

Non-Linear Real Arithmetic

Unit Reasoning

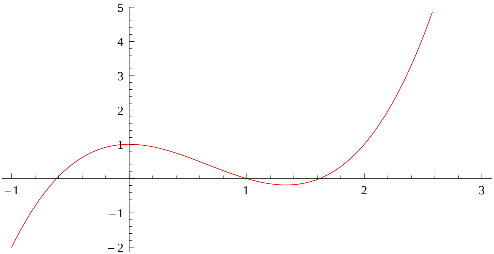
Reason about unit constraints

$$C_1 \equiv (x^2 + y^2 < 1) \qquad\qquad C_2 \equiv (xy > 1) \quad .$$

Unit Non-Linear Constraints

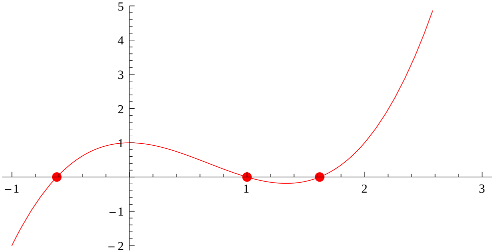
$$x^3 - 2x^2 + 1 > 0$$

Unit Non-Linear Constraints



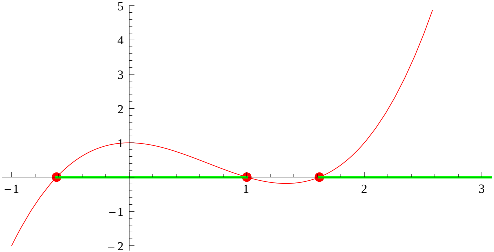
$$x^3 - 2x^2 + 1 > 0$$

Unit Non-Linear Constraints



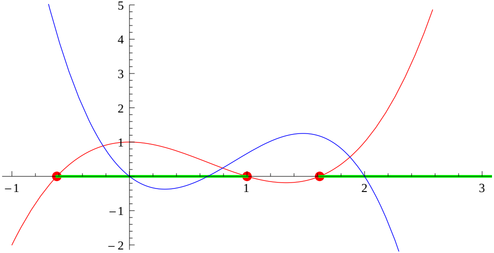
$$x^3 - 2x^2 + 1 > 0$$

Unit Non-Linear Constraints



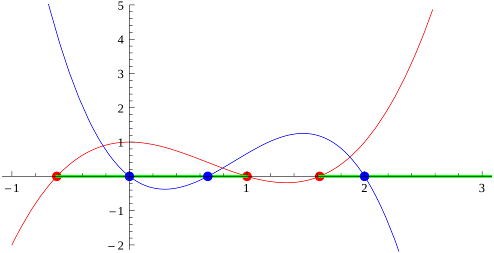
$$x^3 - 2x^2 + 1 > 0$$

Unit Non-Linear Constraints



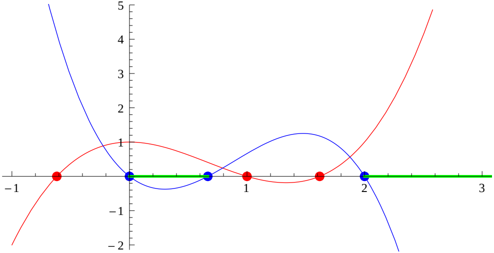
$$x^3 - 2x^2 + 1 > 0 \qquad\qquad -3x^3 + 8x^2 - 4x > 0$$

