Satisfiability Modulo Theories

Summer School on Formal Methods

Menlo College, 2011

Bruno Dutertre and Leonardo de Moura

`bruno@csl.sri.com, leonardo@microsoft.com`

SRI International, Microsoft Research

Roadmap

- Logic Background
- Modern SAT Solvers
- DPLL with Theory Solvers
- Theory Combination
- Equality
- Arithmetic
- Applications

Satisfiability Modulo Theories (SMT)

- In SMT solving, the Boolean atoms represent constraints over individual theory variables (ranging over integer, reals, bit-vectors, datatypes, arrays, etc.).
- The constraints can involve theory operations, equality, and inequality.
- Now, the SAT solver has to interact with theory solvers.
- The constraint solver can detect conflicts involving theory reasoning, e.g., $f(x) \neq f(y), x = y$, or $x - y \leq 2, y - z \leq -1, z - x \leq -3$.
- The theory solver must support incremental assertions, efficient backtracking and propagation, and produce efficient explanations of unsatisfiability.

Theory Solver: Examples

- Equality: $x = y$ (union-find), and offset equalities $x = y + k$.
- Term equality: congruence closure for uninterpreted function symbols.
- Difference constraints: incremental negative cycle detection for inequality constraints of the form $x - y \leq k$.
- Linear arithmetic: Fourier's method, Simplex.

Theory Solver: Rules

- We use $F \models_T G$ to denote the fact that F *entails* G in theory T.
- Abstract DPLL can be extended with two new rules to deal with theory T:

T-Propagate

$$M \parallel F \implies M\, l_{(\neg l_1 \vee \ldots \vee \neg l_n \vee l)} \parallel F \quad \textbf{if} \begin{cases} l \text{ occurs in } F, \\ l \text{ is undefined in } M, \\ l_1 \wedge \ldots \wedge l_n \models_T l, \\ l_1, \ldots, l_n \in \textit{lits}(M) \end{cases}$$

T-Conflict

$$M \parallel F \implies M \parallel F \parallel \neg l_1 \vee \ldots \vee \neg l_n \quad \textbf{if} \begin{cases} l_1 \wedge \ldots \wedge l_n \models_T \textit{false}, \\ l_1, \ldots, l_n \in \textit{lits}(M) \end{cases}$$

DPLL + Theory Solver

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

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

DPLL + Theory Solver

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

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

DPLL + Theory Solver

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

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

$\underbrace{3 < x}_{p}$ implies $\neg \underbrace{x < 0}_{q}$

DPLL + Theory Solver

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

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

DPLL + Theory Solver

$$p \equiv 3 < x$$
$$q \equiv x < 0$$
$$r \equiv x < y$$
$$s \equiv y < 0$$

$$\| \quad p,\ q \vee r,\ s \vee \neg r \quad \Rightarrow \quad \text{(UnitPropagate)}$$
$$p_p \ \| \quad p,\ q \vee r,\ s \vee \neg r \quad \Rightarrow \quad \text{(T-Propagate)}$$
$$p_p\ \neg q_{\neg p \vee \neg q} \ \| \quad p,\ q \vee r,\ s \vee \neg r \quad \Rightarrow \quad \text{(UnitPropagate)}$$
$$p_p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r} \ \| \quad p,\ q \vee r,\ s \vee \neg r \quad \Rightarrow \quad \text{(UnitPropagate)}$$
$$p_p\ \neg q_{\neg p \vee \neg q}\ r_{q \vee r}\ s_{s \vee \neg r} \ \| \quad p,\ q \vee r,\ s \vee \neg r$$

DPLL + Theory Solver

$$p \equiv 3 < x$$
$$q \equiv x < 0$$
$$r \equiv x < y$$
$$s \equiv y < 0$$

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

$\underbrace{3 < x}_{p},\ \underbrace{x < y}_{r},\ \underbrace{y < 0}_{s}$ implies false

DPLL + Theory Solver

- Do we need T-Propagate?

DPLL + Theory Solver

- Do we need T-Propagate?
  - No
  - Trade-off between precision and performance.

DPLL + Theory Solver

- Do we need T-Propagate?
  - No
  - Trade-off between precision and performance.
- What is the minimal functionality of a theory solver?

DPLL + Theory Solver

- Do we need T-Propagate?
  - No
  - Trade-off between precision and performance.
- What is the minimal functionality of a theory solver?
  - Check the unsatisfiability of conjunction of literals.

DPLL + Theory Solver

- Do we need T-Propagate?
  - No
  - Trade-off between precision and performance.
- What is the minimal functionality of a theory solver?
  - Check the unsatisfiability of conjunction of literals.
- Efficiently mining T-justifications

T-Propagate

$$M \parallel F \implies M\, l_{(\neg l_1 \vee \ldots \vee \neg l_n \vee l)} \parallel F \quad \textbf{if} \begin{cases} l \text{ occurs in } F, \\ l \text{ is undefined in } M, \\ l_1 \wedge \ldots \wedge l_n \models_T l, \\ l_1, \ldots, l_n \in \textit{lits}(M) \end{cases}$$

T-Conflict

$$M \parallel F \implies M \parallel F \parallel \neg l_1 \vee \ldots \vee \neg l_n \quad \textbf{if} \begin{cases} l_1 \wedge \ldots \wedge l_n \models_T \textit{false}, \\ l_1, \ldots, l_n \in \textit{lits}(M) \end{cases}$$

The Ideal Theory Solver

- Incremental
- Efficient Backtracking
- Efficient T-Propagate
- Precise T-Justifications

Roadmap

- Logic Background
- Modern SAT Solvers
- DPLL with Theory Solvers
- Theory Combination
- Equality
- Arithmetic
- Applications

Combination of Theories

- In practice, we need a combination of theories.
- Example:
  - $x + 2 = y \Rightarrow f(\textit{read}(\textit{write}(a, x, 3), y - 2)) = f(y - x + 1)$
- Given

$$\begin{array}{rcl} \Sigma & = & \Sigma_1 \cup \Sigma_2 \\ \mathcal{T}_1, \mathcal{T}_2 & : & \textit{theories over } \Sigma_1, \Sigma_2 \\ \mathcal{T} & = & \textit{DC}(\mathcal{T}_1 \cup \mathcal{T}_2) \end{array}$$

- Is $\mathcal{T}$ consistent?
- Given satisfiability procedures for conjunction of literals of $\mathcal{T}_1$ and $\mathcal{T}_2$, how to decide the satisfiability of $\mathcal{T}$?

Preamble

- Disjoint signatures: $\Sigma_1 \cap \Sigma_2 = \emptyset$.
- Purification
- Stably-Infinite Theories.
- Convex Theories.

Purification

- Goal: convert a formula ϕ into $\phi_1 \wedge \phi_2$, where ϕ_1 is in $\mathcal{T}_1$'s language and ϕ_2 is in $\mathcal{T}_2$'s language.
  So ϕ_1 and ϕ_2 have no common symbols, except variables.
- Purification step: replace term t by a fresh variable x
  $\phi \wedge F(\ldots, s[t], \ldots) \rightsquigarrow \phi \wedge F(\ldots, s[x], \ldots) \wedge x = t$,
- Purification is satisfiability preserving and terminating.

