

Die abstrakten Kanten-Effekte $\llbracket k \rrbracket^\sharp$ setzen wir wieder zu den Effekten von Pfaden $\pi = k_1 \dots k_r$ zusammen durch:

$$\llbracket \pi \rrbracket^\sharp = \llbracket k_r \rrbracket^\sharp \circ \dots \circ \llbracket k_1 \rrbracket^\sharp \quad : \mathbb{D} \rightarrow \mathbb{D}$$

Idee zur Korrektheit:

Abstrakte Interpretation

Cousot, Cousot 1977

Aufstellen einer Beschreibungsrelation Δ zwischen **konkreten** Werten und deren Beschreibungen mit:

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

Konkretisierung: $\gamma a = \{x \mid x \Delta a\}$

// liefert Menge der beschriebenen Werte :-)

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

$$z \Delta a \quad \text{gdw.} \quad z = a \vee a = \top$$

Konkretisierung:

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

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

$$z \Delta a \text{ gdw. } z = a \vee a = \top$$

Konkretisierung:

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

(2) **Variablenbelegungen:** $\Delta \subseteq (\text{Vars} \rightarrow \mathbb{Z}) \times (\text{Vars} \rightarrow \mathbb{Z}^\top)_\perp$

$$(\rho, \mu) \Delta D \text{ gdw. } D \neq \perp \wedge \rho x \sqsubseteq D x \quad (x \in \text{Vars})$$

Konkretisierung:

$$\gamma D = \begin{cases} \emptyset & \text{falls } D = \perp \\ \{\rho \mid \forall x : (\rho x) \Delta (D x)\} & \text{sonst} \end{cases}$$

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

(3) Zustände:

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

$$(\rho, \mu) \Delta D \quad \text{gdw.} \quad \rho \Delta D$$

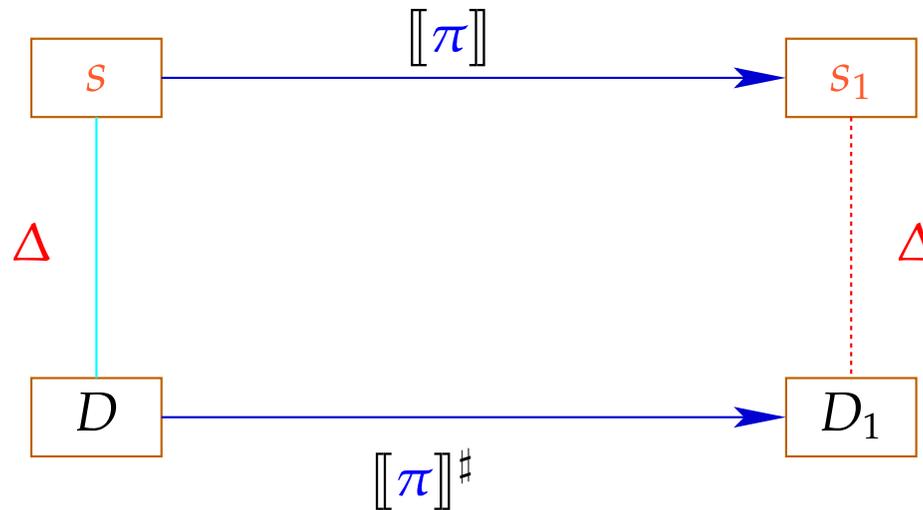
Konkretisierung:

$$\gamma D = \begin{cases} \emptyset & \text{falls } D = \perp \\ \{(\rho, \mu) \mid \forall x : (\rho x) \Delta (D x)\} & \text{sonst} \end{cases}$$

Wir zeigen:

(*) Gilt $s \Delta D$ und ist $[[\pi]]s$ definiert, dann gilt auch:

$$([[\pi]] s) \Delta ([[\pi]]^\# D)$$



Die abstrakte Semantik simuliert die konkrete :-)

Insbesondere gilt:

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

Die abstrakte Semantik simuliert die konkrete :-)

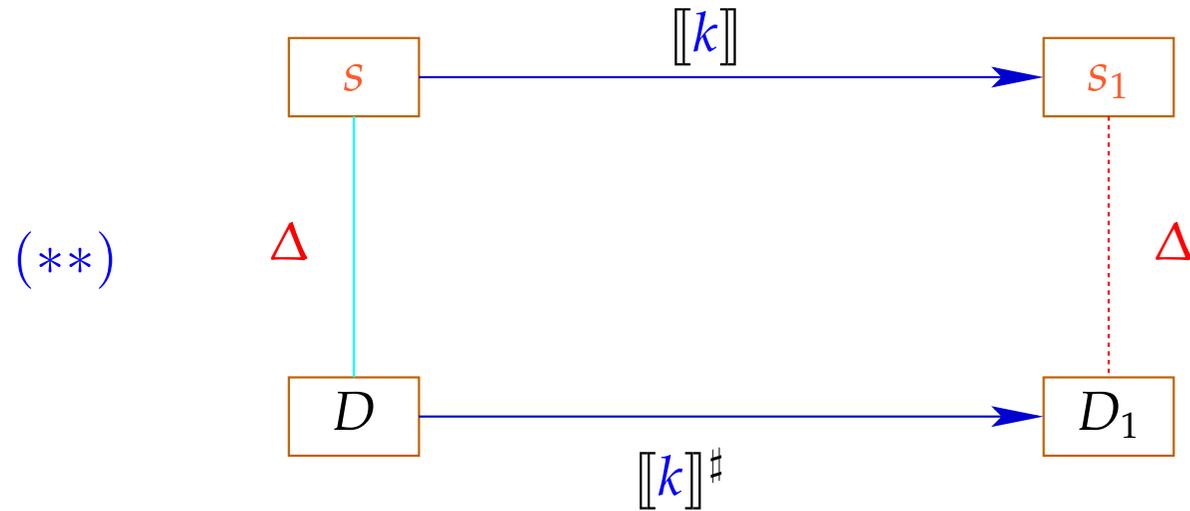
Insbesondere gilt:

$$\llbracket \pi \rrbracket s \in \gamma (\llbracket \pi \rrbracket^\# D)$$

Praktisch heißt das z.B., dass für $D x = -7$ gilt:

$$\begin{aligned} \rho' x &= -7 \text{ für alle } \rho' \in \gamma D \\ \implies \rho_1 x &= -7 \text{ für } (\rho_1, _) = \llbracket \pi \rrbracket s \end{aligned}$$

