Idea:

Design an analysis which for every u,

- determines the values which variables definitely have;
- tells whether u can be reached at all :-)

Idea:

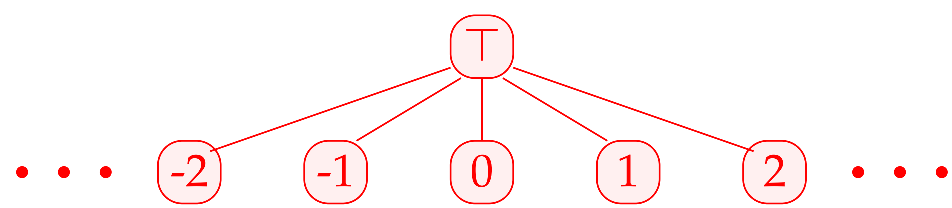
Design an analysis which for every u,

- 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*:

$$\mathbb{Z}^\top = \mathbb{Z} \cup \{\top\} \qquad \text{with} \quad x \sqsubseteq y \quad \text{iff } y = \top \text{ or } x = y$$



Warning: $\mathbb{Z}^\top$ is not a complete lattice in itself :-(

(2) $\mathbb{D} = (\mathit{Vars} \to \mathbb{Z}^\top)_\perp = (\mathit{Vars} \to \mathbb{Z}^\top) \cup \{\perp\}$

// $\perp$ denotes: "not reachable" :-))

with $D_1 \sqsubseteq D_2$ iff $\perp = D_1$ or

$D_1\, x \sqsubseteq D_2\, x \quad (x \in \mathit{Vars})$

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

Warning: $\mathbb{Z}^\top$ is **not** a complete lattice in itself :-(

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

// $\bot$ denotes: "not reachable" :-))

with $D_1 \sqsubseteq D_2$ iff $\bot = D_1$ or

$D_1\, x \sqsubseteq D_2\, x \quad (x \in \mathit{Vars})$

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

Consider $X \subseteq \mathbb{D}$. W.l.o.g., $\bot \not\in X$.

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

If $X = \emptyset$, then $\bigsqcup X = \bot \in \mathbb{D}$:-)

If $\quad X \neq \emptyset \quad$, then $\quad \bigsqcup X = D \quad$ with

$$
\begin{array}{lll}
D\, x & = & \bigsqcup\{f\, x \mid f \in X\} \\
& = & \left\{ \begin{array}{ll} z & \text{if} \quad f\, x = z \quad (f \in X) \\ \top & \text{otherwise} \end{array} \right.
\end{array}
$$

:-))

If $\quad X \neq \emptyset \quad$, then $\quad \bigsqcup X = D \quad$ with

$$
\begin{array}{lll}
D\,x & = & \bigsqcup\{f\,x \mid f \in X\} \\
 & = & \left\{ \begin{array}{ll} z & \text{if} \quad f\,x = z \quad (f \in X) \\ \top & \text{otherwise} \end{array} \right.
\end{array}
$$

:-))

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

Obviously, $\quad [\![lab]\!]^\sharp \perp = \perp \quad$ for all $\quad lab \quad$:-)

Now let $\quad \perp \neq D \in \mathit{Vars} \to \mathbb{Z}^\top$.

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 $\top$:-)

$\Longrightarrow$

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

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

Idea:

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

  $\Longrightarrow$

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

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

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

$$[\![e]\!]^\sharp \; : \; (\mathit{Vars} \to \mathbb{Z}^\top) \to \mathbb{Z}^\top$$

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

$$\begin{array}{lll} [\![c]\!]^\sharp\, D & = & c \\ [\![e_1 \,\Box\, e_2]\!]^\sharp\, D & = & [\![e_1]\!]^\sharp\, D \,\Box^\sharp\, [\![e_2]\!]^\sharp\, D \end{array}$$

... analogously for unary operators :-)

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

$$
\begin{array}{lll}
[\![c]\!]^\sharp\, D & = & c \\
[\![e_1 \,\Box\, e_2]\!]^\sharp\, D & = & [\![e_1]\!]^\sharp\, D \,\Box^\sharp\, [\![e_2]\!]^\sharp\, D
\end{array}
$$