Purification

- Goal: convert a formula ϕ into $\phi_1 \wedge \phi_2$, where ϕ_1 is in $\mathcal{T}_1$'s language and ϕ_2 is in $\mathcal{T}_2$'s language.
  So ϕ_1 and ϕ_2 have no common symbols, except variables.
- Purification step: replace term t by a fresh variable x
  $\phi \wedge F(\ldots, s[t], \ldots) \leadsto \phi \wedge F(\ldots, s[x], \ldots) \wedge x = t$,
- Purification is satisfiability preserving and terminating.
- Example:

$$f(x - 1) - 1 = x, f(y) + 1 = y \leadsto$$

Purification

- Goal: convert a formula ϕ into $\phi_1 \wedge \phi_2$, where ϕ_1 is in $\mathcal{T}_1$'s language and ϕ_2 is in $\mathcal{T}_2$'s language.
  So ϕ_1 and ϕ_2 have no common symbols, except variables.
- Purification step: replace term t by a fresh variable x
  $\phi \wedge F(\ldots, s[t], \ldots) \leadsto \phi \wedge F(\ldots, s[x], \ldots) \wedge x = t$,
- Purification is satisfiability preserving and terminating.
- Example:

$$f(x-1)-1=x, f(y)+1=y \leadsto$$

$$f(u_1)-1=x, f(y)+1=y, u_1=x-1 \leadsto$$

Purification

- Goal: convert a formula ϕ into $\phi_1 \wedge \phi_2$, where ϕ_1 is in $\mathcal{T}_1$'s language and ϕ_2 is in $\mathcal{T}_2$'s language.
  So ϕ_1 and ϕ_2 have no common symbols, except variables.
- Purification step: replace term t by a fresh variable x
  $\phi \wedge F(\ldots, s[t], \ldots) \rightsquigarrow \phi \wedge F(\ldots, s[x], \ldots) \wedge x = t$,
- Purification is satisfiability preserving and terminating.
- Example:

$$f(x-1) - 1 = x, f(y) + 1 = y \rightsquigarrow$$

$$f(u_1) - 1 = x, f(y) + 1 = y, u_1 = x - 1 \rightsquigarrow$$

$$u_2 - 1 = x, f(y) + 1 = y, u_1 = x - 1, u_2 = f(u_1) \rightsquigarrow$$

Purification

- Goal: convert a formula ϕ into $\phi_1 \wedge \phi_2$, where ϕ_1 is in $\mathcal{T}_1$'s language and ϕ_2 is in $\mathcal{T}_2$'s language.
  So ϕ_1 and ϕ_2 have no common symbols, except variables.
- Purification step: replace term t by a fresh variable x
  $\phi \wedge F(\ldots, s[t], \ldots) \leadsto \phi \wedge F(\ldots, s[x], \ldots) \wedge x = t$,
- Purification is satisfiability preserving and terminating.
- Example:

$$f(x-1) - 1 = x, f(y) + 1 = y \leadsto$$
$$f(u_1) - 1 = x, f(y) + 1 = y, u_1 = x - 1 \leadsto$$
$$u_2 - 1 = x, f(y) + 1 = y, u_1 = x - 1, u_2 = f(u_1) \leadsto$$
$$u_2 - 1 = x, u_3 + 1 = y, u_1 = x - 1, u_2 = f(u_1), u_3 = f(y)$$

Purification

- Goal: convert a formula ϕ into $\phi_1 \wedge \phi_2$, where ϕ_1 is in $\mathcal{T}_1$'s language and ϕ_2 is in $\mathcal{T}_2$'s language.

  So ϕ_1 and ϕ_2 have no common symbols, except variables.
- Purification step: replace term t by a fresh variable x

  $\phi \wedge F(\ldots, s[t], \ldots) \rightsquigarrow \phi \wedge F(\ldots, s[x], \ldots) \wedge x = t,$
- Purification is satisfiability preserving and terminating.
- Example:

$$f(x - 1) - 1 = x, f(y) + 1 = y \rightsquigarrow$$

$$f(u_1) - 1 = x, f(y) + 1 = y, u_1 = x - 1 \rightsquigarrow$$

$$u_2 - 1 = x, f(y) + 1 = y, u_1 = x - 1, u_2 = f(u_1) \rightsquigarrow$$

$$u_2 - 1 = x, u_3 + 1 = y, u_1 = x - 1, u_2 = f(u_1), u_3 = f(y)$$

After Purification

$x = f(z), f(x) \neq f(y), 0 \leq x \leq 1, 0 \leq y \leq 1, z = y - 1$

After Purification

$$x = f(z), f(x) \neq f(y), 0 \leq x \leq 1, 0 \leq y \leq 1, z = y - 1$$

Red Model	*Blue Model*
$\lvert R \rvert = \{*_1, \ldots, *_6\}$	$\lvert B \rvert = \{\ldots, -1, 0, 1, \ldots\}$
$R(x) = *_1$	$B(x) = 0$
$R(y) = *_2$	$B(y) = 0$
$R(z) = *_3$	$B(z) = -1$
$R(f) = \{*_1 \mapsto *_4,$	
$*_2 \mapsto *_5,$	
$*_3 \mapsto *_1,$	
else $\mapsto *_6\}$	

Stably-Infinite Theories

- A theory is stably infinite if every satisfiable QFF is satisfiable in an infinite model.
- Example. Theories with only finite models are not stably infinite. $\mathcal{T}_2 = DC(\forall x, y, z.\, (x = y) \vee (x = z) \vee (y = z))$.
- The union of two consistent, disjoint, stably infinite theories is consistent.

Convexity

- A theory $\mathcal{T}$ is convex iff

  for all finite sets Γ of literals and

  for all non-empty disjunctions $\bigvee_{i \in I} x_i = y_i$ of variables,

  $\Gamma \models_{\mathcal{T}} \bigvee_{i \in I} x_i = y_i$ iff $\Gamma \models_{\mathcal{T}} x_i = y_i$ for some $i \in I$.
- Every convex theory $\mathcal{T}$ with non trivial models (i.e., $\models_T \exists x, y.\ x \neq y$) is stably infinite.
- All Horn theories are convex – this includes all (conditional) equational theories.
- Linear rational arithmetic is convex.

Convexity (cont.)

- Many theories are not convex:
  - Linear integer arithmetic.

$$y = 1, z = 2, 1 \leq x \leq 2 \models x = y \vee x = z$$

  - Nonlinear arithmetic.

$$x^2 = 1, y = 1, z = -1 \models x = y \vee x = z$$

  - Theory of Bit-vectors.
  - Theory of Arrays.

$$v_1 = \mathit{read}(\mathit{write}(a, i, v_2), j), v_3 = \mathit{read}(a, j) \models v_1 = v_2 \vee v_1 = v_3$$

Nelson-Oppen Combination

- Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be consistent, stably infinite theories over disjoint (countable) signatures. Assume satisfiability of conjunction of literals can decided in $O(T_1(n))$ and $O(T_2(n))$ time respectively. Then,
  1. The combined theory $\mathcal{T}$ is consistent and stably infinite.
  2. Satisfiability of quantifier free conjunction of literals in $\mathcal{T}$ can be decided in $O(2^{n^2} \times (T_1(n) + T_2(n))$.
  3. If $\mathcal{T}_1$ and $\mathcal{T}_2$ are convex, then so is $\mathcal{T}$ and satisfiability in $\mathcal{T}$ is in $O(n^3 \times (T_1(n) + T_2(n)))$.

Nelson-Oppen Combination Procedure

- The combination procedure:

Initial State: ϕ is a conjunction of literals over $\Sigma_1 \cup \Sigma_2$.

Purification: Preserving satisfiability transform ϕ into $\phi_1 \wedge \phi_2$, such that, $\phi_i \in \Sigma_i$.

