On Designing and Implementing Satisfiability Modulo Theory (SMT) Solvers
Summer School 2009, Nancy Verification Technology, Systems and Applications

Leonardo de Moura
Microsoft Research
Many approaches

- Graph-based for difference logic: $a - b \leq 3$
- Fourier-Motzkin elimination:

  \[ t_1 \leq ax, \ bx \leq t_2 \implies bt_1 \leq at_2 \]

- Standard Simplex
- General Form Simplex
Difference Logic: \( a - b \leq 5 \)

Very useful in practice!

Most arithmetical constraints in software verification/analysis are in this fragment.

\[
x := x + 1 \\
x_1 = x_0 + 1 \\
x_1 - x_0 \leq 1, \ x_0 - x_1 \leq -1
\]
**Job shop scheduling**

<table>
<thead>
<tr>
<th>$d_{i,j}$</th>
<th>Machine 1</th>
<th>Machine 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Job 2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Job 3</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

$\text{max} = 8$

**Solution**

$t_{1,1} = 5$, $t_{1,2} = 7$, $t_{2,1} = 2$, $t_{2,2} = 6$, $t_{3,1} = 0$, $t_{3,2} = 3$

**Encoding**

\[
(t_{1,1} \geq 0) \land (t_{1,2} \geq t_{1,1} + 2) \land (t_{1,2} + 1 \leq 8) \land \\
(t_{2,1} \geq 0) \land (t_{2,2} \geq t_{2,1} + 3) \land (t_{2,2} + 1 \leq 8) \land \\
(t_{3,1} \geq 0) \land (t_{3,2} \geq t_{3,1} + 2) \land (t_{3,2} + 3 \leq 8) \land \\
((t_{1,1} \geq t_{2,1} + 3) \lor (t_{2,1} \geq t_{1,1} + 2)) \land \\
((t_{1,1} \geq t_{3,1} + 2) \lor (t_{3,1} \geq t_{1,1} + 2)) \land \\
((t_{2,1} \geq t_{3,1} + 2) \lor (t_{3,1} \geq t_{2,1} + 3)) \land \\
((t_{1,2} \geq t_{2,2} + 1) \lor (t_{2,2} \geq t_{1,2} + 1)) \land \\
((t_{1,2} \geq t_{3,2} + 3) \lor (t_{3,2} \geq t_{1,2} + 1)) \land \\
((t_{2,2} \geq t_{3,2} + 3) \lor (t_{3,2} \geq t_{2,2} + 1))
\]
Chasing negative cycles!

Algorithms based on Bellman-Ford ($O(mn)$).
Many solvers (e.g., ICS, Simplify) are based on the Standard Simplex.

\[ a - d + 2e = 3 \]
\[ b - d = 1 \]
\[ c + d - e = -1 \]
\[ a, b, c, d, e \geq 0 \]
Many solvers (e.g., ICS, Simplify) are based on the Standard Simplex.

\[
\begin{align*}
 a - d + 2e &= 3 \\
 b - d &= 1 \\
 c + d - e &= -1 \\
 a, b, c, d, e &\geq 0
\end{align*}
\]

\[
\begin{pmatrix}
 1 & 0 & 0 & -1 & 2 \\
 0 & 1 & 0 & -1 & 0 \\
 0 & 0 & 1 & 1 & -1
\end{pmatrix}
\begin{pmatrix}
 a \\
 b \\
 c \\
 d \\
 e
\end{pmatrix}
= \begin{pmatrix}
 3 \\
 1 \\
 -1
\end{pmatrix}
\]

\[Ax = b \text{ and } x \geq 0.\]
Many solvers (e.g., ICS, Simplify) are based on the Standard Simplex.

\[
\begin{align*}
\text{We say } a, b, c \text{ are the basic (or dependent) variables}
\end{align*}
\]

\[
\begin{align*}
a - d + 2e &= 3 \\
b - d &= 1 \\
c + d - e &= -1 \\
a, b, c, d, e &\geq 0
\end{align*}
\]

\[
\begin{bmatrix}
1 & 0 & 0 & -1 & 2 \\
0 & 1 & 0 & -1 & 0 \\
0 & 0 & 1 & 1 & -1
\end{bmatrix}
\begin{bmatrix}
a \\
b \\
c \\
d \\
e
\end{bmatrix}
= \begin{bmatrix}
3 \\
1 \\
-1
\end{bmatrix}
\]

\[Ax = b \text{ and } x \geq 0.\]
Many solvers (e.g., ICS, Simplify) are based on the Standard Simplex.

\[
\begin{align*}
    a - d + 2e &= 3 \\
    b - d &= 1 \\
    c + d - e &= -1 \\
    a, b, c, d, e &\geq 0
\end{align*}
\]

We say \(a,b,c\) are the basic (or dependent) variables.

\[
\begin{pmatrix}
    1 & 0 & 0 & -1 & 2 \\
    0 & 1 & 0 & -1 & 0 \\
    0 & 0 & 1 & 1 & -1
\end{pmatrix}
\begin{pmatrix}
    a \\
    b \\
    c \\
    d \\
    e
\end{pmatrix}
= 
\begin{pmatrix}
    3 \\
    1 \\
    -1
\end{pmatrix}
\]

We say \(d,e\) are the non-basic (or non-dependent) variables.

\(Ax = b\) and \(x \geq 0\).
Incrementality: add/remove equations
Slow backtracking
No theory propagation
Fast Linear Arithmetic

- Simplex General Form
- Algorithm based on the dual simplex
- Non redundant proofs
- Efficient backtracking
- Efficient theory propagation
- Support for string inequalities: $t > 0$
- Preprocessing step
- Integer problems:
  - Gomory cuts, Branch & Bound, GCD test
General Form: \( Ax = 0 \) and \( l_j \leq x_j \leq u_j \)

Example:

\[
x \geq 0, (x + y \leq 2 \lor x + 2y \geq 6), (x + y = 2 \lor x + 2y > 4)
\]

\[
\Leftrightarrow
\]

\[
s_1 \equiv x + y, s_2 \equiv x + 2y,
\]

\[
x \geq 0, (s_1 \leq 2 \lor s_2 \geq 6), (s_1 = 2 \lor s_2 > 4)
\]

Only bounds (e.g., \( s_1 \leq 2 \)) are asserted during the search.

Unconstrained variables can be eliminated before the beginning of the search.
\[ s_1 \equiv x + y, \quad s_2 \equiv x + 2y \]
From Definitions to a Tableau

\[ s_1 \equiv x + y, \quad s_2 \equiv x + 2y \]

\[ s_1 = x + y, \quad s_2 = x + 2y \]
From Definitions to a Tableau

\[ s_1 \equiv x + y, \quad s_2 \equiv x + 2y \]

\[ s_1 = x + y, \]
\[ s_2 = x + 2y \]

\[ s_1 - x - y = 0 \]
\[ s_2 - x - 2y = 0 \]
From Definitions to a Tableau

\[ s_1 \equiv x + y, \quad s_2 \equiv x + 2y \]

\[ s_1 = x + y, \quad s_2 = x + 2y \]

\[ s_1 - x - y = 0 \quad s_1, s_2 \text{ are basic (dependent)} \]

\[ s_2 - x - 2y = 0 \quad x, y \text{ are non-basic} \]
A way to swap a basic with a non-basic variable!

It is just equational reasoning.

Key invariant: a basic variable occurs in only one equation.

Example: swap $s_1$ and $y$

\[
\begin{align*}
    s_1 - x - y &= 0 \\
    s_2 - x - 2y &= 0
\end{align*}
\]
A way to swap a basic with a non-basic variable!

It is just equational reasoning.

Key invariant: a basic variable occurs in only one equation.

Example: swap \(s_1\) and \(y\)

\[
\begin{align*}
  s_1 - x - y &= 0 \\
  s_2 - x - 2y &= 0 \\
\end{align*}
\]

\[
\begin{align*}
  -s_1 + x + y &= 0 \\
  s_2 - x - 2y &= 0 \\
\end{align*}
\]
A way to swap a basic with a non-basic variable!
It is just equational reasoning.
Key invariant: a basic variable occurs in only one equation.

Example: swap $s_1$ and $y$

\[
\begin{align*}
    s_1 - x - y &= 0 \\
    s_2 - x - 2y &= 0 \\
    -s_1 + x + y &= 0 \\
    s_2 - x - 2y &= 0 \\
    -s_1 + x + y &= 0 \\
    s_2 - 2s_1 + x &= 0
\end{align*}
\]
A way to swap a basic with a non-basic variable!
It is just equational reasoning.
Key invariant: a basic variable occurs in only one equation.
Example: swap $s_1$ and $y$

\[
\begin{align*}
    s_1 - x - y &= 0 \\
    s_2 - x - 2y &= 0 \\
    -s_1 + x + y &= 0 \\
    s_2 - x - 2y &= 0 \\
    -s_1 + x + y &= 0 \\
    s_2 - 2s_1 + x &= 0
\end{align*}
\]

It is just substituting equals by equals.
Pivoting

A way to swap a basic with a non-basic variable!
It is just equational reasoning.
Key invariant: a basic variable occurs in only one equation.
Example: swap \( s_1 \) and \( y \)

\[
\begin{align*}
  s_1 - x - y &= 0 \\
  s_2 - x - 2y &= 0 \\
  -s_1 + x + y &= 0 \\
  s_2 - x - 2y &= 0 \\
  -s_1 + x + y &= 0 \\
  s_2 - 2s_1 + x &= 0
\end{align*}
\]

It is just substituting equals by equals.

Definition:
An assignment (model) is a mapping from variables to values

Key Property:
If an assignment satisfies the equations before a pivoting step, then it will also satisfy them after!
A way to swap a basic with a non-basic variable!

It is just equational reasoning.

Key invariant: a basic variable occurs in only one equation.

Example: swap $s_2$ and $y$

$$s_1 - x - y = 0$$
$$s_2 - x - 2y = 0$$
$$-s_1 + x + y = 0$$
$$s_2 - x - 2y = 0$$
$$-s_1 + x + y = 0$$
$$s_2 - 2s_1 + x = 0$$

It is just substituting equals by equals.

Definition:
An assignment (model) is a mapping from variables to values.

Example:
<table>
<thead>
<tr>
<th>$M(x)$</th>
<th>$M(y)$</th>
<th>$M(s_1)$</th>
<th>$M(s_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Key Property:
If an assignment satisfies the equations before a pivoting step, then it will also satisfy them after!
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: \( x = y - z, \ y \leq 2, \ z \geq 3 \Rightarrow x \leq -1 \).

The new bound may be inconsistent with the already known bounds.