Zum Beweis von $(*)$ zeigen wir für jede Kante k :



Dann folgt $(*)$ mittels Induktion $:-)$

Zum Beweis von $(**)$ zeigen wir für jeden Ausdruck e :

$(***)$ $(\llbracket e \rrbracket \rho) \Delta (\llbracket e \rrbracket^\# D)$ sofern nur $\rho \Delta D$

Zum Beweis von $(**)$ zeigen wir für jeden Ausdruck e :

$(***)$ $(\llbracket e \rrbracket \rho) \Delta (\llbracket e \rrbracket^\# D)$ sofern nur $\rho \Delta D$

Zum Beweis von $(***)$ zeigen wir für jeden Operator \square :

$(x \square y) \Delta (x^\# \square^\# y^\#)$ sofern $x \Delta x^\# \wedge y \Delta y^\#$

Zum Beweis von $(**)$ zeigen wir für jeden Ausdruck e :

$$(***) \quad (\llbracket e \rrbracket \rho) \Delta (\llbracket e \rrbracket^\# D) \quad \text{sofern nur } \rho \Delta D$$

Zum Beweis von $(***)$ zeigen wir für jeden Operator \square :

$$(x \square y) \Delta (x^\# \square^\# y^\#) \quad \text{sofern } x \Delta x^\# \wedge y \Delta y^\#$$

So hatten wir die Operatoren $\square^\#$ aber gerade definiert :-)

Nun zeigen wir (**) durch Fallunterscheidung nach der Kanten-Beschriftung *lab*.

Sei $s = (\rho, \mu) \Delta D$. Insbesondere ist $\perp \neq D : Vars \rightarrow \mathbb{Z}^\top$

Fall $x = e;$:

$$\rho_1 = \rho \oplus \{x \mapsto \llbracket e \rrbracket \rho\} \quad \mu_1 = \mu$$

$$D_1 = D \oplus \{x \mapsto \llbracket e \rrbracket^\# D\}$$

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

Fall $x = M[R];$:

$$\rho_1 = \rho \oplus \{x \mapsto \mu(\rho R)\} \quad \mu_1 = \mu$$

$$D_1 = D \oplus \{x \mapsto \top\}$$

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

Fall $M[R_1] = R_2;$:

$$\rho_1 = \rho \quad \mu_1 = \mu \oplus \{\rho R_1 \mapsto \rho R_2\}$$

$$D_1 = D$$

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

Fall $\boxed{\text{Neg}(e)}$:

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

$$0 = \llbracket e \rrbracket \rho$$

$$\Delta \llbracket e \rrbracket^\# D$$

$$\implies 0 \sqsubseteq \llbracket e \rrbracket^\# D$$

$$\implies \perp \neq D_1 = D$$

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

Fall $\boxed{\text{Pos}(e)}$: $(\rho_1, \mu_1) = s$, wobei:

$$0 \neq \llbracket e \rrbracket \rho$$

$$\Delta \llbracket e \rrbracket^\# D$$

$$\implies 0 \neq \llbracket e \rrbracket^\# D$$

$$\implies \perp \neq D_1 = D$$

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

:-)

Wir schließen: Die Behauptung $(*)$ stimmt :-))

Die MOP-Lösung:

$$\mathcal{D}^*[v] = \bigsqcup \{ \llbracket \pi \rrbracket^\# D_0 \mid \pi : start \rightarrow^* v \}$$

wobei $D_0 x = \top$ ($x \in Vars$).

Wir schließen: Die Behauptung $(*)$ stimmt $(:-))$

Die MOP-Lösung:

$$\mathcal{D}^*[v] = \bigsqcup \{ \llbracket \pi \rrbracket^\# D_0 \mid \pi : start \rightarrow^* v \}$$

wobei $D_0 x = \top$ ($x \in Vars$).

Wegen $(*)$ gilt für alle Anfangszustände s und alle Berechnungen π , die v erreichen:

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

Wir schließen: Die Behauptung $(*)$ stimmt $:-))$

Die MOP-Lösung:

$$\mathcal{D}^*[v] = \bigsqcup \{ \llbracket \pi \rrbracket^\# D_0 \mid \pi : start \rightarrow^* v \}$$

