Example: \( \{ x \mapsto 1, y \mapsto -7 \} \Delta \{ x \mapsto 1, y \mapsto -7 \} \)

(3) States:

\[ \Delta \subseteq ((\text{Vars} \rightarrow \mathbb{Z}) \times (\mathbb{N} \rightarrow \mathbb{Z})) \times (\text{Vars} \rightarrow \mathbb{Z}^\top) \perp \]

\((\rho, \mu) \Delta D \quad \text{iff} \quad \rho \Delta D \]

Concretization:

\[ \gamma D = \begin{cases} 
\emptyset & \text{if} \quad D = \perp \\
\{(\rho, \mu) \mid \forall x : (\rho x) \Delta (D x)\} & \text{otherwise}
\end{cases} \]
We show:

\((\ast)\) If \(s \Delta D\) and \([\pi] s\) is defined, then:

\(((\pi) s) \Delta ([\pi]^\# D)\)
The abstract semantics simulates the concrete semantics  :-) 
In particular:

$$\llbracket \pi \rrbracket s \in \gamma (\llbracket \pi \rrbracket^\# D)$$
The abstract semantics simulates the concrete semantics :-)

In particular:

\[ \pi \] \in \gamma (\pi^\# D) 

In practice, this means, e.g., that \( D x = -7 \) implies:

\[ \rho' x = -7 \quad \text{for all} \quad \rho' \in \gamma D \]

\[ \implies \quad \rho_1 x = -7 \quad \text{for} \quad (\rho_1, _) = \pi \]
To prove (\(\ast\)), we show for every edge \(k\) :

\[
\begin{align*}
\Delta & \quad [k] \\
D & \quad [k]'\#
\end{align*}
\]

Then (\(\ast\)) follows by induction :-)

290
To prove (**) we show for every expression $e$:

(***) $(\llbracket e \rrbracket \rho) \Delta (\llbracket e \rrbracket^\# D)$ whenever $\rho \Delta D$
To prove (**) we show for every expression $e$:

\[(***) \quad ([e] \rho) \triangle ([e]^\# D) \quad \text{whenever} \quad \rho \triangle D\]

To prove (***) we show for every operator $\square$:

\[(\square y) \triangle (\square^\# y^\#) \quad \text{whenever} \quad x \triangle x^\# \land y \triangle y^\#\]
To prove (**) we show for every expression $e$:

$$(***) \quad ([e] \rho) \Delta ([e]^{\#} D) \quad \text{whenever} \quad \rho \Delta D$$

To prove (***) we show for every operator $\Box$:

$$(\Box y) \Delta (\Box^{\#} y^{\#}) \quad \text{whenever} \quad x \Delta x^{\#} \land y \Delta y^{\#}$$

This precisely was how we have defined the operators $\Box^{\#}$ :-)}
Now, (***) is proved by case distinction on the edge labels \( \text{lab} \).

Let \( s = (\rho, \mu) \triangle D \). In particular, \( \bot \neq D : \text{Vars} \rightarrow \mathbb{Z}^{\top} \)

Case \( x = e; \):

\[
\begin{align*}
\rho_1 &= \rho \oplus \{ x \mapsto [e] \rho \} \quad \mu_1 = \mu \\
D_1 &= D \oplus \{ x \mapsto [e]^\# D \}
\end{align*}
\]

\( \implies (\rho_1, \mu_1) \triangle D_1 \)
Case \( x = M[e]; \):

\[
\begin{align*}
\rho_1 &= \rho \oplus \{ x \mapsto \mu (\llbracket e \rrbracket^\# \rho) \} \\
D_1 &= D \oplus \{ x \mapsto \top \}
\end{align*}
\]

\[\implies (\rho_1, \mu_1) \Delta D_1\]