Example: \( x \leq -1, \ x \geq 0 \).
If the assignment of a non-basic variable does not satisfy a bound, then fix it and propagate the change to all dependent variables.

\[
\begin{align*}
    a & = c - d \\
    b & = c + d \\
    M(a) & = 0 \\
    M(b) & = 0 \\
    M(c) & = 0 \\
    M(d) & = 0 \\
    1 & \leq c
\end{align*}
\]
If the assignment of a non-basic variable does not satisfy a bound, then fix it and propagate the change to all dependent variables. Of course, we may introduce new “problems”.

\[
\begin{align*}
a &= c - d \\
b &= c + d \\
M(a) &= 0 \\
M(b) &= 0 \\
M(c) &= 0 \\
M(d) &= 0 \\
1 &\leq c \\
a &\leq 0
\end{align*}
\]
If the assignment of a basic variable does not satisfy a bound, then pivot it, fix it, and propagate the change to its new dependent variables.

\[
\begin{align*}
  a &= c - d \\
  b &= c + d \\
  M(a) &= 0 \\
  M(b) &= 0 \\
  M(c) &= 0 \\
  M(d) &= 0 \\
  1 &\leq a
\end{align*}
\]

\[
\begin{align*}
  c &= a + d \\
  b &= a + 2d \\
  M(a) &= 0 \\
  M(b) &= 0 \\
  M(c) &= 0 \\
  M(d) &= 0 \\
  1 &\leq a
\end{align*}
\]

\[
\begin{align*}
  c &= a + d \\
  b &= a + 2d \\
  M(a) &= 1 \\
  M(b) &= 1 \\
  M(c) &= 1 \\
  M(d) &= 0 \\
  1 &\leq a
\end{align*}
\]
Sometimes, a model cannot be repaired. It is pointless to pivot.

\[ a = b - c \]
\[ a \leq 0, \ 1 \leq b, \ c \leq 0 \]
\[ M(a) = 1 \]
\[ M(b) = 1 \]
\[ M(c) = 0 \]

The value of M(a) is too big. We can reduce it by:
- reducing M(b)
  not possible b is at lower bound
- increasing M(c)
  not possible c is at upper bound
Extracting proof from failed repair attempts is easy.

$s_1 \equiv a + d$, $s_2 \equiv c + d$

$a = s_1 - s_2 + c$

$a \leq 0$, $1 \leq s_1$, $s_2 \leq 0$, $0 \leq c$

$M(a) = 1$

$M(s_1) = 1$

$M(s_2) = 0$

$M(c) = 0$
Extracting proof from failed repair attempts is easy.

\[ s_1 = a + d, \quad s_2 = c + d \]
\[ a = s_1 - s_2 + c \]
\[ a \leq 0, \ 1 \leq s_1, \ s_2 \leq 0, \ 0 \leq c \]
\[ M(a) = 1 \]
\[ M(s_1) = 1 \]
\[ M(s_2) = 0 \]
\[ M(c) = 0 \]

\{ a \leq 0, \ 1 \leq s_1, \ s_2 \leq 0, \ 0 \leq c \} \text{ is inconsistent}


"Repairing Models"

Extracting proof from failed repair attempts is easy.

\[ s_1 \equiv a + d, \ s_2 \equiv c + d \]
\[ a = s_1 - s_2 + c \]
\[ a \leq 0, \ 1 \leq s_1, \ s_2 \leq 0, \ 0 \leq c \]
\[ M(a) = 1 \]
\[ M(s_1) = 1 \]
\[ M(s_2) = 0 \]
\[ M(c) = 0 \]

\{ a \leq 0, \ 1 \leq s_1, \ s_2 \leq 0, \ 0 \leq c \} \text{ is inconsistent}

\{ a \leq 0, \ 1 \leq a + d, \ c + d \leq 0, \ 0 \leq c \} \text{ is inconsistent}
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. $x < 1 \Rightarrow x \leq 0$.

For rational/real problems, strict inequalities can be converted into non-strict inequalities using a small $\delta$. $x < 1 \Rightarrow x \leq 1 - \delta$.

We do not compute a $\delta$, we treat it symbolically.

$\delta$ is an infinitesimal parameter: $(c, k) = c + k\delta$
**Example**

- **Initial state**

\[ s \geq 1, \ x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 0 )</td>
<td>( s = x + y )</td>
<td>( M(u) = 0 )</td>
</tr>
<tr>
<td>( M(y) = 0 )</td>
<td>( u = x + 2y )</td>
<td>( M(v) = 0 )</td>
</tr>
<tr>
<td>( M(s) = 0 )</td>
<td>( v = x - y )</td>
<td>( M(u) = 0 )</td>
</tr>
<tr>
<td>( M(u) = 0 )</td>
<td></td>
<td>( M(v) = 0 )</td>
</tr>
</tbody>
</table>
Example

- Asserting $s \geq 1$

  $s \geq 1, x \geq 0$

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 0$</td>
<td>$s = x + y$</td>
<td></td>
</tr>
<tr>
<td>$M(y) = 0$</td>
<td>$u = x + 2y$</td>
<td></td>
</tr>
<tr>
<td>$M(s) = 0$</td>
<td>$v = x - y$</td>
<td></td>
</tr>
<tr>
<td>$M(u) = 0$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = 0$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

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

$s \geq 1, x \geq 0$

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 0$</td>
<td>$s = x + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 0$</td>
<td>$u = x + 2y$</td>
<td></td>
</tr>
<tr>
<td>$M(s) = 0$</td>
<td>$v = x - y$</td>
<td></td>
</tr>
<tr>
<td>$M(u) = 0$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = 0$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

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

\[ s \geq 1, \ x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x)$ = 0</td>
<td>$s = x + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y)$ = 0</td>
<td>$u = x + 2y$</td>
<td></td>
</tr>
<tr>
<td>$M(s)$ = 0</td>
<td>$v = x - y$</td>
<td></td>
</tr>
<tr>
<td>$M(u)$ = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v)$ = 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Asserting \( s \geq 1 \) pivot \( s \) and \( x \) (\( s \) is a dependent variable).

\[
\begin{align*}
  s & \geq 1, x \geq 0 \\
  (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1)
\end{align*}
\]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 0 )</td>
<td>( x = s - y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 0 )</td>
<td>( u = x + 2y )</td>
<td></td>
</tr>
<tr>
<td>( M(s) = 0 )</td>
<td>( v = x - y )</td>
<td></td>
</tr>
<tr>
<td>( M(u) = 0 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M(v) = 0 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

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

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

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 0$</td>
<td>$x = s - y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 0$</td>
<td>$u = s + y$</td>
<td></td>
</tr>
<tr>
<td>$M(s) = 0$</td>
<td>$v = s - 2y$</td>
<td></td>
</tr>
<tr>
<td>$M(u) = 0$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = 0$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Example

- Asserting $s \geq 1$ update assignment.

\[ s \geq 1, x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 0$</td>
<td>$x = s - y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 0$</td>
<td>$u = s + y$</td>
<td></td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = s - 2y$</td>
<td></td>
</tr>
<tr>
<td>$M(u) = 0$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = 0$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Example**

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

\[
s \geq 1, \quad x \geq 0,
\]
\[
(y \leq 1 \lor u \geq 2), \quad (v \leq -2 \lor v \geq 0), \quad (v \leq -2 \lor u \leq -1)
\]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 1$</td>
<td>$x = s - y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 0$</td>
<td>$u = s + y$</td>
<td></td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = s - 2y$</td>
<td></td>
</tr>
<tr>
<td>$M(u) = 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = 1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

 Asserting $x \geq 0$

\[
\begin{align*}
  s &\geq 1, x \geq 0 \\
  (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1)
\end{align*}
\]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 1$</td>
<td>$x = s - y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 0$</td>
<td>$u = s + y$</td>
<td></td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = s - 2y$</td>
<td></td>
</tr>
<tr>
<td>$M(u) = 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = 1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Asserting \( x \geq 0 \) assignment satisfies new bound.

\[
s \geq 1, \ x \geq 0
\]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 1 )</td>
<td>( x = s - y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 0 )</td>
<td>( u = s + y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 )</td>
<td>( v = s - 2y )</td>
<td></td>
</tr>
<tr>
<td>( M(u) = 1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M(v) = 1 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case split $\neg y \leq 1$

\[ s \geq 1, \ x \geq 0 \]
\[ (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1) \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 1$</td>
<td>$x = s - y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 0$</td>
<td>$u = s + y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = s - 2y$</td>
<td></td>
</tr>
<tr>
<td>$M(u) = 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = 1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

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

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

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 1$</td>
<td>$x = s - y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 0$</td>
<td>$u = s + y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = s - 2y$</td>
<td>$y &gt; 1$</td>
</tr>
<tr>
<td>$M(u) = 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = 1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case split $-y \leq 1$ update assignment.

$s \geq 1, x \geq 0$

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 1$</td>
<td>$x = s - y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1 + \delta$</td>
<td>$u = s + y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = s - 2y$</td>
<td>$y &gt; 1$</td>
</tr>
<tr>
<td>$M(u) = 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = 1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Case split \(-y \leq 1\) update dependent variables assignment.

\[ s \geq 1, x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>(M(x)) = (-\delta)</td>
<td>(x = s - y)</td>
<td>(s \geq 1)</td>
</tr>
<tr>
<td>(M(y)) = (1 + \delta)</td>
<td>(u = s + y)</td>
<td>(x \geq 0)</td>
</tr>
<tr>
<td>(M(s)) = (1)</td>
<td>(v = s - 2y)</td>
<td>(y &gt; 1)</td>
</tr>
<tr>
<td>(M(u)) = (2 + \delta)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M(v)) = (-1 - 2\delta)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- **Bound violation**

\[ s \geq 1, x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) ) = (-\delta)</td>
<td>( x = s - y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) ) = (1 + \delta)</td>
<td>( u = s + y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) ) = (1)</td>
<td>( v = s - 2y )</td>
<td>( y &gt; 1 )</td>
</tr>
<tr>
<td>( M(u) ) = (2 + \delta)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M(v) ) = (-1 - 2\delta)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

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

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

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x)$ = $-\delta$</td>
<td>$x = s - y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1 + \delta$</td>
<td>$u = s + y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = s - 2y$</td>
<td>$y &gt; 1$</td>
</tr>
<tr>
<td>$M(u) = 2 + \delta$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = -1 - 2\delta$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Example**

- **Bound violation pivot** $x$ and $s$ ($x$ is a dependent variable).

\[
s \geq 1, \quad x \geq 0
\]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = -\delta$</td>
<td>$s = x + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1 + \delta$</td>
<td>$u = s + y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = s - 2y$</td>
<td>$y &gt; 1$</td>
</tr>
<tr>
<td>$M(u) = 2 + \delta$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = -1 - 2\delta$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

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

\[
s \geq 1, \quad x \geq 0
\]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = -\delta$</td>
<td>$s = x + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1 + \delta$</td>
<td>$u = x + 2y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = x - y$</td>
<td></td>
</tr>
<tr>
<td>$M(u) = 2 + \delta$</td>
<td></td>
<td>$y &gt; 1$</td>
</tr>
<tr>
<td>$M(v) = -1 - 2\delta$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Bound violation update assignment.

\[ s \geq 1, \ x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 0 )</td>
<td>( s = x + y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 + \delta )</td>
<td>( u = x + 2y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 )</td>
<td>( v = x - y )</td>
<td>( y &gt; 1 )</td>
</tr>
<tr>
<td>( M(u) = 2 + \delta )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M(v) = -1 - 2\delta )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Bound violation update dependent variables assignment.