Interaction: Guess a partition of $\mathcal{V}(\phi_1) \cap \mathcal{V}(\phi_2)$ into disjoint subsets. Express it as conjunction of literals ψ.

Example. The partition $\{x_1\}, \{x_2, x_3\}, \{x_4\}$ is represented as $x_1 \neq x_2, x_1 \neq x_4, x_2 \neq x_4, x_2 = x_3$.

Component Procedures : Use individual procedures to decide whether $\phi_i \wedge \psi$ is satisfiable.

Return: If both return yes, return yes. No, otherwise.

NO procedure: soundness

- Each step is satisfiability preserving.
- Say ϕ is satisfiable (in the combination).
  - Purification: $\phi_1 \wedge \phi_2$ is satisfiable.

NO procedure: soundness

- Each step is satisfiability preserving.
- Say ϕ is satisfiable (in the combination).
  - Purification: $\phi_1 \wedge \phi_2$ is satisfiable.
  - Iteration: for some partition ψ, $\phi_1 \wedge \phi_2 \wedge \psi$ is satisfiable.

NO procedure: soundness

- Each step is satisfiability preserving.
- Say ϕ is satisfiable (in the combination).
  - Purification: $\phi_1 \wedge \phi_2$ is satisfiable.
  - Iteration: for some partition ψ, $\phi_1 \wedge \phi_2 \wedge \psi$ is satisfiable.
  - Component procedures: $\phi_1 \wedge \psi$ and $\phi_2 \wedge \psi$ are both satisfiable in component theories.

NO procedure: soundness

- Each step is satisfiability preserving.
- Say ϕ is satisfiable (in the combination).
  - Purification: $\phi_1 \wedge \phi_2$ is satisfiable.
  - Iteration: for some partition ψ, $\phi_1 \wedge \phi_2 \wedge \psi$ is satisfiable.
  - Component procedures: $\phi_1 \wedge \psi$ and $\phi_2 \wedge \psi$ are both satisfiable in component theories.
  - Therefore, if the procedure return unsatisfiable, then ϕ is unsatisfiable.

NO procedure: correctness

- Suppose the procedure returns satisfiable.
  - Let ψ be the partition and A and B be models of $\mathcal{T}_1 \wedge \phi_1 \wedge \psi$ and $\mathcal{T}_2 \wedge \phi_2 \wedge \psi$.

NO procedure: correctness

- Suppose the procedure returns satisfiable.
  - Let ψ be the partition and A and B be models of $\mathcal{T}_1 \wedge \phi_1 \wedge \psi$ and $\mathcal{T}_2 \wedge \phi_2 \wedge \psi$.
  - The component theories are stably infinite. So, assume the models are infinite (of same cardinality).

NO procedure: correctness

- Suppose the procedure returns satisfiable.
  - Let ψ be the partition and A and B be models of $\mathcal{T}_1 \wedge \phi_1 \wedge \psi$ and $\mathcal{T}_2 \wedge \phi_2 \wedge \psi$.
  - The component theories are stably infinite. So, assume the models are infinite (of same cardinality).
  - Let h be a bijection between $|A|$ and $|B|$ such that $h(A(x)) = B(x)$ for each shared variable.

NO procedure: correctness

- Suppose the procedure returns satisfiable.
  - Let ψ be the partition and A and B be models of $\mathcal{T}_1 \wedge \phi_1 \wedge \psi$ and $\mathcal{T}_2 \wedge \phi_2 \wedge \psi$.
  - The component theories are stably infinite. So, assume the models are infinite (of same cardinality).
  - Let h be a bijection between $|A|$ and $|B|$ such that $h(A(x)) = B(x)$ for each shared variable.
  - Extend B to $\bar{B}$ by interpretations of symbols in Σ_1: $\bar{B}(f)(b_1, \ldots, b_n) = h(A(f)(h^{-1}(b_1), \ldots, h^{-1}(b_n)))$

NO procedure: correctness

- Suppose the procedure returns satisfiable.
  - Let ψ be the partition and A and B be models of $\mathcal{T}_1 \wedge \phi_1 \wedge \psi$ and $\mathcal{T}_2 \wedge \phi_2 \wedge \psi$.
  - The component theories are stably infinite. So, assume the models are infinite (of same cardinality).
  - Let h be a bijection between $|A|$ and $|B|$ such that $h(A(x)) = B(x)$ for each shared variable.
  - Extend B to $\bar{B}$ by interpretations of symbols in Σ_1: $\bar{B}(f)(b_1, \ldots, b_n) = h(A(f)(h^{-1}(b_1), \ldots, h^{-1}(b_n)))$
  - $\bar{B}$ is a model of: $\mathcal{T}_1 \wedge \phi_1 \wedge \mathcal{T}_2 \wedge \phi_2 \wedge \psi$

NO deterministic procedure

- Instead of guessing, we can deduce the equalities to be shared.

  Purification: no changes.

  Interaction: Deduce an equality $x = y$:

  $$\mathcal{T}_1 \vdash (\phi_1 \Rightarrow x = y)$$

  Update $\phi_2 := \phi_2 \wedge x = y$. And vice-versa. Repeat until no further changes.

  Component Procedures : Use individual procedures to decide whether ϕ_i is satisfiable.

- Remark: $\mathcal{T}_i \vdash (\phi_i \Rightarrow x = y)$ iff $\phi_i \wedge x \neq y$ is not satisfiable in $\mathcal{T}_i$.

NO deterministic procedure: correctness

- Assume the theories are convex.
  - Suppose ϕ_i is satisfiable.

NO deterministic procedure: correctness

- Assume the theories are convex.
  - Suppose ϕ_i is satisfiable.
  - Let E be the set of equalities $x_j = x_k$ ($j \neq k$) such that, $\mathcal{T}_i \not\vdash \phi_i \Rightarrow x_j = x_k$.

NO deterministic procedure: correctness

- Assume the theories are convex.
  - Suppose ϕ_i is satisfiable.
  - Let E be the set of equalities $x_j = x_k$ ($j \neq k$) such that, $\mathcal{T}_i \not\models \phi_i \Rightarrow x_j = x_k$.
  - By convexity, $\mathcal{T}_i \not\models \phi_i \Rightarrow \bigvee_E x_j = x_k$.

NO deterministic procedure: correctness

- Assume the theories are convex.
  - Suppose ϕ_i is satisfiable.
  - Let E be the set of equalities $x_j = x_k$ ($j \neq k$) such that, $\mathcal{T}_i \not\models \phi_i \Rightarrow x_j = x_k$.
  - By convexity, $\mathcal{T}_i \not\models \phi_i \Rightarrow \bigvee_E x_j = x_k$.
  - $\phi_i \wedge \bigwedge_E x_j \neq x_k$ is satisfiable.

NO deterministic procedure: correctness

- Assume the theories are convex.
  - Suppose ϕ_i is satisfiable.
  - Let E be the set of equalities $x_j = x_k$ $(j \neq k)$ such that, $\mathcal{T}_i \not\models \phi_i \Rightarrow x_j = x_k$.
  - By convexity, $\mathcal{T}_i \not\models \phi_i \Rightarrow \bigvee_E x_j = x_k$.
  - $\phi_i \wedge \bigwedge_E x_j \neq x_k$ is satisfiable.
  - The proof now is identical to the nondeterministic case.

