1.2 Beseitigung überflüssiger Zuweisungen

Beispiel:

$$\begin{array}{ll} 1: & x = y + 2; \\ 2: & y = 5; \\ 3: & x = y + 3; \end{array}$$

Der Wert von x an den Programmpunkten $1, 2$ wird überschrieben, bevor er benutzt werden kann.

Die Variable x nennen wir deshalb an diesen Programmpunkten tot :-)

Beachte:

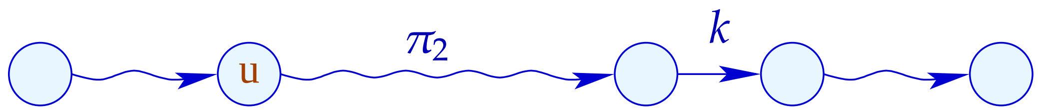
$\longrightarrow$ Zuweisungen an tote Variablen können wir uns schenken ;-)

$\longrightarrow$ Solche Ineffizienzen können u.a. durch andere Transformationen hervorgerufen werden.

Formale Definition:

Die Variable x heißt lebendig an u entlang des Pfads π, falls sich π zerlegen lässt in $\pi = \pi_1\,\pi_2\,k\,\pi_3$ so dass gilt:

- π_1 erreicht u;
- k ist eine Benutzung von x;
- π_2 enthält keine Überschreibung von x.

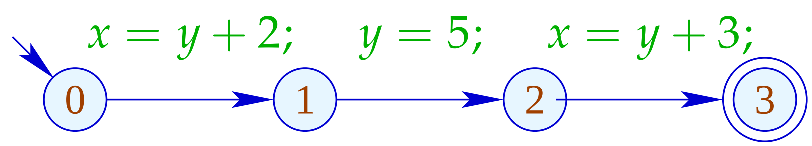


Die Menge der an einer Kante $k = (\_, lab, \_)$ benutzten bzw. überschriebenen Variablen ist dabei gegeben durch:

lab	benutzt	überschrieben
$;$	$\emptyset$	$\emptyset$
$\text{Pos}\,(e)$	$Vars\,(e)$	$\emptyset$
$\text{Neg}\,(e)$	$Vars\,(e)$	$\emptyset$
$R = e;$	$Vars\,(e)$	$\{R\}$
$R_1 = M[R_2];$	$\{R_2\}$	$\{R_1\}$
$M[R_1] = R_2;$	$\{R_1, R_2\}$	$\emptyset$

Eine Variable x, die nicht lebendig an u entlang π ist, heißt tot an u entlang π.

Beispiel:



Wir bemerken:

	lebendig	tot
0	$\{y\}$	$\{x\}$
1	$\emptyset$	$\{x, y\}$
2	$\{y\}$	$\{x\}$
3	$\emptyset$	$\{x, y\}$

Die Variable x ist lebendig an u falls x lebendig ist an u entlang irgend eines Pfads. Andernfalls ist x tot an u.

Die Variable x ist lebendig an u falls x lebendig ist an u entlang irgend eines Pfads. Andernfalls ist x tot an u .

Frage:

Wie berechnet man für jedes u die Menge der dort lebendigen/toten Variablen ???

Die Variable x ist lebendig an u falls x lebendig ist an u entlang irgend eines Pfads. Andernfalls ist x tot an u .

Frage:

Wie berechnet man für jedes u die Menge der dort lebendigen/toten Variablen ???

Idee:

Definiere für jede Kante $k = (u, \_, v)$ eine Funktion $[\![k]\!]^\sharp$, die die Menge der an v lebendigen Variablen in die Menge der an u lebendigen Variablen transformiert ...

Sei $\mathbb{L} = 2^{Vars}$.

Für $k = (\_, lab, \_)$ definieren wir $[\![k]\!]^\sharp = [\![lab]\!]^\sharp$ durch:

$$
\begin{array}{lll}
[\![;]\!]^\sharp\, L & = & L \\
[\![\mathrm{Pos}(e)]\!]^\sharp\, L & = & [\![\mathrm{Neg}(e)]\!]^\sharp\, L \quad = \quad L \cup Vars(e) \\
[\![x = e;]\!]^\sharp\, L & = & (L \backslash \{x\}) \cup Vars(e) \\
[\![R_1 = M[R_2];]\!]^\sharp\, L & = & (L \backslash \{R_1\}) \cup \{R_2\} \\
[\![M[R_1] = R_2;]\!]^\sharp\, L & = & L \cup \{R_1, R_2\}
\end{array}
$$

Sei $\mathbb{L} = 2^{Vars}$.