Case \( M[e_1] = e_2; \):

\[
\begin{align*}
\rho_1 &= \rho \\
\mu_1 &= \mu \oplus \{ \llbracket e_1 \rrbracket^\# \rho \mapsto \llbracket e_2 \rrbracket^\# \rho \}
\end{align*}
\]

\[\implies (\rho_1, \mu_1) \Delta D_1\]
Case \( \text{Neg}(e) \):

\[(\rho_1, \mu_1) = s \quad \text{where:} \]

\[
0 = [e] \rho \\
\Delta [e]^\# D \\
\Rightarrow 0 \subseteq [e]^\# D \\
\Rightarrow \perp \neq D_1 = D \\
\Rightarrow (\rho_1, \mu_1) \Delta D_1
\]
Case $\text{Pos}(e)$: 

$$(\rho_1, \mu_1) = s$$

where:

$$0 \neq \mathbf{[e]} \rho$$

$$\Delta \mathbf{[e]}^\# D$$

$$\implies 0 \neq \mathbf{[e]}^\# D$$

$$\implies \bot \neq D_1 = D$$

$$\implies (\rho_1, \mu_1) \Delta D_1$$

:-)
We conclude: The assertion \((*)\) is true \(:-))\)

The MOP-Solution:

\[
D^*[v] = \bigcup\{[[\pi]]^\# \mid D^*_{\top} \mid \pi : start \rightarrow^* v\}
\]

where \(D^*_{\top} x = \top\) \((x \in Vars)\).
**We conclude:** The assertion \((\ast)\) is true \(:-)\)

The MOP-Solution:

\[
\mathcal{D}^*[v] = \bigsqcup \{ [[\pi]] \# D_{\top} \mid \pi : \text{start} \rightarrow^* v\}
\]

where \(D_{\top} x = \top \quad (x \in Vars)\).

By \((\ast)\), we have for all initial states \(s\) and all program executions \(\pi\) which reach \(v\):

\[
([[\pi]] s) \Delta (\mathcal{D}^*[v])
\]
We conclude: The assertion \((\ast)\) is true \(\rightarrow:\)}

The MOP-Solution

\[
D^*[v] = \bigcup \{ [[\pi]]^\# D_\top | \pi : \text{start} \rightarrow^* v \}
\]

where \(D_\top x = \top \quad (x \in \text{Vars})\).

By \((\ast)\), we have for all initial states \(s\) and all program executions \(\pi\) which reach \(v\):

\[
([\pi] s) \Delta (D^*[v])
\]

In order to approximate the MOP, we use our constraint system \(\rightarrow:\)}\)
Example:

\[
x = 10;
\]

\[
y = 1;
\]

\[
M[R] = y;
\]

\[
y = x \ast y;
\]

\[
x = x - 1;
\]
Example:

\[ x = 10; \]
\[ y = 1; \]
\[ M[R] = y; \]
\[ x = x - 1; \]

Neg(\(x > 1\))

Pos(\(x > 1\))

\[ y = x \cdot y; \]

\[
\begin{array}{|c|c|c|}
\hline
& x & y \\
\hline
0 & \top & \top \\
1 & 10 & \top \\
2 & 10 & 1 \\
3 & 10 & 1 \\
4 & 10 & 10 \\
5 & 9 & 10 \\
6 & \bot & \bot \\
7 & \bot & \bot \\
\hline
\end{array}
\]
Example:

\[
x = 10;\]
\[
y = 1;\]
\[
M[R] = y;\]
\[
y = x \times y;\]
\[
x = x - 1;\]
\[
{\text{Neg}}(x > 1)\]
\[
{\text{Pos}}(x > 1)\]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>⊥</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>⊥</td>
<td></td>
</tr>
</tbody>
</table>
Example:

\[ x = 10; \]

\[ y = 1; \]

\[ M[R] = y; \]

\[ y = x \cdot y; \]

\[ x = x - 1; \]

\[ \text{Neg}(x > 1) \]

\[ \text{Pos}(x > 1) \]