... analogously for unary operators :-)

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

$$
\begin{array}{lll}
[\![x + 7]\!]^\sharp\, D & = & [\![x]\!]^\sharp\, D +^\sharp [\![7]\!]^\sharp\, D \\
 & = & 2 +^\sharp 7 \\
 & = & 9 \\
[\![x - y]\!]^\sharp\, D & = & 2 -^\sharp \top \\
 & = & \top
\end{array}
$$

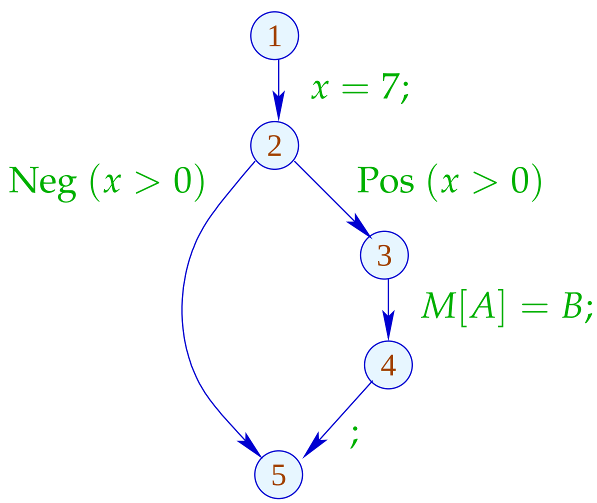
Thus, we obtain the following effects of edges $[\![lab]\!]^\sharp$:

$$
\begin{array}{lll}
[\![;]\!]^\sharp\, D & = & D \\
[\![\mathrm{Pos}\,(e)]\!]^\sharp\, D & = & \left\{\begin{array}{ll} \bot & \text{if} \quad 0 = [\![e]\!]^\sharp\, D \\ D & \text{otherwise} \end{array}\right. \\
[\![\mathrm{Neg}\,(e)]\!]^\sharp\, D & = & \left\{\begin{array}{ll} D & \text{if} \quad 0 \sqsubseteq [\![e]\!]^\sharp\, D \\ \bot & \text{otherwise} \end{array}\right. \\
[\![x = e;]\!]^\sharp\, D & = & D \oplus \{x \mapsto [\![e]\!]^\sharp\, D\} \\
[\![x = M[e];]\!]^\sharp\, D & = & D \oplus \{x \mapsto \top\} \\
[\![M[e_1] = e_2;]\!]^\sharp\, D & = & D
\end{array}
$$

... whenever $\quad D \neq \bot \quad$:-)

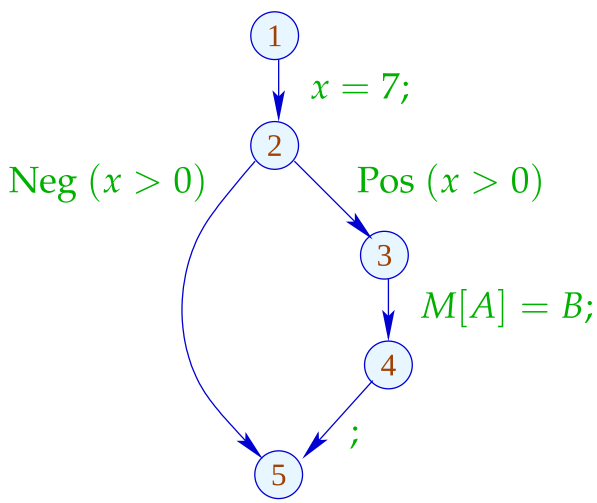
At *start*, we have $D_\top = \{x \mapsto \top \mid x \in \mathit{Vars}\}$.

Example:



At *start*, we have $D_\top = \{x \mapsto \top \mid x \in Vars\}$.