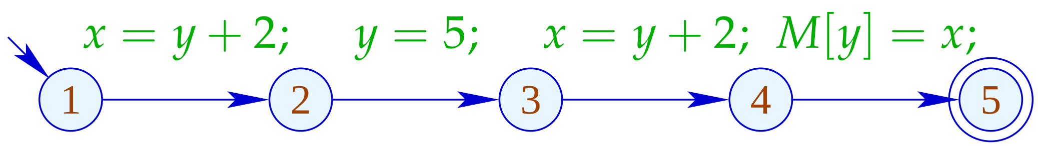
Für $k = (\_, lab, \_)$ definieren wir $[\![k]\!]^\sharp = [\![lab]\!]^\sharp$ durch:

$$\begin{array}{lll}
[\![;]\!]^\sharp\, L & = & L \\
[\![\text{Pos}(e)]\!]^\sharp\, L & = & [\![\text{Neg}(e)]\!]^\sharp\, L \quad = \quad L \cup Vars(e) \\
[\![x = e;]\!]^\sharp\, L & = & (L \backslash \{x\}) \cup Vars(e) \\
[\![R_1 = M[R_2];]\!]^\sharp\, L & = & (L \backslash \{R_1\}) \cup \{R_2\} \\
[\![M[R_1] = R_2;]\!]^\sharp\, L & = & L \cup \{R_1, R_2\}
\end{array}$$

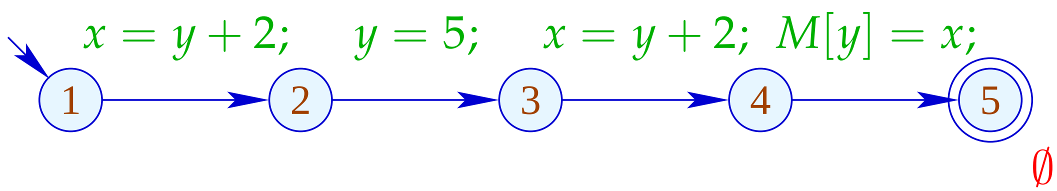
$[\![k]\!]^\sharp$ können wir wieder zu Effekten $[\![\pi]\!]^\sharp$ ganzer Pfade $\pi = k_1 \ldots k_r$ fortsetzen durch:

$$[\![\pi]\!]^\sharp = [\![k_1]\!]^\sharp \circ \ldots \circ [\![k_r]\!]^\sharp$$

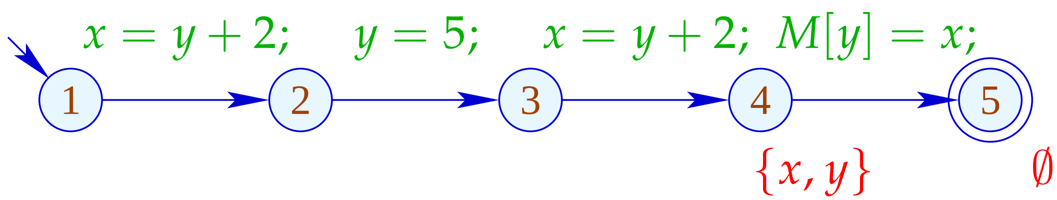
Wir vergewissern uns, dass diese Definitionen vernünftig sind :-)



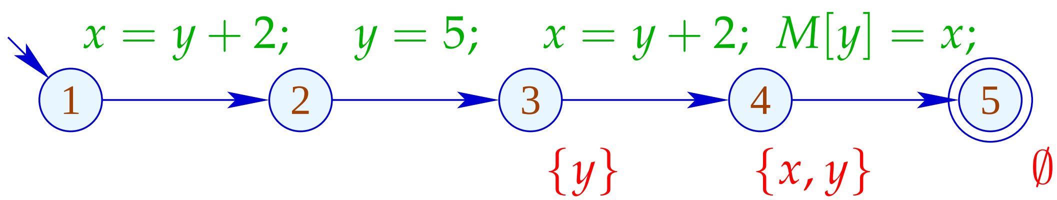
Wir vergewissern uns, dass diese Definitionen vernünftig sind :-)