Unit Non-Linear Constraints



$$x^3 - 2x^2 + 1 > 0 \qquad\qquad - 3x^3 + 8x^2 - 4x > 0$$

Unit Non-Linear Constraints



$$x^3 - 2x^2 + 1 > 0 \qquad\qquad -3x^3 + 8x^2 - 4x > 0$$

Non-Linear Real Arithmetic

Model Construction

Build partial model by assigning variables to values

$$[\![\ldots, C_1, C_2, \ldots, x \mapsto \sqrt{2}/2, \ldots]\!] \ .$$

Unit Reasoning

Reason about unit constraints

$$C_1 \equiv (x^2 + y^2 < 1) \qquad\qquad C_2 \equiv (xy > 1) \ .$$

Explain Conflicts

Explain conflicts using valid clausal reasons

$$(\overline{C_1} \vee \overline{C_2} \vee x \leq 0 \vee x \geq 1) \ .$$

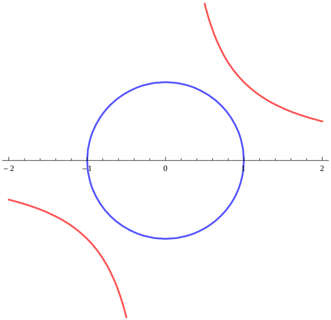
Non-Linear Real Arithmetic

Explain Conflicts

Explain conflicts using valid clausal reasons

$$(\overline{C_1} \vee \overline{C_2} \vee x \leq 0 \vee x \geq 1) \quad .$$

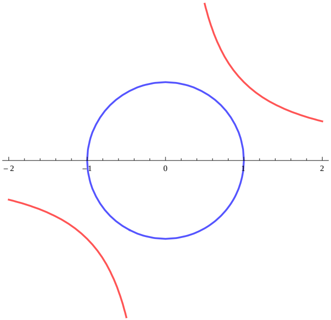
Non-Linear Real Arithmetic



$$\overbrace{x^2 + y^2 < 1}^{c_1} \;\wedge\; \overbrace{xy > 1}^{c_2}$$

$$\llbracket\rrbracket$$

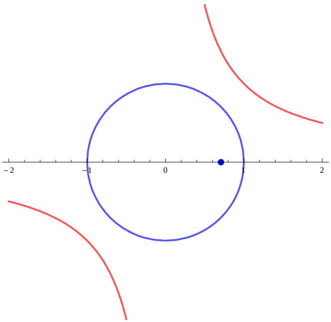
Non-Linear Real Arithmetic



$$\overbrace{x^2 + y^2 < 1}^{c_1} \;\wedge\; \overbrace{xy > 1}^{c_2}$$

$$[\![c_1, c_2]\!]$$

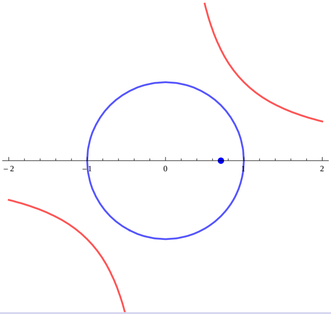
Non-Linear Real Arithmetic



$$\overbrace{x^2 + y^2 < 1}^{c_1} \;\wedge\; \overbrace{xy > 1}^{c_2}$$

$$[\![c_1, c_2, x \mapsto \sqrt{2}/2]\!]$$

Non-Linear Real Arithmetic



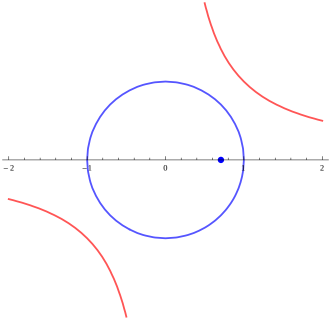
Unit Constraint Reasoning

$$x^2 + y^2 < 1 \Rightarrow -\sqrt{3/2} < y < \sqrt{3/2}$$

$$-2y - x + 4 < 0 \Rightarrow y > \sqrt{2}$$

$$\llbracket C_1, C_2, x \mapsto \sqrt{2}/2 \rrbracket$$

Non-Linear Real Arithmetic

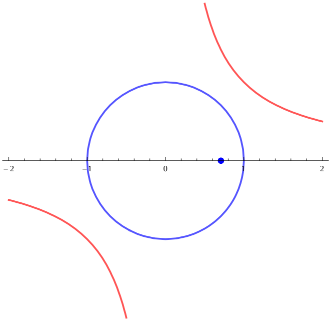


$$\overbrace{x^2 + y^2 < 1}^{C_1} \;\wedge\; \overbrace{xy > 1}^{C_2}$$

$$[\![C_1, C_2, x \mapsto \sqrt{2}/2]\!]$$

Explanation $C_1 \wedge C_2 \Rightarrow x \neq \sqrt{2}/2$

Non-Linear Real Arithmetic