NO deterministic procedure: correctness

- Assume the theories are convex.
  - Suppose ϕ_i is satisfiable.
  - Let E be the set of equalities $x_j = x_k$ ($j \neq k$) such that, $\mathcal{T}_i \not\models \phi_i \Rightarrow x_j = x_k$.
  - By convexity, $\mathcal{T}_i \not\models \phi_i \Rightarrow \bigvee_E x_j = x_k$.
  - $\phi_i \wedge \bigwedge_E x_j \neq x_k$ is satisfiable.
  - The proof now is identical to the nondeterministic case.
  - Sharing equalities is sufficient, because a theory $\mathcal{T}_1$ can assume that $x^B \neq y^B$ whenever $x = y$ is not implied by $\mathcal{T}_2$ and vice versa.

NO procedure: example

$x + 2 = y \wedge f(\textit{read}(\textit{write}(a, x, 3), y - 2)) \neq f(y - x + 1)$

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$

Purifying

NO procedure: example

$$f(\textit{read}(\textit{write}(a, x, 3), y - 2)) \neq f(y - x + 1)$$

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
	$x + 2 = y$	

Purifying

NO procedure: example

$$f(\textit{read}(\textit{write}(a, x, u_1), y - 2)) \neq f(y - x + 1)$$

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
	$x + 2 = y$ $u_1 = 3$	

Purifying

NO procedure: example

$f(\textit{read}(\textit{write}(a, x, u_1), u_2)) \neq f(y - x + 1)$

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
	$x + 2 = y$ $u_1 = 3$ $u_2 = y - 2$	

Purifying

NO procedure: example

$f(u_3) \neq f(y - x + 1)$

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
	$x + 2 = y$ $u_1 = 3$ $u_2 = y - 2$	$u_3 =$ *read*$($*write*$(a, x, u_1), u_2)$

Purifying

NO procedure: example

$f(u_3) \neq f(u_4)$

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
	$x + 2 = y$ $u_1 = 3$ $u_2 = y - 2$ $u_4 = y - x + 1$	$u_3 =$ $\textit{read}(\textit{write}(a, x, u_1), u_2)$

Purifying

NO procedure: example

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
$f(u_3) \neq f(u_4)$	$x + 2 = y$ $u_1 = 3$ $u_2 = y - 2$ $u_4 = y - x + 1$	$u_3 =$ $\mathit{read}(\mathit{write}(a, x, u_1), u_2)$

Solving $\mathcal{T}_A$

NO procedure: example

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
$f(u_3) \neq f(u_4)$	$y = x + 2$ $u_1 = 3$ $u_2 = x$ $u_4 = 3$	$u_3 =$ $\textit{read}(\textit{write}(a, x, u_1), u_2)$

Propagating $u_2 = x$

NO procedure: example

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
$f(u_3) \neq f(u_4)$	$y = x + 2$	$u_3 =$
$u_2 = x$	$u_1 = 3$	*read*$($*write*$(a, x, u_1), u_2)$
	$u_2 = x$	$u_2 = x$
	$u_4 = 3$	

Solving $\mathcal{T}_{Ar}$

NO procedure: example

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
$f(u_3) \neq f(u_4)$	$y = x + 2$	$u_3 = u_1$
$u_2 = x$	$u_1 = 3$	$u_2 = x$
	$u_2 = x$	
	$u_4 = 3$	

Propagating $u_3 = u_1$

NO procedure: example

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
$f(u_3) \neq f(u_4)$	$y = x + 2$	$u_3 = u_1$
$u_2 = x$	$u_1 = 3$	$u_2 = x$
$u_3 = u_1$	$u_2 = x$	
	$u_4 = 3$	
	$u_3 = u_1$	

Propagating $u_1 = u_4$

NO procedure: example

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
$f(u_3) \neq f(u_4)$	$y = x + 2$	$u_3 = u_1$
$u_2 = x$	$u_1 = 3$	$u_2 = x$
$u_3 = u_1$	$u_2 = x$	
$u_4 = u_1$	$u_4 = 3$	
	$u_3 = u_1$	

Congruence $u_3 = u_1 \wedge u_4 = u_1 \Rightarrow f(u_3) = f(u_4)$

NO procedure: example

$\mathcal{T}_E$	$\mathcal{T}_A$	$\mathcal{T}_{Ar}$
$f(u_3) \neq f(u_4)$	$y = x + 2$	$u_3 = u_1$
$u_2 = x$	$u_1 = 3$	$u_2 = x$
$u_3 = u_1$	$u_2 = x$	
$u_4 = u_1$	$u_4 = 3$	
$f(u_3) = f(u_4)$	$u_3 = u_1$	

Unsatisfiable!

NO deterministic procedure

- Deterministic procedure does not work for non convex theories.
- Example (integer arithmetic):

$$0 \leq x, y, z \leq 1, f(x) \neq f(y), f(x) \neq f(z), f(y) \neq f(z)$$

- (Expensive) solution: deduce disjunctions of equalities.

Combining theories in practice

- Propagate all implied equalities.
  - Deterministic Nelson-Oppen.
  - Complete only for convex theories.
  - It may be expensive for some theories.
- Delayed Theory Combination.
  - Nondeterministic Nelson-Oppen.
  - Create set of interface equalities ($x = y$) between shared variables.
  - Use SAT solver to guess the partition.
  - Disadvantage: the number of additional equality literals is quadratic in the number of shared variables.

Combining theories in practice (cont.)

- Common to these methods is that they are pessimistic about which equalities are propagated.
- Model-based Theory Combination
  - Optimistic approach.
  - Use a candidate model M_i for one of the theories $\mathcal{T}_i$ and propagate all equalities implied by the candidate model, hedging that other theories will agree.

$$\textbf{if } M_i \models \mathcal{T}_i \cup \Gamma_i \cup \{u = v\} \textbf{ then } \text{propagate } u = v \ .$$

  - If not, use backtracking to fix the model.
  - It is cheaper to enumerate equalities that are true in a particular model than the equalities implied by all models.

Model based theory combination: Example

$$x = f(y - 1), f(x) \neq f(y), 0 \leq x \leq 1, 0 \leq y \leq 1$$

Purifying

Model based theory combination: Example

$$x = f(z), f(x) \neq f(y), 0 \leq x \leq 1, 0 \leq y \leq 1, z = y - 1$$

Model based theory combination: Example

$\mathcal{T}_E$			$\mathcal{T}_A$	
Literals	*Eq. Classes*	*Model*	*Literals*	*Model*
$x = f(z)$	$\{x, f(z)\}$	$E(x) = *_1$	$0 \leq x \leq 1$	$A(x) = 0$
$f(x) \neq f(y)$	$\{y\}$	$E(y) = *_2$	$0 \leq y \leq 1$	$A(y) = 0$
	$\{z\}$	$E(z) = *_3$	$z = y - 1$	$A(z) = -1$
	$\{f(x)\}$	$E(f) = \{*_1 \mapsto *_4,$		
	$\{f(y)\}$	$*_2 \mapsto *_5,$		
		$*_3 \mapsto *_1,$		
		else $\mapsto *_6\}$		

Assume x = y

Model based theory combination: Example

$\mathcal{T}_E$			$\mathcal{T}_A$	
Literals	*Eq. Classes*	*Model*	*Literals*	*Model*
$x = f(z)$	$\{x, y, f(z)\}$	$E(x) = *_1$	$0 \leq x \leq 1$	$A(x) = 0$
$f(x) \neq f(y)$	$\{z\}$	$E(y) = *_1$	$0 \leq y \leq 1$	$A(y) = 0$
$x = y$	$\{f(x), f(y)\}$	$E(z) = *_2$	$z = y - 1$	$A(z) = -1$
		$E(f) = \{*_1 \mapsto *_3,$	$x = y$	
		$*_2 \mapsto *_1,$		
		else $\mapsto *_4\}$		

Unsatisfiable

Model based theory combination: Example

$\mathcal{T}_E$			$\mathcal{T}_A$	
Literals	*Eq. Classes*	*Model*	*Literals*	*Model*
$x = f(z)$	$\{x, f(z)\}$	$E(x) = *_1$	$0 \leq x \leq 1$	$A(x) = 0$
$f(x) \neq f(y)$	$\{y\}$	$E(y) = *_2$	$0 \leq y \leq 1$	$A(y) = 0$
$x \neq y$	$\{z\}$	$E(z) = *_3$	$z = y - 1$	$A(z) = -1$
	$\{f(x)\}$	$E(f) = \{*_1 \mapsto *_4,$	$x \neq y$	
	$\{f(y)\}$	$*_2 \mapsto *_5,$		
		$*_3 \mapsto *_1,$		
		else $\mapsto *_6\}$		

Backtrack, and assert $x \neq y$.