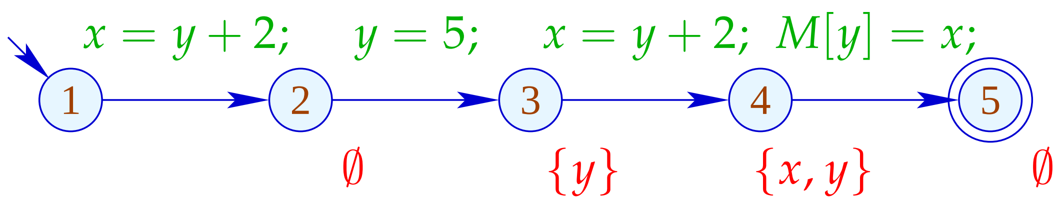
Wir vergewissern uns, dass diese Definitionen vernünftig sind :-)



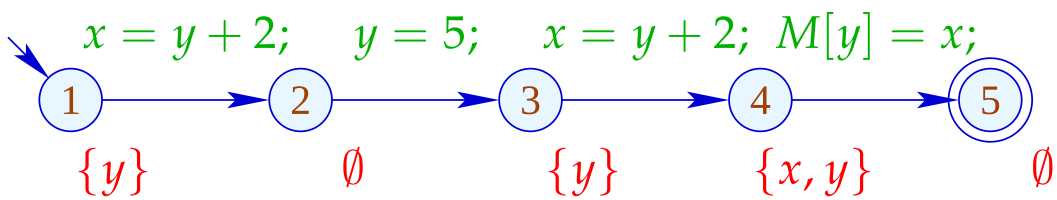
Wir vergewissern uns, dass diese Definitionen vernünftig sind :-)



Wir vergewissern uns, dass diese Definitionen vernünftig sind :-)



Wir vergewissern uns, dass diese Definitionen vernünftig sind :-)



Die Menge der an u lebendigen Variablen ist dann:

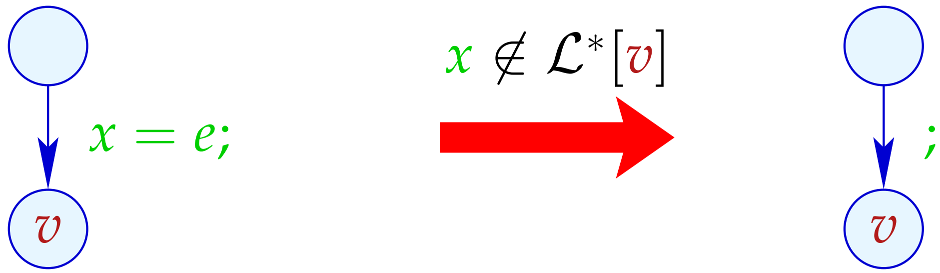
$$\mathcal{L}^*[u] \;=\; \bigcup\{[\![\pi]\!]^\sharp\,\emptyset \mid \pi : u \to^* stop\}$$

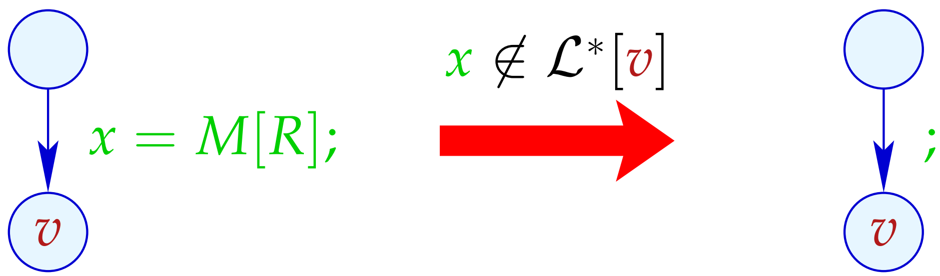
... in Worten:

- Die Pfade starten in u :-)
- x ist lebendig, wenn es nur entlang irgend eines Pfads lebendig ist :-)

  $\Longrightarrow$ Als Halbordnung für $\mathbb{L}$ benötigen wir $\sqsubseteq\, =\, \subseteq$.
- Am Programmende ist keine Variable mehr lebendig :-)

Transformation 3:





Zur Korrektheit zeigt man:

→ Korrektheit der Kanten-Effekte: Falls L die Menge der legendigen Variablen am Ende eines Pfads π sind, dann ist $[\![\pi]\!]^\sharp L$ die Menge der am Anfang lebendigen Variablen :-)

→ Korrektheit der Transformation auf einem Pfad: Wird auf den Wert einer Variable zugegriffen, ist diese stets lebendig. Der Wert toter Variablen ist darum egal :-)

→ Korrektheit der Transformation: Bei Ausführung des transformierten Programms haben bei jedem Besuch eines Programmpunkts die lebendigen Variablen den gleichen Wert :-))

Berechnung der Mengen $\mathcal{L}^*[u]$:

(1) Aufstellen des Constraint-Systems:

$$\begin{array}{lll} \mathcal{L}[stop] & \supseteq & \emptyset \\ \mathcal{L}[u] & \supseteq & [\![k]\!]^\sharp(\mathcal{L}[v]) \qquad k = (u, \_, v) \quad \text{Kante} \end{array}$$

(2) Lösen des Constraint-Systems mittels RR-Iteration.