Example:



1	$\{x \mapsto \top\}$
2	$\{x \mapsto 7\}$
3	$\{x \mapsto 7\}$
4	$\{x \mapsto 7\}$
5	$\bot \sqcup \{x \mapsto 7\} = \{x \mapsto 7\}$

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

$$[\![\pi]\!]^\sharp = [\![k_r]\!]^\sharp \circ \ldots \circ [\![k_1]\!]^\sharp \quad : \mathbb{D} \to \mathbb{D}$$

Idea for Correctness: Abstract Interpretation

Cousot, Cousot 1977

Patrick Cousot, ENS, Paris

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

$$[\![\pi]\!]^\sharp = [\![k_r]\!]^\sharp \circ \ldots \circ [\![k_1]\!]^\sharp \quad : \mathbb{D} \to \mathbb{D}$$

Idea for Correctness: Abstract Interpretation

Cousot, Cousot 1977

Establish a description relation Δ between theconcrete values and their descriptions with:

$$x \,\Delta\, a_1 \quad \wedge \quad a_1 \sqsubseteq a_2 \quad \Longrightarrow \quad x \,\Delta\, a_2$$

Concretization: $\gamma\, a = \{x \mid x \,\Delta\, a\}$

// returns the set of described values :-)

(1) Values: $\Delta \subseteq \mathbb{Z} \times \mathbb{Z}^{\top}$

$$z \, \Delta \, a \quad \text{iff} \quad z = a \vee a = \top$$

Concretization:

$$\gamma \, a = \begin{cases} \{a\} & \text{if} \quad a \sqsubset \top \\ \mathbb{Z} & \text{if} \quad a = \top \end{cases}$$

(1) Values: $\Delta \subseteq \mathbb{Z} \times \mathbb{Z}^\top$

$$z \, \Delta \, a \quad \text{iff} \quad z = a \vee a = \top$$

Concretization:

$$\gamma \, a = \begin{cases} \{a\} & \text{if} \quad a \sqsubset \top \\ \mathbb{Z} & \text{if} \quad a = \top \end{cases}$$

(2) Variable Assignments: $\Delta \subseteq (\mathit{Vars} \to \mathbb{Z}) \times (\mathit{Vars} \to \mathbb{Z}^\top)_\perp$

$$\rho \, \Delta \, D \quad \text{iff} \quad D \neq \perp \wedge \rho \, x \sqsubseteq D \, x \quad (x \in \mathit{Vars})$$

Concretization:

$$\gamma \, D = \begin{cases} \emptyset & \text{if} \quad D = \perp \\ \{\rho \mid \forall \, x : \; (\rho \, x) \, \Delta \, (D \, x)\} & \text{otherwise} \end{cases}$$

Example: $\{x \mapsto 1, y \mapsto -7\} \;\Delta\; \{x \mapsto \top, y \mapsto -7\}$

(3) States:

$$\Delta \;\subseteq\; ((\mathit{Vars} \to \mathbb{Z}) \times (\mathbb{N} \to \mathbb{Z})) \times (\mathit{Vars} \to \mathbb{Z}^{\top})_{\perp}$$

$$(\rho, \mu) \;\Delta\; D \qquad \text{iff} \qquad \rho \;\Delta\; D$$

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:

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

$$([\![\pi]\!]\, s) \;\; \Delta \;\; ([\![\pi]\!]^\sharp\, D)$$

$$\begin{tikzcd}
s \arrow[rr, "{[\![\pi]\!]}"] \arrow[d, dash, "\Delta"'] & & s_1 \arrow[d, dash, dotted, "\Delta"] \\
D \arrow[rr, "{[\![\pi]\!]^\sharp}"'] & & D_1
\end{tikzcd}$$

The abstract semantics simulates the concrete semantics :-)

In particular:

$$[\![\pi]\!]\, s \in \gamma\left([\![\pi]\!]^\sharp\, D\right)$$

The abstract semantics simulates the concrete semantics :-)

In particular:

$$[\![\pi]\!]\, s \in \gamma\,([\![\pi]\!]^\sharp\, D)$$

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

$$\begin{array}{lrcll} & \rho'\,x & = & -7 & \text{for all} \quad \rho' \in \gamma\,D \\ \Longrightarrow & \rho_1\,x & = & -7 & \text{for} \quad (\rho_1, \_) = [\![\pi]\!]\, s \end{array}$$