$\mathcal{T}_A$ model need to be fixed.

Model based theory combination: Example

$\mathcal{T}_E$			$\mathcal{T}_A$	
Literals	*Eq. Classes*	*Model*	*Literals*	*Model*
$x = f(z)$	$\{x, f(z)\}$	$E(x) = *_1$	$0 \leq x \leq 1$	$A(x) = 0$
$f(x) \neq f(y)$	$\{y\}$	$E(y) = *_2$	$0 \leq y \leq 1$	$A(y) = 1$
$x \neq y$	$\{z\}$	$E(z) = *_3$	$z = y - 1$	$A(z) = 0$
	$\{f(x)\}$	$E(f) = \{*_1 \mapsto *_4,$	$x \neq y$	
	$\{f(y)\}$	$*_2 \mapsto *_5,$		
		$*_3 \mapsto *_1,$		
		else $\mapsto *_6\}$		

Assume x = z

Model based theory combination: Example

$\mathcal{T}_E$			$\mathcal{T}_A$	
Literals	*Eq. Classes*	*Model*	*Literals*	*Model*
$x = f(z)$	$\{x, z,$	$E(x) = *_1$	$0 \le x \le 1$	$A(x) = 0$
$f(x) \neq f(y)$	$f(x), f(z)\}$	$E(y) = *_2$	$0 \le y \le 1$	$A(y) = 1$
$x \neq y$	$\{y\}$	$E(z) = *_1$	$z = y - 1$	$A(z) = 0$
$x = z$	$\{f(y)\}$	$E(f) = \{*_1 \mapsto *_1,$	$x \neq y$	
		$*_2 \mapsto *_3,$	$x = z$	
		else $\mapsto *_4\}$		

Satisfiable

Model based theory combination: Example

$\mathcal{T}_E$			$\mathcal{T}_A$	
Literals	*Eq. Classes*	*Model*	*Literals*	*Model*
$x = f(z)$	$\{x, z,$	$E(x) = *_1$	$0 \leq x \leq 1$	$A(x) = 0$
$f(x) \neq f(y)$	$f(x), f(z)\}$	$E(y) = *_2$	$0 \leq y \leq 1$	$A(y) = 1$
$x \neq y$	$\{y\}$	$E(z) = *_1$	$z = y - 1$	$A(z) = 0$
$x = z$	$\{f(y)\}$	$E(f) = \{*_1 \mapsto *_1,$	$x \neq y$	
		$*_2 \mapsto *_3,$	$x = z$	
		else $\mapsto *_4\}$		

Let h be the bijection between $\lvert E \rvert$ and $\lvert A \rvert$.

$$h = \{*_1 \mapsto 0, *_2 \mapsto 1, *_3 \mapsto -1, *_4 \mapsto 2, \ldots\}$$

Model based theory combination: Example

$\mathcal{T}_E$		$\mathcal{T}_A$	
Literals	*Model*	*Literals*	*Model*
$x = f(z)$	$E(x) = *_1$	$0 \leq x \leq 1$	$A(x) = 0$
$f(x) \neq f(y)$	$E(y) = *_2$	$0 \leq y \leq 1$	$A(y) = 1$
$x \neq y$	$E(z) = *_1$	$z = y - 1$	$A(z) = 0$
$x = z$	$E(f) = \{*_1 \mapsto *_1,$	$x \neq y$	$A(f) = \{0 \mapsto 0$
	$*_2 \mapsto *_3,$	$x = z$	$1 \mapsto -1$
	else $\mapsto *_4\}$		*else* $\mapsto 2\}$

Extending A using h.

$$h = \{*_1 \mapsto 0, *_2 \mapsto 1, *_3 \mapsto -1, *_4 \mapsto 2, \ldots\}$$

Model mutation

- Sometimes $M(x) = M(y)$ by accident.

$$\bigwedge_{i=1}^{N} f(x_i) \geq 0 \;\wedge\; x_i \geq 0$$

- Model mutation: diversify the current model.

Roadmap

- Logic Background
- Modern SAT Solvers
- DPLL with Theory Solvers
- Theory Combination
- Equality
- Arithmetic
- Applications

Theory of Equality: Axioms

Reflexivity $x = x$

Symmetry $x = y \Rightarrow y = x$

Transitivity $x = y, y = z \Rightarrow x = z$

Congruence

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

Example

$$f(f(a)) = a,\ b = f(a),\ \neg f(f(f(a))) = b$$

Example

$$f(f(a)) = a,\ b = f(a),\ \neg f(f(f(a))) = b$$

congruence $\rightsquigarrow f(f(f(a))) = f(a)$

Example

$f(f(a)) = a,\ b = f(a),\ \neg f(f(f(a))) = b,$
$f(f(f(a))) = f(a)$

symmetry $\rightsquigarrow f(a) = b$

Example

$f(f(a)) = a,\ b = f(a),\ \neg f(f(f(a))) = b,$

$f(f(f(a))) = f(a),\ f(a) = b$

transitivity $\rightsquigarrow\ f(f(f(a))) = b$

Example

$$f(f(a)) = a,\ b = f(a),\ \neg f(f(f(a))) = b,$$
$$f(f(f(a))) = f(a),\ f(a) = b,\ f(f(f(a))) = b$$

unsatisfiable

Example

- A conjunction of equalities is trivially satisfiable.
- Example: $f(x) = y,\ x = y,\ g(x) = z,\ f(y) = f(z)$

Example

- A conjunction of equalities is trivially satisfiable.
- Example: $f(x) = y,\ x = y,\ g(x) = z,\ f(y) = f(z)$
- Model:
  - $|M| = \{*_1\}$
  - $M(x) = M(y) = M(z) = *_1$
  - $M(f)(*_1) = *_1$
  - $M(g)(*_1) = *_1$

Variable equality

- Assume the problem has not function symbols.
- Use union-find data structure to represent equalities.
- The state consists of a find structure F that maintains equivalence classes and a set of disequalities D.
- Initially, $F(x) = x$ for each variable x.
- $F^*(x)$ is the root of the equivalence class containing x:

$$F^*(x) = \begin{cases} x, & \text{if } F(x) = x \\ F^*(F(x)) & \text{otherwise} \end{cases}$$

- Let *sz*(F, x) denote the size of the equivalence class containing x.

Variable equality: union

- An equality $x = y$ is processed by merging distinct equivalence classes using the *union* operation:

$$\textit{union}(F, x, y) \quad = \quad \begin{cases} F[x' := y'], & \textit{sz}(F, x) < \textit{sz}(F, y) \\ F[y' := x'], & \text{otherwise} \end{cases}$$

where $x' \equiv F^*(x) \not\equiv F^*(y) \equiv y'$

- Optimization: path compression, update F when executing $F^*(x)$. $F[x := F^*(x)]$

Processing equalities

- The entire inference system consists of operations for adding equalities, disequalities, and dectecting unsatisfiability.

