Satisfiability Modulo Theories
Equalities + Uninterpreted Functions (EUF)
Linear Arithmetic

Summer School on Formal Techniques
Menlo Park, May 2012

Bruno Dutertre	Leonardo de Moura
SRI International	Microsoft Research

Outline

SAT Solving: check satisfiability of Boolean formulas (propositional logic)

SMT Solving: extends SAT solving to first-order theories

Lecture Content:

- SMT: CDCL + Theory Solvers
- Theory Solvers for Equality + Uninterpreted Functions
- Theory Solvers for Linear Arithmetic

SMT Solving:
CDCL + Theory Solver

Decision Procedures

Definition

- Algorithm to determine whether a formula ϕ (in a first-order theory T) is satisfiable.

Examples

- Congruence closure: for quantifier-free formulas, uninterpreted functions
- Simplex methods for quantifier-free linear arithmetic
- Cylindrical algebraic decomposition for real closed fields

More useful versions

- Decision procedures for combinations of theories:

$$2.\mathtt{car}(x) - 3.\mathtt{cdr}(x) = f(\mathtt{cdr}(x)) \Rightarrow$$
$$g(\mathtt{cons}(4.\mathtt{car}(x) - 2.f(\mathtt{cdr}(x)), y)) = g(\mathtt{cons}(6.\mathtt{cdr}(x)), y)$$

Dealing with Boolean Structure

Many decision procedures (e.g., congruence closure, simplex) work on conjunctions of literals

They can still be applied to arbitrary formula ϕ. For example, write ϕ in DNF:

$$(a_{11} \wedge \ldots \wedge a_{1n}) \vee \ldots \vee (a_{m1} \wedge \ldots \wedge a_{mp})$$

Problem: this is highly inefficient

- DNF can explode
- If several conjuncts share identical literals, we prove the same thing many time:

$$(f(x,y) \neq f(y,x) \wedge z = 3x + 1 \wedge x = y \wedge z < 0) \;\vee$$
$$(t > g(y) \wedge x = y \wedge z + 3 \leqslant 0 \wedge f(x,y) \neq f(y,x)) \;\vee\; \ldots$$

Better approach: use a Boolean SAT solver to enumerate the conjuncts

- This is Satisfiability Modulo Theories: efficient combination of SAT solver and decision procedures

Basic SMT Solving

$$x + y \geqslant 0 \wedge (x = z \Rightarrow z + y = -1) \wedge z > 3t$$

1) Replace atoms by Boolean variables

$$\begin{array}{lllllll} a & \mapsto & x + y \geqslant 0 & \quad & b & \mapsto & x = z \\ c & \mapsto & z + y = -1 & \quad & d & \mapsto & z > 3t \end{array}$$

2) Ask for a model of $a \wedge (b \Rightarrow c) \wedge d$ using a SAT solver

- Boolean model: $\{a, b, c, d\}$
- Convert the model back to arithmetic

$$x + y \geqslant 0 \wedge x = z \wedge z + y = -1 \wedge z > 3t$$

and check its consistency

Answer: not consistent

Explanation: $Arithmetic \models \neg(x + y \geqslant 0 \wedge x = z \wedge z + y = -1)$

Basic SMT Solving (continued)

3) Feed the explanation to the SAT solver:

- add the clause $(\neg a \vee \neg b \vee \neg c)$

4) Get a model of $(a \wedge (b \Rightarrow c) \wedge d) \wedge (\neg a \vee \neg b \vee \neg c)$

- Boolean model: $\{a, \neg b, c, d\}$
- Convert back to arithmetic:

$$x + y \geqslant 0 \wedge \neg(x = z) \wedge z + y = -1 \wedge z > 3t$$

- Check consistency: satisfiable

$$x = 1, \quad y = 1, \quad z = -2, \quad t = -1$$

Conclusion: The original formula is satisfiable

Improvements to the Basic Approach

Make it incremental

- Don't wait for a full Boolean model to check consistency: interleave Boolean propagation and calls to the theory solver

Theory propagation

- Example: given partial model $\{a, d, c\}$ (i.e., $x + y \geqslant 0, z + y = -1, z > 3t$) the linear arithmetic solver can deduce that b must be false (since $Arithmetic \models x + y \geqslant 0 \wedge z + y = -1 \Rightarrow \neg(x = z)$)
- Theory propagation: detect this and assign $\neg b$ in the SAT solver.

Benefit of these improvements: prune the SAT solver search space

Approaches to SMT Solving

Eager Methods

- Convert SMT problem into an equisatisfiable SAT problem
- Example theories: bitvectors, difference logic, equality

Lazy Methods

- Close integration of a SAT solver and decision procedures
- More widely applicable that eager methods
- Most common approach: CDCL SAT solver combined with a theory solver

Theory Solver

Notation

- We assume a theory T (quantifier-free for now)
- We use $T \vdash A$ to denote that formula A is valid in T

Theory Solver

- A decision procedure for T specialized to interact with a CDCL SAT solver
- Implements two new rules T-Propagation and T-Conflict
- T-Conflict:
  - given a set of literals M (i.e., a partial model), find a clause C such that $M \models \neg C$ and $T \vdash C$.
- T-Propagation:
  - given a set of literals M, find C and ℓ such that ℓ is not assigned in M, and $M \models \neg C$, and $T \vdash C \vee \ell$.

Abstract CDCL(T)

$$M \parallel F \quad \Longrightarrow \quad M\ell \parallel F \qquad \textbf{if} \begin{cases} \ell \text{ or } \overline{\ell} \text{ occurs in } F \\ \ell \text{ unassigned in } M \end{cases} \qquad \text{(Decide)}$$

$$M \parallel F, C \vee \ell \quad \Longrightarrow \quad M\ell_{C\vee\ell} \parallel F \qquad \textbf{if} \begin{cases} \ell \text{ unassigned in } M \\ M \models \neg C \end{cases} \qquad \text{(UnitPropagate)}$$

$$M \parallel F \quad \Longrightarrow \quad M\ell_{C\vee\ell} \parallel F \qquad \textbf{if} \begin{cases} \ell \text{ unassigned in } M \\ \ell \text{ or } \overline{\ell} \text{ occurs in } F \\ T \vdash C \vee \ell \\ M \models \neg C \end{cases} \qquad (T\text{-Propagate})$$

Abstract CDCL(T) continued

$M \parallel F, C$	$\Longrightarrow M \parallel F, C \parallel C$	**if** $M \models \neg C$	(Conflict)
$M \parallel F$	$\Longrightarrow M \parallel F \parallel C$	**if** $\begin{cases} T \vdash C \\ M \models \neg C \end{cases}$	(T-Conflict)
$M \parallel F \parallel C' \vee \overline{\ell}$	$\Longrightarrow M \parallel F \parallel C \vee C'$	**if** $\ell_{C \vee \ell} \in M$	(Resolve)
$M \parallel F \parallel C$	$\Longrightarrow M \parallel F, C \parallel C$	**if** $C \notin F$	(Learn)
$M \ell_0 M' \parallel F \parallel C \vee \ell$	$\Longrightarrow M \ell_{C \vee \ell} \parallel F$	**if** $\begin{cases} M \models \neg C \\ \ell \text{ unassigned in } M \end{cases}$	(Backjump)
$M \parallel F \parallel \square$	$\Longrightarrow$ `unsat`		(Unsat)