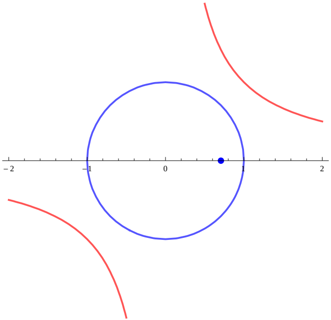


$$\overbrace{x^2 + y^2 < 1}^{C_1} \;\wedge\; \overbrace{xy > 1}^{C_2}$$

$$[\![C_1, C_2, x \mapsto \sqrt{2}/2]\!]$$

Explanation $C_1 \wedge C_2 \Rightarrow$

Non-Linear Real Arithmetic

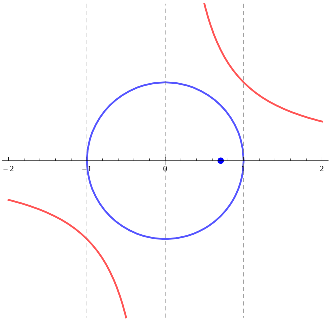


$$\overbrace{x^2 + y^2 < 1}^{c_1} \;\wedge\; \overbrace{xy > 1}^{c_2}$$

CAD Projection

$$\mathsf{P} = \{x, -4 + 4x^2, 1 - x^2 + x^4\}$$

Non-Linear Real Arithmetic

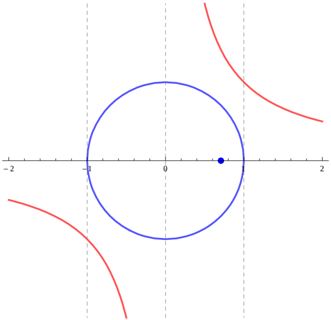


$$\overbrace{x^2 + y^2 < 1}^{c_1} \;\wedge\; \overbrace{xy > 1}^{c_2}$$

CAD Projection

$$\mathsf{P} = \{x, -4 + 4x^2, 1 - x^2 + x^4\}$$

Non-Linear Real Arithmetic

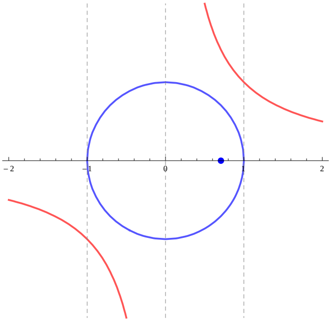


$$\overbrace{x^2 + y^2 < 1}^{C_1} \wedge \overbrace{xy > 1}^{C_2}$$

$$[\![C_1, C_2, x \mapsto \sqrt{2}/2]\!]$$

Explanation $C_1 \wedge C_2 \Rightarrow x \leq 0 \vee x \geq 1$

Non-Linear Real Arithmetic

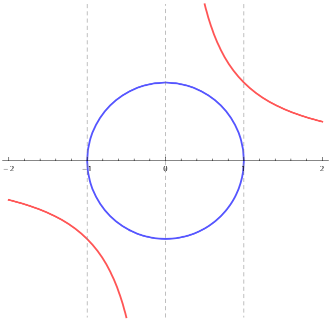


$$\overbrace{x^2 + y^2 < 1}^{C_1} \;\wedge\; \overbrace{xy > 1}^{C_2}$$

$$[\![C_1, C_2, x \mapsto \sqrt{2}/2]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee x \leq 0 \vee x \geq 1$

Non-Linear Real Arithmetic

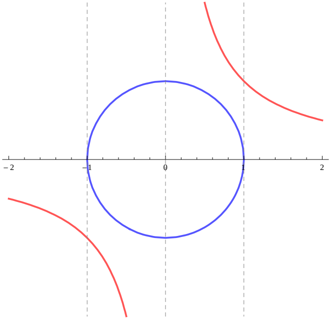


$$\overbrace{x^2 + y^2 < 1}^{C_1} \;\wedge\; \overbrace{xy > 1}^{C_2}$$

$$[\![C_1, C_2]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee x \leq 0 \vee x \geq 1$

Non-Linear Real Arithmetic

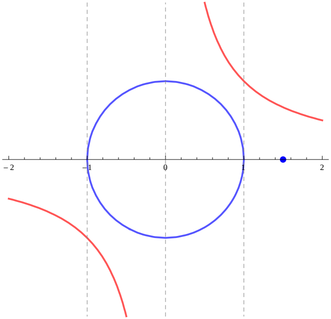


$$\overbrace{x^2 + y^2 < 1}^{C_1} \;\wedge\; \overbrace{xy > 1}^{C_2}$$

$$[\![C_1, C_2, x \geq 1]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee x \leq 0 \vee x \geq 1$

Non-Linear Real Arithmetic

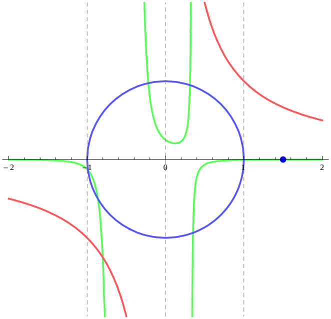


$$\overbrace{x^2 + y^2 < 1}^{C_1} \;\wedge\; \overbrace{xy > 1}^{C_2}$$

$$[\![C_1, C_2, x \geq 1, x \mapsto 3/2]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee x \leq 0 \vee x \geq 1$