To prove $(*)$, we show for every edge k :

$$
(**) \qquad
\begin{tikzcd}
s \arrow[r, "[\![k]\!]"] \arrow[d, dash, "\Delta"'] & s_1 \arrow[d, dash, dotted, "\Delta"] \\
D \arrow[r, "[\![k]\!]^\sharp"'] & D_1
\end{tikzcd}
$$

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

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

$$(* * *) \quad ([\![e]\!]\,\rho) \;\; \Delta \;\; ([\![e]\!]^\sharp\,D) \quad \text{whenever} \quad \rho\,\Delta\,D$$

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

$(***)$ $\quad ([\![e]\!]\,\rho) \;\Delta\; ([\![e]\!]^\sharp\,D) \quad$ whenever $\quad \rho\,\Delta\,D$

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

$$(x \,\Box\, y) \;\Delta\; (x^\sharp \,\Box^\sharp\, y^\sharp) \qquad \text{whenever} \quad x\,\Delta\,x^\sharp \wedge y\,\Delta\,y^\sharp$$

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

$(***) \quad ([\![e]\!]\,\rho) \;\Delta\; ([\![e]\!]^\sharp\,D)$ whenever $\rho \,\Delta\, D$

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

$$(x \Box y) \;\Delta\; (x^\sharp \Box^\sharp y^\sharp) \qquad \text{whenever} \quad x \,\Delta\, x^\sharp \wedge y \,\Delta\, y^\sharp$$

This precisely was how we have defined the operators $\Box^\sharp$:-)

Now, $(**)$ is proved by case distinction on the edge labels lab .

Let $s = (\rho, \mu) \;\Delta\; D$. In particular, $\perp \neq D \;:\; Vars \to \mathbb{Z}^\top$

Case $\boxed{x = e;}$:

$$\begin{array}{lll} \rho_1 & = & \rho \oplus \{x \mapsto [\![e]\!]\,\rho\} \qquad \mu_1 \;=\; \mu \\ D_1 & = & D \oplus \{x \mapsto [\![e]\!]^\sharp\,D\} \end{array}$$

$$\Longrightarrow \qquad (\rho_1, \mu_1)\;\Delta\;D_1$$

Case $\boxed{x = M[e];}$:

$$\rho_1 = \rho \oplus \{x \mapsto \mu\left(\llbracket e \rrbracket^\sharp \rho\right)\} \qquad \mu_1 = \mu$$
$$D_1 = D \oplus \{x \mapsto \top\}$$

$$\Longrightarrow \qquad (\rho_1, \mu_1) \;\Delta\; D_1$$

Case $\boxed{M[e_1] = e_2;}$:

$$\rho_1 = \rho \qquad \mu_1 = \mu \oplus \{\llbracket e_1 \rrbracket^\sharp \rho \mapsto \llbracket e_2 \rrbracket^\sharp \rho\}$$
$$D_1 = D$$

$$\Longrightarrow \qquad (\rho_1, \mu_1) \;\Delta\; D_1$$

Case $\boxed{\mathrm{Neg}(e)}$: $\quad (\rho_1, \mu_1) = s \quad$ where:

$$
\begin{array}{rlll}
& 0 & = & [\![e]\!]\,\rho \\
& & \Delta & [\![e]\!]^\sharp\,D \\
\Longrightarrow & 0 & \sqsubseteq & [\![e]\!]^\sharp\,D \\
\Longrightarrow & \bot & \neq & D_1 \;=\; D \\
\Longrightarrow & \multicolumn{3}{l}{(\rho_1, \mu_1)\;\Delta\;D_1}
\end{array}
$$

Case $\boxed{\mathrm{Pos}(e)}$: $(\rho_1, \mu_1) = s$ where:

$$
\begin{array}{rlll}
 & 0 & \neq & [\![e]\!]\,\rho \\
 & & \Delta & [\![e]\!]^\sharp\,D \\
\Longrightarrow & 0 & \neq & [\![e]\!]^\sharp\,D \\
\Longrightarrow & \perp & \neq & D_1 \;=\; D \\
\Longrightarrow & (\rho_1, \mu_1)\;\Delta\;D_1 & &
\end{array}
$$

:-)

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

The MOP-Solution:

$$\mathcal{D}^*[v] \;=\; \bigsqcup\{[\![\pi]\!]^\sharp\, D_\top \mid \pi : start \rightarrow^* v\}$$

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

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

The MOP-Solution:

$$\mathcal{D}^*[v] = \bigsqcup \{ [\![\pi]\!]^\sharp \, D_\top \mid \pi : start \to^* v \}$$

where $D_\top \, x = \top \qquad (x \in Vars)$.