CDCL + Theory: Example

$$
\begin{aligned}
p &\equiv 3 < x \\
q &\equiv x < 0 \\
r &\equiv x < y \\
s &\equiv y < 0
\end{aligned}
$$

$$\| \;\; p,\; q \vee r,\; s \vee \neg r$$

CDCL + Theory: Example

$$\begin{aligned} p &\equiv 3 < x \\ q &\equiv x < 0 \\ r &\equiv x < y \\ s &\equiv y < 0 \end{aligned}$$

$$\begin{array}{rl} \| & p,\ q \vee r,\ s \vee \neg r \ \Rightarrow \ \text{(UnitPropagate)} \\ p \ \| & p,\ q \vee r,\ s \vee \neg r \end{array}$$

CDCL + Theory: Example

$$\begin{array}{lcl} p & \equiv & 3 < x \\ q & \equiv & x < 0 \\ r & \equiv & x < y \\ s & \equiv & y < 0 \end{array}$$

$$\begin{array}{rcl} & \parallel & p,\ q \vee r,\ s \vee \neg r \ \Rightarrow \ \text{(UnitPropagate)} \\ p & \parallel & p,\ q \vee r,\ s \vee \neg r \end{array}$$

$\underbrace{3 < x}_{p}$ implies $\neg \underbrace{x < 0}_{q}$ so $T \vdash \neg p \vee \neg q$

CDCL + Theory: Example

$$\begin{aligned} p &\equiv 3 < x \\ q &\equiv x < 0 \\ r &\equiv x < y \\ s &\equiv y < 0 \end{aligned}$$

$$\begin{aligned} &\parallel \; p,\; q \vee r,\; s \vee \neg r \;\Rightarrow\; \text{(UnitPropagate)} \\ p &\parallel \; p,\; q \vee r,\; s \vee \neg r \;\Rightarrow\; \text{(T-Propagate)} \\ p\; \neg q_{\neg p \vee \neg q} &\parallel \; p,\; q \vee r,\; s \vee \neg r \end{aligned}$$

CDCL + Theory: Example

$$
\begin{aligned}
p &\equiv 3 < x \\
q &\equiv x < 0 \\
r &\equiv x < y \\
s &\equiv y < 0
\end{aligned}
$$

$$
\begin{array}{rcll}
 & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\
p & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(T-Propagate)} \\
p\ \neg q_{\neg p \vee \neg q} & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\
p_p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r} & \| & p,\ q \vee r,\ s \vee \neg r &
\end{array}
$$

CDCL + Theory: Example

$$
\begin{aligned}
p &\equiv 3 < x\\
q &\equiv x < 0\\
r &\equiv x < y\\
s &\equiv y < 0
\end{aligned}
$$

$$
\begin{array}{rll}
\parallel & p,\ q \vee r,\ s \vee \neg r \ \Rightarrow & \text{(UnitPropagate)}\\
p \parallel & p,\ q \vee r,\ s \vee \neg r \ \Rightarrow & \text{(T-Propagate)}\\
p\ \neg q_{\neg p \vee \neg q} \parallel & p,\ q \vee r,\ s \vee \neg r \ \Rightarrow & \text{(UnitPropagate)}\\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r} \parallel & p,\ q \vee r,\ s \vee \neg r \ \Rightarrow & \text{(UnitPropagate)}\\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} \parallel & p,\ q \vee r,\ s \vee \neg r &
\end{array}
$$

CDCL + Theory: Example

$$\begin{aligned} p &\equiv 3 < x \\ q &\equiv x < 0 \\ r &\equiv x < y \\ s &\equiv y < 0 \end{aligned}$$

$$\begin{array}{rcll} & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\ p & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(T-Propagate)} \\ p\ \neg q_{\neg p \vee \neg q} & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\ p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r} & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\ p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} & \| & p,\ q \vee r,\ s \vee \neg r & \end{array}$$

$\underbrace{3 < x}_{p} \wedge \underbrace{x < y}_{r} \wedge \underbrace{y < 0}_{s}$ is false so $T \vdash \neg p \vee \neg r \vee \neg s$

CDCL + Theory: Example

$$
\begin{aligned}
p &\equiv 3 < x \\
q &\equiv x < 0 \\
r &\equiv x < y \\
s &\equiv y < 0
\end{aligned}
$$

$$
\begin{array}{rcl}
 \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\
p \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(T-Propagate)} \\
p\ \neg q_{\neg p \vee \neg q} \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r} \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(T-Conflict)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} \| & p,\ q \vee r,\ s \vee \neg r & \| \quad \neg p \vee \neg r \vee \neg s
\end{array}
$$

CDCL + Theory: Example

$$
\begin{aligned}
p &\equiv 3 < x \\
q &\equiv x < 0 \\
r &\equiv x < y \\
s &\equiv y < 0
\end{aligned}
$$

$$
\begin{array}{rlll}
 & \| \; p, \; q \vee r, \; s \vee \neg r & \Rightarrow & \text{(UnitPropagate)} \\
p & \| \; p, \; q \vee r, \; s \vee \neg r & \Rightarrow & \text{(T-Propagate)} \\
p \; \neg q_{\neg p \vee \neg q} & \| \; p, \; q \vee r, \; s \vee \neg r & \Rightarrow & \text{(UnitPropagate)} \\
p \; \neg q_{\neg p \vee \neg q} \; r_{q \vee r} & \| \; p, \; q \vee r, \; s \vee \neg r & \Rightarrow & \text{(UnitPropagate)} \\
p \; \neg q_{\neg p \vee \neg q} \; r_{q \vee r} \; s_{s \vee \neg r} & \| \; p, \; q \vee r, \; s \vee \neg r & \Rightarrow & \text{(T-Conflict)} \\
p \; \neg q_{\neg p \vee \neg q} \; r_{q \vee r} \; s_{s \vee \neg r} & \| \; p, \; q \vee r, \; s \vee \neg r & \| & \neg p \vee \neg r \vee \neg s \quad \Rightarrow \; \text{(Resolve)} \\
p \; \neg q_{\neg p \vee \neg q} \; r_{q \vee r} \; s_{s \vee \neg r} & \| \; p, \; q \vee r, \; s \vee \neg r & \| & \neg p \vee \neg r
\end{array}
$$