\[ s \geq 1, \ x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 0 )</td>
<td>( s = x + y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 + \delta )</td>
<td>( u = x + 2y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 + \delta )</td>
<td>( v = x - y )</td>
<td>( y &gt; 1 )</td>
</tr>
<tr>
<td>( M(u) = 2 + 2\delta )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M(v) = -1 - \delta )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Theory propagation \( x \geq 0, y > 1 \leadsto u > 2 \)

\[
\begin{align*}
  s \geq 1, x \geq 0 \\
  (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1)
\end{align*}
\]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) ) = 0</td>
<td>( s = x + y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 + \delta )</td>
<td>( u = x + 2y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 + \delta )</td>
<td>( v = x - y )</td>
<td>( y &gt; 1 )</td>
</tr>
<tr>
<td>( M(u) = 2 + 2\delta )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M(v) = -1 - \delta )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Theory propagation \( u > 2 \Leftrightarrow -u \leq -1 \)

\[ s \geq 1, \ x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) ) = 0</td>
<td>( s = x + y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 + \delta )</td>
<td>( u = x + 2y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 + \delta )</td>
<td>( v = x - y )</td>
<td>( y &gt; 1 )</td>
</tr>
<tr>
<td>( M(u) = 2 + 2\delta )</td>
<td></td>
<td>( u &gt; 2 )</td>
</tr>
<tr>
<td>( M(v) = -1 - \delta )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Boolean propagation $-y \leq 1 \leadsto v \geq 2$

$s \geq 1, x \geq 0$

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 0$</td>
<td>$s = x + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1 + \delta$</td>
<td>$u = x + 2y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1 + \delta$</td>
<td>$v = x - y$</td>
<td>$y &gt; 1$</td>
</tr>
<tr>
<td>$M(u) = 2 + 2\delta$</td>
<td></td>
<td>$u &gt; 2$</td>
</tr>
<tr>
<td>$M(v) = -1 - \delta$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Theory propagation: $v \geq 2 \leadsto -v \leq -2$

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

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x)$</td>
<td>$s = x + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y)$</td>
<td>$u = x + 2y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s)$</td>
<td>$v = x - y$</td>
<td>$y &gt; 1$</td>
</tr>
<tr>
<td>$M(u)$</td>
<td></td>
<td>$u &gt; 2$</td>
</tr>
<tr>
<td>$M(v)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Example

- **Conflict empty clause**

\[ s \geq 1, x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 0 )</td>
<td>( s = x + y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 + \delta )</td>
<td>( u = x + 2y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 + \delta )</td>
<td>( v = x - y )</td>
<td>( y &gt; 1 )</td>
</tr>
<tr>
<td>( M(u) = 2 + 2\delta )</td>
<td></td>
<td>( u &gt; 2 )</td>
</tr>
<tr>
<td>( M(v) = -1 - \delta )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

Backtracking

\[ s \geq 1, \ x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) ) = 0</td>
<td>( s = x + y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 + \delta )</td>
<td>( u = x + 2y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 + \delta )</td>
<td>( v = x - y )</td>
<td></td>
</tr>
<tr>
<td>( M(u) = 2 + 2\delta )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M(v) = -1 - \delta )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Asserting $y \leq 1$

\[
s \geq 1, \ x \geq 0
\]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 0$</td>
<td>$s = x + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1 + \delta$</td>
<td>$u = x + 2y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1 + \delta$</td>
<td>$v = x - y$</td>
<td></td>
</tr>
<tr>
<td>$M(u) = 2 + 2\delta$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = -1 - \delta$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

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

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

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 0$</td>
<td>$s = x + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1 + \delta$</td>
<td>$u = x + 2y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1 + \delta$</td>
<td>$v = x - y$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u) = 2 + 2\delta$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = -1 - \delta$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Asserting $y \leq 1$ update assignment.

\[ s \geq 1, x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x)$ = 0</td>
<td>$s = x + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y)$ = 1</td>
<td>$u = x + 2y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1 + \delta$</td>
<td>$v = x - y$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u) = 2 + 2\delta$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = -1 - \delta$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

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

\[ s \geq 1, \ x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 0$</td>
<td>$s = x + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1$</td>
<td>$u = x + 2y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = x - y$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u) = 2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M(v) = -1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

Theory propagation \( s \geq 1, y \leq 1 \iff v \geq -1 \)

\[
\begin{align*}
  s \geq 1, x \geq 0 \\
  (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1)
\end{align*}
\]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 0 )</td>
<td>( x = s - y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 )</td>
<td>( u = s + y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 )</td>
<td>( v = s - 2y )</td>
<td>( y \leq 1 )</td>
</tr>
<tr>
<td>( M(u) = 2 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M(v) = -1 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Theory propagation \( v \geq -1 \Rightarrow -v \leq -2 \)

\[
\begin{align*}
\text{Model} & \quad \text{Equations} & \quad \text{Bounds} \\
M(x) = 0 & \quad x = s - y & \quad s \geq 1 \\
M(y) = 1 & \quad u = s + y & \quad x \geq 0 \\
M(s) = 1 & \quad v = s - 2y & \quad y \leq 1 \\
M(u) = 2 & \quad & \quad v \geq -1 \\
M(v) = -1 & \quad &
\end{align*}
\]
Example

- Boolean propagation
  \[ \neg v \leq -2 \implies v \geq 0 \]
  \[ s \geq 1, \ x \geq 0 \]
  \[ (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1) \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 0 )</td>
<td>( x = s - y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 )</td>
<td>( u = s + y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 )</td>
<td>( v = s - 2y )</td>
<td>( y \leq 1 )</td>
</tr>
<tr>
<td>( M(u) = 2 )</td>
<td></td>
<td>( v \geq -1 )</td>
</tr>
<tr>
<td>( M(v) = -1 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- **Bound violation** assignment does not satisfy new bound.

\[
\begin{align*}
\ s & \geq 1, \ x \geq 0 \\
(y \leq 1 \lor v \geq 2), (v \leq -2 \lor u \geq 0), (v \leq -2 \lor u \leq -1)
\end{align*}
\]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) ) = 0</td>
<td>( x = s - y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) ) = 1</td>
<td>( u = s + y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) ) = 1</td>
<td>( v = s - 2y )</td>
<td>( y \leq 1 )</td>
</tr>
<tr>
<td>( M(u) ) = 2</td>
<td></td>
<td>( v \geq 0 )</td>
</tr>
<tr>
<td>( M(v) ) = -1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bound violation pivot $u$ and $s$ ($u$ is a dependent variable).

$$\begin{align*}
  s & \geq 1, \quad x \geq 0 \\
  (y \leq 1 \lor u \geq 2), \quad (v \leq -2 \lor v \geq 0), \quad (v \leq -2 \lor u \leq -1)
\end{align*}$$

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 0$</td>
<td>$x = s - y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1$</td>
<td>$u = s + y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$v = s - 2y$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u) = 2$</td>
<td>$v \geq 0$</td>
<td></td>
</tr>
<tr>
<td>$M(v) = -1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

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

\[
\begin{align*}
    s & \geq 1, x \geq 0 \\
    (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1)
\end{align*}
\]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x)$ = 0</td>
<td>$x = s - y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y)$ = 1</td>
<td>$u = s + y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s)$ = 1</td>
<td>$s = v + 2y$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u)$ = 2</td>
<td></td>
<td>$v \geq 0$</td>
</tr>
<tr>
<td>$M(v)$ = -1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

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

\[ s \geq 1, \; x \geq 0 \]
\[(y \leq 1 \vee v \geq 2), (v \leq -2 \vee v \geq 0), (v \leq -2 \vee u \leq -1)\]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 0$</td>
<td>$x = v + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1$</td>
<td>$u = v + 3y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 1$</td>
<td>$s = v + 2y$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u) = 2$</td>
<td></td>
<td>$v \geq 0$</td>
</tr>
<tr>
<td>$M(v) = -1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Bound violation: update assignment.

\[ s \geq 1, \quad x \geq 0 \]
\[ (y \leq 1 \lor v \geq 2), \quad (v \leq -2 \lor v \geq 0), \quad (v \leq -2 \lor u \leq -1) \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 0 )</td>
<td>( x = v + y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 )</td>
<td>( u = v + 3y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 )</td>
<td>( s = v + 2y )</td>
<td>( y \leq 1 )</td>
</tr>
<tr>
<td>( M(u) = 2 )</td>
<td></td>
<td>( v \geq 0 )</td>
</tr>
<tr>
<td>( M(v) = 0 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bound violation  update dependent variables assignment.

\[ s \geq 1, x \geq 0 \]
\[ (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1) \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 1 )</td>
<td>( x = v + y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 )</td>
<td>( u = v + 3y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 2 )</td>
<td>( s = v + 2y )</td>
<td>( y \leq 1 )</td>
</tr>
<tr>
<td>( M(u) = 3 )</td>
<td></td>
<td>( v \geq 0 )</td>
</tr>
<tr>
<td>( M(v) = 0 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Boolean propagation: $-v \leq -2 \iff u \leq -1$

\[ s \geq 1, \quad x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 1$</td>
<td>$x = v + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1$</td>
<td>$u = v + 3y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 2$</td>
<td>$s = v + 2y$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u) = 3$</td>
<td></td>
<td>$v \geq 0$</td>
</tr>
<tr>
<td>$M(v) = 0$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

- Bound violation assignment does not satisfy new bound.

\[ s \geq 1, \quad x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 1 )</td>
<td>( x = v + y )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 )</td>
<td>( u = v + 3y )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 2 )</td>
<td>( s = v + 2y )</td>
<td>( y \leq 1 )</td>
</tr>
<tr>
<td>( M(u) = 3 )</td>
<td></td>
<td>( v \geq 0 )</td>
</tr>
<tr>
<td>( M(v) = 0 )</td>
<td></td>
<td>( u \leq -1 )</td>
</tr>
</tbody>
</table>
Example

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

\[
s \geq 1, \ x \geq 0
\]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equation</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 1$</td>
<td>$x = v + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1$</td>
<td>$u = v + 3y$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 2$</td>
<td>$s = v + 2y$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u) = 3$</td>
<td></td>
<td>$v \geq 0$</td>
</tr>
<tr>
<td>$M(v) = 0$</td>
<td></td>
<td>$u \leq -1$</td>
</tr>
</tbody>
</table>
Example

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

\[
s \geq 1, \ x \geq 0
\]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 1$</td>
<td>$x = v + y$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1$</td>
<td>$y = \frac{1}{3}u - \frac{1}{3}v$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 2$</td>
<td>$s = v + 2y$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u) = 3$</td>
<td></td>
<td>$v \geq 0$</td>
</tr>
<tr>
<td>$M(v) = 0$</td>
<td></td>
<td>$u \leq -1$</td>
</tr>
</tbody>
</table>
Example

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

\[
s \geq 1, \quad x \geq 0
\]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = 1$</td>
<td>$x = \frac{1}{3}u + \frac{2}{3}v$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = 1$</td>
<td>$y = \frac{1}{3}u - \frac{1}{3}v$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = 2$</td>
<td>$s = \frac{2}{3}u + \frac{1}{3}v$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u) = 3$</td>
<td></td>
<td>$v \geq 0$</td>
</tr>
<tr>
<td>$M(v) = 0$</td>
<td></td>
<td>$u \leq -1$</td>
</tr>
</tbody>
</table>
Example

- Bound violation

\[ s \geq 1, \ x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 1 )</td>
<td>( x = \frac{1}{3}u + \frac{2}{3}v )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = 1 )</td>
<td>( y = \frac{1}{3}u - \frac{1}{3}v )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 2 )</td>
<td>( s = \frac{2}{3}u + \frac{1}{3}v )</td>
<td>( y \leq 1 )</td>
</tr>
<tr>
<td>( M(u) = -1 )</td>
<td>( v \geq 0 )</td>
<td></td>
</tr>
<tr>
<td>( M(v) = 0 )</td>
<td>( u \leq -1 )</td>
<td></td>
</tr>
</tbody>
</table>
Example

- Bound violation
- update dependent variables assignment.