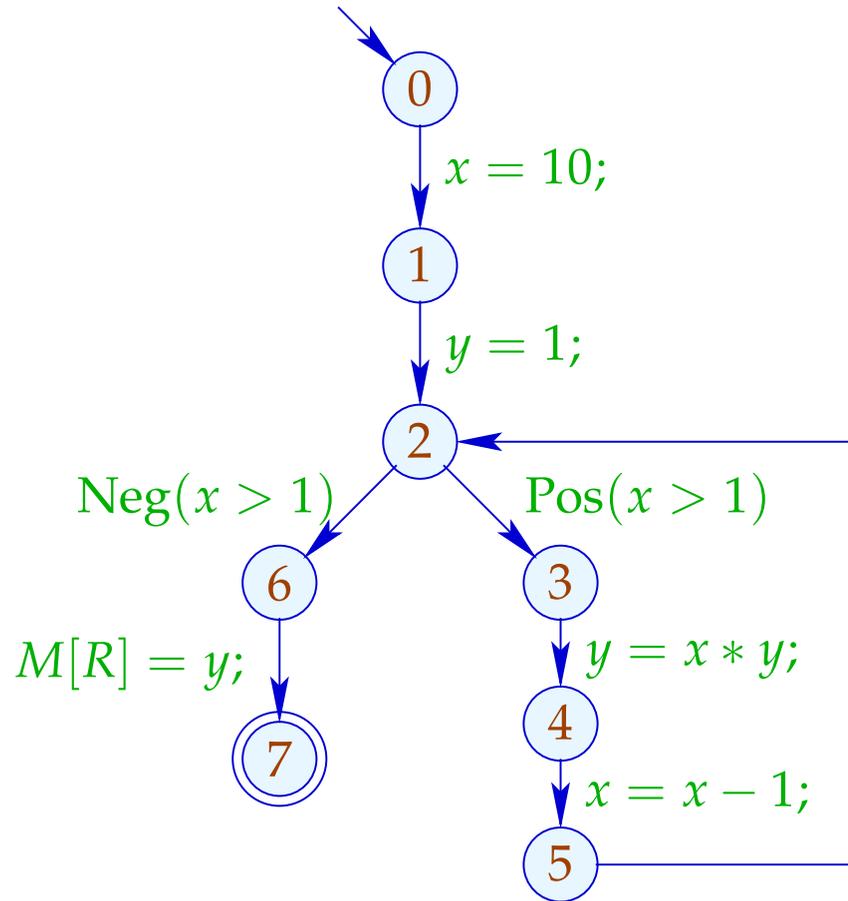
wobei $D_0 x = \top$ ($x \in Vars$).

Wegen $(*)$ gilt für alle Anfangszustände s und alle Berechnungen π , die v erreichen:

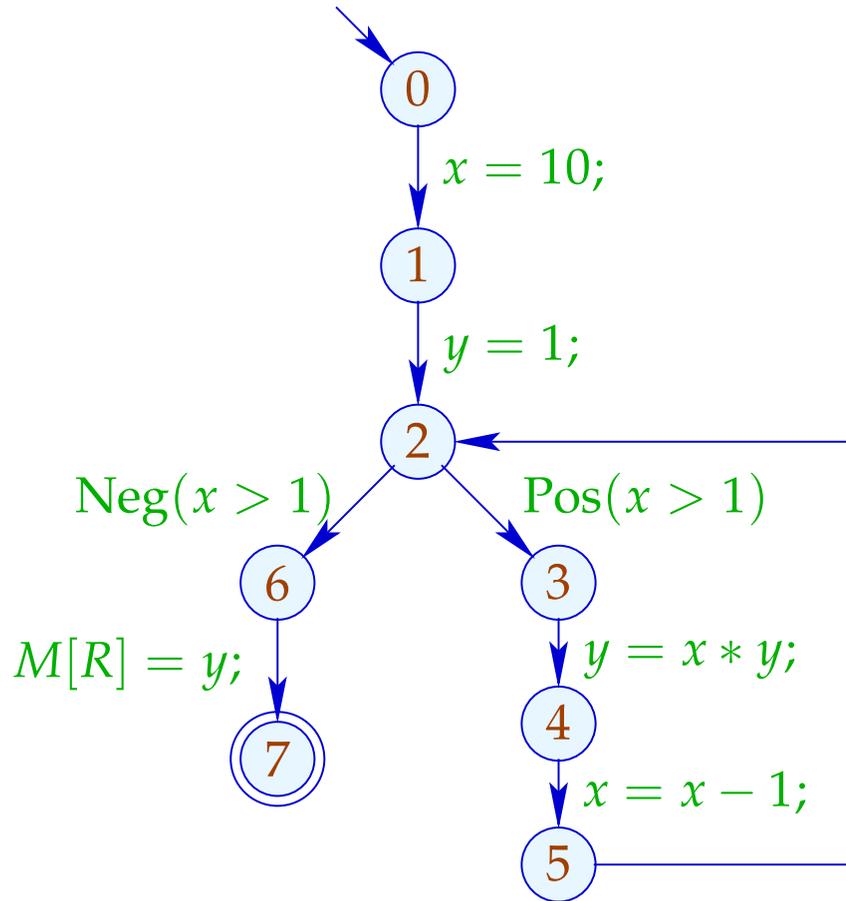
$$(\llbracket \pi \rrbracket s) \Delta (\mathcal{D}^*[v])$$

Zur Approximation des MOP benutzen wir unser Constraint-System $:-))$

Beispiel:

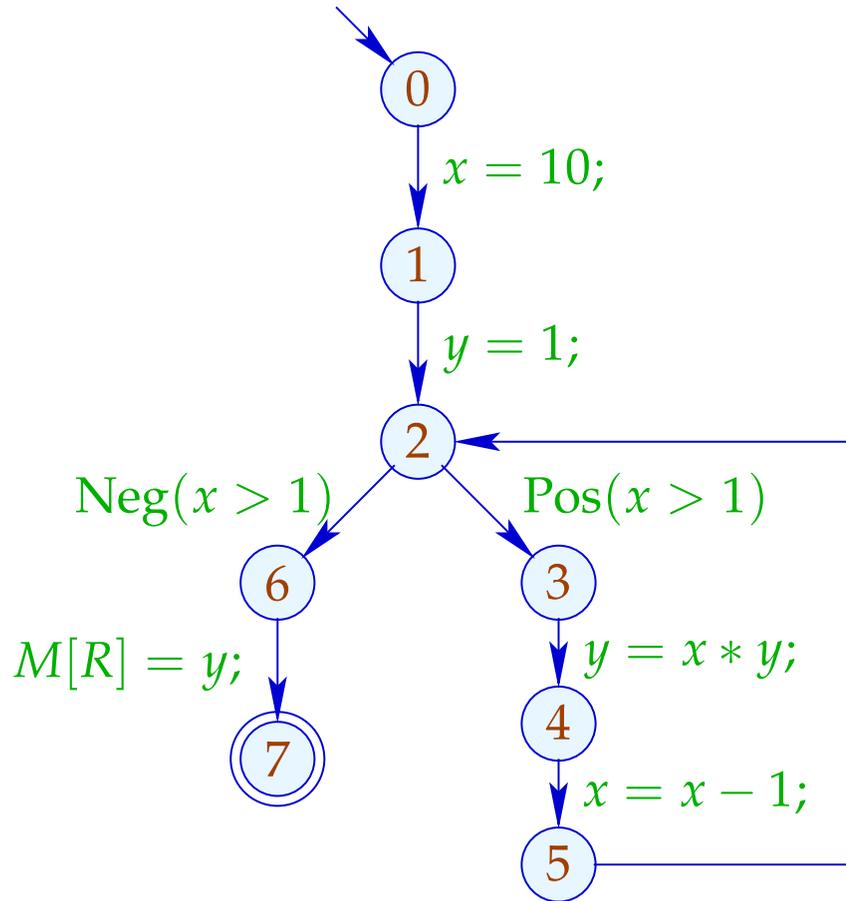


Beispiel:



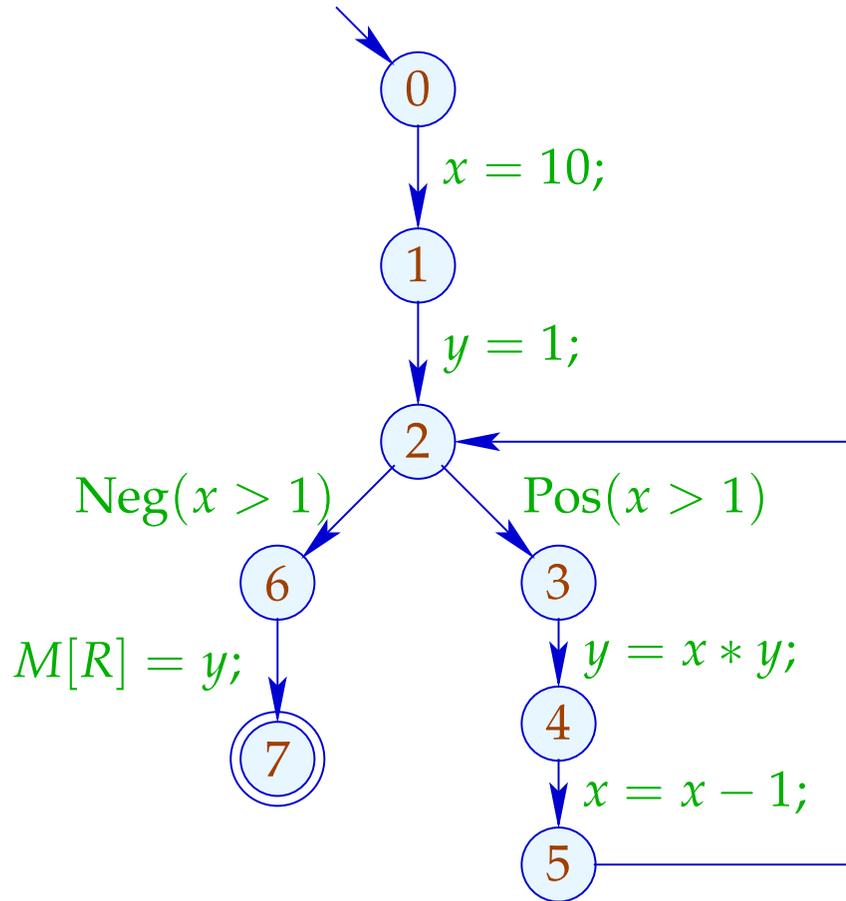
	1	
	x	y
0	⊤	⊤
1	10	⊤
2	10	1
3	10	1
4	10	10
5	9	10
6	⊥	
7	⊥	

Beispiel:



	1		2	
	x	y	x	y
0	⊤	⊤	⊤	⊤
1	10	⊤	10	⊤
2	10	1	⊤	⊤
3	10	1	⊤	⊤
4	10	10	⊤	⊤
5	9	10	⊤	⊤
6	⊥		⊤	⊤
7	⊥		⊤	⊤

Beispiel:



	1		2		3	
	x	y	x	y	x	y
0	⊤	⊤	⊤	⊤		
1	10	⊤	10	⊤		
2	10	1	⊤	⊤		
3	10	1	⊤	⊤		
4	10	10	⊤	⊤	dito	
5	9	10	⊤	⊤		
6	⊥		⊤	⊤		
7	⊥		⊤	⊤		

Fazit:

Obwohl wir mit konkreten Zahlen rechnen, kriegen wir nicht **alles** raus :-)

Dafür terminiert die Fixpunkt-Iteration garantiert:

Für n Programmpunkte und m Variablen benötigen wir maximal: $n \cdot (m + 1)$ Runden :-)

Achtung:

Die Kanten-Effekte sind **nicht distributiv !!!**

Gegenbeispiel: $f = \llbracket x = x + y; \rrbracket^\#$

Sei $D_1 = \{x \mapsto 2, y \mapsto 3\}$

$$D_2 = \{x \mapsto 3, y \mapsto 2\}$$

Dann $f D_1 \sqcup f D_2 = \{x \mapsto 5, y \mapsto 3\} \sqcup \{x \mapsto 5, y \mapsto 2\}$

$$= \{x \mapsto 5, y \mapsto \top\}$$

$$\neq \{x \mapsto \top, y \mapsto \top\}$$

$$= f \{x \mapsto \top, y \mapsto \top\}$$

$$= f (D_1 \sqcup D_2)$$

:-((

Wir schließen:

Die kleinste Lösung \mathcal{D} des Constraint-Systems liefert i.a. nur eine **obere Approximation** des MOP, d.h.:

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

Wir schließen:

Die kleinste Lösung \mathcal{D} des Constraint-Systems liefert i.a. nur eine **obere Approximation** des MOP, d.h.:

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

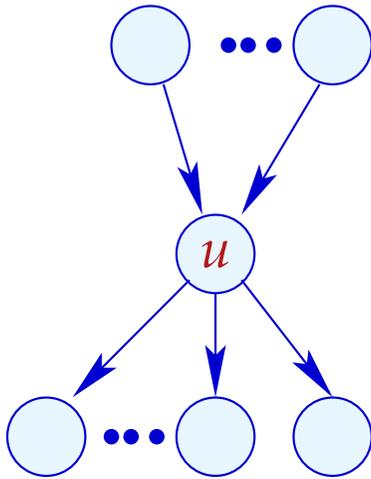
Als obere Approximation **beschreibt** $\mathcal{C}[v]$ trotzdem das Ergebnis jeder Berechnung π , die in v endet:

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

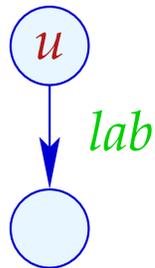
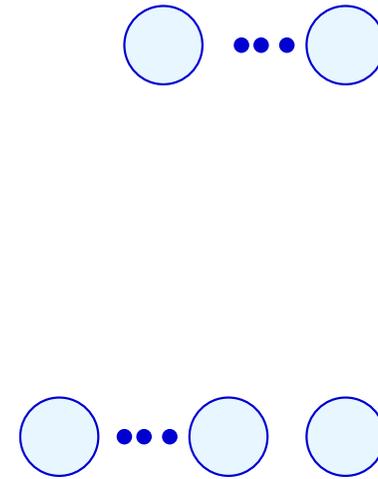
wann immer $\llbracket \pi \rrbracket (\rho, \mu)$ definiert ist **;-))**

Transformation 5:

Beseitigung von totem Code



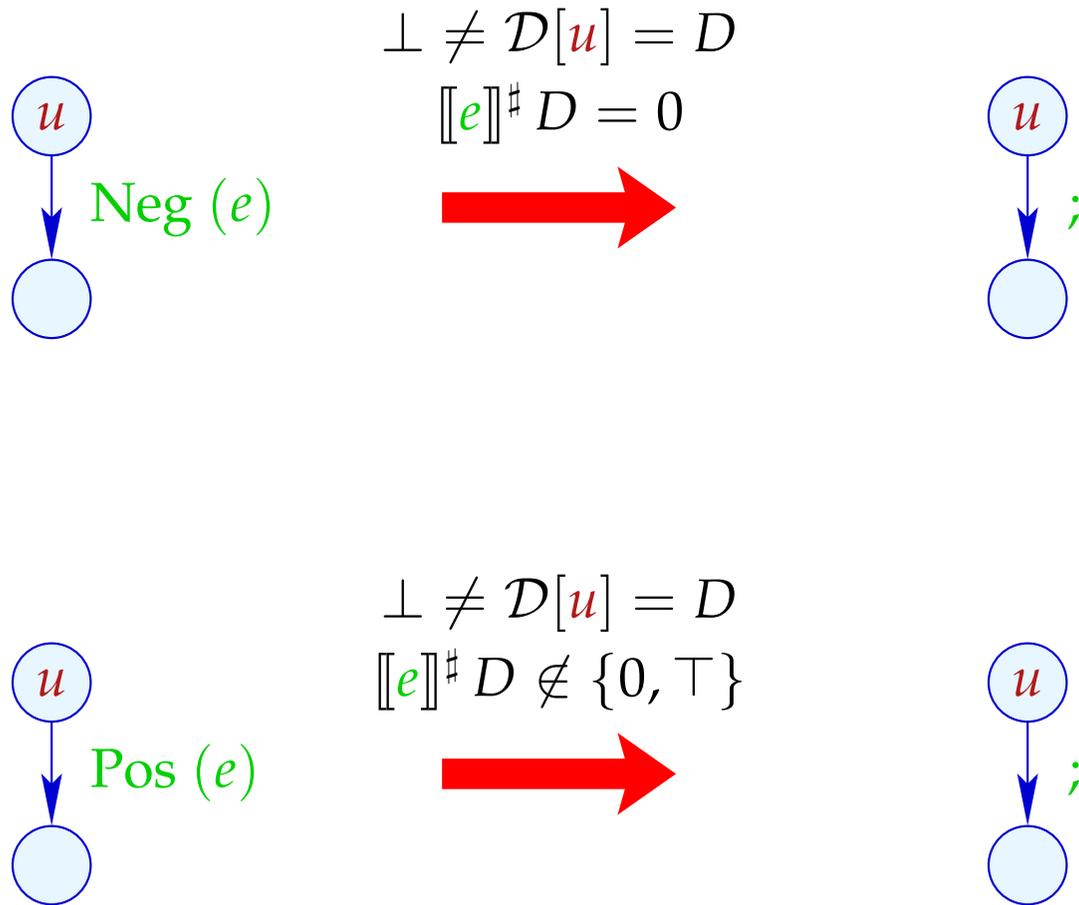
$$\mathcal{D}[u] = \perp$$

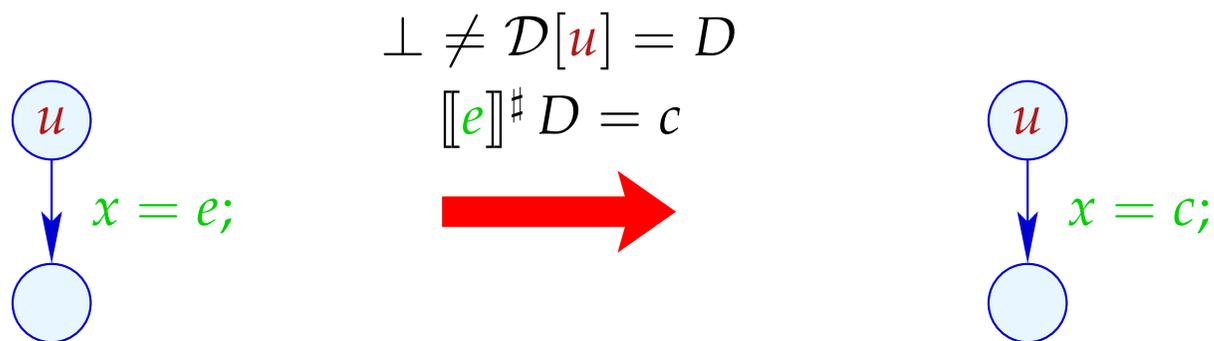
$$[[lab]]^\#(\mathcal{D}[u]) = \perp$$




Transformation 5 (Forts.): Beseitigung von totem Code



Transformation 5 (Forts.): Vereinfachte Zuweisungen



Erweiterungen:

- Statt ganzer rechter Seiten kann man auch Teilausdrücke vereinfachen:

$$x + (3 * y) \xrightarrow{\{x \mapsto \top, y \mapsto 5\}} x + 15$$

... und weitere Vereinfachungsregeln anwenden, etwa:

$$x * 0 \implies 0$$

$$x * 1 \implies x$$

$$x + 0 \implies x$$

$$x - 0 \implies x$$

...

- Bisher haben wir die Information von **Bedingungen** nicht optimal ausgenutzt:

```

if (x == 7)
    y = x + 3;

```

Selbst wenn wir den Wert von x vor der if-Abfrage nicht kennen, wissen wir doch, dass **bei Betreten** des then-Teils x stets den Wert 7 hat :-)

Wir könnten darum definieren:

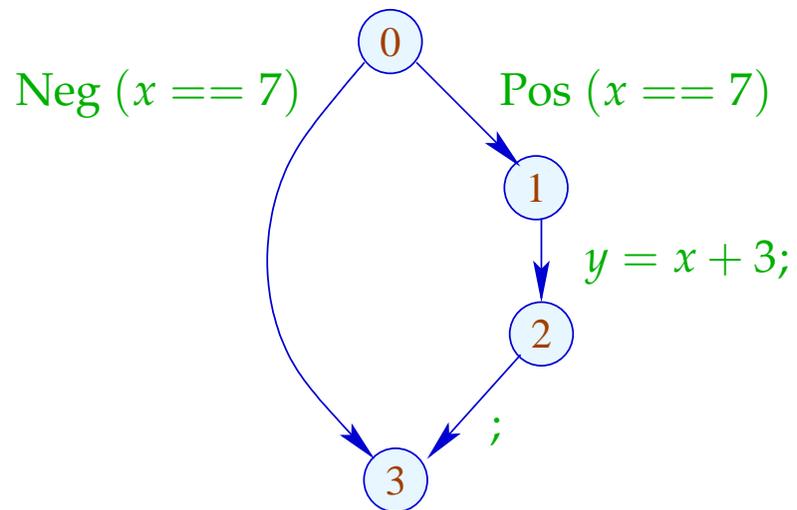
$$[[\text{Pos}(x == e)]]^\# D = \begin{cases} D & \text{falls } [[x == e]]^\# D = 1 \\ \perp & \text{falls } [[x == e]]^\# D = 0 \\ D_1 & \text{sonst} \end{cases}$$

wobei

$$D_1 = D \oplus \{x \mapsto (D \ x \sqcap [[e]]^\# D)\}$$

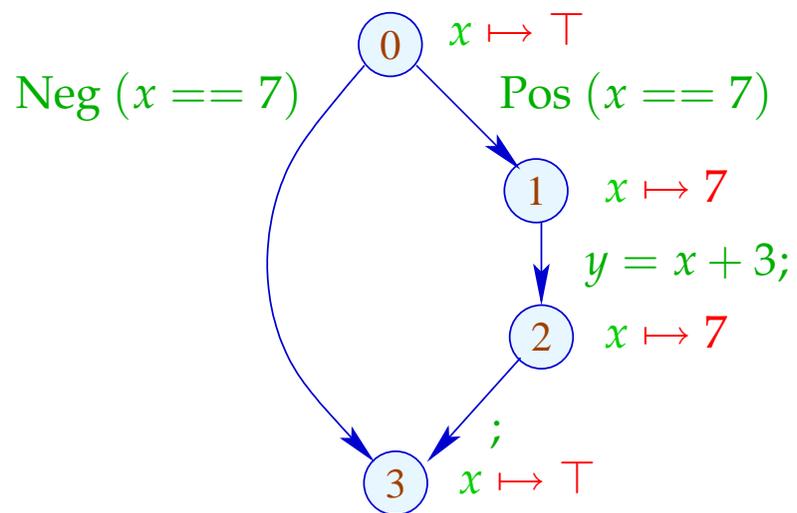
Analog sieht der Kanteneffekt für $\text{Neg}(x \neq e)$ aus :-)