CDCL + Theory: Example

$$\begin{aligned} p &\equiv 3 < x \\ q &\equiv x < 0 \\ r &\equiv x < y \\ s &\equiv y < 0 \end{aligned}$$

$$\begin{array}{rcll}
 & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\
p & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(T-Propagate)} \\
p\ \neg q_{\neg p \vee \neg q} & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r} & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(UnitPropagate)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow \text{(T-Conflict)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} & \| & p,\ q \vee r,\ s \vee \neg r & \|\ \ \neg p \vee \neg r \vee \neg s \quad \Rightarrow \text{(Resolve)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} & \| & p,\ q \vee r,\ s \vee \neg r & \|\ \ \neg p \vee \neg r \quad \Rightarrow \text{(Resolve)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} & \| & p,\ q \vee r,\ s \vee \neg r & \|\ \ \neg p
\end{array}$$

CDCL + Theory: Example

$$
\begin{aligned}
p &\equiv 3 < x\\
q &\equiv x < 0\\
r &\equiv x < y\\
s &\equiv y < 0
\end{aligned}
$$

$$
\begin{array}{rcllll}
 & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow & \text{(UnitPropagate)} & \\
p & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow & \text{(T-Propagate)} & \\
p\ \neg q_{\neg p \vee \neg q} & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow & \text{(UnitPropagate)} & \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r} & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow & \text{(UnitPropagate)} & \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} & \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow & \text{(T-Conflict)} & \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} & \| & p,\ q \vee r,\ s \vee \neg r & \| & \neg p \vee \neg r \vee \neg s & \Rightarrow\ \text{(Resolve)}\\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} & \| & p,\ q \vee r,\ s \vee \neg r & \| & \neg p \vee \neg r & \Rightarrow\ \text{(Resolve)}\\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} & \| & p,\ q \vee r,\ s \vee \neg r & \| & \neg p & \Rightarrow\ \text{(Resolve)}\\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} & \| & p,\ q \vee r,\ s \vee \neg r & \| & \square &
\end{array}
$$

CDCL + Theory: Example

$$
\begin{aligned}
p &\equiv 3 < x \\
q &\equiv x < 0 \\
r &\equiv x < y \\
s &\equiv y < 0
\end{aligned}
$$

$$
\begin{array}{rlll}
 \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow & \text{(UnitPropagate)} \\
p \ \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow & \text{(T-Propagate)} \\
p\ \neg q_{\neg p \vee \neg q} \ \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow & \text{(UnitPropagate)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r} \ \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow & \text{(UnitPropagate)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} \ \| & p,\ q \vee r,\ s \vee \neg r & \Rightarrow & \text{(T-Conflict)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} \ \| & p,\ q \vee r,\ s \vee \neg r \ \| \ \neg p \vee \neg r \vee \neg s & \Rightarrow & \text{(Resolve)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} \ \| & p,\ q \vee r,\ s \vee \neg r \ \| \ \neg p \vee \neg r & \Rightarrow & \text{(Resolve)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} \ \| & p,\ q \vee r,\ s \vee \neg r \ \| \ \neg p & \Rightarrow & \text{(Resolve)} \\
p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} \ \| & p,\ q \vee r,\ s \vee \neg r \ \| \ \square & \Rightarrow & \text{(Unsat)}
\end{array}
$$

unsat

More on Theory Propagation

T-Propagation is optional

- Without it, the Decide rule may branch the wrong way but this will lead to a T-Conflict (detected later)

T-Propagation can be expensive

- There's a tradeoff between the cost of theory propagation and the search-space reduction it provides
- In practice, theory solvers use incomplete forms of theory propagation. They find some literals that are implied by the current assignment M, but not necessarily all of them

Minimal Explanations

T-Conflict

- the theory solver produces a clause C such that

$$M \models \neg C \text{ and } T \vdash C$$

T-Propagate:

- the theory solver finds ℓ and C such that

$$M \models \neg C \text{ and } T \vdash C \vee \ell$$

Precise Explanations:

- In both cases, there may be several such clauses C
- If C contains irrelevant literals, then the rules are still sound, but performance is worse
- For best pruning, the theory solver should produce minimal explanations C

Ideal Properties of Theory Solver

Incrementality

- Theory solvers process successive sets of literals $M_0 \subset M_1 \subset \ldots \subset M_k$
  Should try to reuse work: save results from processing M_i to accelerate processing of M_{i+1}

Fast Backtracking

Efficient Theory Propagation

Precise Theory Explanations

Equality with Uninterpreted Functions

Theory Solver for EUF

Equality Axioms

- Reflexivity: $x = x$
- Symmetry: $x = y \Rightarrow y = x$
- Transitivity: $x = y \wedge y = z \Rightarrow x = z$
- Congruence: $x_1 = y_1 \wedge \ldots \wedge x_n = y_n \Rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$

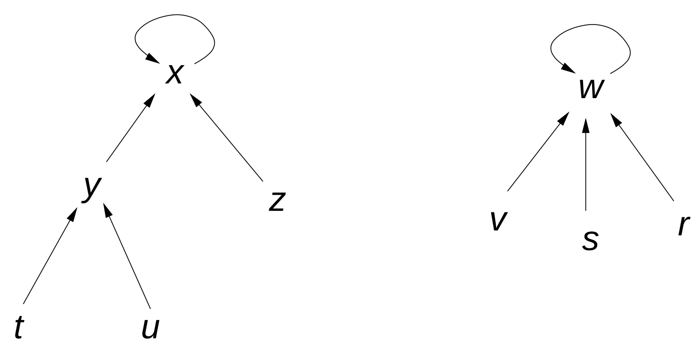
Theory Solver

- Given a set M of equalities and disequalities between terms:
  - Check whether M is consistent, if not find a minimal explanation
  - Propagate implied equalities and disequalities

A Simple Case: No Function Symbols

Union-Find Data Structure

- The theory solver state consists of
  - a find structure F that maintains equivalence classes
  - a set of disequalities D
- F defines a set of merge trees



- $F(x) = x$ if x is a root, otherwise $F(x)$ is the parent of x

Union-Find

Equivalence Relation

- Let $F^*(x)$ denote the root of the tree containing x, then
  x and y are equal if $F^*(x) = F^*(y)$ (i.e., x and y are in the same tree)

Union Operation

- Processing equality $x = y$ amounts to merging the classes of x and y
- Let $\mathsf{sz}(F, x)$ denote the size of the equivalence class that contains x then

$$\mathsf{union}(F, x, y) \;=\; \begin{cases} F & \text{if } x' = y' \\ F[x' := y'] & \text{if } x' \neq y' \text{ and } \mathsf{sz}(F, x) < \mathsf{sz}(F, y) \\ F[y' := x'] & \text{otherwise} \end{cases}$$

where $x' = F^*(x)$ and $y' = F^*(y)$

- Optimization: path compression, update F when computing $F^*(x)$.