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

$$([\![\pi]\!] \, s) \;\; \Delta \;\; (\mathcal{D}^*[v])$$

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

The MOP-Solution

$$\mathcal{D}^*[v] \;=\; \bigsqcup\{[\![\pi]\!]^\sharp\, D_\top \mid \pi : start \to^* v\}$$

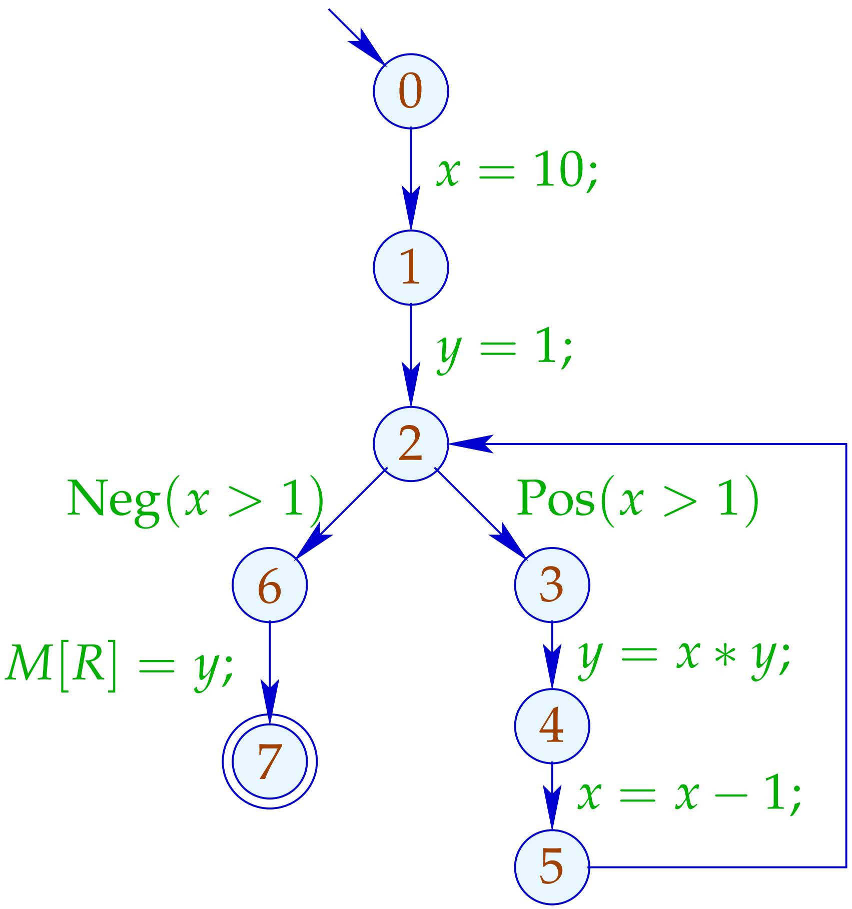
where $\quad D_\top\, x = \top \qquad (x \in Vars)\;.$