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

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

where $F^*(x) \not\equiv F^*(y)$

$F' = \textit{union}(F, x, y)$

Processing disequalities

$$\textit{addneq}(x \neq y, F, D) := \textbf{unsat}, \text{ if } F^*(x) \equiv F^*(y)$$

$$\textit{addneq}(x \neq y, F, D) := \langle F, D\rangle, \text{ if } F^*(x) = F^*(u), F^*(y) = F^*(v), \text{ for } u \neq v \in D \text{ or } v \neq u \in D$$

$$\textit{addneq}(x \neq y, F, D) := \langle F, D \cup \{x \neq y\}\rangle, \text{ otherwise}$$

Example

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

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

Example

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

$$F = \{x_1 \mapsto x_1, x_2 \mapsto x_2, x_3 \mapsto x_3, x_4 \mapsto x_4, x_5 \mapsto x_5\}$$

$$D = \{\}$$

Merge equivalence classes of x_1 and x_2.

Example

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

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

Example

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

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

Merge equivalence classes of x_1 and x_3.

Example

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

$$F = \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_4, x_5 \mapsto x_5\}$$

$$D = \{\}$$

Example

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

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

Skip equality

Example

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

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

Add disequality

Example

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

$$F = \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_4, x_5 \mapsto x_5\}$$

$$D = \{x_2 \neq x_4\}$$

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 equivalence 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

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

$$F = \{x_1 \mapsto x_1, x_2 \mapsto x_1, x_3 \mapsto x_1, x_4 \mapsto x_4, x_5 \mapsto x_4\}$$

$$D = \{x_2 \neq x_4\}$$

Model M:

$|M| = \{*_1, *_2\}$

$M(x_1), M(x_2), M(x_3) = *_1$

$M(x_4), M(x_5) = *_2$

Equality with offsets

- Many terms are equal modulo a numeric offset (e.g., $x = y + 1$).
- If these are placed in separate equivalence classes, then the equality reasoning on these terms must invoke the arithmetic module.
- We can modify the *find* data structure so that $F(x)$ returns $y + c$, and similarly $F^*(x)$.
- Example: $x_1 \neq x_2 + c$ if $F^*(x_1) = y + c_1$ and $F^*(x_2) = y + c_2$, where $c \neq c_1 - c_2$.

Retracting assertions

- Checkpointing the find data structure can be expensive.
- A disequality can be retracted by just deleting it from D.
- Retracting equality assertions is more difficult, the history of the merge operations have to be maintained.
- On retraction, the find values have to be restored.

Congruence Closure

- Equivalence is extended to *congruence* with the rule that for each n-ary function f, $f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)$ if $s_i = t_i$ for each $1 \leq 1 \leq n$.
- New index: $\pi(t)$ is the set of parents of the equivalence class rooted by t (aka use-list).
- Example:

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

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

Congruence Closure (cont.)

- As with equivalence, the *find* roots $s' = F^*(s)$ and $t' = F^*(t)$ are merged. The use lists $\pi(s')$ and $\pi(t')$ are also merged.
- How to merge use-lists?
  1. Use-lists are circular lists:
     - Constant time merge and unmerge.
  2. Use-lists are vectors:
     - Linear time merge: copy $\pi(s')$ to $\pi(t')$.
     - Constant time unmerge: shrink the vector.
  3. Do not merge: to traverse the set of parents, traverse the equivalence class.
- Any pair p_1 in $\pi(s')$ and p_2 in $\pi(t')$ that are congruent in F is added to a queue of equalities to be merged.

Congruence Closure (cont.)

- Any pair p_1 in $\pi(s')$ and p_2 in $\pi(t')$ that are congruent in F is added to a queue of equalities to be merged.
  - Naïve solution: for each p_i of $\pi(s')$ traverse $\pi(t')$ looking for a congruence p_j.
  - Efficient solution: congruence table.
    - Hashtable of ground terms.
    - Hash of $f(t_1, \ldots, t_n)$ is based on f, $F^*(t_1), \ldots, F^*(t_n)$
    - $f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)$ if $F^*(s_1) = F^*(t_1), \ldots, F^*(s_n) = F^*(t_n)$
    - The operation $F[x' := y']$ affects the hashcode of $\pi(x')$, before executing it remove terms in $\pi(x')$ from the table, and reinsert them back after.
    - Detect new congruences during reinsertion.

Example

$$f(g(a)) = c, c \neq f(g(b)), a = b$$

$$\begin{aligned}
F &= \{a \mapsto a,\ b \mapsto b,\ c \mapsto c,\ g(a) \mapsto g(a),\ g(b) \mapsto g(b) \\
&\quad f(g(a)) \mapsto f(g(a)),\ f(g(b)) \mapsto f(g(b))\} \\
D &= \{\} \\
\pi(a) &= \{g(a)\} \\
\pi(b) &= \{g(b)\} \\
\pi(g(a)) &= \{f(g(a))\} \\
\pi(g(b)) &= \{f(g(b))\}
\end{aligned}$$

Example

$$f(g(a)) = c, c \neq f(g(b)), a = b$$

$$\begin{aligned}
F &= \{a \mapsto a,\ b \mapsto b,\ c \mapsto c,\ g(a) \mapsto g(a),\ g(b) \mapsto g(b) \\
&\quad f(g(a)) \mapsto f(g(a)),\ f(g(b)) \mapsto f(g(b))\} \\
D &= \{\} \\
\pi(a) &= \{g(a)\} \\
\pi(b) &= \{g(b)\} \\
\pi(g(a)) &= \{f(g(a))\} \\
\pi(g(b)) &= \{f(g(b))\}
\end{aligned}$$

Merge equivalence classes of $f(g(a))$ and c.

Example

$$f(g(a)) = c, c \neq f(g(b)), a = b$$

$$\begin{aligned}
F &= \{a \mapsto a,\ b \mapsto b,\ c \mapsto c,\ g(a) \mapsto g(a),\ g(b) \mapsto g(b) \\
&\quad f(g(a)) \mapsto c,\ f(g(b)) \mapsto f(g(b))\} \\
D &= \{\} \\
\pi(a) &= \{g(a)\} \\
\pi(b) &= \{g(b)\} \\
\pi(g(a)) &= \{f(g(a))\} \\
\pi(g(b)) &= \{f(g(b))\}
\end{aligned}$$

Example

$$f(g(a)) = c, c \neq f(g(b)), a = b$$

$$\begin{aligned}
F &= \{a \mapsto a,\ b \mapsto b,\ c \mapsto c,\ g(a) \mapsto g(a),\ g(b) \mapsto g(b) \\
&\quad f(g(a)) \mapsto c,\ f(g(b)) \mapsto f(g(b))\} \\
D &= \{\} \\
\pi(a) &= \{g(a)\} \\
\pi(b) &= \{g(b)\} \\
\pi(g(a)) &= \{f(g(a))\} \\
\pi(g(b)) &= \{f(g(b))\}
\end{aligned}$$

Add disequality

Example

$$f(g(a)) = c, c \neq f(g(b)), a = b$$

$$\begin{aligned}
F &= \{a \mapsto a,\ b \mapsto b,\ c \mapsto c,\ g(a) \mapsto g(a),\ g(b) \mapsto g(b) \\
&\quad f(g(a)) \mapsto c,\ f(g(b)) \mapsto f(g(b))\} \\
D &= \{c \neq f(g(b))\} \\
\pi(a) &= \{g(a)\} \\
\pi(b) &= \{g(b)\} \\
\pi(g(a)) &= \{f(g(a))\} \\
\pi(g(b)) &= \{f(g(b))\}
\end{aligned}$$

Example

$$f(g(a)) = c, c \neq f(g(b)), a = b$$

$$\begin{aligned}
F &= \{a \mapsto a,\ b \mapsto b,\ c \mapsto c,\ g(a) \mapsto g(a),\ g(b) \mapsto g(b) \\
&\quad f(g(a)) \mapsto c,\ f(g(b)) \mapsto f(g(b))\} \\
D &= \{c \neq f(g(b))\} \\
\pi(a) &= \{g(a)\} \\
\pi(b) &= \{g(b)\} \\
\pi(g(a)) &= \{f(g(a))\} \\
\pi(g(b)) &= \{f(g(b))\}
\end{aligned}$$

Merge equivalence classes of a and b.