Non-Linear Real Arithmetic

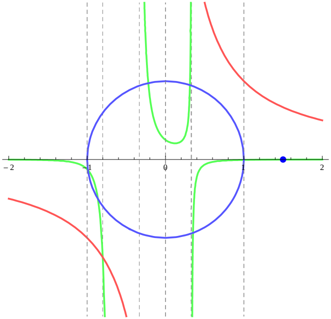


$$\overbrace{x^2 + y^2 < 1}^{C_1} \;\wedge\; \overbrace{xy > 1}^{C_2}$$

$$[\![C_1, C_2, {\color{red} x \geq 1}, {\color{red} x \mapsto 3/2}]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee x \leq 0 \vee x \geq 1$

Non-Linear Real Arithmetic

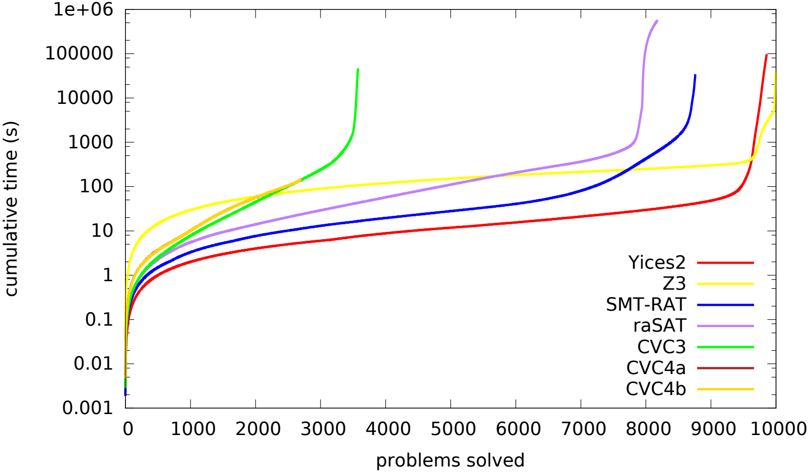


$$\overbrace{x^2 + y^2 < 1}^{C_1} \;\wedge\; \overbrace{xy > 1}^{C_2}$$

$$[\![C_1, C_2, x \geq 1, x \mapsto 3/2]\!]$$

Explanation $\overline{C_1} \vee \overline{C_2} \vee x \leq 0 \vee x \geq 1$

Non-Linear Real Arithmetic: SMT-COMP 2015



Model-Based Procedures

Linear Real Arithmetic

- Generalizing DPLL to Richer Logics [MKS09]
- Conflict Resolution [KTV09]
- Natural Domain SMT [Cot10]

Linear Integer Arithmetic

- Cutting to the Chase: Solving Linear Integer Arithmetic [JDM11]

Non-Linear Real Arithmetic

- Solving Non-Linear Arithmetic [JDM12, Jov12]

General Framework

- Model-Constructing Satisfiability Calculus [DMJ13, JBdM13]

Outline

Theory Combination

Combination of Theories

Given individual decision procedures for (quantifier free) first-order theories T_1 and T_2 how can we combine them in a modular fashion into a decision procedure for a theory $T_1 \oplus T_2$?

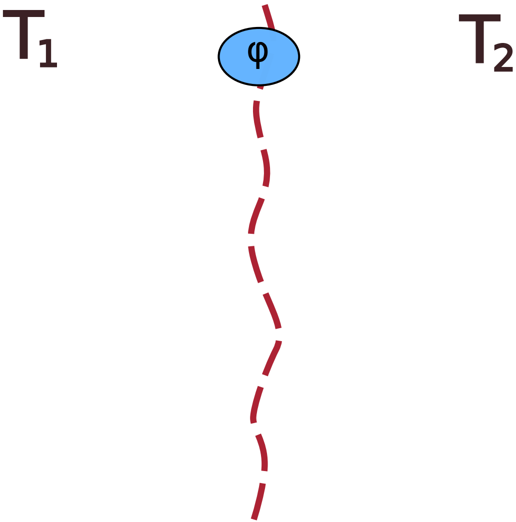
Theory Combination

Combination of Theories

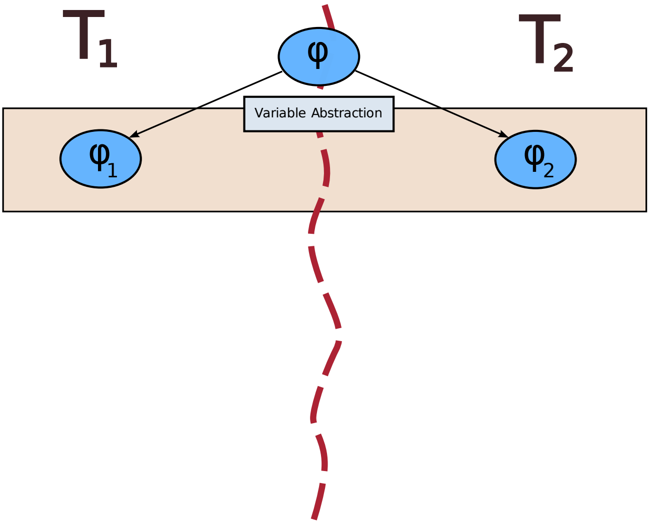
Given individual decision procedures for (quantifier free) first-order theories T_1 and T_2 how can we combine them in a modular fashion into a decision procedure for a theory $T_1 \oplus T_2$?

