1.4 Constant Propagation

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

Example:
Obviously, $x$ has always the value $7$  :)
Thus, the memory access is always executed  :))

Goal:

Generalization:  Partial Evaluation

Neil D. Jones, DIKU, Kopenhagen

Idea:

Design an analysis which for every $u$,
- determines the values which variables definitely have;
- tells whether $u$ can be reached at all  :)

Goal:
Idea:

Design an analysis which for every \( u \),
- determines the values which variables definitely have;
- tells whether \( u \) can be reached at all \( : ( \)

The complete lattice is constructed in two steps.

1. The potential values of variables:
   \[
   Z^T = \mathbb{Z} \cup \{ T \} \quad \text{with} \quad x \subseteq y \iff y = T \text{ or } x = y
   \]

   ![Diagram of lattice](Image)

   - \( \cdots \ 2 \ 1 \ 0 \ 1 \ 2 \ \cdots \)

Caveat: \( Z^T \) is not a complete lattice in itself \( : ( \)

\[
2. \quad \mathcal{D} = (\text{Vars} \rightarrow Z^T)_\perp = (\text{Vars} \rightarrow Z^T) \cup \{ \perp \} \\
\quad \quad \quad \quad \quad \text{// } \perp \text{ denotes: “not reachable” } : ()
\]

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

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

Caveat: \( Z^T \) is not a complete lattice in itself \( : ( \)

\[
2. \quad \mathcal{D} = (\text{Vars} \rightarrow Z^T)_\perp = (\text{Vars} \rightarrow Z^T) \cup \{ \perp \} \\
\quad \quad \quad \quad \quad \text{// } \perp \text{ denotes: “not reachable” } : ()
\]

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

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

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

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

If \( X = \emptyset \), then \( \bigcup X = \perp \in \mathcal{D} \) \( : ( \)
If $X \neq \emptyset$, then $\bigcup X = D$ with
\[
D_x = \bigcup \{ f \mid f \in X \}
= \begin{cases}
  z & \text{if } f_x = z \ (f \in X) \\
  \top & \text{otherwise}
\end{cases}
\]

For every edge $k = (_, \{\text{lab}\}, _)$, construct an effect function
$[k]^2 = [\{\text{lab}\}]^2 : D \to D$ which simulates the concrete computation.

Obviously, $[\{\text{lab}\}]^2 \bot = \bot$ for all $\{\text{lab}\} \vdash$

Now let $\bot \neq D \in \text{Vars} \to \mathbb{Z}^\dagger$.

\[\text{271}
\]

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

For every edge $k = (_, \{\text{lab}\}, _)$, construct an effect function
$[k]^2 = [\{\text{lab}\}]^2 : D \to D$ which simulates the concrete computation.

Obviously, $[\{\text{lab}\}]^2 \bot = \bot$ for all $\{\text{lab}\} \vdash$

Now let $\bot \neq D \in \text{Vars} \to \mathbb{Z}^\dagger$.

\[\text{271}
\]

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

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

$$\prod X \times \bigcirc D \uparrow \{ x \mapsto \mathcal{S}, \varphi \mapsto \top \} \quad \cong \quad \top$$

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

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

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

$$[e]^D : (\text{Vars} \to \mathcal{Z}^\top) \to \mathcal{Z}^\top$$

273

---

Idea:

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

$$\prod X \times \bigcirc D \uparrow \{ x \mapsto \mathcal{S}, \varphi \mapsto \top \} \quad \cong \quad \top$$

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

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

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

$$[e]^D : (\text{Vars} \to \mathcal{Z}^\top) \to \mathcal{Z}^\top$$

274

---

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

$$[e]^D D = e$$
$$[e_1 \Box^D e_2]^D D = [e_1]^D D \Box^D [e_2]^D D$$

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

\[
[x]_D^1 = c \\
[e_1 \lor e_2]_D^1 = [e_1]_D^1 \lor [e_2]_D^1 \\
\text{... analogously for unary operators} \quad :-) 
\]

**Example:**

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

\[
[x + 7]_D^1 = [x]_D^1 + 7^1 1
\]

\[* 9
\]

\[
[x - y]_D^1 = 2 - 3 \top
\]

\[* T
\]

Thus, we obtain the following effects of edges \([lab]^2\):

\[
[]_D^1 = D \\
[\text{Pos}(e)]_D^1 = \begin{cases} \bot & \text{if } 0 = [e]_D^1 \\ D & \text{otherwise} \end{cases} \\
[\text{Neg}(e)]_D^1 = \begin{cases} D & \text{if } 0 \sqsubseteq [e]_D^1 \\ \bot & \text{otherwise} \end{cases} \\
[x = e]_D^1 = D \oplus \{ x \mapsto [e]_D^1 \} \\
[x = M[e]\_D^1] = D \oplus \{ x \mapsto \top \} \\
[M[e]_D = e_2]_D^1 = D
\]

\[\text{... whenever } D \neq \bot \quad :-) \]

At **start**, we have \( D = \{ x \mapsto \top \mid x \in \text{Vars} \}\).

