The abstract effects of edges $[k]^♯$ are again composed to the effects of paths $\pi = k_1 \ldots k_r$ by:

$$[\pi]^♯ = [k_r]^♯ \circ \ldots \circ [k_1]^♯ : \mathbb{D} \rightarrow \mathbb{D}$$

Idea for Correctness: Abstract Interpretation
Cousot, Cousot 1977

Establish a description relation $\Delta$ between the concrete values and their descriptions with:

$$x \Delta a_1 \land a_1 \sqsubseteq a_2 \implies x \Delta a_2$$

Concretization: $\gamma a = \{x \mid x \Delta a\}$
// returns the set of described values :-}
Values: \[ \Delta \subseteq \mathbb{Z} \times \mathbb{Z}^\top \]

\[ z \Delta a \quad \text{iff} \quad z = a \lor a = \top \]

Concretization:

\[ \gamma a = \begin{cases} 
\{a\} & \text{if } a \sqsubseteq \top \\
\mathbb{Z} & \text{if } a = \top 
\end{cases} \]
(1) Values: \[ \Delta \subseteq \mathbb{Z} \times \mathbb{Z}^\top \]

\[ z \Delta a \iff z = a \lor a = \top \]

Concretization:

\[ \gamma a = \begin{cases} 
\{a\} & \text{if } a \subseteq \top \\
\mathbb{Z} & \text{if } a = \top 
\end{cases} \]

(2) Variable Assignments: \[ \Delta \subseteq (Vars \rightarrow \mathbb{Z}) \times (Vars \rightarrow \mathbb{Z}^\top)_{\perp} \]

\[ \rho \Delta D \iff D \neq \bot \land \rho x \subseteq D x \quad (x \in Vars) \]

Concretization:

\[ \gamma D = \begin{cases} 
\emptyset & \text{if } D = \bot \\
\{ \rho \mid \forall x : (\rho x) \Delta (D x) \} & \text{otherwise} 
\end{cases} \]
Example: \[\{x \mapsto 1, y \mapsto -7\} \Delta \{x \mapsto \top, y \mapsto -7\}\]

(3) States:

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

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

Concretization:

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

\[(\ast) \text{ If } s \Delta D \text{ and } \lceil \pi \rceil s \text{ is defined, then:} \]

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

In particular:

$$\langle \pi \rangle \; s \in \gamma (\[\pi\] D)$$
The abstract semantics simulates the concrete semantics :-)

In particular:

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

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

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

\(\implies\)

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

\[
\begin{align*}
S & \rightarrow [k] \\
D & \rightarrow [k]^\#
\end{align*}
\]

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

290
To prove \((**\)), we show for every expression \(e\) :
\[(***\) \((\llbracket e \rrbracket \rho) \triangle (\llbracket e \rrbracket^{\dagger} D)\) whenever \(\rho \triangle D\)\]
To prove (**), we show for every expression $e$:

$\left( [[e] \rho] \triangle [[e]^# D] \right)$ whenever $\rho \triangle D$ 

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

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

To prove (***) \( \equiv \), we show for every operator \( \square \):
\[ (x \square y) \triangle (x \square y) \quad \text{whenever} \quad x \triangle x \land y \triangle y \]

This precisely was how we have defined the operators \( \square \quad \equiv \)
Now, \((**\)) is proved by case distinction on the edge labels \(lab\).

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

Case \(x = e;\):

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

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

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

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

Case $M[e_1] = e_2;$

\[
\begin{align*}
\rho_1 &= \rho \\
\mu_1 &= \mu \oplus \{ [e_1]\#\rho \mapsto [e_2]\#\rho \} \\
D_1 &= D
\end{align*}
\]

$\implies (\rho_1, \mu_1) \Delta D_1$
Case \textbf{Neg}(e): \quad (\rho_1, \mu_1) = s \quad \text{where:}

\[
\begin{align*}
0 &= \llbracket e \rrbracket \rho \\
\Delta &= \llbracket e \rrbracket \# D \\
\implies 0 &\subseteq \llbracket e \rrbracket \# D \\
\implies \bot &\neq D_1 = D \\
\implies (\rho_1, \mu_1) &\Delta D_1
\end{align*}
\]
Case \textbf{Pos}(e): \quad (\rho_1, \mu_1) = s \quad \text{where:}

\[
\begin{align*}
0 & \neq [e] \rho \\
\Delta & \quad [e]^# D \\
\implies & \quad 0 \neq [e]^# D \\
\implies & \quad \bot \neq D_1 = D \\
\implies & \quad (\rho_1, \mu_1) \Delta D_1 \\
\end{align*}
\]
We conclude: The assertion \((\ast)\) is true \(:-))\)

The MOP-Solution:

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

where \(D_\top x = \top\) \((x \in Vars)\).
We conclude: The assertion \((\ast)\) is true :-()

The MOP-Solution:

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

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 \(v\):

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

The MOP-Solution

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

where \( D_\top x = \top (x \in Vars) \).