- Nelson-Oppen Method [NO79]
- Allows one to decide the combination using decision procedures for T_1 and T_2 as black-boxes
- Most SMT Solvers that involve more than one theory use a combination method based on Nelson-Oppen

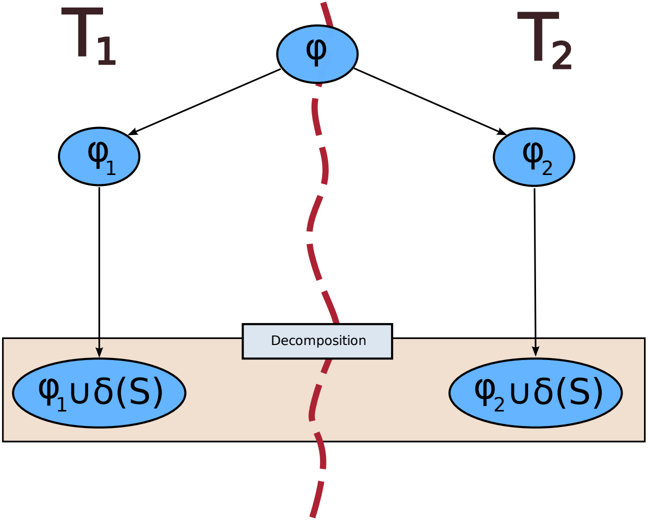
Theory Combination: Nelson-Oppen



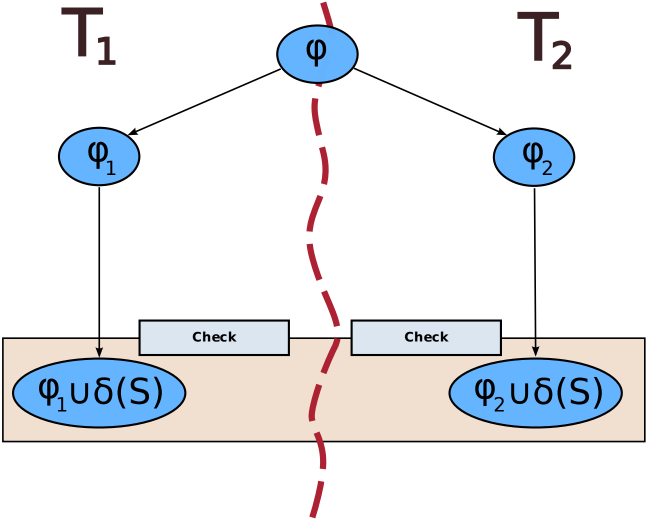
Theory Combination: Nelson-Oppen



Theory Combination: Nelson-Oppen



Theory Combination: Nelson-Oppen

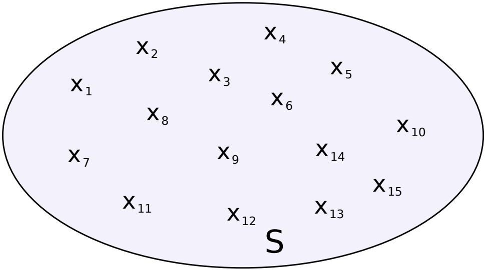


Theory Combination: Nelson-Oppen Complexity

Complexity

Search for a suitable arrangement over the shared variables introduces a heavy layer of complexity:

$$O(\mathcal{T}_1(n)) \oplus O(\mathcal{T}_2(n)) \Rightarrow O(2^{n^2} \times (\mathcal{T}_1(n) + \mathcal{T}_2(n)))\ .$$

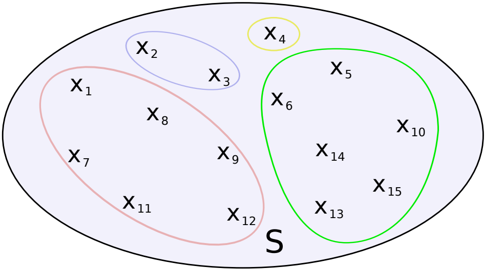


Theory Combination: Nelson-Oppen Complexity

Complexity

Search for a suitable arrangement over the shared variables introduces a heavy layer of complexity:

$$O(\mathcal{T}_1(n)) \oplus O(\mathcal{T}_2(n)) \Rightarrow O(2^{n^2} \times (\mathcal{T}_1(n) + \mathcal{T}_2(n))) \quad .$$