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

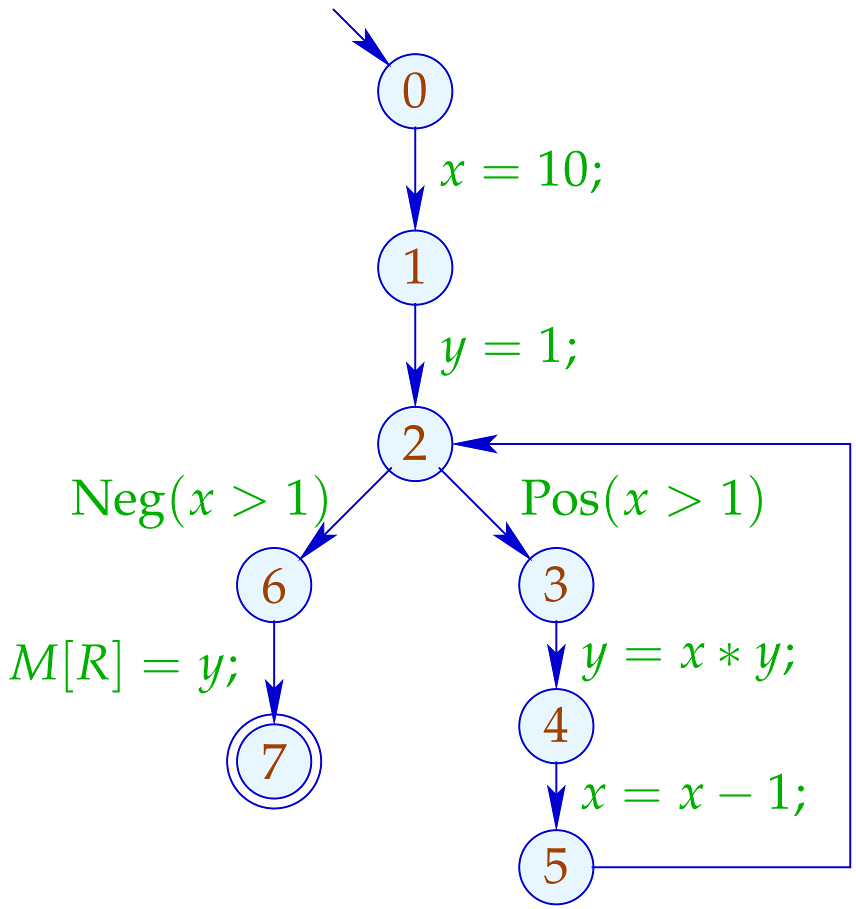
$$([\![\pi]\!]\; s) \;\;\Delta\;\; (\mathcal{D}^*[v])$$

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

Example:

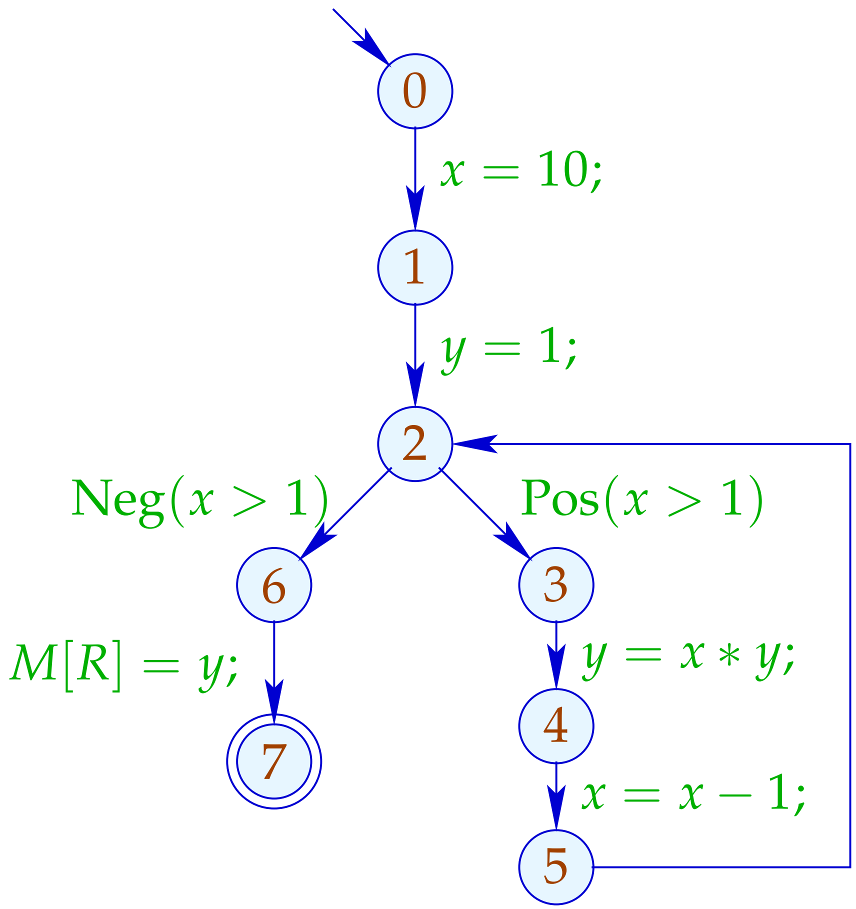


Example:



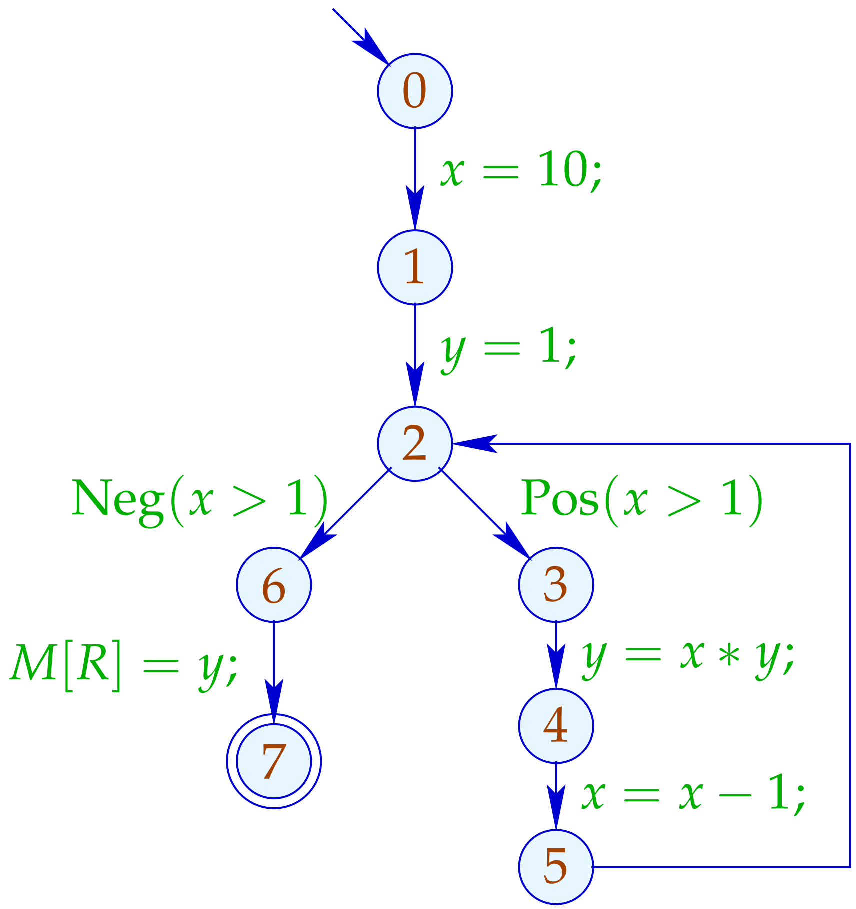
	1	
	x	y
0	$\top$	$\top$
1	10	$\top$
2	10	1
3	10	1
4	10	10
5	9	10
6	$\bot$	
7	$\bot$	

Example:



	1		2	
	x	y	x	y
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	$\perp$		$\top$	$\top$
7	$\perp$		$\top$	$\top$

Example:



	1		2		3	
	x	y	x	y	x	y
0	$\top$	$\top$	$\top$	$\top$		
1	10	$\top$	10	$\top$		
2	10	1	$\top$	$\top$		
3	10	1	$\top$	$\top$		
4	10	10	$\top$	$\top$	dito	
5	9	10	$\top$	$\top$		
6	$\perp$		$\top$	$\top$		
7	$\perp$		$\top$	$\top$		

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: $n \cdot (m + 1)$ rounds :-)

Warning:

The effects of edge are not distributive !!!