\[ s \geq 1, x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>(M(x) = -\frac{1}{3})</td>
<td>(x = \frac{1}{3}u + \frac{2}{3}v)</td>
<td>(s \geq 1)</td>
</tr>
<tr>
<td>(M(y) = -\frac{1}{3})</td>
<td>(y = \frac{1}{3}u - \frac{1}{3}v)</td>
<td>(x \geq 0)</td>
</tr>
<tr>
<td>(M(s) = -\frac{2}{3})</td>
<td>(s = \frac{2}{3}u + \frac{1}{3}v)</td>
<td>(y \leq 1)</td>
</tr>
<tr>
<td>(M(u) = -1)</td>
<td>(v \geq 0)</td>
<td></td>
</tr>
<tr>
<td>(M(v) = 0)</td>
<td>(u \leq -1)</td>
<td></td>
</tr>
</tbody>
</table>
## Example

- **Bound violations**

\[
s \geq 1, \quad x \geq 0
\]

\[
(y \leq 1 \lor v \geq 2), \quad (v \leq -2 \lor v \geq 0), \quad (v \leq -2 \lor u \leq -1)
\]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>(M(x)) = (-\frac{1}{3})</td>
<td>(x = \frac{1}{3}u + \frac{2}{3}v)</td>
<td>(s \geq 1)</td>
</tr>
<tr>
<td>(M(y)) = (-\frac{1}{3})</td>
<td>(y = \frac{1}{3}u - \frac{1}{3}v)</td>
<td>(x \geq 0)</td>
</tr>
<tr>
<td>(M(s)) = (-\frac{2}{3})</td>
<td>(s = \frac{2}{3}u + \frac{1}{3}v)</td>
<td></td>
</tr>
<tr>
<td>(M(u)) = (-1)</td>
<td></td>
<td>(y \leq 1)</td>
</tr>
<tr>
<td>(M(v)) = (0)</td>
<td></td>
<td>(v \geq 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(u \leq -1)</td>
</tr>
</tbody>
</table>
Example

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

\[ s \geq 1, \ x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x)$</td>
<td>$x = \frac{1}{3} u + \frac{2}{3} v$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y)$</td>
<td>$y = \frac{1}{3} u - \frac{1}{3} v$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s)$</td>
<td>$s = \frac{2}{3} u + \frac{1}{3} v$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u)$</td>
<td>$u \leq -1$</td>
<td></td>
</tr>
<tr>
<td>$M(v)$</td>
<td>$v \geq 0$</td>
<td></td>
</tr>
</tbody>
</table>
Example

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

\[ s \geq 1, \ x \geq 0 \]
\[ (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1) \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x)$</td>
<td>$x = \frac{1}{3}u + \frac{2}{3}v$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y)$</td>
<td>$y = \frac{1}{3}u - \frac{1}{3}v$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s)$</td>
<td>$v = 3s - 2u$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u)$</td>
<td></td>
<td>$v \geq 0$</td>
</tr>
<tr>
<td>$M(v)$</td>
<td></td>
<td>$u \leq -1$</td>
</tr>
</tbody>
</table>
Example

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

\[ s \geq 1, \ x \geq 0 \]
\[ (y \leq 1 \lor v \geq 2), \ (v \leq -2 \lor v \geq 0), \ (v \leq -2 \lor u \leq -1) \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(x) = -\frac{1}{3}$</td>
<td>$x = 2s - u$</td>
<td>$s \geq 1$</td>
</tr>
<tr>
<td>$M(y) = -\frac{1}{3}$</td>
<td>$y = -s + u$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$M(s) = -\frac{2}{3}$</td>
<td>$v = 3s - 2u$</td>
<td>$y \leq 1$</td>
</tr>
<tr>
<td>$M(u) = -1$</td>
<td></td>
<td>$v \geq 0$</td>
</tr>
<tr>
<td>$M(v) = 0$</td>
<td></td>
<td>$u \leq -1$</td>
</tr>
</tbody>
</table>
**Example**

- **Bound violations update assignment.**

  \[ s \geq 1, x \geq 0 \]
  \[ (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1) \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = -\frac{1}{3} )</td>
<td>( x = 2s - u )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = -\frac{1}{3} )</td>
<td>( y = -s + u )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 )</td>
<td>( v = 3s - 2u )</td>
<td>( y \leq 1 )</td>
</tr>
<tr>
<td>( M(u) = -1 )</td>
<td></td>
<td>( v \geq 0 )</td>
</tr>
<tr>
<td>( M(v) = 0 )</td>
<td></td>
<td>( u \leq -1 )</td>
</tr>
</tbody>
</table>
Example

- Bound violations update dependent variables assignment.

\[ s \geq 1, x \geq 0 \]
\[ (y \leq 1 \lor v \geq 2), (v \leq -2 \lor v \geq 0), (v \leq -2 \lor u \leq -1) \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 3 )</td>
<td>( x = 2s - u )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = -2 )</td>
<td>( y = -s + u )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 )</td>
<td>( v = 3s - 2u )</td>
<td>( y \leq 1 )</td>
</tr>
<tr>
<td>( M(u) = -1 )</td>
<td></td>
<td>( v \geq 0 )</td>
</tr>
<tr>
<td>( M(v) = 5 )</td>
<td></td>
<td>( u \leq -1 )</td>
</tr>
</tbody>
</table>
Example

- Found satisfying assignment

\[ s \geq 1, \quad x \geq 0 \]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Equations</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M(x) = 3 )</td>
<td>( x = 2s - u )</td>
<td>( s \geq 1 )</td>
</tr>
<tr>
<td>( M(y) = -2 )</td>
<td>( y = -s + u )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( M(s) = 1 )</td>
<td>( v = 3s - 2u )</td>
<td>( y \leq 1 )</td>
</tr>
<tr>
<td>( M(u) = -1 )</td>
<td>( v \geq 0 )</td>
<td>( u \leq -1 )</td>
</tr>
<tr>
<td>( M(v) = 5 )</td>
<td>( u \leq -1 )</td>
<td>( u \leq -1 )</td>
</tr>
</tbody>
</table>
Correctness

Completeness: trivial
Soundness: also trivial
Termination: non trivial.

We cannot choose arbitrary variable to pivot.
Assume the variables are ordered.
Bland’s rule: select the smallest basic variable $c$ that does not satisfy its bounds, then select the smallest non-basic in the row of $c$ that can be used for pivoting.

Too technical.

Uses the fact that a tableau has a finite number of configurations. Then, any infinite trace will have cycles.
Array of rows (equations).

Each row is a dynamic array of tuples:
(coefficient, variable, pos_in_occs, is_dead)

Each variable x has a “set” (dynamic array) of occurrences:
(row_idx, pos_in_row, is_dead)

Each variable x has a “field” row[x]
row[x] is -1 if x is non basic
otherwise, row[x] contains the idx of the row containing x

Each variable x has “fields”: lower[x], upper[x], and value[x]
rows: array of rows (equations).
   Each row is a dynamic array of tuples:
   (coefficient, variable, pos_in_occs, is_dead)

occs[x]: Each variable x has a “set” (dynamic array) of occurrences:
   (row_idx, pos_in_row, is_dead)

row[x]:
   row[x] is -1 if x is non basic
   otherwise, row[x] contains the idx of the row containing x

Other “fields”: lower[x], upper[x], and value[x]

atoms[x]: atoms (assigned/unassigned) that contains x
Data-structures

\[ s_1 \equiv a + b, \quad s_2 \equiv c - b \]
\[ p_1 \equiv a \leq 0, \quad p_2 \equiv 1 \leq s_1, \quad p_3 \equiv 1 \leq s_2 \]
p_1, p_2 were already assigned
\[ a - s_1 + s_2 + c = 0 \]
\[ b - c + s_2 = 0 \]
\[ a \leq 0, \quad 1 \leq s_1 \]
\[ M(a) = 0 \quad \text{value}[a] = 0 \]
\[ M(b) = -1 \quad \text{value}[a] = -1 \]
\[ M(c) = 0 \quad \text{value}[c] = 0 \]
\[ M(s_1) = 1 \quad \text{value}[s_1] = 1 \]
\[ M(s_2) = 1 \quad \text{value}[s_2] = 1 \]

\[ \text{rows} = [ \]
\[ [(1, a, 0, t), (-1, s_1, 0, t), (1, s_2, 1, t), (1, c, 0, t)], \]
\[ [(1, b, 0, t), (-1, c, 1, t), (1, s_2, 2, t)] ] \]

\[ \text{occs}[a] = [(0, 0, f)] \]
\[ \text{occs}[b] = [(1,0,f)] \]
\[ \text{occs}[c] = [(0,3,f), (1,1,f)] \]
\[ \text{occs}[s_1] = [(0,1,f)] \]
\[ \text{occs}[s_2] = [(0,0,t), (0,2,f), (1,2,f)] \]

\[ \text{row}[a] = 0, \quad \text{row}[b] = 1, \quad \text{row}[c] = -1, \ldots \]
\[ \text{upper}[a] = 0, \quad \text{lower}[s_1] = 1 \]
\[ \text{atoms}[a] = \{ p_1 \}, \quad \text{atoms}[s_1] = \{ p_2 \}, \ldots \]
In practice, we need a combination of theories.

\[ b + 2 = c \quad \text{and} \quad f(\text{read(\text{write}(a, b, 3), c-2)}) \neq f(c-b+1) \]

A theory is a set (potentially infinite) of first-order sentences.

**Main questions:**
Is the union of two theories \( T_1 \cup T_2 \) consistent?

Given a solvers for \( T_1 \) and \( T_2 \), how can we build a solver for \( T_1 \cup T_2 \)?
Two theories are disjoint if they do not share function/constant and predicate symbols.

= is the only exception.

Example:
The theories of arithmetic and arrays are disjoint.

Arithmetic symbols: \{0, -1, 1, -2, 2, ..., +, -, *, >, <, \geq, \leq\}
Array symbols: \{read, write\}
It is a different name for our “naming” subterms procedure.

\[ b + 2 = c, \ f(\text{read(write(a,b,3), c-2)}) \neq f(c-b+1) \]

\[ b + 2 = c, \ v_6 \neq v_7 \]
\[ v_1 \equiv 3, \ v_2 \equiv \text{write}(a, b, v_1), \ v_3 \equiv c-2, \ v_4 \equiv \text{read}(v_2, v_3), \]
\[ v_5 \equiv c-b+1, \ v_6 \equiv f(v_4), \ v_7 \equiv f(v_5) \]
Purification

It is a different name for our “naming” subterms procedure.

\[ b + 2 = c, \; f(\text{read}(\text{write}(a,b,3), c-2)) \neq f(c-b+1) \]

\[ b + 2 = c, \; v_6 \neq v_7 \]
\[ v_1 \equiv 3, \; v_2 \equiv \text{write}(a, b, v_1), \; v_3 \equiv c-2, \; v_4 \equiv \text{read}(v_2, v_3), \]
\[ v_5 \equiv c-b+1, \; v_6 \equiv f(v_4), \; v_7 \equiv f(v_5) \]

\[ b + 2 = c, \; v_1 \equiv 3, \; v_3 \equiv c-2, \; v_5 \equiv c-b+1, \]
\[ v_2 \equiv \text{write}(a, b, v_1), \; v_4 \equiv \text{read}(v_2, v_3), \]
\[ v_6 \equiv f(v_4), \; v_7 \equiv f(v_5), \; v_6 \neq v_7 \]
A theory is stably infinite if every satisfiable QFF is satisfiable in an infinite model.