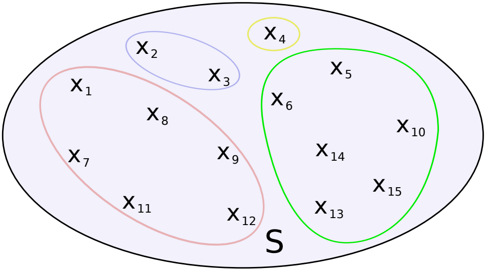
Theory Combination: Nelson-Oppen Complexity

Complexity

Search for a suitable arrangement over the shared variables introduces a heavy layer of complexity:

$$O(\mathcal{T}_1(n)) \oplus O(\mathcal{T}_2(n)) \Rightarrow O(2^{n^2} \times (\mathcal{T}_1(n) + \mathcal{T}_2(n))) \ \ .$$

$\phi_1 \cup \delta(S)$



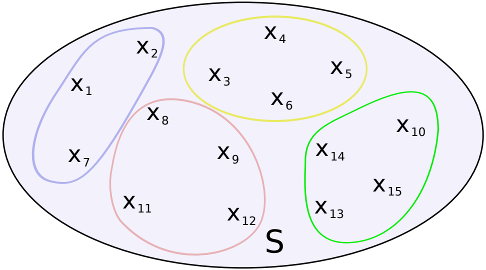
$\phi_2 \cup \delta(S)$

Theory Combination: Nelson-Oppen Complexity

Complexity

Search for a suitable arrangement over the shared variables introduces a heavy layer of complexity:

$$O(\mathcal{T}_1(n)) \oplus O(\mathcal{T}_2(n)) \Rightarrow O(2^{n^2} \times (\mathcal{T}_1(n) + \mathcal{T}_2(n))) \ .$$



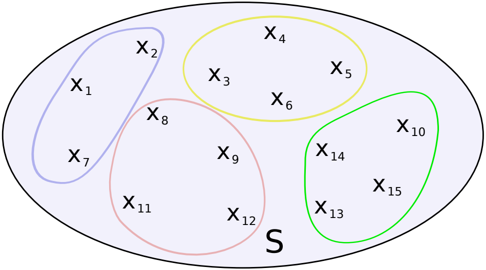
Theory Combination: Nelson-Oppen Complexity

Complexity

Search for a suitable arrangement over the shared variables introduces a heavy layer of complexity:

$$O(\mathcal{T}_1(n)) \oplus O(\mathcal{T}_2(n)) \Rightarrow O(2^{n^2} \times (\mathcal{T}_1(n) + \mathcal{T}_2(n))) \quad .$$

$\phi_1 \cup \delta(S)$



$\phi_2 \cup \delta(S)$

Theory Combination: Nelson-Oppen Complexity

Again, two modules have a restricted interface to each other!

Model-based theory combination [dMB08]

Uninterpreted Functions

$$x = y \qquad\qquad x \neq y \qquad\qquad x = f(y, z)$$

DPLL(T): Congruence Closure

- Incremental algorithms for congruence closure.
- Propagation of entailed equalities.
- Combination through Nelson-Oppen style procedures.

Uninterpreted Functions

$$x = y \qquad x \neq y \qquad x = f(y, z)$$

DPLL(T): Congruence Closure

- Incremental algorithms for congruence closure.
- Propagation of entailed equalities.
- Combination through Nelson-Oppen style procedures.

Alternative: Ackermannization [Ack54]

$$\frac{}{x_1 = y_1 \wedge x_2 = y_2 \Rightarrow f(x_1, x_2) = f(y_1, y_2)}$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![\,]\!]$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y)]\!]$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y), \;\; f(x) \mapsto 0]\!]$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y), \;\; f(x) \mapsto 0, \;\; f(y) \mapsto 1]\!]$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y),\ \ f(x) \mapsto 0,\ \ f(y) \mapsto 1,\ \ x \mapsto 0]\!]$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y),\ \ f(x) \mapsto 0,\ \ f(y) \mapsto 1,\ \ x \mapsto 0,\ \ y \mapsto 0]\!]$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y),\ \ f(x) \mapsto 0,\ \ f(y) \mapsto 1,\ \ x \mapsto 0,\ \ y \mapsto 0]\!]$$