Da $\mathbb{L}$ endlich ist, terminiert die Iteration :-)

(3) Die kleinste Lösung $\mathcal{L}$ des Constraint-Systems ist gleich $\mathcal{L}^*$ da alle $[\![k]\!]^\sharp$ distributiv sind :-))

Berechnung der Mengen $\mathcal{L}^*[u]$:

(1) Aufstellen des Constraint-Systems:

$$\begin{array}{lll} \mathcal{L}[stop] & \supseteq & \emptyset \\ \mathcal{L}[u] & \supseteq & [\![k]\!]^\sharp (\mathcal{L}[v]) \qquad k = (u, \_, v) \quad \text{Kante} \end{array}$$

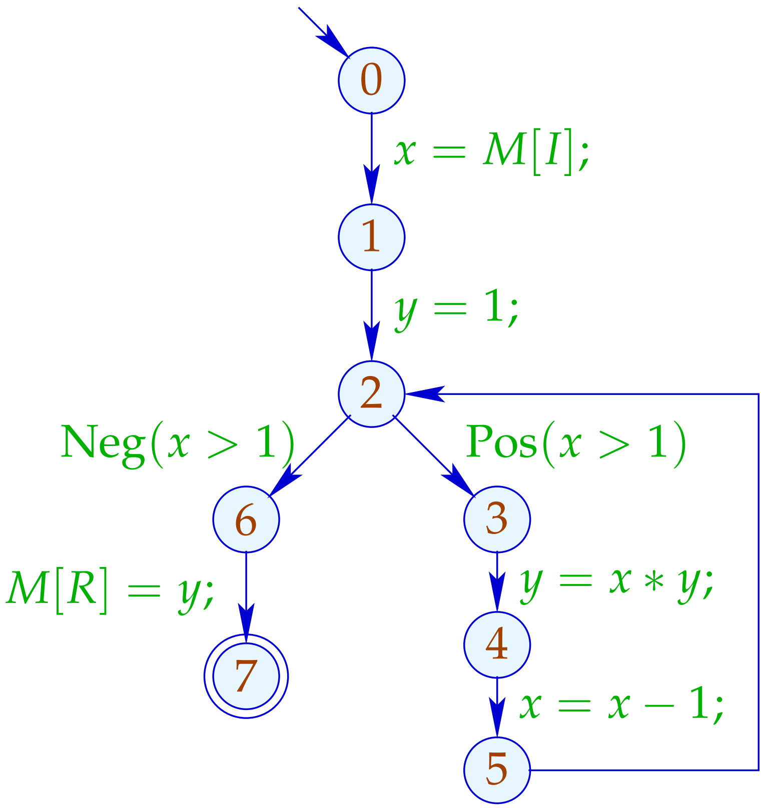
(2) Lösen des Constraint-Systems mittels RR-Iteration.

Da $\mathbb{L}$ endlich ist, terminiert die Iteration :-)

(3) Die kleinste Lösung $\mathcal{L}$ des Constraint-Systems ist gleich $\mathcal{L}^*$ da alle $[\![k]\!]^\sharp$ distributiv sind :-))