Theory Solver for Variable Equalities

State

- F: the find structure
- D: a set of disequalities
- Initially:
  - $F(x) = x$ for all x
  - $D = \emptyset$
- The state is inconsistent iff there are x and y such that $F^*(x) = F^*(y)$ and $(x \neq y) \in D$

Operations

- $\mathsf{addeq}(x = y, F, D)$: add an equality
- $\mathsf{addneq}(x \neq y, F, D)$: add a disequality

Both operations either report unsatisfiability or return a new state $\langle F', D' \rangle$

Processing Equalities

$$\mathsf{addeq}(x = y, F, D) \;:=\; \langle F, D\rangle \text{ if } F^*(x) = F^*(y)$$

$$\mathsf{addeq}(x = y, F, D) \;:=\; \begin{cases} \mathsf{unsat} & \text{if } F'^*(u) \equiv F'^*(v) \text{ for some} \\ & u \neq v \in D \\ \langle F', D\rangle & \text{otherwise} \end{cases}$$

where $F^*(x) \not\equiv F^*(y)$ and $F' = \mathsf{union}(F, x, y)$

Processing Disequalities

$$\begin{array}{lcl} \mathsf{addneq}(x \neq y, F, D) & := & \mathsf{unsat} \text{ if } F^*(x) \equiv F^*(y) \\ \mathsf{addneq}(x \neq y, F, D) & := & \langle F, D \rangle \text{ if there is } u \neq v \in D \text{ or } v \neq u \in D \\ & & \text{such that } F^*(x) = F^*(u) \text{ and } F^*(y) = F^*(v) \\ \mathsf{addneq}(x \neq y, F, D) & := & \langle F, D \cup \{x \neq y\} \rangle \text{ otherwise} \end{array}$$

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_2, x_3 \mapsto x_3, x_4 \mapsto x_4, x_5 \mapsto x_5\} \\ D &= \emptyset \end{aligned}$$

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_2, x_3 \mapsto x_3, x_4 \mapsto x_4, x_5 \mapsto x_5\} \\ D &= \emptyset \end{aligned}$$

Merge classes of x_1 and x_2

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_3, x_4 \mapsto x_4, x_5 \mapsto x_5\} \\ D &= \emptyset \end{aligned}$$

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_3, x_4 \mapsto x_4, x_5 \mapsto x_5\} \\ D &= \emptyset \end{aligned}$$

Merge classes of x_1 and x_3

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_4, x_5 \mapsto x_5\} \\ D &= \emptyset \end{aligned}$$

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_4, x_5 \mapsto x_5\} \\ D &= \emptyset \end{aligned}$$

No change: x_2 and x_3 are already in the same class

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_4, x_5 \mapsto x_5\} \\ D &= \emptyset \end{aligned}$$

Add disequality

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_4, x_5 \mapsto x_5\} \\ D &= \{x_2 \neq x_4\} \end{aligned}$$

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_4, x_5 \mapsto x_5\} \\ D &= \{x_2 \neq x_4\} \end{aligned}$$

Merge classes of x_4 and x_5

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_4, x_5 \mapsto x_4\} \\ D &= \{x_2 \neq x_4\} \end{aligned}$$

Example

$$x_1 = x_2,\ x_1 = x_3,\ x_2 = x_3,\ x_2 \neq x_4,\ x_4 = x_5$$

$$\begin{aligned} F &= \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_4, x_5 \mapsto x_4\} \\ D &= \{x_2 \neq x_4\} \end{aligned}$$

Model Found

- domain $E = \{a, b\}$ (two equivalence classes in F)
- $M(x_1) = M(x_2) = M(x_3) = a$
- $M(x_4) = M(x_5) = b$

General Case: Function Symbols

Congruence Closure

- The data structure must now define a congruence-closed equivalence relation
  For any n-ary function F, we must have

$$x_1 = y_1, \ldots, x_1 = y_n \Rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$$

Use Lists

- We add a new index to the solver state: $\pi(t)$ is the set of terms that contain t or any term in the same class as t
- Example:

$$\{f(f(a)),\ g(a),\ a,\ g(b)\} \quad F = \{b \mapsto a, g(a) \mapsto g(b), \ldots\}$$

$$\begin{array}{lcl} \pi(a) & = & \{f(a),\ g(a),\ g(b)\} \\ \pi(f(a)) & = & \{f(f(a))\} \end{array} \qquad \begin{array}{rcl} \pi(g(a)) & = & \emptyset \\ \pi(f(f(a))) & = & \emptyset \end{array}$$

Congruence Closure

Processing Equality

- When we merge the classes of s and t, we must do more than before
  - Merge the use lists $\pi(s')$ and $\pi(t')$, where $s' = F^*(s)$ and $t' = F^*(t)$
  - Find equalities implied by $s = t$ by congruence:
    - Find u in $\pi(s')$ and v in $\pi(t;)$ such that u and v are congruent in F
    - Add $u = v$ to a queue of equalities to process
  - Process all equalities in the queue

Possible Implementations of Use Lists

Circular Lists

- constant time merge and split

Vectors

- linear-time merge: add $\pi(s')$ to $\pi(t')$
- constant-time split: shrink vector $\pi(t')$

No Explicit Merge

- keep the use lists fixed
- scan the equivalence classes for finding congruent terms

Implementation: Congruence Table

To quickly find congruent terms u and v, use a hash table

- Hash of $f(t_1, \ldots, t_n)$ is based on the term's signature:

$$\sigma(f(t_1, \ldots, t_n) = \langle f, F^*(t_1), \ldots, F^*(t_n) \rangle$$

- Property: $\sigma(u) = \sigma(v)$ iff u and v are congruent in F
- The hash table stores one term per signature
- When searching for congruences:
  - scan $\pi(s')$
  - recompute the signature $\sigma(u)$ for every u in $\pi(s')$
  - check whether there's a term v with the same signature in the hash table
  - if not add u to the hash table

Theory Explanations for EUF

Problem: Given two terms u and v that are in the same equivalence class (i.e., $F^*(u) = F^*(v)$) find a set of equalities that imply $u = v$

For variable equalities

- Use the merge tree: find the shortest path between u and v
- This ensures that the explanation is non-redundant

General case

- Label edges in the merge tree with a local explanation: $x = y$ may be asserted or implied by congruence
- Explanation generation:
  - find path between u and v
  - collect edges and recursively build explanation for the congruence edges
- Issue: not guaranteed to generate minimal explanations

Explanations: Example

Input Equalities

$$
\begin{array}{rclclcl}
f_1(x_1) & = & x_1 & = & x_2 & = & f_1(x_{n+1}) \\
f_2(x_1) & = & x_2 & = & x_3 & = & f_2(x_{n+1}) \\
 & & & \vdots & & & \\
f_n(x_1) & = & x_n & = & x_{n+1} & = & f_n(x_{n+1})
\end{array}
$$