<table>
<thead>
<tr>
<th></th>
<th>( x )</th>
<th>( y )</th>
<th>( x )</th>
<th>( y )</th>
<th>( x )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( \top )</td>
<td>( \top )</td>
<td>( \top )</td>
<td>( \top )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>( \top )</td>
<td>10</td>
<td>( \top )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>1</td>
<td>( \top )</td>
<td>( \top )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1</td>
<td>( \top )</td>
<td>( \top )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>10</td>
<td>( \top )</td>
<td>( \top )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>10</td>
<td>( \top )</td>
<td>( \top )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>( \bot )</td>
<td>( \top )</td>
<td>( \top )</td>
<td>( \top )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>( \bot )</td>
<td>( \top )</td>
<td>( \top )</td>
<td>( \top )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\text{dito}
Conclusion:

Although we compute with concrete values, we fail to compute everything :-(

The fixpoint iteration, at least, is guaranteed to terminate:

For \( n \) program points and \( m \) variables, we maximally need:
\[
n \cdot (m + 1) \text{ rounds} \quad :-)
\]

Caveat:

The effects of edge are not distributive !!!
Counter Example: \[ f = \left[ x = x + y; \right] \] 

Let \[ D_1 = \{ x \mapsto 2, y \mapsto 3 \} \] 
\[ D_2 = \{ x \mapsto 3, y \mapsto 2 \} \] 

Dann \[ f D_1 \sqcup f D_2 = \{ x \mapsto 5, y \mapsto 3 \} \sqcup \{ x \mapsto 5, y \mapsto 2 \} \] 
\[ = \{ x \mapsto 5, y \mapsto \top \} \] 
\[ \neq \{ x \mapsto \top, y \mapsto \top \} \] 
\[ = f \{ x \mapsto \top, y \mapsto \top \} \] 
\[ = f (D_1 \sqcup D_2) \] 
\[ :-(( \)
We conclude:

The least solution $\mathcal{D}$ of the constraint system in general yields only an **upper approximation** of the MOP, i.e.,

$$\mathcal{D}^*[v] \sqsubseteq \mathcal{D}[v]$$
We conclude:

The least solution $D$ of the constraint system in general yields only an upper approximation of the MOP, i.e.,

$$D^*[v] \subseteq D[v]$$

As an upper approximation, $D[v]$ nonetheless describes the result of every program execution $\pi$ which reaches $v$:

$$([\pi](\rho, \mu)) \triangle (D[v])$$

whenever $[\pi](\rho, \mu)$ is defined ;-))
Transformation 4: Removal of Dead Code

\[ D[u] = \bot \]

\[ [\text{lab}]^\#(D[u]) = \bot \]
Transformation 4 (cont.): Removal of Dead Code

\[ \bot \neq D[u] = D \]
\[ [e]^{\#} D = 0 \]

\[ \bot \neq D[u] = D \]
\[ [e]^{\#} D \notin \{0, \top\} \]
Transformation 4 (cont.): Simplified Expressions

\[ \bot \neq \mathcal{D}[u] = D \]
\[ [e]^\# D = c \]

\[ u \rightarrow x = e; \]
\[ u \rightarrow x = c; \]
Extensions:

- Instead of complete right-hand sides, also subexpressions could be simplified:

\[ x + (3 \times y) \xrightarrow{\{x \mapsto T, y \mapsto 5\}} x + 15 \]

... and further simplifications be applied, e.g.:

\[ x \times 0 \quad \implies \quad 0 \]
\[ x \times 1 \quad \implies \quad x \]
\[ x + 0 \quad \implies \quad x \]
\[ x - 0 \quad \implies \quad x \]

...
So far, the information of **conditions** has not yet be optimally exploited:

\[
\text{if } (x == 7) \\
y = x + 3;
\]

Even if the value of \( x \) before the if statement is unknown, we at least know that \( x \) definitely has the value 7 — whenever the then-part is **entered** :-)

Therefore, we can define:

\[
\lbrack \text{Pos} (x == e) \rbrack \# D = \begin{cases} 
D & \text{if } \lbrack x == e \rbrack \# D = 1 \\
\bot & \text{if } \lbrack x == e \rbrack \# D = 0 \\
D_1 & \text{otherwise}
\end{cases}
\]

where

\[
D_1 = D \oplus \{ x \mapsto (D x \sqcap \lbrack e \rbrack \# D) \}
\]
The effect of an edge labeled \( \text{Neg} \ (x \neq e) \) is analogous \( :-) \)

