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:

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

   $$\mathbb{Z}^\top = \mathbb{Z} \cup \{\top\}$$

   with $x \sqsubseteq y$ iff $y = \top$ or $x = y$

   \[
   \begin{array}{ccccccc}
   \cdots & -2 & -1 & 0 & 1 & 2 & \cdots \\
   \end{array}
   \]
Caveat: \( \mathbb{Z}^\top \) is not a complete lattice in itself

(2) \( \mathbb{D} = (\text{Vars} \rightarrow \mathbb{Z}^\top) \bot = (\text{Vars} \rightarrow \mathbb{Z}^\top) \cup \{ \bot \} \)

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

with \( D_1 \sqsubseteq D_2 \) iff \( \bot = D_1 \) or
\[
D_1 x \sqsubseteq D_2 x \quad (x \in \text{Vars})
\]

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

(2) $\mathbb{D} = (\text{Vars} \to \mathbb{Z}^\top) \bot = (\text{Vars} \to \mathbb{Z}^\top) \cup \{\bot\}$

// $\bot$ denotes: “not reachable”

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

Remark: $\mathbb{D}$ is a complete lattice

Consider $X \subseteq \mathbb{D}$. W.l.o.g., $\bot \notin X$.

Then $X \subseteq \text{Vars} \to \mathbb{Z}^\top$.

If $X = \emptyset$, then $\bigcup X = \bot \in \mathbb{D}$
If $X \neq \emptyset$, then $\bigsqcup X = D$ with

$$
D_x = \bigsqcup \{ f_x | f \in X \}
$$

$$
= \begin{cases}
  z & \text{if } f_x = z \quad (f \in X) \\
  \top & \text{otherwise}
\end{cases}
$$
If \( X \neq \emptyset \), then \( \bigsqcup X = D \) with
\[
D x = \bigsqcup \{ f x \mid f \in X \}
\]
\[
= \begin{cases} 
    z & \text{if } f x = z \ (f \in X) \\
    \top & \text{otherwise}
\end{cases}
\]

For every edge \( k = (_, lab, _) \), construct an effect function
\[
[k]^\# = [lab]^\# : D \to D
\]
which simulates the concrete computation.

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

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

  $\top = \top 

  \Rightarrow

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

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

- The abstract operators allow to define an \texttt{abstract} evaluation of expressions:

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

\[
[c] D = c
\]

\[
[e_1 \square e_2] D = [e_1] D \square [e_2] D
\]

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

\[
\begin{align*}
[c] \# D &= c \\
[e_1 \bigcirc e_2] \# D &= [e_1] \# D \bigcirc [e_2] \# D
\end{align*}
\]

... analogously for unary operators

Example: \( D = \{ x \mapsto 2, y \mapsto \top \} \)

\[
\begin{align*}
&= 2 + 7 \\
&= 9 \\
[x - y] \# D &= 2 - \top \\
&= \top
\end{align*}
\]
Thus, we obtain the following effects of edges $[lab]^\#$:

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

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

\[ \begin{array}{c}
\text{Example:} \\
\end{array} \]

\[ \begin{array}{c}
1 & \xrightarrow{x = 7;} & 2 \\
\text{Neg} \ (x > 0) & \xrightarrow{\text{Pos} \ (x > 0)} & 3 \\
\text{Neg} \ (x > 0) & \xrightarrow{M[A] = B;} & 4 \\
\xrightarrow{\text{Pos} \ (x > 0)} & 2 \\
\xrightarrow{\text{Neg} \ (x > 0)} & 5 \\
\end{array} \]
At \textit{start}, we have \( D_{\top} = \left\{ x \mapsto \top \mid x \in Vars \right\} \).

Example:

\begin{itemize}
  \item \( x = 7 \);
  \item Neg \((x > 0)\)
  \item Pos \((x > 0)\)
  \item \( M[A] = B \);
  \item \( \bot \sqcup \left\{ x \mapsto 7 \right\} = \left\{ x \mapsto 7 \right\} \)
\end{itemize}
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:** 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:**

\[
\begin{align*}
\gamma a & = \{ x \mid x \Delta a \} \\
// & \text{ returns the set of described values}
\end{align*}
\]
(1) Values: \[ \Delta \subseteq \mathbb{Z} \times \mathbb{Z}^\top \]

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

Concretization:

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

(2) **Variable Bindings:** \[ \Delta \subseteq (\text{Vars} \to \mathbb{Z}) \times (\text{Vars} \to \mathbb{Z}^\top)_\bot \]
\[
\rho \Delta D \iff D \neq \bot \land \rho x \sqsubseteq D x \quad (x \in \text{Vars})
\]
Concretization:
\[
\gamma D = \begin{cases} 
\emptyset & \text{if } D = \bot \\
\{\rho \mid \forall x : (\rho x) \Delta (D x)\} & \text{otherwise}
\end{cases}
\]