Implied Equality

$$
g(f_1(x_1), \ldots, f_n(x_1)) \;=\; g(f_1(x_{n+1}), \ldots, f_n(x_{n+1}))
$$

Extensions of the Basic Union-Find Techniques

Dynamic Ackermann Lemmas

- Ackermann Trick: if we explicitly add all instances of the congruence axiom, then equality + Boolean reasoning are enough (no need for congruence closure)
  This is usually too expensive to be done eagerly
- But it helps to heuristically generate some instances and add them dynamically during the search:

$$u_1 = t_1 \wedge \ldots \wedge u_n = t_n \;\Rightarrow\; f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)$$

- Benefit: this new lemma improves theory propagation
  Example from $f(x, y) \neq f(y, x)$, congruence closure can't deduce $x \neq y$ but the SAT solver can do it, if we add the lemma

$$x = y \;\Rightarrow f(x, y) = f(y, x)$$

Extensions of the Basic Union-Find Techniques

Offset Equalities

- Offset equalities are of the form $x = y + c$ where c is a rational (or integer) constant
- Union-Find + congruence closure algorithms can be extended to handle them

Array Theory

- Can be implemented on top of EUF: by instantiating the array axioms

$$
\begin{aligned}
read(write(a,i,v),i) &= v \\
read(write(a,i,v),j) &= read(a,j) \text{ if } i \neq j
\end{aligned}
$$

Linear Arithmetic

Linear Arithmetic Solvers

Linear Arithmetic

- Atoms are of the form $a_1x_1 + \ldots + a_nx_n \bowtie b$ where
  - $a_1, \ldots, a_n$ and b are rational constants
  - $\bowtie$ is one of the predicates $\leqslant$, $<$, $=$, etc.
  - $x_1, \ldots, x_n$ are real or integer variables

- Variants:
  - Difference Logic: atoms are of the form $x - y \leqslant b$
  - Linear Integer Arithmetic: all variables are integer
  - Linear Real Arithmetic: all variables are real
  - Mixed Arithmetic: mixed both real and integer variables

Algorithms for Linear Arithmetic Solvers

Difference Logic

- Graph-based algorithms (to detect negative circuits)

General Case

- Fourier-Motzkin Elimination: eliminate variables using rules such as
  $t_1 \leqslant ax, \quad bx \leqslant t_2 \quad \Rightarrow \quad bt_1 \leqslant at_2$ (provided $a > 0$ and $b > 0$)
- Simplex (generally more scalable than Fourier-Motzkin)

Main issue: how to adapt Simplex to SMT solving?

- efficiently support addition/retraction of constraints
- generate (precise) explanations
- support theory propagation

Simplex in Standard Form

Standard Form:

$$Ax = b \text{ and } x \geqslant 0$$

where A is a matrix, b is a constant vector and x is a vector of variables

limits of this form for SMT

- to solve incremental problems: add rows to A (expensive)
- slow backtracking (same issue: need to remove rows from A)
- no theory propagation

Fast Linear Arithmetic Solver for SMT

Use Simplex in General Form

Algorithm is based on the Dual Simplex

Gives precise theory explanations

Efficient backtracking

Efficient theory propagation

Support strict inequalities (e.g., $x > 0$)

Allow presimplification step

To deal with integer problems: Gomory cuts, Branch & Bound, GCD test

General Form Simplex

General Form: $Ax = 0$ and $l_j \leqslant x_j \leqslant u_j$

We can always convert linear arithmetic problems to this form

Example:

$$
\begin{array}{l}
x \geqslant 0, (x + y \leqslant 2 \vee x + 2y \geqslant 6), (x + y = 2 \vee x + 2y > 4) \\
\rightsquigarrow \\
s_1 = x + y, s_2 = x + 2y, \\
x \geqslant 0, (s_1 \leqslant 2 \vee s_2 \geqslant 6), (s_1 = 2 \vee s_2 > 4)
\end{array}
$$

Main Benefits

- The matrix A is fixed: no need to add or remove rows
- Incrementality means adding/removing bounds on variables (e.g., $s_1 \leqslant 2$)
- Unconstrained variables can be eliminated before the search

Tableau and Assignment

Tableau

- Simplex turns s $Ax = 0$ into the following form (called a tableau)

$$\begin{aligned} y_1 &= a_{11}x_1 + \ldots + a_{1n}x_n \\ &\vdots \\ y_m &= a_{m1}x_1 + \ldots + a_{mn}x_n \end{aligned}$$

$y_1, \ldots, y_m$ are basic (or dependent) variables
$x_1, \ldots, x_n$ are non-basic (or independent) variables

Assignment

- An assignment (model) is a mapping from variables to values
- The value of dependent variables is computed form the assignment of independent variables

Algorithm Properties

The algorithm maintains an assignment that satisfies all equations and bounds

To process new constraints

- the assignment and tableau are updated using pivoting
- pivoting swaps one basic and one non-basic variable
- when pivoting fails to produce satisfying assignment, we get a conflict explanation from one equation (one row of the tableau)

Backtracking is very cheap: just remove bounds (the assignment and tableau don't change)

Theory propagation: use bounds and equations to derive new bounds on variables

- Example: $x = y - z,\ y \leqslant 2,\ z \geqslant 3 \leadsto x \leqslant -1$

Main Procedure

Solver State

- Equations (i.e., tableau)

$$\begin{array}{rcl} y_1 & = & a_{11}x_1 + \ldots + a_{1n}x_n \\ & \vdots & \\ y_m & = & a_{m1}x_1 + \ldots + a_{mn}x_n \end{array}$$

- Bounds: $l_i \leqslant x_1 \leqslant u_i$ and $l'_j \leqslant y_j \leqslant u_j$
- Assignment: M assigns values to $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$
- Invariant: all bounds on $x_1 \ldots, x_n$ are satisfied

Procedure

- Assume some of the bounds on are violated by M: say $M(y_1) < l'_1$
- How do we fix the tableau and assignment?

Main Procedure

First row in the tableau

$$y_1 \;=\; a_{11}x_1 + \ldots + a_{1n}x_n$$

To satisfy $l'_1 \leqslant M(y_1)$, we want to increase $M(y_1)$

- If $a_{1i} > 0$ and $M(x_i) < u_i$ then $M(x_i)$ can increase and this makes $M(y_1)$ increase
- If $a_{1i} < 0$ and $M(x_i) > l_i$ then $M(x_i)$ can decrease and this makes $M(y_i)$ increase

In either cases, we can pivot x_i and y:

- Rewrite the first row to

$$x_i \;=\; \frac{1}{a_{1i}}y - \frac{a_{11}}{a_{1i}}x_1 - \ldots - \frac{a_{1n}}{a_{1i}}x_n$$