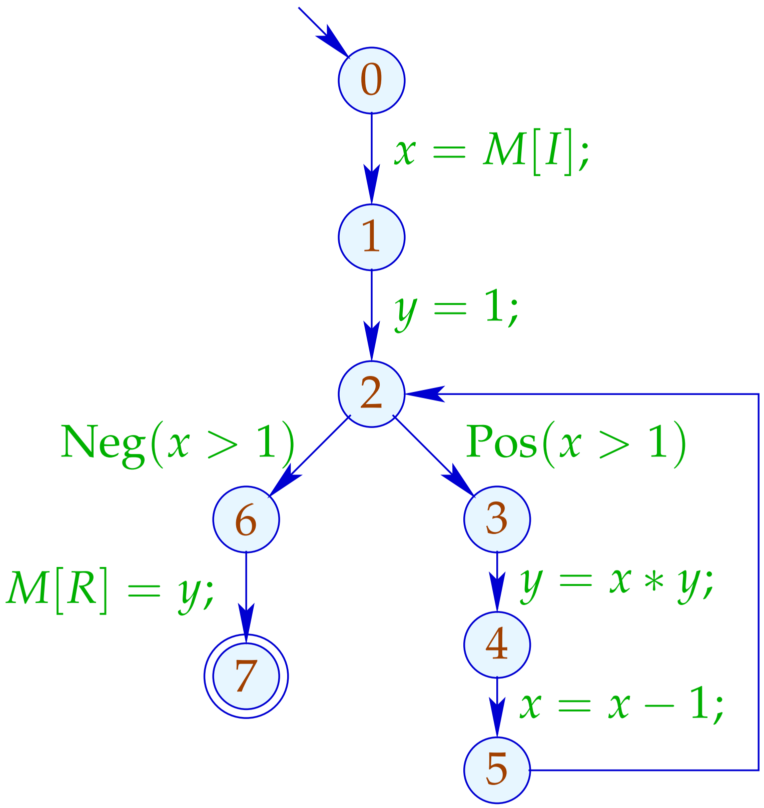
Achtung: Die Information wird rückwärts propagiert !!!

Beispiel:



$$
\begin{array}{lll}
\mathcal{L}[0] & \supseteq & (\mathcal{L}[1] \backslash \{x\}) \cup \{I\} \\
\mathcal{L}[1] & \supseteq & \mathcal{L}[2] \backslash \{y\} \\
\mathcal{L}[2] & \supseteq & (\mathcal{L}[6] \cup \{x\}) \cup (\mathcal{L}[3] \cup \{x\}) \\
\mathcal{L}[3] & \supseteq & (\mathcal{L}[4] \backslash \{y\}) \cup \{x, y\} \\
\mathcal{L}[4] & \supseteq & (\mathcal{L}[5] \backslash \{x\}) \cup \{x\} \\
\mathcal{L}[5] & \supseteq & \mathcal{L}[2] \\
\mathcal{L}[6] & \supseteq & \mathcal{L}[7] \cup \{y, R\} \\
\mathcal{L}[7] & \supseteq & \emptyset
\end{array}
$$

Beispiel:

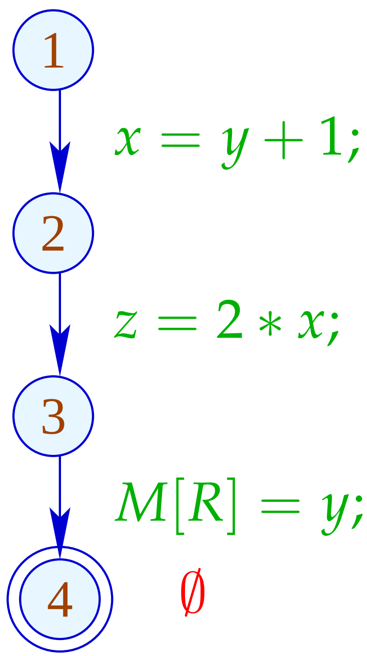


	1	2
7	$\emptyset$	
6	$\{y, R\}$	
2	$\{x, y, R\}$	dito
5	$\{x, y, R\}$	
4	$\{x, y, R\}$	
3	$\{x, y, R\}$	
1	$\{x, R\}$	
0	$\{I, R\}$	

Bei keiner Zuweisung ist die linke Variable tot :-)

Achtung:

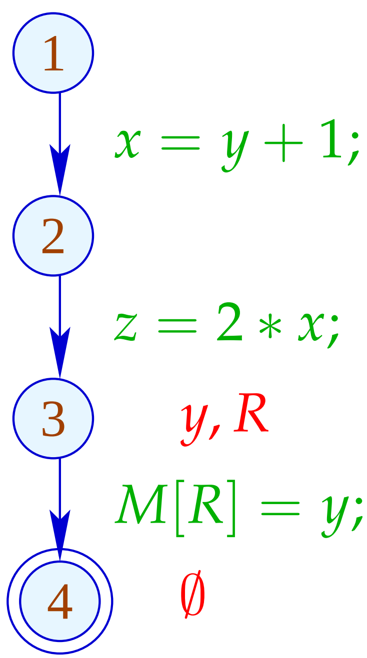
Beseitigung von Zuweisungen an tote Variablen kann weitere Variablen töten:



Bei keiner Zuweisung ist die linke Variable tot :-)