Example

$$f(g(a)) = c, c \neq f(g(b)), a = b, g(a) = g(b)$$

$$\begin{aligned}
F &= \{a \mapsto a,\ b \mapsto a,\ c \mapsto c,\ g(a) \mapsto g(a),\ g(b) \mapsto g(b) \\
&\quad f(g(a)) \mapsto c,\ f(g(b)) \mapsto f(g(b))\} \\
D &= \{c \neq f(g(b))\} \\
\pi(a) &= \{g(a), g(b)\} \\
\pi(b) &= \{g(b)\} \\
\pi(g(a)) &= \{f(g(a))\} \\
\pi(g(b)) &= \{f(g(b))\}
\end{aligned}$$

Example

$$f(g(a)) = c, c \neq f(g(b)), a = b, g(a) = g(b)$$

$$\begin{aligned}
F &= \{a \mapsto a,\ b \mapsto a,\ c \mapsto c,\ g(a) \mapsto g(a),\ g(b) \mapsto g(b) \\
&\quad f(g(a)) \mapsto c,\ f(g(b)) \mapsto f(g(b))\} \\
D &= \{c \neq f(g(b))\} \\
\pi(a) &= \{g(a), g(b)\} \\
\pi(b) &= \{g(b)\} \\
\pi(g(a)) &= \{f(g(a))\} \\
\pi(g(b)) &= \{f(g(b))\}
\end{aligned}$$

Merge equivalence classes of $g(a)$ and $g(b)$.

Example

$$f(g(a)) = c, c \neq f(g(b)), a = b, g(a) = g(b), f(g(a)) = f(g(b))$$

$$\begin{aligned}
F &= \{a \mapsto a,\ b \mapsto a,\ c \mapsto c,\ g(a) \mapsto g(b),\ g(b) \mapsto g(b) \\
&\quad f(g(a)) \mapsto c,\ f(g(b)) \mapsto f(g(b))\} \\
D &= \{c \neq f(g(b))\} \\
\pi(a) &= \{g(a), g(b)\} \\
\pi(b) &= \{g(b)\} \\
\pi(g(a)) &= \{f(g(a))\} \\
\pi(g(b)) &= \{f(g(b)), f(g(a))\}
\end{aligned}$$

Example

$$f(g(a)) = c, c \neq f(g(b)), a = b, g(a) = g(b), f(g(a)) = f(g(b))$$

$$\begin{aligned}
F &= \{a \mapsto a,\ b \mapsto a,\ c \mapsto c,\ g(a) \mapsto g(b),\ g(b) \mapsto g(b) \\
&\quad f(g(a)) \mapsto c,\ f(g(b)) \mapsto f(g(b))\} \\
D &= \{c \neq f(g(b))\} \\
\pi(a) &= \{g(a), g(b)\} \\
\pi(b) &= \{g(b)\} \\
\pi(g(a)) &= \{f(g(a))\} \\
\pi(g(b)) &= \{f(g(b)), f(g(a))\}
\end{aligned}$$

Merge equivalence classes of $f(g(a))$ and $f(g(b))$ $\rightsquigarrow$ **unsat**.

Example: Satisfiable Version

$$f(g(a)) = c, a \neq f(g(b)), a = b, g(a) = g(b), f(g(a)) = f(g(b))$$

$$\begin{aligned} F &= \{a \mapsto a,\ b \mapsto a,\ c \mapsto c,\ g(a) \mapsto g(b),\ g(b) \mapsto g(b) \\ &\quad f(g(a)) \mapsto c,\ f(g(b)) \mapsto c\} \\ D &= \{a \neq f(g(b))\} \end{aligned}$$

Example: Satisfiable Version

$$f(g(a)) = c, a \neq f(g(b)), a = b, g(a) = g(b), f(g(a)) = f(g(b))$$

$$\begin{aligned} F &= \{a \mapsto a,\ b \mapsto a,\ c \mapsto c,\ g(a) \mapsto g(b),\ g(b) \mapsto g(b) \\ &\qquad f(g(a)) \mapsto c,\ f(g(b)) \mapsto c\} \\ D &= \{a \neq f(g(b))\} \end{aligned}$$

Model: $|M| = \{*_1, *_2, *_3\}$ One value for each eq. class root.

$M(a) = M(b) = *_1$

$M(c) = *_2$

$M(g) = \{*_1 \mapsto *_3, \textit{else} \mapsto *_?\}$ $*_?$ can be any value.

$M(f) = \{*_3 \mapsto *_2, \textit{else} \mapsto *_?\}$

Example: Satisfiable Version

$$f(g(a)) = c, a \neq f(g(b)), a = b, g(a) = g(b), f(g(a)) = f(g(b))$$

$$\begin{aligned} F &= \{a \mapsto a,\ b \mapsto a,\ c \mapsto c,\ g(a) \mapsto g(b),\ g(b) \mapsto g(b) \\ &\qquad f(g(a)) \mapsto c,\ f(g(b)) \mapsto c\} \\ D &= \{a \neq f(g(b))\} \end{aligned}$$

Model: $|M| = \{*_1, *_2, *_3\}$ One value for each eq. class root.

$M(a) = M(b) = *_1$

$M(c) = *_2$

$M(g) = \{*_1 \mapsto *_3, \textit{else} \mapsto *_?\}$ $*_?$ can be any value.

$M(f) = \{*_3 \mapsto *_2, \textit{else} \mapsto *_?\}$

Equality: T-Justifications

- A T-Justification for F is a set of literals S such that $S \models_T F$.
- S is a non-redudant if there is no $S' \subset S$ such that $S' \models_T F$.
- Non-redundant T-Justifications for variable equalities is easy: shortest-path between two variables.
- With uninterpreted functions the problem is more difficult:
- Example:

$$
\begin{array}{ccccccc}
f_1(x_1) & = & x_1 & = & x_2 & = & f_1(x_{n+1}), \\
\ldots, \\
f_n(x_1) & = & x_n & = & x_{n+1} & = & f_n(x_{n+1}),
\end{array}
$$

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

Roadmap

- Logic Background
- Modern SAT Solvers
- DPLL with Theory Solvers
- Theory Combination
- Equality
- Arithmetic
- Applications

Linear Arithmetic

- Algorithms:
  - Graph based for difference logic ($x \leq y - k$).
  - Fourier-Motzkin elimination.

    $$t_1 \leq ax, \;\; bx \leq t_2 \;\; \Rightarrow \;\; bt_1 \leq at_2$$
  - Standard Simplex.
- Standard Simplex based solvers:
  - Standard Form: $Ax = b$ and $x \geq 0$.
  - Incremental: add/remove equations (i.e., rows).
  - Slow backtracking.
  - No theory propagation.