293
Example: \( \{ x \mapsto 1, y \mapsto -7 \} \triangleright \{ x \mapsto \top, y \mapsto -7 \} \)

(3) States:

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

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

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 local correctness:

\[ (*) \quad \text{If} \quad s \Delta D \quad \text{and} \quad [\pi] s \quad \text{is defined, then:} \]

\[ ([\pi] s) \Delta ([\pi] \# D) \]
The abstract semantics simulates the concrete semantics

In particular:

\[ [\pi] s \in \gamma ([\pi]^# D) \]
The abstract semantics simulates the concrete semantics.

In particular:

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

In practice, this means for example that $Dx = -7$ implies:

$$\rho' x = -7 \text{ for all } \rho' \in \gamma D$$

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

\[
s \xrightarrow{[k]} S_1
\]

\[D \xrightarrow{[k]} D_1\]

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

\[(***) \quad ([e] \rho) \Delta ([e]^D) \quad \text{whenever} \quad \rho \Delta D\]
To prove (**)\textsuperscript{,} we show for every expression \( e \):

\[(**\textsuperscript{**}) \quad ([e] \rho) \Delta ([e] D) \text{ whenever } \rho \Delta D\]

To prove (**\textsuperscript{***})\textsuperscript{,} we show for every operator \( \Box \):

\[(**\textsuperscript{***}) \quad (x \Box y) \Delta (x \Box y) \text{ whenever } x \Delta x \land y \Delta y\]
To prove \((**\))\), we show for every expression \(e\):
\[(***) \quad ([e] \rho) \triangleright ([e]^{\#} D) \quad \text{whenever} \quad \rho \triangleright D\]

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

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

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

Case \(\boxed{x = e;}\):

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

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

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

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

Case $M[e_1] = e_2$; :

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

\[\implies (\rho_1, \mu_1) \triangleq D_1\]
Case \textbf{Neg}(e) : \hspace{1cm} (\rho_1, \mu_1) = s \hspace{0.5cm} \text{where:}

\[
0 = [e] \rho \\
\Delta [e]^\# D \\
\implies 0 \subseteq [e]^\# D \\
\implies \bot \neq D_1 = D \\
\implies (\rho_1, \mu_1) \Delta D_1
\]
Case \text{Pos}(e) : \quad (\rho_1, \mu_1) = s \quad \text{where:}

\begin{align*}
0 & \neq [e] \rho \\
\Delta & \neq [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 \((*)\) is true

The MOP-Solution:

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

where \(D_\top x = \top \quad (x \in \text{Vars})\) .
We conclude: The assertion $(\ast)$ is true

The MOP-Solution:

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

where $D_T x = \top \quad (x \in Vars)$.

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

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

The MOP-Solution

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

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

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

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

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:

\[
x = 10;
\]

\[
y = 1;
\]

\[
M[R] = y;
\]

\[
y = x \times y;
\]

\[
x = x - 1;
\]

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

$M[R] = y$;

$\text{Neg}(x > 1)$

$\text{Pos}(x > 1)$

$x = 10$

$y = 1$

$y = x \times y$

$x = x - 1$

$y = 1$

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

Dito
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: \[ f = [x = x + y;]^{\#} \]

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.,

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

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

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

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

\[ D[u] = \bot \]

\[ \lceil lab \rceil^\#(D[u]) = \bot \]
Transformation CF (cont.): Removal of Dead Code

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

\[
\begin{align*}
\text{Neg } (e) & \quad \rightarrow \\
\text{Pos } (e) & \quad \rightarrow
\end{align*}
\]
Transformation CF (cont.):  Simplified Expressions

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

\[ x = e; \]

\[ x = c; \]
Extensions:

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

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

... 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 been 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 \( \text{Neg} \left( x \neq e \right) \) is analogous

Our Example:

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

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

Our Example:

\[
\begin{align*}
\text{Neg}(x == 7) & & \text{Pos}(x == 7) & & \text{Neg}(x == 7) \\
0 & \rightarrow 1 & \rightarrow 2 & \rightarrow 3 & \rightarrow 0 \\
\text{Neg}(x == 7) & \rightarrow \text{Pos}(x == 7) \\
0 & \rightarrow 1 & \rightarrow 2 & \rightarrow 3 & \rightarrow 0
\end{align*}
\]

\[
y = x + 3; \\
y = 10;
\]