Explain Conflict: Ackermanization

$$\frac{}{x = y \Rightarrow f(x) = f(y)}$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y),\ \ f(x) \mapsto 0,\ \ f(y) \mapsto 1,\ \ x \mapsto 0,\ \ y \mapsto 0]\!]$$

Explain Conflict: Ackermanization

$$\frac{}{x \neq y \vee f(x) = f(y)}$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y),\ \ f(x) \mapsto 0,\ \ f(y) \mapsto 1]\!]$$

Explain Conflict: Ackermanization

$$\frac{}{x \neq y \vee f(x) = f(y)}$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y),\ \ f(x) \mapsto 0,\ \ f(y) \mapsto 1,\ \ x \neq y]\!]$$

Explain Conflict: Ackermanization

$$\frac{}{x \neq y \vee f(x) = f(y)}$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y),\ \ f(x) \mapsto 0,\ \ f(y) \mapsto 1,\ \ x \neq y,\ \ x \mapsto 0]\!]$$

Explain Conflict: Ackermanization

$$\frac{}{x \neq y \vee f(x) = f(y)}$$

Uninterpreted Functions: Example

$$f(x) < f(y)$$

$$[\![f(x) < f(y),\ \ f(x) \mapsto 0,\ \ f(y) \mapsto 1,\ \ x \neq y,\ \ x \mapsto 0,\ \ y \mapsto 1]\!]$$

Explain Conflict: Ackermanization

$$\frac{}{x \neq y \vee f(x) = f(y)}$$

Uninterpreted Functions: Comparison to DPLL(T)

	mcsat		cvc4		z3		mathsat5		yices	
set	solved	time (s)	solved	time (s)	solved	time (s)	solved	time (s)	solved	time (s)
EufLaArithmetic (33)	33	39.57	33	49.11	**33**	**2.53**	33	20.18	33	4.61
Hash (198)	198	34.81	198	10.60	198	7.18	198	1330.88	**198**	**2.64**
RandomCoupled (400)	400	68.04	400	35.90	400	31.44	**400**	**18.56**	384	39903.78
RandomDecoupled (500)	500	34.95	500	40.63	500	30.98	**500**	**21.86**	500	3863.79
Wisa (223)	223	9.18	223	87.35	223	10.80	223	65.27	**223**	**2.80**
wisas (108)	**108**	**40.17**	108	5221.37	108	443.36	106	1737.41	108	736.98
	1462	**226.72**	1462	5444.96	1462	526.29	1460	3194.16	1446	44514.60

Uninterpreted Functions: Comparison to DPLL(T)

DPLL(T) CC (CVC4)

```
Total Physical Source Lines of Code (SLOC)                = 5,727
Development Effort Estimate, Person-Years (Person-Months) = 1.25 (15.00)
 (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                         = 0.58 (7.00)
 (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 2.14
Total Estimated Cost to Develop                           = $ 168,836
 (average salary = $56,286/year, overhead = 2.40).
```

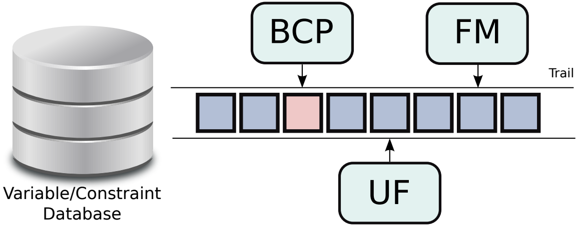
MCSAT Ackermanization (CVC4)

```
Total Physical Source Lines of Code (SLOC)                = 247
Development Effort Estimate, Person-Years (Person-Months) = 0.05 (0.55)
 (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                         = 0.17 (2.00)
 (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 0.28
Total Estimated Cost to Develop                           = $ 6,223
 (average salary = $56,286/year, overhead = 2.40).
```

Generated using David A. Wheeler's 'SLOCCount'.

Outline

MCSAT: Simple Architecture



Each plugins reasons about their domain:

1. Track when a constraint becomes unit or fully assigned [MMZ+01].
2. Unit constraints imply feasible sets of individual variables.
3. Propagate any constraints/variables whose value is implied.
4. Explain any unit conflicts with clausal explanations.
5. When asked, decide unassigned variable to feasible value.

MCSAT: Implementations

- QF_NRA in Yices2 ▸ Link
- QF_NRA in Z3 ▸ Link
- QF_UFLRA in CVC4 ▸ Link
- LibPoly: Library for polynomial manipulation ▸ Link

MCSAT: Appeal

Compared to DPPL(T)

- Only need to reason about constraints in one variable.
- Easy to add new decision procedures.
- Can generate new facts in a conflict directed manner.
- Simple combination mechanism: build the model.
- Performs well on practical problems.