Fast Linear Arithmetic

- Simplex General Form.
- Algorithm based on the Dual Simplex.
- Non-redundant T-Justifications.
- Efficient Backtracking.
- Efficient T-Propagate.
- Support for strict inequalities ($t > 0$).
- Presimplification step.
- Integer problems: Gomory cuts, Branch & Bound, GCD test.

General Form

- General Form: $Ax = 0$ and $l_j \leq x_j \leq u_j$
- Example:

$$x \geq 0, (x + y \leq 2 \vee x + 2y \geq 6), (x + y = 2 \vee x + 2y > 4)$$

$$\rightsquigarrow$$

$$s_1 = x + y, s_2 = x + 2y,$$

$$x \geq 0, (s_1 \leq 2 \vee s_2 \geq 6), (s_1 = 2 \vee s_2 > 4)$$

- Only bounds (e.g., $s_1 \leq 2$) are asserted during the search.
- Unconstrained variables can be eliminated before the beginning of the search.

Model + Equations + Bounds

- An assignment (model) is a mapping from variables to values.
- We maintain an assignment that satisfies all equations and bounds.
- The assignment of non dependent variables implies the assignment of dependent variables.
- Equations + Bounds can be used to derive new bounds.
- Example: $x = y - z,\ y \leq 2,\ z \geq 3 \leadsto x \leq -1$.
- The new bound may be inconsistent with the already known bounds.
- Example: $x \leq -1,\ x \geq 0$.

Strict Inequalities

- The method described only handles non-strict inequalities (e.g., $x \leq 2$).
- For integer problems, strict inequalities can be converted into non-strict inequalities. $x < 1 \rightsquigarrow x \leq 0$.
- For rational/real problems, strict inequalities can be converted into non-strict inequalities using a small δ. $x < 1 \rightsquigarrow x \leq 1 - \delta$.
- We do not compute a δ, we treat it symbolically.
- δ is an infinitesimal parameter: $(c, k) = c + k\delta$

Example

- Initial state

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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 $s \geq 1$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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 $s \geq 1$ assignment does not satisfy new bound.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $s \geq 1$ pivot s and x (s is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $s \geq 1$ pivot s and x (s is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $s \geq 1$ pivot s and x (s is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $s \geq 1$ update assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $s \geq 1$ update dependent variables assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $x \geq 0$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $x \geq 0$ assignment satisfies new bound.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Case split $\neg y \leq 1$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Case split $\neg y \leq 1$ assignment does not satisfies new bound.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Case split $\neg y \leq 1$ update assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Case split $\neg y \leq 1$ update dependent variables assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

Model			Equations			Bounds		
$M(x)$	$=$	$-\delta$	x	$=$	$s - y$	s	$\geq$	1
$M(y)$	$=$	$1 + \delta$	u	$=$	$s + y$	x	$\geq$	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 (x is a dependent variables).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

Model			Equations			Bounds		
$M(x)$	$=$	$-\delta$	x	$=$	$s - y$	s	$\geq$	1
$M(y)$	$=$	$1 + \delta$	u	$=$	$s + y$	x	$\geq$	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 (x is a dependent variables).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation pivot x and s (x is a dependent variables).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation   update assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation update dependent variables assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Theory propagation $x \geq 0, y > 1 \rightsquigarrow u > 2$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Theory propagation $u > 2 \rightsquigarrow \neg u \leq -1$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1)$$

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

Example

- Boolean propagation $\neg y \leq 1 \rightsquigarrow v \geq 2$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Theory propagation $v \geq 2 \rightsquigarrow \neg v \leq -2$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Conflict empty clause

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Backtracking

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $y \leq 1$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $y \leq 1$ assignment does not satisfy new bound.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $y \leq 1$ update assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Asserting $y \leq 1$ update dependent variables assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Theory propagation $s \geq 1, y \leq 1 \rightsquigarrow v \geq -1$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Theory propagation $v \geq -1 \leadsto \neg v \leq -2$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Boolean propagation $\neg v \le -2 \rightsquigarrow v \ge 0$

$$s \ge 1, x \ge 0$$

$$(y \le 1 \lor v \ge 2), (v \le -2 \lor v \ge 0), (v \le -2 \lor u \le -1)$$

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

Example

- Bound violation assignment does not satisfy new bound.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation pivot u and s (u is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation pivot u and s (u is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation pivot u and s (u is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation update assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation update dependent variables assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Boolean propagation $\neg v \leq -2 \rightsquigarrow u \leq -1$

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation assignment does not satisfy new bound.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation pivot u and y (u is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation pivot u and y (u is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation pivot u and y (u is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation update assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violation   update dependent variables assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violations

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violations pivot s and v (s is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violations pivot s and v (s is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violations pivot s and v (s is a dependent variable).

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violations update assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Bound violations update dependent variables assignment.

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Example

- Found satisfying assignment

$$s \geq 1, x \geq 0$$

$$(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)$$

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

Opportunistic equality propagation

- Efficient (and incomplete) methods for propagating equalities.
- Notation
  - A variable x_i is fixed iff $l_i = u_i$.
  - A linear polynomial $\sum_{x_j \in \mathcal{V}} a_{ij} x_j$ is fixed iff x_j is fixed or $a_{ij} = 0$.
  - Given a linear polynomial $P = \sum_{x_j \in \mathcal{V}} a_{ij} x_j$, and a model M:
    $M(P)$ denotes $\sum_{x_j \in \mathcal{V}} a_{ij} M(x_j)$.

Opportunistic equality propagation

- Equality propagation in arithmetic:

FixedEq

$$l_i \le x_i \le u_i, \;\; l_j \le x_j \le u_j \Longrightarrow \;\; x_i = x_j \;\; \textbf{if} \;\; l_i = u_i = l_j = u_j$$

EqRow

$$x_i = x_j + P \qquad \Longrightarrow \;\; x_i = x_j \;\; \textbf{if} \;\; P \text{ is fixed, and } M(P) = 0$$

EqOffsetRows

$$\begin{array}{l} x_i = x_k + P_1 \\ x_j = x_k + P_2 \end{array} \qquad \Longrightarrow \;\; x_i = x_j \;\; \textbf{if} \;\; \left\{ \begin{array}{l} P_1 \text{ and } P_2 \text{ are fixed, and} \\ M(P_1) = M(P_2) \end{array} \right.$$

EqRows

$$\begin{array}{l} x_i = P + P_1 \\ x_j = P + P_2 \end{array} \qquad \Longrightarrow \;\; x_i = x_j \;\; \textbf{if} \;\; \left\{ \begin{array}{l} P_1 \text{ and } P_2 \text{ are fixed, and} \\ M(P_1) = M(P_2) \end{array} \right.$$

Opportunistic theory/equality propagation

- These rules can miss some implied equalities.
- Example: $z = w$ is detected, but $x = y$ is not because w is not a fixed variable.

$$
\begin{aligned}
x &= y + w + s \\
z &= w + s \\
0 &\leq z \\
& w \leq 0 \\
0 &\leq s \leq 0
\end{aligned}
$$

- Remark: bound propagation can be used imply the bound $0 \leq w$, making w a fixed variable.

Linear Integer Arithmetic

- GCD test
- Gomory Cuts
- Branch and Bound

Beyond Linear Arithmetic

- Gröbner Basis
- Cylindric Algebraic Decomposition