Counter Example: $\quad f \;=\; [\![x = x + y;]\!]^\sharp$

$$\begin{array}{lrcl}
\text{Let} & D_1 & = & \{x \mapsto 2, y \mapsto 3\} \\
& D_2 & = & \{x \mapsto 3, y \mapsto 2\} \\
\text{Dann} & f\,D_1 \sqcup f\,D_2 & = & \{x \mapsto 5, y \mapsto 3\} \sqcup \{x \mapsto 5, y \mapsto 2\} \\
& & = & \{x \mapsto 5, y \mapsto \top\} \\
& & \neq & \{x \mapsto \top, y \mapsto \top\} \\
& & = & f\,\{x \mapsto \top, y \mapsto \top\} \\
& & = & f\,(D_1 \sqcup D_2)
\end{array}$$

:-((

We conclude:

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

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

We conclude:

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

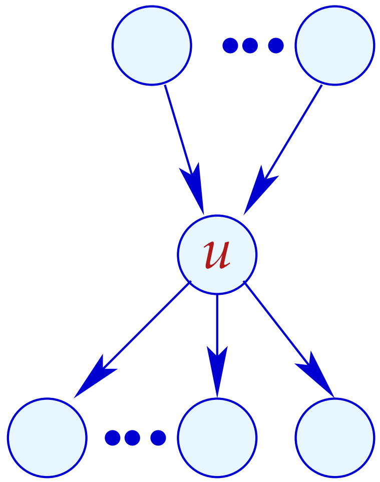
$$\mathcal{D}^*[v] \quad \sqsubseteq \quad \mathcal{D}[v]$$

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

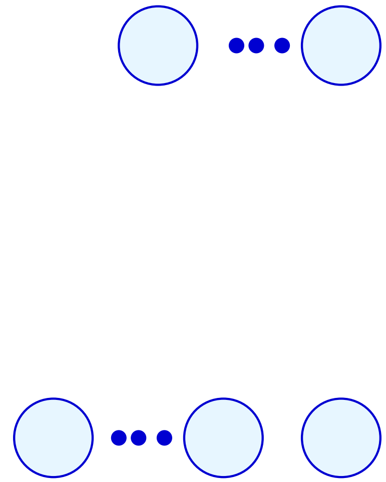
$$(\llbracket \pi \rrbracket\, (\rho, \mu)) \;\; \Delta \;\; (\mathcal{D}[v])$$

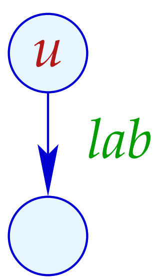
whenever $\llbracket \pi \rrbracket\, (\rho, \mu)$ is defined ;-))

Transformation 4: Removal of Dead Code

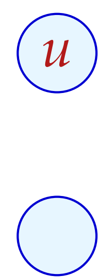


$$\mathcal{D}[u] = \bot$$

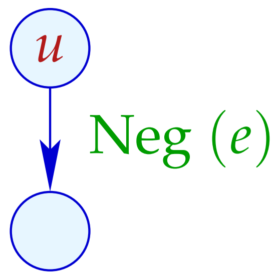




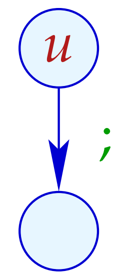
$$[\![lab]\!]^\sharp(\mathcal{D}[u]) = \bot$$

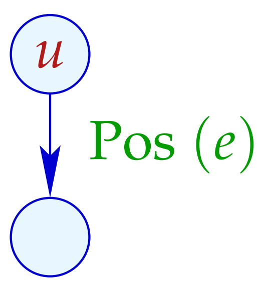


Transformation 4 (cont.): Removal of Dead Code

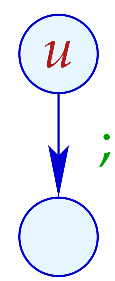


$$\bot \neq \mathcal{D}[u] = D$$
$$[\![e]\!]^\sharp\, D = 0$$

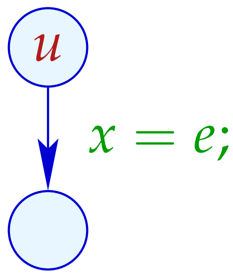




$$\bot \neq \mathcal{D}[u] = D$$
$$[\![e]\!]^\sharp\, D \not\in \{0, \top\}$$



Transformation 4 (cont.): Simplified Expressions



$$\bot \neq \mathcal{D}[u] = D$$
$$[\![e]\!]^\sharp\, D = c$$