EUF and arithmetic are stably infinite.

Bit-vectors are not.
The union of two consistent, disjoint, stably infinite theories is consistent.
A theory $T$ is convex iff

for all finite sets $S$ of literals and
for all $a_1 = b_1 \lor \ldots \lor a_n = b_n$

$S$ implies $a_1 = b_1 \lor \ldots \lor a_n = b_n$

iff

$S$ implies $a_i = b_i$ for some $1 \leq i \leq n$
Convexity: Results

Every convex theory with non trivial models is stably infinite.

All Horn equational theories are convex.

formulas of the form $s_1 \neq r_1 \lor ... \lor s_n \neq r_n \lor t = t'$

Linear rational arithmetic is convex.
Convexity: Negative Results

Linear integer arithmetic is not convex
\[ 1 \leq a \leq 2, \ b = 1, \ c = 2 \ \text{implies} \ a = b \lor a = c \]

Nonlinear arithmetic
\[ a^2 = 1, \ b = 1, \ c = -1 \ \text{implies} \ a = b \lor a = c \]

Theory of bit-vectors

Theory of arrays
\[ c_1 = \text{read(write}(a, \ i, \ c_2), \ j), \ c_3 = \text{read}(a, \ j) \]
\[ \text{implies} \ c_1 = c_2 \lor c_1 = c_3 \]
EUF is convex (O(n log n))
IDL is non-convex (O(nm))

EUF \cup IDL is NP-Complete
Reduce 3CNF to EUF \cup IDL
For each boolean variable \( p_i \) add \( 0 \leq a_i \leq 1 \)
For each clause \( p_1 \lor \neg p_2 \lor p_3 \) add
\[ f(a_1, a_2, a_3) \neq f(0, 1, 0) \]
Combination of non-convex theories

EUF is convex (O(n log n))
IDL is non-convex (O(nm))

EUF \cup IDL is NP-Complete
Reduce 3CNF to EUF \cup IDL
For each boolean variable \( p_i \) add \( 0 \leq a_i \leq 1 \)
For each clause \( p_1 \lor \neg p_2 \lor p_3 \) add
\[ f(a_1, a_2, a_3) \neq f(0, 1, 0) \]
implies
\[ a_1 \neq 0 \lor a_2 \neq 1 \lor a_3 \neq 0 \]
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 $O(n^3 \times (T_1(n) + T_2(n)))$. 
The combination procedure:

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

**Purification:** Preserving satisfiability transform $\phi$ into $\phi_1 \land \phi_2$, such that, $\phi_i \in \Sigma_i$.

**Interaction:** Guess a partition of $\forall(\phi_1) \cap \forall(\phi_2)$ into disjoint subsets. Express it as conjunction of literals $\psi$.

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 $\phi_i \land \psi$ is satisfiable.

**Return:** If both return yes, return yes. No, otherwise.
Each step is satisfiability preserving.

Say $\phi$ is satisfiable (in the combination).

- Purification: $\phi_1 \land \phi_2$ is satisfiable.
- Iteration: for some partition $\psi$, $\phi_1 \land \phi_2 \land \psi$ is satisfiable.
- Component procedures: $\phi_1 \land \psi$ and $\phi_2 \land \psi$ are both satisfiable in component theories.
- Therefore, if the procedure return unsatisfiable, then $\phi$ is unsatisfiable.
Suppose the procedure returns satisfiable.

- Let $\psi$ be the partition and $A$ and $B$ be models of $\mathcal{T}_1 \land \phi_1 \land \psi$
  and $\mathcal{T}_2 \land \phi_2 \land \psi$.

- 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 $\Sigma_1$:
  $\bar{B}(f)(b_1, \ldots, b_n) = h(A(f)(h^{-1}(b_1), \ldots, h^{-1}(b_n)))$.

- $\bar{B}$ is a model of:
  $\mathcal{T}_1 \land \phi_1 \land \mathcal{T}_2 \land \phi_2 \land \psi$.
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 $\phi_2 := \phi_2 \land x = y$. And vice-versa. Repeat until no further changes.

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

Remark: $\mathcal{T}_i \vdash (\phi_i \Rightarrow x = y)$ iff $\phi_i \land x \neq y$ is not satisfiable in $\mathcal{T}_i$. 
Assume the theories are convex.

- Suppose $\phi_i$ is satisfiable.
- Let $E$ be the set of equalities $x_j = x_k$ ($j \neq k$) such that, $T_i \models \phi_i \Rightarrow x_j = x_k$.
- By convexity, $T_i \models \phi_i \Rightarrow \bigvee_E x_j = x_k$.
- $\phi_i \land \bigwedge_E x_j \neq x_k$ is satisfiable.
- The proof now is identical to the nondeterministic case.
- Sharing equalities is sufficient, because a theory $T_1$ can assume that $x^B \neq y^B$ whenever $x = y$ is not implied by $T_2$ and vice versa.
b + 2 = c, f(read(write(a, b, 3), c-2)) ≠ f(c-b+1)

Arithmetic
b + 2 = c,
v_1 ≡ 3,
v_3 ≡ c-2,
v_5 ≡ c-b+1

Arrays
v_2 ≡ write(a, b, v_1),
v_4 ≡ read(v_2, v_3)

EUF
v_6 ≡ f(v_4),
v_7 ≡ f(v_5),
v_6 ≠ v_7
b + 2 = c, \( f(\text{read}(\text{write}(a, b, 3), c-2)) \neq f(c-b+1) \)

**Arithmetic**

\[ b + 2 = c, \]
\[ v_1 \equiv 3, \]
\[ v_3 \equiv c-2, \]
\[ v_5 \equiv c-b+1 \]

**Arrays**

\[ v_2 \equiv \text{write}(a, b, v_1), \]
\[ v_4 \equiv \text{read}(v_2, v_3) \]

**EUF**

\[ v_6 \equiv f(v_4), \]
\[ v_7 \equiv f(v_5), \]
\[ v_6 \neq v_7 \]

Substituting \( c \)
NO procedure: Example

\[
b + 2 = c, \quad f(\text{read(\text{write}(a,b,3), c-2)}) \neq f(c-b+1)
\]

**Arithmetic**
- \(b + 2 = c,\)
- \(v_1 \equiv 3,\)
- \(v_3 \equiv b,\)
- \(v_5 \equiv 3\)

**Arrays**
- \(v_2 \equiv \text{write}(a, b, v_1),\)
- \(v_4 \equiv \text{read}(v_2, v_3),\)

**EUF**
- \(v_6 \equiv f(v_4),\)
- \(v_7 \equiv f(v_5),\)
- \(v_6 \neq v_7\)

**Propagating**
- \(v_3 = b\)
**NO procedure: Example**

\[ b + 2 = c, \quad f(\text{read}(\text{write}(a, b, 3), c-2)) \neq f(c-b+1) \]

**Arithmetic**
- \[ b + 2 = c, \]
- \[ v_1 \equiv 3, \]
- \[ v_3 \equiv b, \]
- \[ v_5 \equiv 3 \]

**Arrays**
- \[ v_2 \equiv \text{write}(a, b, v_1), \]
- \[ v_4 \equiv \text{read}(v_2, v_3), \]
- \[ v_3 = b \]

**EUF**
- \[ v_6 \equiv f(v_4), \]
- \[ v_7 \equiv f(v_5), \]
- \[ v_6 \neq v_7, \]
- \[ v_3 = b \]

Deducing \[ v_4 = v_1 \]
b + 2 = c, f(read(write(a, b, 3), c-2)) ≠ f(c-b+1)

Arithmetic
b + 2 = c,
v_1 ≜ 3,
v_3 ≜ b,
v_5 ≜ 3

Arrays
v_2 ≜ write(a, b, v_1),
v_4 ≜ read(v_2, v_3),
v_3 = b,
v_4 = v_1

EUF
v_6 ≜ f(v_4),
v_7 ≜ f(v_5),
v_6 ≠ v_7,
v_3 = b

Propagating v_4 = v_1
**NO procedure: Example**

\[ b + 2 = c, \ f(\text{read}(\text{write}(a,b,3), c-2)) \neq f(c-b+1) \]

**Arithmetic**
\[
\begin{align*}
b + 2 &= c, \\
v_1 &= 3, \\
v_3 &= b, \\
v_5 &= 3, \\
v_4 &= v_1
\end{align*}
\]

**Arrays**
\[
\begin{align*}
v_2 &= \text{write}(a, b, v_1), \\
v_4 &= \text{read}(v_2, v_3), \\
v_3 &= b, \\
v_4 &= v_1
\end{align*}
\]

**EUF**
\[
\begin{align*}
v_6 &= f(v_4), \\
v_7 &= f(v_5), \\
v_6 &\neq v_7, \\
v_3 &= b, \\
v_4 &= v_1
\end{align*}
\]

**Propagating**
\[ v_5 = v_1 \]
NO procedure: Example

\[ b + 2 = c, \ f(\text{read(write(a,b,3), c-2)}) \neq f(c-b+1) \]

**Arithmetic**
\[ b + 2 = c, \]
\[ v_1 \equiv 3, \]
\[ v_3 \equiv b, \]
\[ v_5 \equiv 3, \]
\[ v_4 = v_1 \]

**Arrays**
\[ v_2 \equiv \text{write(a, b, v_1)}, \]
\[ v_4 \equiv \text{read(v_2, v_3)}, \]
\[ v_3 = b, \]
\[ v_4 = v_1 \]

**EUF**
\[ v_6 \equiv f(v_4), \]
\[ v_7 \equiv f(v_5), \]
\[ v_6 \neq v_7, \]
\[ v_3 = b, \]
\[ v_4 = v_1, \]
\[ v_5 = v_1 \]

**Congruence:** \( v_6 = v_7 \)
NO procedure: Example

b + 2 = c, f(read(write(a,b,3), c-2)) ≠ f(c-b+1)

Arithmetic
b + 2 = c,
\( v_1 \equiv 3 \),
\( v_3 \equiv b \),
\( v_5 \equiv 3 \),
\( v_4 = v_1 \)

Arrays
\( v_2 \equiv \text{write}(a, b, v_1) \),
\( v_4 \equiv \text{read}(v_2, v_3) \),
\( v_3 = b \),
\( v_4 = v_1 \)

EUF
\( v_6 \equiv f(v_4) \),
\( v_7 \equiv f(v_5) \),
\( v_6 \neq v_7 \),
\( v_3 = b \),
\( v_4 = v_1 \),
\( v_5 = v_1 \),
\( v_6 = v_7 \)

Unsatisfiable
Deterministic procedure may fail for non-convex theories.

\[ 0 \leq a \leq 1, \quad 0 \leq b \leq 1, \quad 0 \leq c \leq 1, \]
\[ f(a) \neq f(b), \]
\[ f(a) \neq f(c), \]
\[ f(b) \neq f(c) \]
Combining Procedures 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 Procedures in Practice

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 $T_i$ and propagate all equalities implied by the candidate model, hedging that other theories will agree.

\[
\text{if } M_i \models T_i \cup \Gamma_i \cup \{u = v\} \text{ then propagate } u = v.
\]

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

\[ x = f(y - 1), \quad f(x) \neq f(y), \quad 0 \leq x \leq 1, \quad 0 \leq y \leq 1 \]

