1.4 Constant Propagation

Idea:
Execute as much of the code at compile-time as possible!

Example:

\[ x = 7; \]
\[ \text{if } (x > 0) \]
\[ M[A] = B; \]
Obviously, $x$ has always the value 7.
Thus, the memory access is always executed.

Goal:

1. $x = 7$;
2. Neg ($x > 0$) Pos ($x > 0$)
3. $M[A] = B$;
4.
5.
Obviously, $x$ has always the value 7.
Thus, the memory access is always executed.

**Goal:**

```
1
 ▼
 2
     ▼
   3   4
      ▼
     5

x = 7;
Neg (x > 0) Pos (x > 0)
M[A] = B;

1
 ▼
 2
     ▼
   3   4
      ▼
     5

; ;
M[A] = B;
```
Idea:

Design an analysis that for every program point \( u \), determines the values that variables definitely have;

As a side effect, it also tells whether \( u \) can be reached at all
Idea:

Design an analysis that for every program \( u \), determines the values that variables definitely have;

As a sideeffect, it also tells whether \( u \) can be reached at all

The complete lattice is constructed in two steps.

(1) The potential values of variables:

\[
\mathbb{Z}^T = \mathbb{Z} \cup \{ \top \} \quad \text{with} \quad x \sqsubseteq y \quad \text{iff} \quad y = \top \text{ or } x = y
\]
Caveat: $\mathbb{Z}^\top$ is not a complete lattice in itself

(2) $\mathcal{D} = \underbrace{(\text{Vars} \rightarrow \mathbb{Z}^\top)}_\perp = (\text{Vars} \rightarrow \mathbb{Z}^\top) \cup \{\perp\}$

// $\perp$ denotes: “not reachable”

with $D_1 \subseteq D_2$ iff $\perp = D_1$ or $D_1 x \subseteq D_2 x$ ($x \in \text{Vars}$)

$\mathcal{D}_1$ knows more values than $\mathcal{D}_2$

Remark: $\mathcal{D}$ is a complete lattice
Caveat: \( \mathbb{Z}^T \) is not a complete lattice in itself

\[(2) \quad \mathbb{D} = (\text{Vars} \rightarrow \mathbb{Z}^T) \perp = (\text{Vars} \rightarrow \mathbb{Z}^T) \cup \{\perp\} \]

// \( \perp \) denotes: “not reachable”

with \( D_1 \subseteq D_2 \) iff \( \perp = D_1 \) or
\( D_1 x \subseteq D_2 x \) \( (x \in \text{Vars}) \)

Remark: \( \mathbb{D} \) is a complete lattice

Consider \( X \subseteq \mathbb{D} \). W.l.o.g., \( \perp \notin X \).

Then \( X \subseteq \text{Vars} \rightarrow \mathbb{Z}^T \).

If \( X = \emptyset \), then \( \bigcup X = \perp \in \mathbb{D} \)
If $X \neq \emptyset$, then $\bigcup X = D$ with

$$D_x = \bigcup \{ f(x) \mid f \in X \} = \begin{cases} z & \text{if } f(x) = z \quad (f \in X) \\ \top & \text{otherwise} \end{cases}$$

$\therefore$
If \( X \neq \emptyset \), then \( \bigcup X = D \) with

\[
D_x = \bigcup \{ f \cdot x \mid f \in X \}
\]

= \begin{cases} 
  z & \text{if } f \cdot x = z \quad (f \in X) \\
  \top & \text{otherwise}
\end{cases}

\[\text{:-)}\]

For every edge \( k = (\_, \text{lab}, \_) \), construct an effect function

\[ [k]^\# = [\text{lab}]^\# : \mathbb{D} \to \mathbb{D} \]

which simulates the \textit{concrete} computation.

Obviously, \[ [\text{lab}]^\# \bot = \bot \] for all \( \text{lab} \)

Now let \( \bot \neq D \in \text{Vars} \to \mathbb{Z}^\top \).
Idea:

- We use $D$ to determine the values of expressions.
Idea:

- We use $D$ to determine the values of expressions.
- For some sub-expressions, we obtain $\top$. 
Idea:

- We use $D$ to determine the values of expressions.
- For some sub-expressions, we obtain $\top$.

We must replace the concrete operators $\square$ by abstract operators $\square^#$ which can handle $\top$:

$$a \square^# b = \begin{cases} \top & \text{if } a = \top \text{ or } b = \top \\ a \square b & \text{otherwise} \end{cases}$$
Idea:

- We use $D$ to determine the values of expressions.
- For some sub-expressions, we obtain $\top$.