Our Example:

\[
\begin{align*}
\text{Neg} \ (x == 7) & \quad \text{Pos} \ (x == 7) \\
0 & \quad 1 \\
& \quad y = x + 3; \\
& \quad 2 \\
& \quad 3
\end{align*}
\]
The effect of an edge labeled \( \text{Neg}(x \neq e) \) is analogous :-)

Our Example:

\[
\begin{align*}
0 & \quad x \mapsto \top \\
1 & \quad x \mapsto 7 \\
2 & \quad y = x + 3; \\
3 & \quad x \mapsto \top \\
\end{align*}
\]
The effect of an edge labeled \( \text{Neg} (x \neq e) \) is analogous :-(

Our Example:

\[ \begin{align*}
\text{Neg} (x == 7) & \quad \text{Pos} (x == 7) & \quad \text{Neg} (x == 7) \\
0 & \quad 1 & \quad 2 \\
1 & \quad y = x + 3; & \\
2 & \quad \; & \\
3 & \quad \; & \\
\end{align*} \]
1.5 Interval Analysis

Observation:

- Programmers often use global constants for switching debugging code on/off.
  
  Constant propagation is useful :-)

- In general, precise values of variables will be unknown — perhaps, however, a tight interval !!!
Example:

```plaintext
for (i = 0; i < 42; i++)
    if (0 ≤ i ∧ i < 42) {
        A₁ = A + i;
        M[A₁] = i;
    }

//   A start address of an array
//   if the array-bound check
```

Obviously, the inner check is superfluous :-)

318
Idea 1:

Determine for every variable $x$ an (as tight as possible :-)) interval of possible values:

$$\mathbb{I} = \{ [l, u] \mid l \in \mathbb{Z} \cup \{-\infty\}, u \in \mathbb{Z} \cup \{+\infty\}, l \leq u \}$$

Partial Ordering:

$$[l_1, u_1] \sqsubseteq [l_2, u_2] \text{ iff } l_2 \leq l_1 \land u_1 \leq u_2$$
Thus:

\[ [l_1, u_1] \sqcup [l_2, u_2] = [l_1 \sqcap l_2, u_1 \sqcup u_2] \]
Thus:

\[
[l_1, u_1] \sqcup [l_2, u_2] = [l_1 \sqcap l_2, u_1 \sqcup u_2]
\]

\[
[l_1, u_1] \sqcap [l_2, u_2] = [l_1 \sqcup l_2, u_1 \sqcap u_2] \quad \text{whenever} \ (l_1 \sqcup l_2) \leq (u_1 \sqcap u_2)
\]
Caveat:

→ $\mathbb{I}$ is not a complete lattice  :-)

→ $\mathbb{I}$ has infinite ascending chains, e.g.,

\[
[0, 0] \sqsubset [0, 1] \sqsubset [-1, 1] \sqsubset [-1, 2] \sqsubset \ldots
\]
Caveat:

→ \( I \) is not a complete lattice  

→ \( I \) has infinite ascending chains, e.g.,

\[
[0, 0] \sqsubseteq [0, 1] \sqsubseteq [-1, 1] \sqsubseteq [-1, 2] \sqsubseteq \ldots
\]

Description Relation:

\[
z \triangleq [l, u] \quad \text{iff} \quad l \leq z \leq u
\]

Concretization:

\[
\gamma [l, u] = \{ z \in \mathbb{Z} \mid l \leq z \leq u \}
\]
Example:

\[
\gamma [0, 7] = \{0, \ldots, 7\} \\
\gamma [0, \infty] = \{0, 1, 2, \ldots, \}
\]

Computing with intervals: Interval Arithmetic :-)

Addition:

\[
[l_1, u_1] +^\# [l_2, u_2] = [l_1 + l_2, u_1 + u_2]
\]

where

\[
-\infty + _- = -\infty \\
+\infty + _- = +\infty
\]

// \( -\infty + \infty \) cannot occur :-)

324