**Example:**

\[\begin{array}{c}
\text{Neg}(x > 0) \\
\text{Pos}(x > 0) \\
M[A] = B;
\end{array}\]

Thus, we obtain the following effects of edges \([lab]^2\):

\[
[]_D^1 = D \\
[\text{Pos}(e)]_D^1 = \begin{cases} \bot & \text{if } 0 = [e]_D^1 \\ D & \text{otherwise} \end{cases} \\
[\text{Neg}(e)]_D^1 = \begin{cases} D & \text{if } 0 \sqsubseteq [e]_D^1 \\ \bot & \text{otherwise} \end{cases} \\
[x = e]_D^1 = D \oplus \{ x \mapsto [e]_D^1 \} \\
[x = M[e]\_D^1] = D \oplus \{ x \mapsto \top \} \\
[M[e]_D = e_2]_D^1 = D
\]

\[\text{... whenever } D \neq \bot \quad :-) \]
At start, we have \( D_T = \{ x \mapsto T \mid x \in \text{Vars} \} \).

Example:

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

\[ [\pi]^i = [k_1]^i \circ \ldots \circ [k_r]^i : D \rightarrow D \]

Idea for Correctness: Abstract Interpretation

Cousot, Cousot 1977
The abstract effects of edges $[k]^d$ are again composed to the effects of paths $\pi = k_1 \ldots k_n$ by:

$$[\pi]^d = [k_n]^d \circ \ldots \circ [k_1]^d : D \to D$$

**Idea for Correctness:**

Abstract Interpretation  
Cousot, Cousot 1977

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

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

**Concretization:**

$$\gamma a = \{ x | x \Delta a \}$$

// returns the set of described values :-)

---

(1) **Values:**

$$\Delta \subseteq Z \times Z^T$$

$$z \Delta a \iff [z = a] \lor [a = T]$$

**Concretization:**

$$\gamma a = \begin{cases} \{a\} & \text{if } a \subseteq T \\ Z & \text{if } a = T \end{cases}$$

---

(2) **Variable Assignments:**

$$\Delta \subseteq (\mathcal{V} \rightarrow \mathbb{Z}) \times (\mathcal{V} \rightarrow \mathbb{Z}^T)^\perp$$

$$\rho \Delta D \iff D \neq \perp \land \rho x \Delta (D_x) \quad (x \in \mathcal{V})$$

**Concretization:**

$$\gamma D = \begin{cases} \emptyset & \text{if } D = \perp \\ \{ \rho | \forall x : (\rho x) \Delta (D_x) \} & \text{otherwise} \end{cases}$$
(1) \( \Delta \subseteq \mathbb{Z} \times \mathbb{Z}^T \)

Concretization:

\[ \{ (\gamma a, \gamma b) \mid \gamma a = a \Rightarrow a = \top \} \]

Concretization:

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

(2) Variable Assignments:

\( \Delta \subseteq (\mathbb{V}ars \rightarrow \mathbb{Z}) \times (\mathbb{V}ars \rightarrow \mathbb{Z}^T) \)

Concretization:

\[ \rho \Delta D \iff D \neq \bot \bigwedge \rho x \subseteq D x \quad (x \in \mathbb{V}ars) \]

Concretization:

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

The abstract semantics simulates the concrete semantics. 

In particular:

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

Example:

\[ \{ x \mapsto 1, y \mapsto -7 \} \Delta \{ x \mapsto \top, y \mapsto -7 \} \]

(3) States:

\( \Delta \subseteq ((\mathbb{V}ars \rightarrow \mathbb{Z}) \times (\mathbb{N} \rightarrow \mathbb{Z})) \times (\mathbb{V}ars \rightarrow \mathbb{Z}^T) \)

Concretization:

\[ \rho \Delta D \iff \rho \Delta D \]

Concretization:

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

We show:

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

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

In particular:

\[ [x] s \in \gamma ([x]^p D) \]

To prove (\#), we show for every edge \( k \):

\[ ([x] s) \leq \Delta [k] \]

Then (\#) follows by induction :)

The abstract semantics simulates the concrete semantics 

In particular:

\[ [x] s \in \gamma ([x]^p D) \]

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

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

\[ \implies \quad \rho x = -7 \quad \text{for} \quad (\rho, x) = [x] s \]

To prove (\#\#), we show for every expression \( e \):

(\#\#) \( ([e], \rho) \Delta ([e]^p D) \) whenever \( \rho \Delta D \)
To prove (**), we show for every expression $e$:

\[(\llbracket e \rrbracket_\rho) \Delta \llbracket e \rrbracket^\dagger D \quad \text{whenever} \quad \rho \Delta D\]

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

\[(x \Box y) \Delta (x^\dagger \Box^\dagger y^\dagger) \quad \text{whenever} \quad x \Delta x^\dagger \land y \Delta y^\dagger\]

This precisely was how we have defined the operators $\Box^\dagger$ :}