Unser Beispiel:



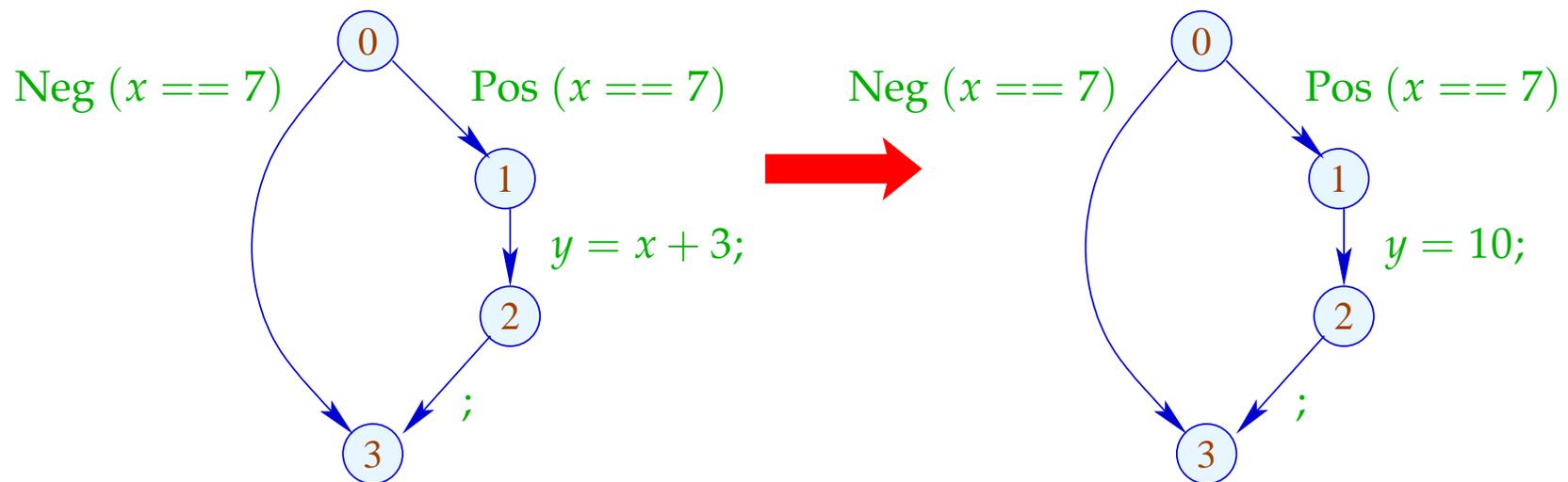
Analog sieht der Kanteneffekt für $\text{Neg}(x \neq e)$ aus :-)

Unser Beispiel:



Analog sieht der Kanteneffekt für $\text{Neg}(x \neq e)$ aus :-)

Unser Beispiel:



1.5 Intervall-Analyse

Beobachtung:

- Programmiererinnen benutzen oft globale Konstanten, um Debug-Code ein oder aus zu schalten



Konstantenpropagation ist hilfreich :-)

- Im allgemeinen wird aber der Wert von Variablen nicht bekannt sein — möglicherweise aber ein **Intervall !!!**

Beispiel:

```
for ( $i = 0; i < 42; i++$ )  
    if ( $0 \leq i \wedge i < 42$ ) {  
         $A_1 = A + i;$   
         $M[A_1] = i;$   
    }  
// A Anfangsadresse eines Felds  
// if ist Array-Bound-Check
```

Offenbar ist die innere Abfrage überflüssig :-)

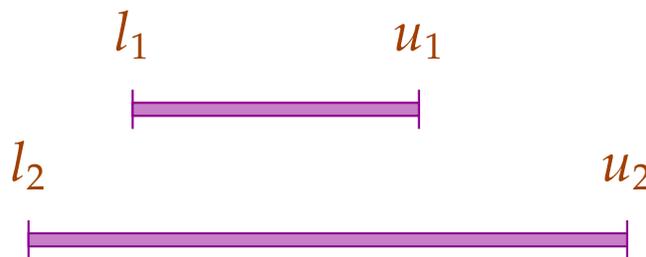
Idee 1:

Bestimme für jede Variable x ein (möglichst kleines :-) Intervall für die möglichen Werte:

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

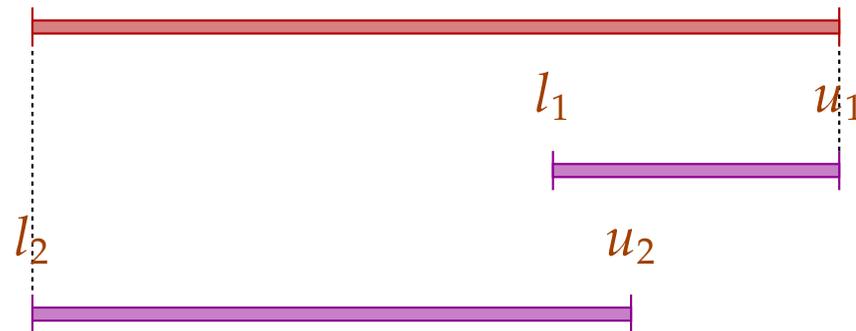
Partielle Ordnung:

$$[l_1, u_1] \sqsubseteq [l_2, u_2] \quad \text{gdw.} \quad l_2 \leq l_1 \wedge u_1 \leq u_2$$