- Update the assignment by setting $M(y_1) := l'_1$

Then we check if some other bounds is violated and iterate

Conflict and Theory Explanation

If there's no suitable x_i

$$y_1 \;=\; a_{11}x_1 + \ldots + a_{1n}x_n$$

- We must have $a_{1i} > 0 \;\Rightarrow\; M(x_i) = u_i$ and $a_{1i} < 0 \;\Rightarrow\; M(x_i) = l_i$
- Then

$$M(y_1) \;=\; (\sum_{a_{1i}>0} a_{1i}u_i) + (\sum_{a_{1i}<0} a_{i1}l_i)$$

  and we have $M(y_1) < l'_1$

We have found a contradiction:

$$\bigvee_{a_{i1}>0} (x_i \leqslant u_i) \;\vee\; \bigvee_{a_{i1}<0} (x_i \geqslant l_i) \;\Rightarrow\; (y_1 < l'_1)$$

This gives us a theory explanation

How to Handle Strict Inequalities

In the general form, all bounds are non-strict (e.g,, $x \leqslant 2$)

For integer problems, that's not an issue:
strict inequalities can be converted to non-strict (e.g., $x < 1 \leadsto x \leqslant 0$)

For real or rational problems:

- introduce a symbolic, infinitesimal parameter δ
- convert $x < c$ to $x \leqslant c - \delta$, and $x > c$ to $x \geqslant c + \delta$)
- now the assignment maps variables to values of the form $c + d\delta$, where c and d are rational
- we use the following ordering relation on these values

$$c_1 + d_1\delta \leqslant c_2 + d_2\delta \text{ iff } c_1 < c_2 \text{ or } (c_1 = c_2 \text{ and } d_1 < d_2)$$

Example

Initial state

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	
$M(y) = 0$	$u = x + 2y$	
$M(s) = 0$	$v = x - y$	
$M(u) = 0$		
$M(v) = 0$		

Example

Asserting $s \geqslant 1$

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	
$M(y) = 0$	$u = x + 2y$	
$M(s) = 0$	$v = x - y$	
$M(u) = 0$		
$M(v) = 0$		

Example

Asserting $s \geqslant 1$: the assignment does not satisfy the new bound

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 0$	$u = x + 2y$	
$M(s) = 0$	$v = x - y$	
$M(u) = 0$		
$M(v) = 0$		

Example

Asserting $s \geqslant 1$: pivot s and x

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 0$	$u = x + 2y$	
$M(s) = 0$	$v = x - y$	
$M(u) = 0$		
$M(v) = 0$		

Example

Asserting $s \geqslant 1$: pivot s and x

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \lor v \geqslant 2), (v \leqslant -2 \lor v \geqslant 0), (v \leqslant -2 \lor u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$x = s - y$	$s \geqslant 1$
$M(y) = 0$	$u = x + 2y$	
$M(s) = 0$	$v = x - y$	
$M(u) = 0$		
$M(v) = 0$		

Example

Asserting $s \geqslant 1$: pivot s and x

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$x = s - y$	$s \geqslant 1$
$M(y) = 0$	$u = s + y$	
$M(s) = 0$	$v = s - 2y$	
$M(u) = 0$		
$M(v) = 0$		

Example

Asserting $s \geqslant 1$: update the assignment for s

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$x = s - y$	$s \geqslant 1$
$M(y) = 0$	$u = s + y$	
$M(s) = 1$	$v = s - 2y$	
$M(u) = 0$		
$M(v) = 0$		

Example

Asserting $s \geqslant 1$: update the dependent variables value

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 1$	$x = s - y$	$s \geqslant 1$
$M(y) = 0$	$u = s + y$	
$M(s) = 1$	$v = s - 2y$	
$M(u) = 1$		
$M(v) = 1$		

Example

Asserting $x \geqslant 0$

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 1$	$x = s - y$	$s \geqslant 1$
$M(y) = 0$	$u = s + y$	
$M(s) = 1$	$v = s - 2y$	
$M(u) = 1$		
$M(v) = 1$		

Example

Asserting $x \geqslant 0$: nothing to do. The bound is satisfied by M

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 1$	$x = s - y$	$s \geqslant 1$
$M(y) = 0$	$u = s + y$	$x \geqslant 0$
$M(s) = 1$	$v = s - 2y$	
$M(u) = 1$		
$M(v) = 1$		

Example

Case split: $\neg(y \leqslant 1)$

$$s \geqslant 1, x \geqslant 0$$

$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 1$	$x = s - y$	$s \geqslant 1$
$M(y) = 0$	$u = s + y$	$x \geqslant 0$
$M(s) = 1$	$v = s - 2y$	
$M(u) = 1$		
$M(v) = 1$		

Example

Case split: $\neg(y \leqslant 1)$: the assignment does not satisfy the new bound

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 1$	$x = s - y$	$s \geqslant 1$
$M(y) = 0$	$u = s + y$	$x \geqslant 0$
$M(s) = 1$	$v = s - 2y$	$y > 1$
$M(u) = 1$		
$M(v) = 1$		

Example

Case split: $\neg(y \leqslant 1)$: update the assignment

$$s \geqslant 1, x \geqslant 0$$

$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 1$	$x = s - y$	$s \geqslant 1$
$M(y) = 1 + \delta$	$u = s + y$	$x \geqslant 0$
$M(s) = 1$	$v = s - 2y$	$y > 1$
$M(u) = 1$		
$M(v) = 1$		

Example

Case split: $\neg(y \leqslant 1)$: update dependent variables

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = -\delta$	$x = s - y$	$s \geqslant 1$
$M(y) = 1 + \delta$	$u = s + y$	$x \geqslant 0$
$M(s) = 1$	$v = s - 2y$	$y > 1$
$M(u) = 2 + \delta$		
$M(v) = -1 - 2\delta$		

Example

Bound violation

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = -\delta$	$x = s - y$	$s \geqslant 1$
$M(y) = 1 + \delta$	$u = s + y$	$x \geqslant 0$
$M(s) = 1$	$v = s - 2y$	$y > 1$
$M(u) = 2 + \delta$		
$M(v) = -1 - 2\delta$		

Example

Bound violation: pivot x and s

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \lor v \geqslant 2), (v \leqslant -2 \lor v \geqslant 0), (v \leqslant -2 \lor u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = -\delta$	$s = x + y$	$s \geqslant 1$
$M(y) = 1 + \delta$	$u = s + y$	$x \geqslant 0$
$M(s) = 1$	$v = s - 2y$	$y > 1$
$M(u) = 2 + \delta$		
$M(v) = -1 - 2\delta$		

Example

Bound violation: pivot x and s

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{aligned} M(x) &= -\delta \\ M(y) &= 1+\delta \\ M(s) &= 1 \\ M(u) &= 2+\delta \\ M(v) &= -1-2\delta \end{aligned}$$

Equations

$$\begin{aligned} s &= x+y \\ u &= x+2y \\ v &= x-y \end{aligned}$$

Bounds

$$\begin{array}{ccc} s & \geqslant & 1 \\ x & \geqslant & 0 \\ \hline y & > & 1 \end{array}$$

Example

Bound violation: update the assignment

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1 + \delta$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1$	$v = x - y$	$y > 1$
$M(u) = 2 + \delta$		
$M(v) = -1 - 2\delta$		