We must replace the concrete operators $\square$ by abstract operators $\square^\#$ which can handle $\top$:

$$a \square^\# b = \begin{cases} 
\top & \text{if } a = \top \text{ or } b = \top \\
\square b & \text{otherwise}
\end{cases}$$

- The abstract operators allow to define an abstract evaluation of expressions:

$$[e]^\# : (Vars \to \mathbb{Z}^\top) \to \mathbb{Z}^\top$$
Abstract evaluation of expressions is like the concrete evaluation — but with abstract values and operators. Here:

\[
\begin{align*}
[c]^D&=c \\
[e_1 \square e_2]^D&=[e_1]^D \square [e_2]^D
\end{align*}
\]

... analogously for unary operators
Abstract evaluation of expressions is like the concrete evaluation — but with abstract values and operators. Here:

\[
[c]^D = c \\
[e_1 \Box e_2]^D = [e_1]^D \Box^D [e_2]^D
\]

... analogously for unary operators

**Example:**

\[
D = \{x \mapsto 2, y \mapsto \top\}
\]

\[
= 2 +^D 7 \\
= 9
\]

\[
[x - y]^D = 2 -^D \top \\
= \top
\]
Thus, we obtain the following effects of edges \([lab]^\#\):

\[
\begin{align*}
[;]^\# \ D &= \ D \\
{\text{Pos}(e)}^\# \ D &= \begin{cases} 
\bot & \text{if } 0 = [e]^\# \ D \\
D & \text{otherwise}
\end{cases} \\
{\text{Neg}(e)}^\# \ D &= \begin{cases} 
D & \text{if } 0 \subseteq [e]^\# \ D \\
\bot & \text{otherwise}
\end{cases} \\
[x = e;]^\# \ D &= D \oplus \{x \mapsto [e]^\# \ D\} \\
[x = M[e];]^\# \ D &= D \oplus \{x \mapsto \top\} \\
[M[e_1] = e_2;]^\# \ D &= \ D
\end{align*}
\]

... whenever \(D \neq \bot\)
At \textit{start}, we have \( D_{\top} = \{ x \mapsto \top \mid x \in \text{Vars} \} \).

Example:

\begin{itemize}
  \item \( x = 7; \)
  \item \( \text{Neg} \ (x > 0) \)
  \item \( \text{Pos} \ (x > 0) \)
  \item \( M[A] = B; \)
  \item \( ; \)
\end{itemize}
At \textit{start}, we have \( D_\top = \{ x \mapsto \top \mid x \in \text{Vars} \} \).

\textbf{Example:}

\begin{equation*}
\begin{array}{c}
1 \\
2 \quad x = 7; \\
3 \quad \text{Neg} (x > 0) \\
4 \quad \text{Pos} (x > 0) \\
5 \\
\end{array}
\end{equation*}

\begin{align*}
1 & \quad \{ x \mapsto \top \} \\
2 & \quad \{ x \mapsto 7 \} \\
3 & \quad \{ x \mapsto 7 \} \\
4 & \quad \{ x \mapsto 7 \} \\
5 & \quad \bot \cup \{ x \mapsto 7 \} = \{ x \mapsto 7 \}
\end{align*}
The abstract effects of edges \([k]\) are again composed to the effects of paths \(\pi = k_1 \ldots k_r\) by:

\[
[\pi] = [k_r] \circ \ldots \circ [k_1] : \mathbb{D} \to \mathbb{D}
\]

**Idea for Correctness:**

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

**Abstract Interpretation**

Cousot, Cousot 1977
Values:

\[ \Delta \subseteq \mathbb{Z} \times \mathbb{Z}^\top \]

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

Concretization:

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

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

Concretization:

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

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

\[ \rho \Delta D \quad \text{iff} \quad D \neq \bot \land \rho x \sqsubseteq D x \quad (x \in \text{Vars}) \]

Concretization:

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

(3) States:

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

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

Concretization:

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

\[(*) \quad \text{If} \quad s \triangle D \quad \text{and} \quad \llbracket \pi \rrbracket s \quad \text{is defined, then:} \quad \llbracket \pi \rrbracket s \triangle \llbracket \pi \rrbracket^\# D\]
The abstract semantics simulates the concrete semantics.

In particular:

\[
[p_\pi] s \in \gamma([p_\pi]^D) \quad \text{if } \land \Delta \models D
\]
The abstract semantics simulates the concrete semantics

In particular:

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

In practice, this means for example that \( D \, x = -7 \) implies:

\[
\begin{align*}
\rho' \, x & = -7 \quad \text{for all} \quad \rho' \in \gamma \, D \\
\implies \rho_1 \, x & = -7 \quad \text{for} \quad (\rho_1, \_ ) = [\pi] \, s
\end{align*}
\]
To prove (*)&, we show for every edge $k$:

\[
\begin{array}{ccc}
S & \xrightarrow{[k]} & S_1 \\
\Delta & \downarrow & \Delta \\
D & \xrightarrow{[k]^\#} & D_1
\end{array}
\]

Then (*)& follows by induction.
To prove (**) we show for every expression $e$:

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

\[(x \Box y) \triangle (x^{\#} \Box^{\#} y^{\#}) \quad \text{whenever} \quad x \triangle x^{\#} \land y \triangle y^{\#}\]
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 \(\square\):

\[\quad (x \square y) \triangle (x \# \square \# y \#) \quad \text{whenever} \quad x \triangle x \# \wedge y \triangle y \#\]

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

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

Case \(x = c;\): 

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

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

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

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

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

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

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

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

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

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

\[
\begin{align*}
0 & \not\equiv \llbracket e \rrbracket \rho \\
\Delta & \llbracket e \rrbracket^\# D \\
\implies & 0 \not\equiv \llbracket e \rrbracket^\# D \\
\implies & \bot \not\equiv D_1 = D \\
\implies & (\rho_1, \mu_1) \Delta D_1
\end{align*}
\]

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

The MOP-Solution:

\[
\mathcal{D}^*[v] = \bigcup \{ [[\pi]]^\# \mid D_T | \pi : start \rightarrow^* v \}
\]

where \(D_T x = T\) \((x \in Vars)\).
We conclude: The assertion \((*)\) is true

The MOP-Solution:

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

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

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

\[
([[\pi]]^s) \triangleq (D^*[v])
\]
We conclude: The assertion \((\ast)\) is true

The MOP-Solution

\[
D^*[v] = \bigcup \{ [\pi]^\# \mid D_T \mid \pi : start \to^* v \}
\]

where \( D_T x = T \quad (x \in Vars) \).

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

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

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

0
\[ x = 10; \]

1
\[ y = 1; \]

2
\[ \text{Neg}(x > 1) \]
\[ \text{Pos}(x > 1) \]

6
\[ M[R] = y; \]

7

3
\[ y = x \times y; \]

4
\[ x = x - 1; \]

5
Example:

Graph:

- Node 0: $x = 10$
- Node 1: $y = 1$
- Node 2: \(\text{Neg}(x > 1)\) and \(\text{Pos}(x > 1)\)
- Node 6: $M[R] = y$
- Node 3: $y = x \ast y$
- Node 4: $x = x - 1$
- Node 7

Table:

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

\[
x = 10;
\]

\[
y = 1;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
\]

\[
M[R] = y;
Example:

```
x = 10;
y = 1;
Neg(x > 1)
```

```
Pos(x > 1)
```

```
M[R] = y;
y = x * y;
```

```
x = x - 1;
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>y</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>x</td>
<td>10</td>
<td>10</td>
<td>T</td>
</tr>
<tr>
<td>y</td>
<td>1</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>x</td>
<td>10</td>
<td>10</td>
<td>T</td>
</tr>
<tr>
<td>y</td>
<td>10</td>
<td>10</td>
<td>T</td>
</tr>
<tr>
<td>x</td>
<td>9</td>
<td>10</td>
<td>T</td>
</tr>
<tr>
<td>y</td>
<td>10</td>
<td>10</td>
<td>T</td>
</tr>
<tr>
<td>x</td>
<td>⊥</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>y</td>
<td>⊥</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

```
dito```

```
312```

```
```
Conclusion:

Although we compute with concrete values, we fail to compute everything.
The fixpoint iteration is guaranteed to terminate:
For $n$ program points and $m$ variables, we maximally need:
$n \cdot (m + 1)$ rounds

Caveat:

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

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 $\pi$ that reaches $v$:

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

whenever $[\pi](\rho, \mu)$ is defined
Transformation CF:

Removal of Dead Code

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

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

\[ \bot \neq D[u] = D \]
\[ \lceil e \rceil \# D = 0 \]

\[ \bot \neq D[u] = D \]
\[ \lceil e \rceil \# D \not\in \{0, \top\} \]
Transformation CF (cont.): Simplified Expressions

\[ \bot \neq \mathcal{D}[u] = D \]
\[ \llbracket e \rrbracket^\sharp D = c \]

\[ x = e; \]

\[ x = c; \]
Extensions:

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

\[ x + (3 \times y) \quad \xrightarrow{\{x \mapsto 1, y \mapsto 5\}} \quad x + 15 \]

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

\[
\begin{align*}
  x \times 0 & \quad \Longrightarrow \quad 0 \\
  x \times 1 & \quad \Longrightarrow \quad x \\
  x + 0 & \quad \Longrightarrow \quad x \\
  x - 0 & \quad \Longrightarrow \quad x \\
  \ldots & \\
\end{align*}
\]
• 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:

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

where

\[
D_1 = D \oplus \{ x \mapsto (D x \cap [e]^\# D) \}
\]
The effect of an edge labeled $\text{Neg}(x \neq e)$ is analogous.

Our Example:

```
Neg(x == 7)  Pos(x == 7)
```

```
y = x + 3;
```

;
The effect of an edge labeled \( \text{Neg} (x \neq e) \) is analogous

Our Example:

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

Our Example:

$$\neg (x == 7) \quad \text{Pos} (x == 7) \quad \neg (x == 7) \quad \text{Pos} (x == 7)$$

$$y = x + 3;$$

$$y = 10;$$