MCSAT: TODO & Research Problems

- Linear arithmetic
  - Simplex vs Fourier-Motzkin?
  - Add integer reasoning.
- Non-linear arithmetic:
  - More natural conflict explanations?
  - Integrate interval reasoning into MCSAT [BDG+14].
  - Add integer reasoning.
- Bit-vectors, floating point, strings…
  - How to reason about unit constraints?
  - How to represent feasible sets?
  - How to explain conflicts?
- Arrays
  - Adapt lemmas on demand to MCSAT [BB08]?

References I

[Ack54] Wilhelm Ackermann.
Solvable cases of the decision problem.
1954.

[BB08] Robert Brummayer and Armin Biere.
Lemmas on demand for the extensional theory of arrays.
Journal on Satisfiability, Boolean Modeling and Computation, 6:165–201, 2008.

[BDdM08] Nikolaj Bjørner, Bruno Dutertre, and Leonardo de Moura.
Accelerating lemma learning using joins-dppl (join).
Proceedings of short papers at LPAR, 8, 2008.

[BDG+14] Martin Brain, Vijay D'Silva, Alberto Griggio, Leopold Haller, and Daniel Kroening.
Deciding floating-point logic with abstract conflict driven clause learning.
Formal Methods in System Design, 45(2):213–245, 2014.

[BNOT06] Clark Barrett, Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli.
Splitting on demand in sat modulo theories.
In *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 512–526, 2006.

[Can88] John Canny.
Some algebraic and geometric computations in pspace.
In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 460–467, 1988.

References II

[Col75] George E Collins.
Quantifier elimination for real closed fields by cylindrical algebraic decomposition.
In *Automata Theory and Formal Languages*, pages 134–183, 1975.

[Cot10] Scott Cotton.
Natural domain SMT: A preliminary assessment.
Formal Modeling and Analysis of Timed Systems, pages 77–91, 2010.

[DDM06] Bruno Dutertre and Leonardo De Moura.
A fast linear-arithmetic solver for DPLL(T).
In *Computer Aided Verification*, pages 81–94, 2006.

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

[dMB08] Leonardo de Moura and Nikolaj Bjørner.
Model-based theory combination.
Electronic Notes in Theoretical Computer Science, 198(2):37–49, 2008.

[DMJ13] Leonardo De Moura and Dejan Jovanović.
A model-constructing satisfiability calculus.
In *Verification, Model Checking, and Abstract Interpretation*, pages 1–12, 2013.

References III

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

[Gri88] D Yu Grigor'ev.
Complexity of deciding tarski algebra.
Journal of symbolic Computation, 5(1):65–108, 1988.

[HAMM14] Julien Henry, Mihail Asavoae, David Monniaux, and Claire Maïza.
How to compute worst-case execution time by optimization modulo theory and a clever encoding of program semantics.
In *Proceedings of the 2014 SIGPLAN/SIGBED conference on Languages, compilers and tools for embedded systems*, pages 43–52. ACM, 2014.

[JBdM13] Dejan Jovanović, Clark Barrett, and Leonardo de Moura.
The design and implementation of the model constructing satisfiability calculus.
Formal Methods in Computer-Aided Design, 2013.

[JDM11] Dejan Jovanović and Leonardo De Moura.
Cutting to the chase: solving linear integer arithmetic.
In *Automated Deduction*, pages 338–353. 2011.

[JDM12] Dejan Jovanović and Leonardo De Moura.
Solving non-linear arithmetic.
In *Automated Reasoning*, pages 339–354. 2012.

References IV

[Jov12] Dejan Jovanović.
SMT Beyond DPLL(T): A New Approach to Theory Solvers and Theory Combination.
PhD thesis, Courant Institute of Mathematical Sciences New York, 2012.

[KTV09] Konstantin Korovin, Nestan Tsiskaridze, and Andrei Voronkov.
Conflict resolution.
In *Principles and Practice of Constraint Programming*, pages 509–523. 2009.

[MKS09] Kenneth L McMillan, Andreas Kuehlmann, and Mooly Sagiv.
Generalizing DPLL to richer logics.
In *Computer Aided Verification*, pages 462–476, 2009.

[MMZ+01] Matthew W Moskewicz, Conor F Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik.
Chaff: Engineering an efficient sat solver.
In *Proceedings of the 38th annual Design Automation Conference*, pages 530–535. ACM, 2001.

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

References V

[SS97] João P Marques Silva and Karem A Sakallah.
Grasp—a new search algorithm for satisfiability.
In *Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design*, pages 220–227. IEEE Computer Society, 1997.

[Tar48] Alfred Tarski.
A decision method for elementary algebra and geometry.
1948.