Example

Bound violation: update dependent variable values

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{aligned} M(x) &= 0 \\ M(y) &= 1+\delta \\ M(s) &= 1+\delta \\ M(u) &= 2+2\delta \\ M(v) &= -1-\delta \end{aligned}$$

Equations

$$\begin{aligned} s &= x+y \\ u &= x+2y \\ v &= x-y \end{aligned}$$

Bounds

$$\begin{array}{l} s \geqslant 1 \\ x \geqslant 0 \\ \hline y > 1 \end{array}$$

Example

Theory propagation: $x \geqslant 0, y > 1 \leadsto u > 2$

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{aligned} M(x) &= 0 \\ M(y) &= 1 + \delta \\ M(s) &= 1 + \delta \\ M(u) &= 2 + 2\delta \\ M(v) &= -1 - \delta \end{aligned}$$

Equations

$$\begin{aligned} s &= x + y \\ u &= x + 2y \\ v &= x - y \end{aligned}$$

Bounds

$$\begin{array}{c} s \geqslant 1 \\ x \geqslant 0 \\ \hline y > 1 \end{array}$$

Example

Theory propagation: $x \geqslant 0, y > 1 \rightsquigarrow u > 2$

$$s \geqslant 1, x \geqslant 0$$

$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1 + \delta$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1 + \delta$	$v = x - y$	$y > 1$
$M(u) = 2 + 2\delta$		$u > 2$
$M(v) = -1 - \delta$		

Example

Theory propagation: $u > 2 \leadsto \neg u \leqslant 1$

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{aligned} M(x) &= 0 \\ M(y) &= 1 + \delta \\ M(s) &= 1 + \delta \\ M(u) &= 2 + 2\delta \\ M(v) &= -1 - \delta \end{aligned}$$

Equations

$$\begin{aligned} s &= x + y \\ u &= x + 2y \\ v &= x - y \end{aligned}$$

Bounds

$$\begin{array}{ccc} s & \geqslant & 1 \\ x & \geqslant & 0 \\ \hline y & > & 1 \\ u & > & 2 \end{array}$$

Example

Boolean propagation: $v \geqslant 2$ must be true

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{aligned} M(x) &= 0 \\ M(y) &= 1+\delta \\ M(s) &= 1+\delta \\ M(u) &= 2+2\delta \\ M(v) &= -1-\delta \end{aligned}$$

Equations

$$\begin{aligned} s &= x+y \\ u &= x+2y \\ v &= x-y \end{aligned}$$

Bounds

$$\begin{array}{rcl} s & \geqslant & 1 \\ x & \geqslant & 0 \\ \hline y & > & 1 \\ u & > & 2 \end{array}$$

Example

Theory propagation: $v \geqslant 2 \rightsquigarrow \neg(v \leqslant -2)$

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1 + \delta$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1 + \delta$	$v = x - y$	$y > 1$
$M(u) = 2 + 2\delta$		$u > 2$
$M(v) = -1 - \delta$		

Example

Conflict: empty clause

$$s \geqslant 1, x \geqslant 0$$

$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{aligned} M(x) &= 0 \\ M(y) &= 1 + \delta \\ M(s) &= 1 + \delta \\ M(u) &= 2 + 2\delta \\ M(v) &= -1 - \delta \end{aligned}$$

Equations

$$\begin{aligned} s &= x + y \\ u &= x + 2y \\ v &= x - y \end{aligned}$$

Bounds

$$\begin{array}{ccc} s & \geqslant & 1 \\ x & \geqslant & 0 \\ \hline y & > & 1 \\ u & > & 2 \end{array}$$

Example

Backtracking

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1 + \delta$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1 + \delta$	$v = x - y$	
$M(u) = 2 + 2\delta$		
$M(v) = -1 - \delta$		

Example

Asserting $y \leqslant 1$

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1 + \delta$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1 + \delta$	$v = x - y$	
$M(u) = 2 + 2\delta$		
$M(v) = -1 - \delta$		

Example

Asserting $y \leqslant 1$: the assignment does not satisfy the new bound

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{aligned} M(x) &= 0 \\ M(y) &= 1 + \delta \\ M(s) &= 1 + \delta \\ M(u) &= 2 + 2\delta \\ M(v) &= -1 - \delta \end{aligned}$$

Equations

$$\begin{aligned} s &= x + y \\ u &= x + 2y \\ v &= x - y \end{aligned}$$

Bounds

$$\begin{aligned} s &\geqslant 1 \\ x &\geqslant 0 \\ \hline y &\leqslant 1 \end{aligned}$$

Example

Asserting $y \leqslant 1$: update the assignment

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1 + \delta$	$v = x - y$	$y \leqslant 1$
$M(u) = 2 + 2\delta$		
$M(v) = -1 - \delta$		

Example

Asserting $y \leqslant 1$: update dependent variable values

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1$	$v = x - y$	$y \leqslant 1$
$M(u) = 2$		
$M(v) = -1$		

Example

Theory propagation: $x \geqslant 0, y \leqslant 1 \leadsto v \geqslant -1$

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{aligned} M(x) &= 0 \\ M(y) &= 1 \\ M(s) &= 1 \\ M(u) &= 2 \\ M(v) &= -1 \end{aligned}$$

Equations

$$\begin{aligned} s &= x + y \\ u &= x + 2y \\ v &= x - y \end{aligned}$$

Bounds

$$\begin{array}{c} s \geqslant 1 \\ x \geqslant 0 \\ \hline y \leqslant 1 \end{array}$$

Example

Theory propagation: $x \geqslant 0, y \leqslant 1 \rightsquigarrow v \geqslant -1$

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1$	$v = x - y$	$y \leqslant 1$
$M(u) = 2$		$v \geqslant -1$
$M(v) = -1$		

Example

Theory propagation: $v \geqslant -1 \leadsto \neg(v \leqslant -2)$

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1$	$v = x - y$	$y \leqslant 1$
$M(u) = 2$		$v \geqslant -1$
$M(v) = -1$		

Example

Boolean propagation: $v \geqslant 0$ must be true

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{aligned} M(x) &= 0 \\ M(y) &= 1 \\ M(s) &= 1 \\ M(u) &= 2 \\ M(v) &= -1 \end{aligned}$$

Equations

$$\begin{aligned} s &= x + y \\ u &= x + 2y \\ v &= x - y \end{aligned}$$

Bounds

$$\begin{aligned} s &\geqslant 1 \\ x &\geqslant 0 \\ \hline y &\leqslant 1 \\ v &\geqslant -1 \end{aligned}$$

Example

Boolean propagation: $v \geqslant 0$ must be true

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1$	$v = x - y$	$y \leqslant 1$
$M(u) = 2$		$v \geqslant 0$
$M(v) = -1$		