Purifying
Example

\[ x = f(z), \ f(x) \neq f(y), \ 0 \leq x \leq 1, \ 0 \leq y \leq 1, \ z = y - 1 \]
### Example

<table>
<thead>
<tr>
<th>Literals</th>
<th>Eq. Classes</th>
<th>Model</th>
<th>Literals</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x = f(z) )</td>
<td>( {x, f(z)} )</td>
<td>( E(x) = *_1 )</td>
<td>( 0 \leq x \leq 1 )</td>
<td>( A(x) = 0 )</td>
</tr>
<tr>
<td>( f(x) \neq f(y) )</td>
<td>( {y} )</td>
<td>( E(y) = *_2 )</td>
<td>( 0 \leq y \leq 1 )</td>
<td>( A(y) = 0 )</td>
</tr>
<tr>
<td></td>
<td>( {z} )</td>
<td>( E(z) = *_3 )</td>
<td>( z = y - 1 )</td>
<td>( A(z) = -1 )</td>
</tr>
<tr>
<td></td>
<td>( {f(x)} )</td>
<td>( E(f) = {*_1 \leftrightarrow *_4, )</td>
<td>( )</td>
<td>( )</td>
</tr>
<tr>
<td></td>
<td>( {f(y)} )</td>
<td>( )</td>
<td>( *_2 \leftrightarrow *_5, )</td>
<td>( )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( )</td>
<td>( *_3 \leftrightarrow *_1, )</td>
<td>( )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( )</td>
<td>( else \leftrightarrow *_6 } )</td>
<td>( )</td>
</tr>
</tbody>
</table>

Assume \( x = y \)
<table>
<thead>
<tr>
<th>Literals</th>
<th>$T_E$</th>
<th>$T_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = f(z)$</td>
<td>${x, y, f(z)}$</td>
<td>${0 \leq x \leq 1}$, $A(x) = 0$</td>
</tr>
<tr>
<td>$f(x) \neq f(y)$</td>
<td>${z}$</td>
<td>$0 \leq y \leq 1$, $A(y) = 0$</td>
</tr>
<tr>
<td>$x = y$</td>
<td>${f(x), f(y)}$</td>
<td>$z = y - 1$, $A(z) = -1$</td>
</tr>
</tbody>
</table>

Model:
- $E(x) = *_1$
- $E(y) = *_1$
- $E(z) = *_2$
- $E(f) = \{*_1 \mapsto *_3, *_2 \mapsto *_1, \text{else} \mapsto *_4\}$

Unsatisfiable
### Example

<table>
<thead>
<tr>
<th>Literals</th>
<th>Eq. Classes</th>
<th>Model</th>
<th>Literals</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = f(z)$</td>
<td>${x, f(z)}$</td>
<td>$E(x) = *_1$</td>
<td>$0 \leq x \leq 1$</td>
<td>$A(x) = 0$</td>
</tr>
<tr>
<td>$f(x) \neq f(y)$</td>
<td>${y}$</td>
<td>$E(y) = *_2$</td>
<td>$0 \leq y \leq 1$</td>
<td>$A(y) = 0$</td>
</tr>
<tr>
<td>$x \neq y$</td>
<td>${z}$</td>
<td>$E(z) = *_3$</td>
<td>$z = y - 1$</td>
<td>$A(z) = -1$</td>
</tr>
<tr>
<td></td>
<td>${f(x)}$</td>
<td>$E(f) = {*_1 \mapsto *_4,$</td>
<td></td>
<td>$x \neq y$</td>
</tr>
<tr>
<td></td>
<td>${f(y)}$</td>
<td>$*_2 \mapsto *_5,$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$*_3 \mapsto *_1,$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$else \mapsto *_6}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Backtrack, and assert $x \neq y$.

$T_A$ model need to be fixed.
**Example**

<table>
<thead>
<tr>
<th>$\mathcal{T}_E$</th>
<th>$\mathcal{T}_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Literals</strong></td>
<td><strong>Model</strong></td>
</tr>
<tr>
<td>$x = f(z)$</td>
<td>$E(x) = *_1$</td>
</tr>
<tr>
<td>$f(x) \neq f(y)$</td>
<td>$E(y) = *_2$</td>
</tr>
<tr>
<td>$x \neq y$</td>
<td>$E(z) = *_3$</td>
</tr>
<tr>
<td></td>
<td>$E(f) = {*_1 \mapsto *_4,$</td>
</tr>
<tr>
<td></td>
<td>$*_2 \mapsto *_5,$</td>
</tr>
<tr>
<td></td>
<td>$*_3 \mapsto *_1,$</td>
</tr>
<tr>
<td></td>
<td>$else \mapsto *_6}$</td>
</tr>
</tbody>
</table>

Assume $x = z$
### Example

<table>
<thead>
<tr>
<th>$T_E$</th>
<th>$T_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Literals</strong></td>
<td><strong>Eq. Classes</strong></td>
</tr>
<tr>
<td>$x = f(z)$</td>
<td>${x, z, f(x), f(z)}$</td>
</tr>
<tr>
<td>$f(x) \neq f(y)$</td>
<td>${y}$</td>
</tr>
<tr>
<td>$x \neq y$</td>
<td>${f(y)}$</td>
</tr>
<tr>
<td>$x = z$</td>
<td></td>
</tr>
</tbody>
</table>

Satisfiable
Example

<table>
<thead>
<tr>
<th>Literals</th>
<th>Eq. Classes</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = f(z)$</td>
<td>${x, z, f(x), f(z)}$</td>
<td>$E(x) = *_1$</td>
</tr>
<tr>
<td>$f(x) \neq f(y)$</td>
<td>${y}$</td>
<td>$E(y) = *_2$</td>
</tr>
<tr>
<td>$x \neq y$</td>
<td>${f(y)}$</td>
<td>$E(z) = *_1$</td>
</tr>
<tr>
<td>$x = z$</td>
<td></td>
<td>$E(f) = {*_1 \mapsto *_1,$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$*_2 \mapsto *_3,$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$else \mapsto *_4}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Literals</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0 \leq x \leq 1$</td>
<td>$A(x) = 0$</td>
</tr>
<tr>
<td>$0 \leq y \leq 1$</td>
<td>$A(y) = 1$</td>
</tr>
<tr>
<td>$z = y - 1$</td>
<td>$A(z) = 0$</td>
</tr>
<tr>
<td>$x \neq y$</td>
<td>$x = z$</td>
</tr>
</tbody>
</table>

Let $h$ be the bijection between $|E|$ and $|A|$.

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

\[ h = \{ *_1 \mapsto 0, *_2 \mapsto 1, *_3 \mapsto -1, *_4 \mapsto 2, \ldots \} \]
Sometimes $M(x) = M(y)$ by accident.

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

**Model mutation:** diversify the current model.
Model mutation without pivoting

For each non basic variable $x_j$ compute $[L_j, U_j]$

Each row containing $x_j$ enforces a limit on how much it can be increased and/or decreased without violating the bounds of the basic variable in the row.
We say a variable is fixed if the lower and upper bound are the same.

\[ 1 \leq x \leq 1 \]

A polynomial \( P \) is fixed if all its variables are fixed.

Given a fixed polynomial \( P \) of the forma \( 2x_1 + x_2 \), we use \( M(P) \) to denote \( 2M(x_1) + M(x_2) \)
Opportunistic Equality Propagation

FixedEq
\[ l_i \leq x_i \leq u_i, \quad l_j \leq x_j \leq u_j \implies x_i = x_j \quad \text{if} \quad l_i = u_i = l_j = u_j \]

EqRow
\[ x_i = x_j + P \quad \implies x_i = x_j \quad \text{if} \quad P \text{ is fixed, and } |M(P)| = 0 \]

EqOffsetRows
\[ x_i = x_k + P_1 \quad x_j = x_k + P_2 \quad \implies x_i = x_j \quad \text{if} \quad \begin{cases} P_1 \text{ and } P_2 \text{ are fixed, and} \\ |M(P_1)| = |M(P_2)| \end{cases} \]

EqRows
\[ x_i = P + P_1 \quad x_j = P + P_2 \quad \implies x_i = x_j \quad \text{if} \quad \begin{cases} P_1 \text{ and } P_2 \text{ are fixed, and} \\ |M(P_1)| = |M(P_2)| \end{cases} \]
Bit-vector theory is not stably-infinite.

How can we support it?

**Solution:** add a predicate $is-bv(x)$ to the bit-vector theory (intuition: $is-bv(x)$ is true iff $x$ is a bitvector).

The result of the bit-vector operation $op(x, y)$ is not specified if $\neg is-bv(x)$ or $\neg is-bv(y)$.

The new bit-vector theory is stably-infinite.
A reduction function reduces the satisfiability problem for a complex theory into the satisfiability problem of a simpler theory.

Ackermannization is a reduction function.
Reduction Functions

Theory of commutative functions.

- $\forall x, y. f(x, y) = f(y, x)$
- Reduction to EUF
- For every $f(a, b)$ in $\phi$, do $\phi := \phi \land f(a, b) = f(b, a)$. 
Verifying Compilers

Annotated Program  
→  Verification Condition $F$

pre/post conditions  
invariants  
and other annotations
Verification conditions: Structure

∀ Axioms (non-ground)

Control & Data Flow

BIG and-or tree (ground)
Main Challenge

- Quantifiers, quantifiers, quantifiers, ...
- Modeling the runtime

\[ \forall h, o, f: \]
\[ \text{IsHeap}(h) \land o \neq \text{null} \land \text{read}(h, o, \text{alloc}) = t \]
\[ \Rightarrow \]
\[ \text{read}(h, o, f) = \text{null} \lor \text{read}(h, \text{read}(h, o, f), \text{alloc}) = t \]
Main Challenge

- Quantifiers, quantifiers, quantifiers, ...
- Modeling the runtime
- Frame axioms

∀ o, f:

\[ o \neq \text{null} \land \text{read}(h_0, o, \text{alloc}) = t \implies \]
\[ \text{read}(h_1, o, f) = \text{read}(h_0, o, f) \lor (o, f) \in M \]
Main Challenge

- Quantifiers, quantifiers, quantifiers, ...
- Modeling the runtime
- Frame axioms
- User provided assertions
  \( \forall \ i, j: \ i \leq j \implies \text{read}(a,i) \leq \text{read}(b,j) \)
Main Challenge

- Quantifiers, quantifiers, quantifiers, ...
- Modeling the runtime
- Frame axioms
- User provided assertions
- Theories
  - $\forall x: p(x,x)$
  - $\forall x,y,z: p(x,y), p(y,z) \implies p(x,z)$
  - $\forall x,y: p(x,y), p(y,x) \implies x = y$
Main Challenge

- Quantifiers, quantifiers, quantifiers, ...
- Modeling the runtime
- Frame axioms
- User provided assertions
- Theories
- Solver must be fast in satisfiable instances.

We want to find bugs!
Some statistics

- Grand challenge: Microsoft Hypervisor
- 70k lines of dense C code
- VCs have several Mb
- Thousands of non ground clauses
- Developers are willing to wait at most 5 min per VC
Many Approaches

- Heuristic quantifier instantiation
- Combining SMT with Saturation provers
- Complete quantifier instantiation
- Decidable fragments
- Model based quantifier instantiation
SMT solvers use **heuristic quantifier instantiation**.

**E-matching** (matching modulo equalities).

**Example:**
\[
\forall x: f(g(x)) = x \{ f(g(x)) \}
\]

\[a = g(b),\]
\[b = c,\]
\[f(a) \neq c\]
SMT solvers use **heuristic quantifier instantiation**.