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

\[ (\llbracket \pi \rrbracket s) \Delta (\mathcal{D}^*[v]) \]

In order to approximate the MOP, we use our constraint system :-))
Example:

\[ x = 10; \]

\[ y = 1; \]

\[ M[R] = y; \]

\[ y = x \times y; \]

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

\[ x = 10; \]

\[ y = 1; \]

\[ M[R] = y; \]

\[ y = x \times y; \]

\[ x = x - 1; \]

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

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

\[ 0 \]

\[ 1 \]

\[ 2 \]

\[ 3 \]

\[ 4 \]

\[ 5 \]

\[ 6 \]

\[ 7 \]

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

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

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(x)</td>
<td>(y)</td>
<td>(x)</td>
</tr>
<tr>
<td>0</td>
<td>(\top)</td>
<td>(\top)</td>
<td>(\top)</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>(\top)</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>1</td>
<td>(\top)</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1</td>
<td>(\top)</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>10</td>
<td>(\top)</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>10</td>
<td>(\top)</td>
</tr>
<tr>
<td>6</td>
<td>(\bot)</td>
<td>(\top)</td>
<td>(\top)</td>
</tr>
<tr>
<td>7</td>
<td>(\bot)</td>
<td>(\top)</td>
<td>(\top)</td>
</tr>
</tbody>
</table>
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: \[ f = [x = x + y;]^{\#} \]

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.,

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

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

$$\mathcal{D}^*[v] \subseteq \mathcal{D}[v]$$

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

$$([\pi](\rho, \mu)) \Delta (\mathcal{D}[v])$$

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

\[
\mathcal{D}[u] = \bot
\]

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

Neg \((e)\):
\[
\downarrow \not\equiv \mathcal{D}[u] = D
\]
\[
\lceil e \rceil^\# D = 0
\]

Pos \((e)\):
\[
\downarrow \not\equiv \mathcal{D}[u] = D
\]
\[
\lceil e \rceil^\# D \not\in \{0, \top\}
\]
Transformation 4 (cont.): Simplified Expressions

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

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

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

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

\[
\begin{align*}
x \times 0 & \quad \Longrightarrow \quad 0 \\
x \times 1 & \quad \Longrightarrow \quad x \\
x + 0 & \quad \Longrightarrow \quad x \\
x - 0 & \quad \Longrightarrow \quad x \\
& \quad \ldots
\end{align*}
\]
So far, the information of *conditions* has not yet be optimally exploited:

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

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:

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

where

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

Our Example:

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

Our Example:

\[
\begin{array}{c}
\text{Neg} \ (x == 7) \\
\text{Pos} \ (x == 7)
\end{array}
\]

\[
\begin{array}{c}
0 \quad x \mapsto \top \\
1 \quad x \mapsto 7 \\
2 \quad x \mapsto 7 \\
3 \quad x \mapsto \top
\end{array}
\]

\[
\begin{array}{c}
y = x + 3; \\
x \mapsto 7
\end{array}
\]
The effect of an edge labeled \( \text{Neg} \left( x \neq e \right) \) is analogous \( \text{:-)} \)

Our Example:

\[
\begin{align*}
\text{Neg} \left( x == 7 \right) & \quad \text{Pos} \left( x == 7 \right) & \quad \text{Neg} \left( x == 7 \right) \\
0 & \quad 1 & \quad 0 \\
1 & \quad 2 & \quad 1 \\
2 & \quad 3 & \quad 2 \\
3 & \quad ; & \quad 3 \\
& \quad y = x + 3; & \quad y = 10; \\
\end{align*}
\]
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:

```c
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  :-)

Idea 1:

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

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

Partial Ordering:

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

→ \( \mathbb{I} \) is not a complete lattice \( \therefore \)

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

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

Description Relation:

\[
z \triangle [l, u] \iff l \leq z \leq u
\]

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 :-(
Negation:

\[-\# [l, u] = [-u, -l]\]

Multiplication:

\[[l_1, u_1] \ast \# [l_2, u_2] = [a, b]\]

where

\[a = l_1 l_2 \cap l_1 u_2 \cap u_1 l_2 \cap u_1 u_2\]

\[b = l_1 l_2 \cup l_1 u_2 \cup u_1 l_2 \cup u_1 u_2\]

Example:

\[[0, 2] \ast \# [3, 4] = [0, 8]\]

\[[{-1, 2}] \ast \# [3, 4] = [-4, 8]\]

\[[{-1, 2}] \ast \# [{-3, 4}] = [-6, 8]\]

\[[{-1, 2}] \ast \# [{-4, -3}] = [-8, 4]\]