Example

Bound violation: the assignment does not satisfy the new bound

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 0$	$s = x + y$	$s \geqslant 1$
$M(y) = 1$	$u = x + 2y$	$x \geqslant 0$
$M(s) = 1$	$v = x - y$	$y \leqslant 1$
$M(u) = 2$		$v \geqslant 0$
$M(v) = -1$		

Example

Bound violation: pivot v and x then update the assignment

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 1$	$s = v + 2y$	$s \geqslant 1$
$M(y) = 1$	$u = v + 3y$	$x \geqslant 0$
$M(s) = 2$	$x = v + y$	$y \leqslant 1$
$M(u) = 3$		$v \geqslant 0$
$M(v) = 0$		

Example

Boolean propagation: $u \leqslant -1$ must be true

$$s \geqslant 1, x \geqslant 0$$

$$(y \leqslant 1 \lor v \geqslant 2), (v \leqslant -2 \lor v \geqslant 0), (v \leqslant -2 \lor u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 1$	$s = v + 2y$	$s \geqslant 1$
$M(y) = 1$	$u = v + 3y$	$x \geqslant 0$
$M(s) = 2$	$x = v + y$	$y \leqslant 1$
$M(u) = 3$		$v \geqslant 0$
$M(v) = 0$		

Example

Boolean propagation: $u \leqslant -1$ must be true

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 1$	$s = v + 2y$	$s \geqslant 1$
$M(y) = 1$	$u = v + 3y$	$x \geqslant 0$
$M(s) = 2$	$x = v + y$	$y \leqslant 1$
$M(u) = 3$		$v \geqslant 0$
$M(v) = 0$		$u \leqslant -1$

Example

Bound violation

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 1$	$s = v + 2y$	$s \geqslant 1$
$M(y) = 1$	$u = v + 3y$	$x \geqslant 0$
$M(s) = 2$	$x = v + y$	$y \leqslant 1$
$M(u) = 3$		$v \geqslant 0$
$M(v) = 0$		$u \leqslant -1$

Example

Bound violation: pivot u and y then update the assignment

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{array}{rcl} M(x) & = & -\frac{1}{3} \\ M(y) & = & -\frac{1}{3} \\ M(s) & = & -\frac{2}{3} \\ M(u) & = & -1 \\ M(v) & = & 0 \end{array}$$

Equations

$$\begin{array}{rcl} s & = & \frac{2}{3}u + \frac{1}{3}v \\ y & = & \frac{1}{3}u - \frac{1}{3}v \\ x & = & \frac{1}{3}u + \frac{1}{3}v \end{array}$$

Bounds

$$\begin{array}{rcl} s & \geqslant & 1 \\ x & \geqslant & 0 \\ \hline y & \leqslant & 1 \\ v & \geqslant & 0 \\ u & \leqslant & -1 \end{array}$$

Example

Bound violations

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model

$$\begin{aligned} M(x) &= -\tfrac{1}{3} \\ M(y) &= -\tfrac{1}{3} \\ M(s) &= -\tfrac{2}{3} \\ M(u) &= -1 \\ M(v) &= 0 \end{aligned}$$

Equations

$$\begin{aligned} s &= \tfrac{2}{3}u + \tfrac{1}{3}v \\ y &= \tfrac{1}{3}u - \tfrac{1}{3}v \\ x &= \tfrac{1}{3}u + \tfrac{1}{3}v \end{aligned}$$

Bounds

$$\begin{array}{lll} s & \geqslant & 1 \\ x & \geqslant & 0 \\ \hline y & \leqslant & 1 \\ v & \geqslant & 0 \\ u & \leqslant & -1 \end{array}$$

Example

Bound violations: pivot s and v then update assignment

$$s \geqslant 1, x \geqslant 0$$

$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = -\frac{1}{3}$	$s = \frac{2}{3}u + \frac{1}{3}v$	$s \geqslant 1$
$M(y) = -\frac{1}{3}$	$y = \frac{1}{3}u - \frac{1}{3}v$	$x \geqslant 0$
$M(s) = -\frac{2}{3}$	$x = \frac{1}{3}u + \frac{1}{3}v$	$y \leqslant 1$
$M(u) = -1$		$v \geqslant 0$
$M(v) = 0$		$u \leqslant -1$

Example

Bound violations: pivot s and v then update assignment

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = -\frac{1}{3}$	$v = 3s - 2u$	$s \geqslant 1$
$M(y) = -\frac{1}{3}$	$y = \frac{1}{3}u - \frac{1}{3}v$	$x \geqslant 0$
$M(s) = -\frac{2}{3}$	$x = \frac{1}{3}u + \frac{1}{3}v$	$y \leqslant 1$
$M(u) = -1$		$v \geqslant 0$
$M(v) = 0$		$u \leqslant -1$

Example

Bound violations: pivot s and v then update assignment

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = -\frac{1}{3}$	$v = 3s - 2u$	$s \geqslant 1$
$M(y) = -\frac{1}{3}$	$y = -s + u$	$x \geqslant 0$
$M(s) = -\frac{2}{3}$	$x = 2s - u$	$y \leqslant 1$
$M(u) = -1$		$v \geqslant 0$
$M(v) = 0$		$u \leqslant -1$

Example

Bound violations: pivot s and v then update assignment

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 3$	$v = 3s - 2u$	$s \geqslant 1$
$M(y) = -2$	$y = -s + u$	$x \geqslant 0$
$M(s) = 1$	$x = 2s - u$	$y \leqslant 1$
$M(u) = -1$		$v \geqslant 0$
$M(v) = 5$		$u \leqslant -1$

Example

Success: found a satisfying assignment

$$s \geqslant 1, x \geqslant 0$$
$$(y \leqslant 1 \vee v \geqslant 2), (v \leqslant -2 \vee v \geqslant 0), (v \leqslant -2 \vee u \leqslant -1)$$

Model	Equations	Bounds
$M(x) = 3$	$v = 3s - 2u$	$s \geqslant 1$
$M(y) = -2$	$y = -s + u$	$x \geqslant 0$
$M(s) = 1$	$x = 2s - u$	$y \leqslant 1$
$M(u) = -1$		$v \geqslant 0$
$M(v) = 5$		$u \leqslant -1$

Other Techniques used for Linear Arithmetic SMT

Opportunistic Equality Propagation

- x_i is fixed if $l_i = u_i$
- propagating this in other rows leads to simple method for detecting some implied variable equalities (i.e., $x_j = y_k$)
- this is efficient but not complete

Extension to Linear Integer Arithmetic

- Use techniques from integer programming: GCD test, Gomory Cuts, Branch & Bound

Summary

CDCL + Theory Solver: generic framework for SMT Solving

Example Theory Solvers

- Equality + Uninterpreted Function: congruence closure algorithms
- Linear Arithmetic: Simplex-based

Main Issues

- Incrementality, Fast backtracking, Good explanations, Theory propagation

Other Relevant Topics

- How to combine multiple theories: Nelson-Oppen method and variants
- Solvers for arrays, recursive datatypes, etc.