**E-matching** (matching modulo equalities).

**Example:**
\[ \forall x: f(g(x)) = x \{ f(g(x)) \} \]

\[ a = g(b), \]

\[ b = c, \]

\[ f(a) \neq c \]

Equalities and ground terms come from the partial model \( M \)
Integrates smoothly with DPLL.

Software verification problems are **big & shallow**.

Decides useful theories:

- Arrays
- Partial orders
- ...
Efficient E-matching

- E-matching is NP-Hard.
- In practice

<table>
<thead>
<tr>
<th>Problem</th>
<th>Indexing Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast retrieval</td>
<td>E-matching code trees</td>
</tr>
<tr>
<td>Incremental E-Matching</td>
<td>Inverted path index</td>
</tr>
</tbody>
</table>
Trigger:
\[ f(x_1, g(x_1, a), h(x_2), b) \]

Instructions:
1. `init(f, 2)`
2. `check(r4, b, 3)`
3. `bind(r2, g, r5, 4)`
4. `compare(r1, r5, 5)`
5. `check(r6, a, 6)`
6. `bind(r3, h, r7, 7)`
7. `yield(r1, r7)`

Similar triggers share several instructions.

Combine code sequences in a code tree.
E-matching: Limitations

- E-matching needs **ground seeds**.
  - \( \forall x: p(x) \),
  - \( \forall x: \text{not } p(x) \)
E-matching: Limitations

- E-matching needs **ground seeds**.
- Bad user provided triggers:
  \[ \forall x: f(g(x)) = x \{ f(g(x)) \} \]
  
  \[ g(a) = c, \]
  \[ g(b) = c, \]
  \[ a \neq b \]

Trigger is too restrictive
E-matching: Limitations

- E-matching needs ground seeds.
- Bad user provided triggers:
  \[ \forall x: f(g(x)) = x \{ g(x) \} \]
  
  \[ g(a) = c, \]
  
  \[ g(b) = c, \]
  
  \[ a \neq b \]

More “liberal” trigger
E-matching: Limitations

- E-matching needs **ground seeds**.
- Bad user provided triggers:
  \[ \forall x: f(g(x)) = x \{ g(x) \} \]
  
  \[ g(a) = c, \]
  
  \[ g(b) = c, \]
  
  \[ a \neq b, \]
  
  \[ f(g(a)) = a, \]
  
  \[ f(g(b)) = b \]
E-matching: Limitations

- E-matching needs ground seeds.
- Bad user provided triggers.
- It is not refutationally complete.

False positives
Tight integration: DPLL + Saturation solver.
Inference rule:

\[ \begin{array}{c}
C_1 & \ldots & C_n \\
\hline \\
C
\end{array} \]

DPLL(Γ) is parametric.

Examples:
- Resolution
- Superposition calculus
- ...

\[ \text{DPLL}(\Gamma) \]
$DPLL(\Gamma)$

- Partial model
- Set of clauses

M | F
p(a) \mid p(a) \lor q(a), \ \forall x: \neg p(x) \lor r(x), \ \forall x: p(x) \lor s(x)
\[
p(a) \mid p(a) \lor q(a), \neg p(x) \lor r(x), p(x) \lor s(x)
\]
\[ p(a) \mid p(a) \lor q(a), \neg p(x) \lor r(x), p(x) \lor s(x) \]

Resolution

\[ p(a) \mid p(a) \lor q(a), \neg p(x) \lor r(x), p(x) \lor s(x), r(x) \lor s(x) \]
Using ground atoms from $M$:

$$M \mid F$$

Main issue: backtracking.

Hypothetical clauses:

$$H \triangleright C$$

Track literals from $M$ used to derive $C$
\[ p(a) \mid p(a) \lor q(a), \neg p(x) \lor r(x) \]

\[ r(a) \]

\[ p(a), \neg p(x) \lor r(x) \]

\[ p(a) \mid p(a) \lor q(a), \neg p(x) \lor r(x), p(a) \triangleright r(a) \]
DPLL(Γ): Backtracking

p(a), r(a) | p(a) ∨ q(a), ¬p(a) ∨ ¬r(a), p(a) ▽ r(a), ...

DPLL(\(\Gamma\)): Backtracking

\[ p(a), r(a) \mid p(a) \lor q(a), \neg p(a) \lor \neg r(a), \quad p(a) \not\rightarrow r(a), \ldots \]

\[ \neg p(a) \mid p(a) \lor q(a), \neg p(a) \lor \neg r(a), \ldots \]

\(p(a)\) is removed from \(M\)
Saturation solver ignores non-unit ground clauses.

\[ p(a) \lor \neg p(a) \lor q(a), \neg p(x) \lor r(x) \]
Saturation solver ignores non-unit ground clauses.

It is still refutationally complete if:

- $\Gamma$ has the reduction property.
Saturation solver ignores non-unit ground clauses.

It is still refutationally complete if:

- $\Gamma$ has the reduction property.
DPLL(Γ): Problem

- Interpreted symbols
  \( \neg (f(a) > 2), \ f(x) > 5 \)

- It is refutationally complete if
  - Interpreted symbols only occur in ground clauses
  - Non ground clauses are variable inactive
  - “Good” ordering is used
\[ \forall x_1, x_2: \neg p(x_1, x_2) \lor f(x_1) = f(x_2) + 1, \]
\[ p(a, b), a < b + 1 \]
\neg p(x_1, x_2) \lor f(x_1) = f(x_2) + 1,
\neg p(a,b), a < b + 1
Variables appear only as arguments of uninterpreted symbols.

\[ f(g(x_1) + a) < g(x_1) \lor h(f(x_1), x_2) = 0 \]

\[ f(x_1 + x_2) \leq f(x_1) + f(x_2) \]
Basic Idea

Given a set of formulas \( F \), build an equisatisfiable set of quantifier-free formulas \( F^* \)

“Domain” of \( f \) is the set of ground terms \( A_f \)
\( t \in A_f \) if there is a ground term \( f(t) \)

Suppose
1. We have a clause \( C[f(x)] \) containing \( f(x) \).
2. We have \( f(t) \).

\( \Rightarrow \)
Instantiate \( x \) with \( t \): \( C[f(t)] \).
<table>
<thead>
<tr>
<th>( F )</th>
<th>( F^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g(x_1, x_2) = 0 \lor h(x_2) = 0, )</td>
<td></td>
</tr>
<tr>
<td>( g(f(x_1),b) + 1 \leq f(x_1), )</td>
<td></td>
</tr>
<tr>
<td>( h(c) = 1, )</td>
<td></td>
</tr>
<tr>
<td>( f(a) = 0 )</td>
<td></td>
</tr>
</tbody>
</table>
Example

F

\[ g(x_1, x_2) = 0 \lor h(x_2) = 0, \]
\[ g(f(x_1), b) + 1 \leq f(x_1), \]
\[ h(c) = 1, \]
\[ f(a) = 0 \]

F*

\[ h(c) = 1, \]
\[ f(a) = 0 \]

Copy quantifier-free formulas

"Domains":

\[ A_f: \{ a \} \]
\[ A_g: \{ \} \]
\[ A_h: \{ c \} \]
Example

F
\[ g(x_1, x_2) = 0 \lor h(x_2) = 0, \]
\[ g(f(x_1), b) + 1 \leq f(x_1), \]
\[ h(c) = 1, \]
\[ f(a) = 0 \]

F*
\[ h(c) = 1, \]
\[ f(a) = 0, \]

“Domains”:
\[ A_f : \{ a \} \]
\[ A_g : \{ \} \]
\[ A_h : \{ c \} \]
Example

\[ g(x_1, x_2) = 0 \lor h(x_2) = 0, \]
\[ g(f(x_1), b) + 1 \leq f(x_1), \]
\[ h(c) = 1, \]
\[ f(a) = 0 \]

F

\[ h(c) = 1, \]
\[ f(a) = 0, \]
\[ g(f(a), b) + 1 \leq f(a) \]

F*

“Domains”:
\[ A_f : \{ a \} \]
\[ A_g : \{ [f(a), b] \} \]
\[ A_h : \{ c \} \]
Example

\( g(x_1, x_2) = 0 \lor h(x_2) = 0, \)
\( g(f(x_1), b) + 1 \leq f(x_1), \)
\( h(c) = 1, \)
\( f(a) = 0 \)

\( F \)

\( F^* \)

“Domains”:

\( A_f : \{ a \} \)
\( A_g : \{ [f(a), b] \} \)
\( A_h : \{ c \} \)
Example

\[ g(x_1, x_2) = 0 \lor h(x_2) = 0, \]
\[ g(f(x_1), b) + 1 \leq f(x_1), \]
\[ h(c) = 1, \]
\[ f(a) = 0 \]

\[ F^* \]
\[ h(c) = 1, \]
\[ f(a) = 0, \]
\[ g(f(a), b) + 1 \leq f(a), \]
\[ g(f(a), b) = 0 \lor h(b) = 0 \]

“Domains”:

\[ A_f : \{ a \} \]
\[ A_g : \{ [f(a), b] \} \]
\[ A_h : \{ c, b \} \]
\[ g(x_1, x_2) = 0 \lor h(x_2) = 0, \]
\[ g(f(x_1), b) + 1 \leq f(x_1), \]
\[ h(c) = 1, \]
\[ f(a) = 0 \]

\[ h(c) = 1, \]
\[ f(a) = 0, \]
\[ g(f(a), b) + 1 \leq f(a), \]
\[ g(f(a), b) = 0 \lor h(b) = 0 \]

“Domains”:
\[ A_f : \{ a \} \]
\[ A_g : \{ [f(a), b] \} \]
\[ A_h : \{ c, b \} \]
Example

\[
g(x_1, x_2) = 0 \lor h(x_2) = 0, \\
g(f(x_1), b) + 1 \leq f(x_1), \\
h(c) = 1, \\
f(a) = 0
\]

\[
F^*
\]

\[
h(c) = 1, \\
f(a) = 0, \\
g(f(a), b) + 1 \leq f(a), \\
g(f(a), b) = 0 \lor h(b) = 0, \\
g(f(a), c) = 0 \lor h(c) = 0
\]

“Domains”:

\[
A_f : \{ a \} \\
A_g : \{ [f(a), b], [f(a), c] \} \\
A_h : \{ c, b \}
\]
Example

\[ g(x_1, x_2) = 0 \lor h(x_2) = 0, \]
\[ g(f(x_1), b) + 1 \leq f(x_1), \]
\[ h(c) = 1, \]
\[ f(a) = 0 \]

\[ F^* \]
\[ h(c) = 1, \]
\[ f(a) = 0, \]
\[ g(f(a), b) + 1 \leq f(a), \]
\[ g(f(a), b) = 0 \lor h(b) = 0, \]
\[ g(f(a), c) = 0 \lor h(c) = 0 \]

\[ M \]
\[ a \rightarrow 2, \ b \rightarrow 2, \ c \rightarrow 3 \]
\[ f \rightarrow \{2 \rightarrow 0, \ldots\} \]
\[ h \rightarrow \{2 \rightarrow 0, \ 3 \rightarrow 1, \ldots\} \]
\[ g \rightarrow \{[0,2] \rightarrow -1, \ [0,3] \rightarrow 0, \ldots\} \]
Basic Idea (cont.)