Damit:

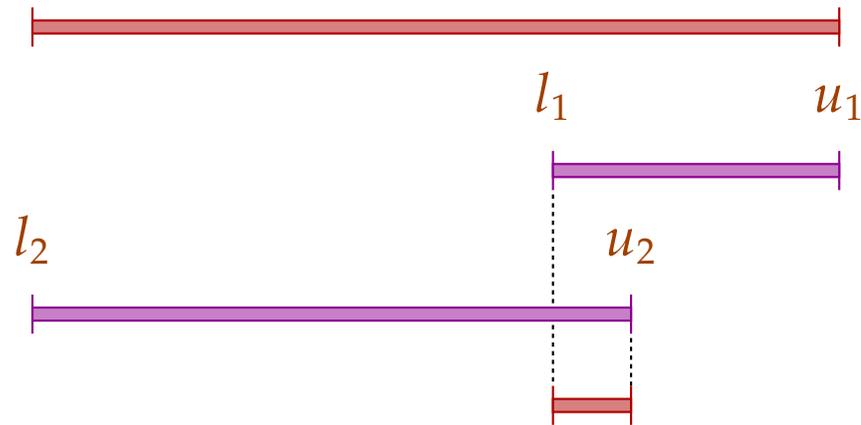
$$[l_1, u_1] \sqcup [l_2, u_2] = [l_1 \sqcap l_2, u_1 \sqcup u_2]$$



Damit:

$$[l_1, u_1] \sqcup [l_2, u_2] = [l_1 \sqcap l_2, u_1 \sqcup u_2]$$

$$[l_1, u_1] \sqcap [l_2, u_2] = [l_1 \sqcup l_2, u_1 \sqcap u_2] \quad \text{sofern } (l_1 \sqcup l_2) \leq (u_1 \sqcap u_2)$$



Achtung:

- \mathbb{I} ist kein vollständiger Verband :-)
- \mathbb{I} besitzt **unendliche aufsteigende Ketten**, z.B.

$$[0, 0] \sqsubset [0, 1] \sqsubset [-1, 1] \sqsubset [-1, 2] \sqsubset \dots$$

Achtung:

- \mathbb{I} ist kein vollständiger Verband :-)
- \mathbb{I} besitzt unendliche aufsteigende Ketten, z.B.

$$[0, 0] \sqsubset [0, 1] \sqsubset [-1, 1] \sqsubset [-1, 2] \sqsubset \dots$$

Beschreibungsrelation:

$$z \Delta [l, u] \quad \text{gdw.} \quad l \leq z \leq u$$

Konkretisierung:

$$\gamma[l, u] = \{z \in \mathbb{Z} \mid l \leq z \leq u\}$$

Beispiel:

$$\begin{aligned}\gamma[0,7] &= \{0, \dots, 7\} \\ \gamma[0, \infty] &= \{0, 1, 2, \dots, \}\end{aligned}$$

Rechnen mit Intervallen: Intervall-Arithmetik :-)

Addition:

$$\begin{aligned}[l_1, u_1] +^\# [l_2, u_2] &= [l_1 + l_2, u_1 + u_2] \quad \text{wobei} \\ -\infty +_- &= -\infty \\ +\infty +_- &= +\infty \\ // \quad -\infty + \infty &\text{ kommt nicht vor :-)}\end{aligned}$$

Negation:

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

Multiplikation:

$$\begin{aligned} [l_1, u_1] *^\# [l_2, u_2] &= [a, b] \quad \text{wobei} \\ a &= l_1 l_2 \sqcap l_1 u_2 \sqcap u_1 l_2 \sqcap u_1 u_2 \\ b &= l_1 l_2 \sqcup l_1 u_2 \sqcup u_1 l_2 \sqcup u_1 u_2 \end{aligned}$$

Beispiel:

$$\begin{aligned} [0, 2] *^\# [3, 4] &= [0, 8] \\ [-1, 2] *^\# [3, 4] &= [-4, 8] \\ [-1, 2] *^\# [-3, 4] &= [-6, 8] \\ [-1, 2] *^\# [-4, -3] &= [-8, 4] \end{aligned}$$

Division: $[l_1, u_1] /^\# [l_2, u_2] = [a, b]$

- Ist 0 **nicht** im Nenner-Intervall enthalten, sei:

$$a = l_1/l_2 \sqcap l_1/u_2 \sqcap u_1/l_2 \sqcap u_1/u_2$$

$$b = l_1/l_2 \sqcup l_1/u_2 \sqcup u_1/l_2 \sqcup u_1/u_2$$

- Gilt: $l_2 \leq 0 \leq u_2$, setzen wir:

$$[a, b] = [-\infty, +\infty]$$

Gleichheit:

$$[l_1, u_1] ==^\# [l_2, u_2] = \begin{cases} [1, 1] & \text{falls } l_1 = u_1 = l_2 = u_2 \\ [0, 0] & \text{falls } u_1 < l_2 \vee u_2 < l_1 \\ [0, 1] & \text{sonst} \end{cases}$$

Gleichheit:

$$[l_1, u_1] ==^\# [l_2, u_2] = \begin{cases} [1, 1] & \text{falls } l_1 = u_1 = l_2 = u_2 \\ [0, 0] & \text{falls } u_1 < l_2 \vee u_2 < l_1 \\ [0, 1] & \text{sonst} \end{cases}$$

Beispiel:

$$\begin{aligned} [42, 42] ==^\# [42, 42] &= [1, 1] \\ [0, 7] ==^\# [0, 7] &= [0, 1] \\ [1, 2] ==^\# [3, 4] &= [0, 0] \end{aligned}$$