Given a model $M$ for $F^*$,
Build a model $M^\pi$ for $F$

Define a projection function $\pi_f$ s.t.
range of $\pi_f$ is $M(A_f)$, and
$\pi_f(v) = v$ if $v \in M(A_f)$

Then,
$M^\pi(f)(v) = M(f)(\pi_f(v))$
Basic Idea (cont.)

\[ M(A_f) \rightarrow M(f(A_f)) \]

\[ M(\pi f) \rightarrow M(f) \rightarrow M(f(A_f)) \]
Given a model $M$ for $F^*$, Build a model $M^\pi$ for $F$

In our example, we have: $h(b)$ and $h(c)$

$\rightarrow A_h = \{ b, c \}$, and $M(A_h) = \{ 2, 3 \}$

$$\pi_h = \{ 2 \rightarrow 2, 3 \rightarrow 3, \text{else} \rightarrow 3 \}$$

$M(h)$

$\{ 2 \rightarrow 0, 3 \rightarrow 1, \ldots \}$

$M^\pi(h)$

$\{ 2 \rightarrow 0, 3 \rightarrow 1, \text{else} \rightarrow 1 \}$

$M^\pi(h) = \lambda x. \text{if}(x=2, 0, 1)$
Example

$F$
\begin{align*}
g(x_1, x_2) &= 0 \lor h(x_2) = 0, \\
g(f(x_1), b) + 1 &\leq f(x_1), \\
h(c) &= 1, \\
f(a) &= 0
\end{align*}

$F^*$
\begin{align*}
h(c) &= 1, \\
f(a) &= 0, \\
g(f(a), b) + 1 &\leq f(a), \\
g(f(a), b) &= 0 \lor h(b) = 0, \\
g(f(a), c) &= 0 \lor h(c) = 0
\end{align*}

$M^\pi$
\begin{align*}
a &\rightarrow 2, \ b \rightarrow 2, \ c \rightarrow 3 \\
f &\rightarrow \lambda x. \ 2 \\
h &\rightarrow \lambda x. \ \text{if}(x=2, \ 0, \ 1) \\
g &\rightarrow \lambda x,y. \ \text{if}(x=0 \land y=2, -1, \ 0)
\end{align*}

$M$
\begin{align*}
a &\rightarrow 2, \ b \rightarrow 2, \ c \rightarrow 3 \\
f &\rightarrow \{ \ 2 \rightarrow 0, \ \ldots \} \\
h &\rightarrow \{ \ 2 \rightarrow 0, \ 3 \rightarrow 1, \ \ldots \} \\
g &\rightarrow \{ \ [0,2] \rightarrow -1, \ [0,3] \rightarrow 0, \ \ldots \}
Example: Model Checking

\( M^\pi \)

a \rightarrow 2, \ b \rightarrow 2, \ c \rightarrow 3
f \rightarrow \lambda x. \ 2
h \rightarrow \lambda x. \ \text{if}(x=2, \ 0, \ 1)
g \rightarrow \lambda x,y. \ \text{if}(x=0 \land y=2,-1, \ 0)

Does \( M^\pi \) satisfies?
\( \forall x_1, x_2: g(x_1, x_2) = 0 \lor h(x_2) = 0 \)

\( \forall x_1, x_2: \text{if}(x_1=0 \land x_2=2,-1,0) = 0 \lor \text{if}(x_2=2,0,1) = 0 \) is valid

\( \exists x_1, x_2: \text{if}(x_1=0 \land x_2=2,-1,0) \neq 0 \land \text{if}(x_2=2,0,1) \neq 0 \) is unsat

\( \text{if}(s_1=0 \land s_2=2,-1,0) \neq 0 \land \text{if}(s_2=2,0,1) \neq 0 \) is unsat
Why does it work?

Suppose \( M^\pi \) does not satisfy \( C[f(x)] \).

Then for some value \( v \),
\( M^\pi \{x \to v\} \) falsifies \( C[f(x)] \).

\( M^\pi \{x \to \pi_f(v)\} \) also falsifies \( C[f(x)] \).

But, there is a term \( t \in A_f \) s.t. \( M(t) = \pi_f(v) \)
Moreover, we instantiated \( C[f(x)] \) with \( t \).

So, \( M \) must not satisfy \( C[f(t)] \).
Contradiction: \( M \) is a model for \( F^* \).
Refinement 1: Lazy construction

- $F^*$ may be very big (or infinite).
- Lazy-construction
  - Build $F^*$ incrementally, $F^*$ is the limit of the sequence $F^0 \subset F^1 \subset \ldots \subset F^k \subset \ldots$
  - If $F^k$ is unsat then $F$ is unsat.
  - If $F^k$ is sat, then build (candidate) $M^\pi$
  - If $M^\pi$ satisfies all quantifiers in $F$ then return sat.
Suppose $M^\pi$ does not satisfy a clause $C[f(x)]$ in $F$.

Add an instance $C[f(t)]$ which “blocks” this spurious model.
Issue: how to find $t$?

Use model checking,
and the “inverse” mapping $\pi_f^{-1}$ from values to terms (in $A_f$).
$\pi_f^{-1}(v) = t$ if $M^\pi(t) = \pi_f(v)$
Model-based instantiation: Example

\[ \forall x_1: f(x_1) < 0, \]
\[ f(a) = 1, \]
\[ f(b) = -1 \]

\[ f(a) = 1, \]
\[ f(b) = -1 \]

\[ a \rightarrow 2, b \rightarrow 3 \]
\[ f \rightarrow \lambda x. \text{if}(x = 2, 1, -1) \]

Model Checking \[ \forall x_1: f(x_1) < 0 \]
not \[ \text{if}(s_1 = 2, 1, -1) < 0 \]

unsat

\[ f(a) = 1, \]
\[ f(b) = -1 \]
\[ f(a) < 0 \]

\[ s_1 \rightarrow 2 \]
\[ \pi_f^{-1}(2) = a \]
Is our procedure refutationally complete?

FOL Compactness

A set of sentences is unsatisfiable iff it contains an unsatisfiable finite subset.

A theory $T$ is a set of sentences, then apply compactness to $F^* \cup T$
Infinite $F^*$: Example

$F$

$\forall x_1: f(x_1) < f(f(x_1))$, 
$\forall x_1: f(x_1) < a$, 
$1 < f(0)$. 

$F^*$

$f(0) < f(f(0))$, $f(f(0)) < f(f(f(0)))$, ..., 
$f(0) < a$, $f(f(0)) < a$, ..., 
$1 < f(0)$

Every finite subset of $F^*$ is satisfiable.

Unsatisfiable
Theory of linear arithmetic $T_Z$ is the set of all first-order sentences that are true in the standard structure $Z$.

$T_Z$ has non-standard models.

F and $F^*$ are satisfiable in a non-standard model.

Alternative: a theory is a class of structures.

Compactness does not hold.

F and $F^*$ are still equisatisfiable.
Given a clause $C_k[x_1, ..., x_n]$

Let

$S_{k,i}$ be the set of ground terms used to instantiate $x_i$ in clause $C_k[x_1, ..., x_n]$

How to characterize $S_{k,i}$?

<table>
<thead>
<tr>
<th>$F$</th>
<th>$\Delta_F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$-th argument of $f$ in $C_k$</td>
<td>system of set constraints</td>
</tr>
<tr>
<td>a ground term $t$</td>
<td>$t \in A_{f,j}$</td>
</tr>
<tr>
<td>$t[x_1, ..., x_n]$</td>
<td>$t[S_{k,1}, ..., S_{k,n}] \subseteq A_{f,j}$</td>
</tr>
<tr>
<td>$x_i$</td>
<td>$S_{k,i} = A_{f,j}$</td>
</tr>
</tbody>
</table>
$\Delta_F$: Example

$F$

\[ g(x_1, x_2) = 0 \lor h(x_2) = 0, \]
\[ g(f(x_1), b) + 1 \leq f(x_1), \]
\[ h(c) = 1, \]
\[ f(a) = 0 \]

$\Delta_F$

\[ S_{1,1} = A_{g,1}, S_{1,2} = A_{g,2}, S_{1,2} = A_{h,1} \]
\[ S_{2,1} = A_{f,1}, f(S_{2,1}) \subseteq A_{g,1}, b \in A_{g,2} \]
\[ c \in A_{h,1} \]
\[ a \in A_{f,1} \]

$\Delta_F$: least solution

\[ S_{1,1} = \{ f(a) \}, S_{1,2} = \{ b, c \} \]
\[ S_{2,1} = \{ a \} \]

Use $\Delta_F$ to generate $F^*$
**Complexity**

- $\Delta_F$ is **stratified** then the least solution (and $F^*$) is finite

<table>
<thead>
<tr>
<th>$t[S_{k,1}, ..., S_{k,n}] \subseteq A_{f,j}$</th>
<th>level($S_{k,i}$) $&lt;$ level($A_{f,j}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{k,i} = A_{f,j}$</td>
<td>level($S_{k,i}$) $=$ level($A_{f,j}$)</td>
</tr>
</tbody>
</table>

- New decidable fragment: NEXPTIME-Hard.
- The least solution of $\Delta_F$ is exponential in the worst case.
  - $a \in S_1, b \in S_1, f_1(S_1, S_1) \subseteq S_2, ..., f_n(S_n, S_n) \subseteq S_{n+1}$
- $F^*$ can be doubly exponential in the size of $F$. 
**Extensions**

Arithmetical literals: $\pi_f$ must be monotonic.

<table>
<thead>
<tr>
<th>Literal of $C_k$</th>
<th>$\Delta_F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg(x_i \leq x_j)$</td>
<td>$S_{k,i} = S_{k,j}$</td>
</tr>
<tr>
<td>$\neg(x_i \leq t), \neg(t \leq x_i)$</td>
<td>$t \in S_{k,i}$</td>
</tr>
<tr>
<td>$x_i = t$</td>
<td>${t+1, t-1} \subseteq S_{k,i}$</td>
</tr>
</tbody>
</table>

Offsets:

<table>
<thead>
<tr>
<th>j-th argument of f in $C_k$</th>
<th>$\Delta_F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_i + r$</td>
<td>$S_{k,i} + r \subseteq A_{f,j}$</td>
</tr>
<tr>
<td></td>
<td>$A_{f,j} + (-r) \subseteq S_{k,i}$</td>
</tr>
</tbody>
</table>
Extensions: Example

Shifting

\( \neg (0 \leq x_1) \lor \neg (x_1 \leq n) \lor f(x_1) = g(x_1 + 2) \)
More Extensions

- Many-sorted logic
- Pseudo-Macros

\[ 0 \leq g(x_1) \lor f(g(x_1)) = x_1, \]
\[ 0 \leq g(x_1) \lor h(g(x_1)) = 2x_1, \]
\[ g(a) < 0 \]
Conclusion

Powerful, mature, and versatile tools like SMT solvers can now be exploited in very useful ways.

The construction and application of satisfiability procedures is an active research area with exciting challenges.

SMT is hot at Microsoft.

Z3 is a new SMT solver.

Main applications:

- Test-case generation.
- Verifying compiler.
- Model Checking & Predicate Abstraction.
Books

- Bradley & Manna: The Calculus of Computation
- Kroening & Strichman: Decision Procedures, An Algorithmic Point of View
- Chapter in the Handbook of Satisfiability
Z3:
http://research.microsoft.com/projects/z3

http://research.microsoft.com/~leonardo

- Slides & Papers

http://www.smtlib.org

http://www.smtcomp.org
References


References


References


References