Achtung:

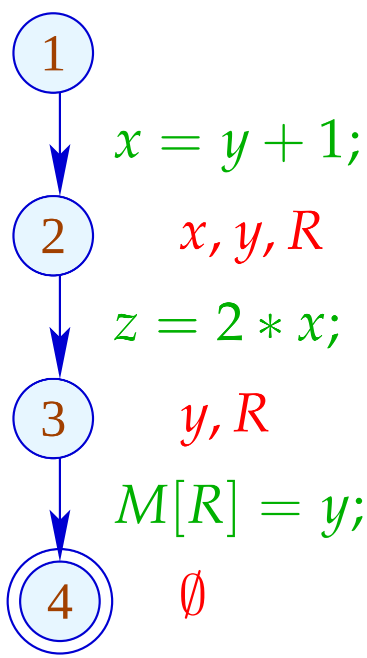
Beseitigung von Zuweisungen an tote Variablen kann weitere Variablen töten:



Bei keiner Zuweisung ist die linke Variable tot :-)

Achtung:

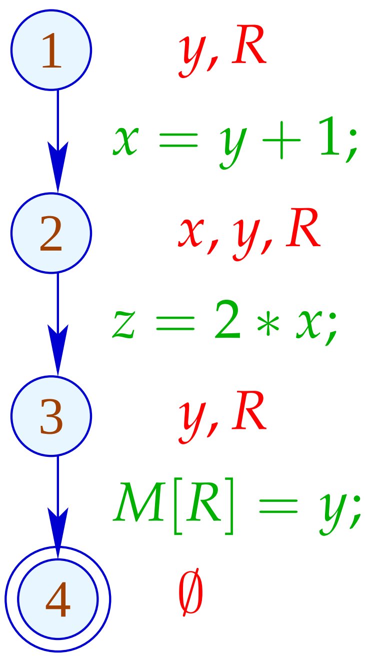
Beseitigung von Zuweisungen an tote Variablen kann weitere Variablen töten:



Bei keiner Zuweisung ist die linke Variable tot :-)

Achtung:

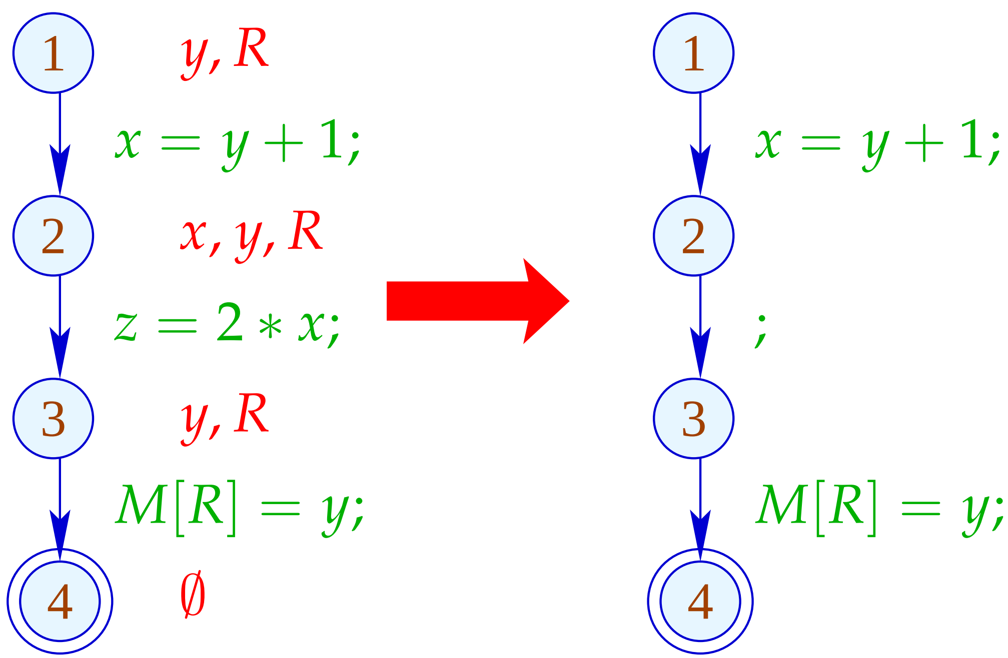
Beseitigung von Zuweisungen an tote Variablen kann weitere Variablen töten:



Bei keiner Zuweisung ist die linke Variable tot :-)

Achtung:

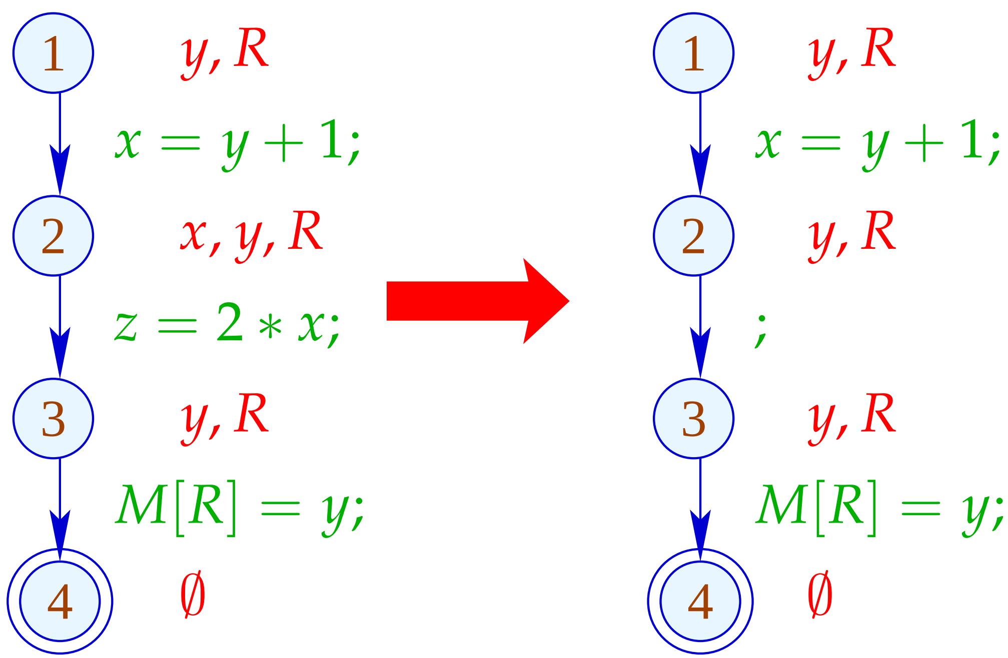
Beseitigung von Zuweisungen an tote Variablen kann weitere Variablen töten:



Bei keiner Zuweisung ist die linke Variable tot :-)

Achtung:

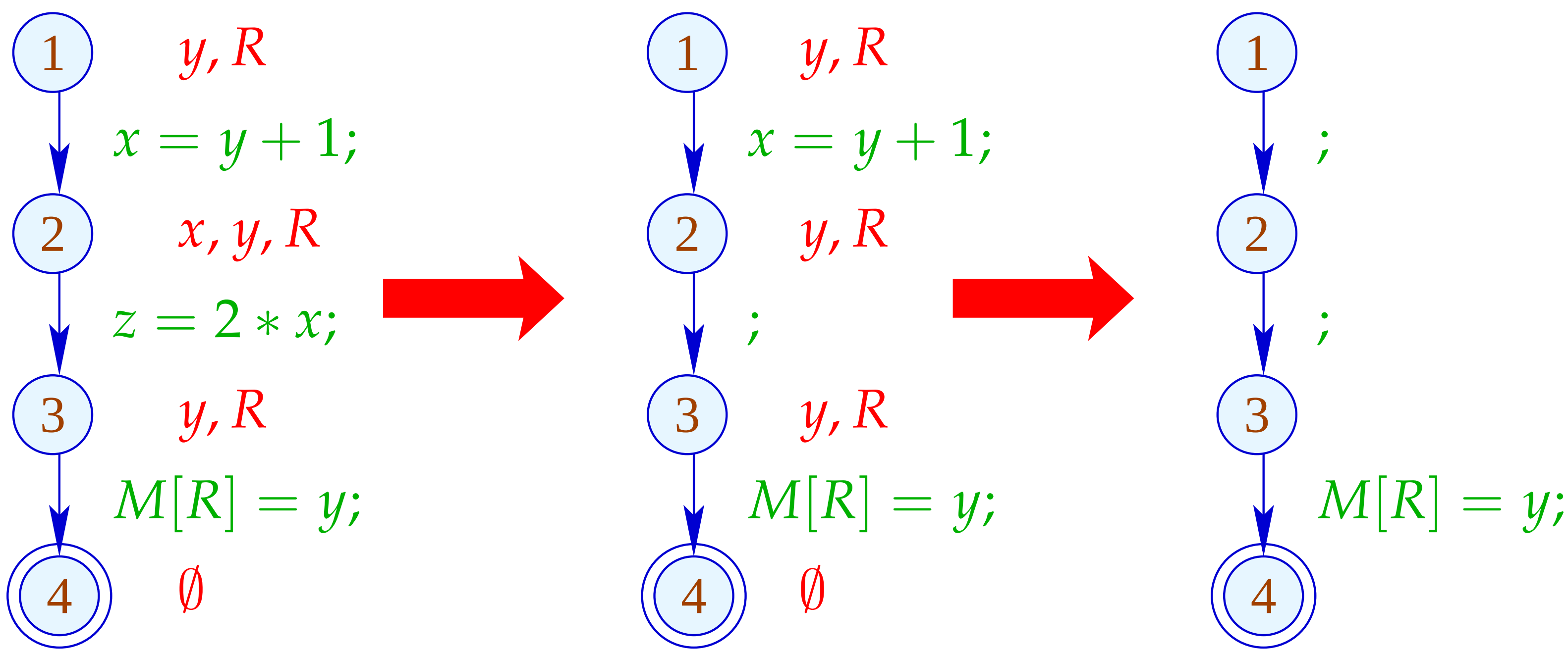
Beseitigung von Zuweisungen an tote Variablen kann weitere Variablen töten:



Bei keiner Zuweisung ist die linke Variable tot :-)

Achtung:

Beseitigung von Zuweisungen an tote Variablen kann weitere Variablen töten:

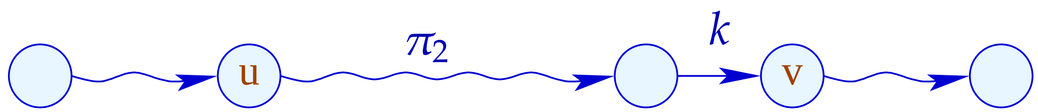


Das Programm mehrmals zu analysieren, ist hässlich :-(

Idee: Analysiere echte Lebendigkeit!

x heißt echt lebendig an u entlang eines Pfads π, falls sich π zerlegen lässt in $\pi = \pi_1\, \pi_2\, k\, \pi_3$ so dass gilt:

- π_1 erreicht u ;
- k ist eine echte Benutzung von x ;
- π_2 enthält keine Überschreibung von x.

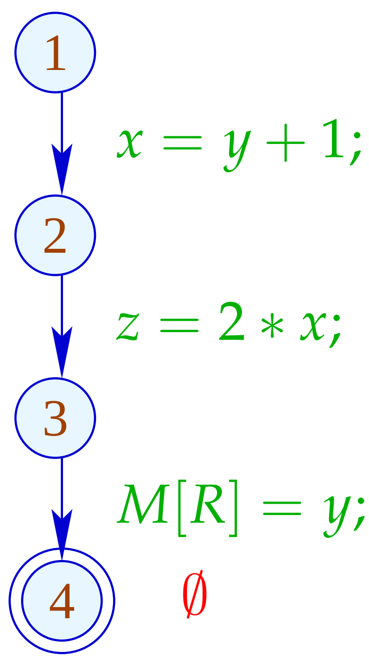


Die Menge der an einer Kante $k = (\_, lab, v)$ echt benutzten Variablen ist gegeben durch:

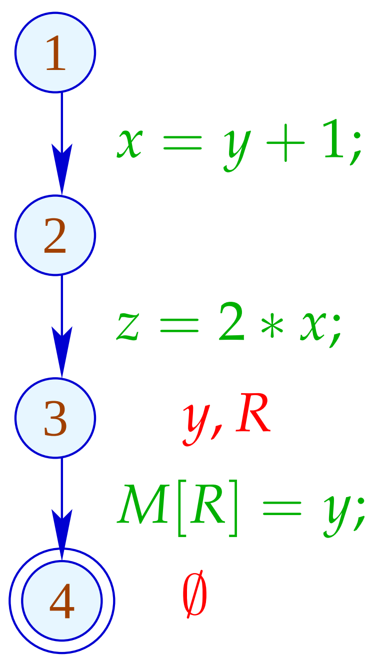
lab	echt benutzt	
$;$	$\emptyset$	
$\text{Pos}\,(e)$	$Vars\,(e)$	
$\text{Neg}\,(e)$	$Vars\,(e)$	
$x = e;$	$Vars\,(e)$	$(*)$
$x = M[R];$	$\{R\}$	$(*)$
$M[R_1] = R_2;$	$\{R_1, R_2\}$	

$(*)$ – sofern x an v echt lebendig ist :-)

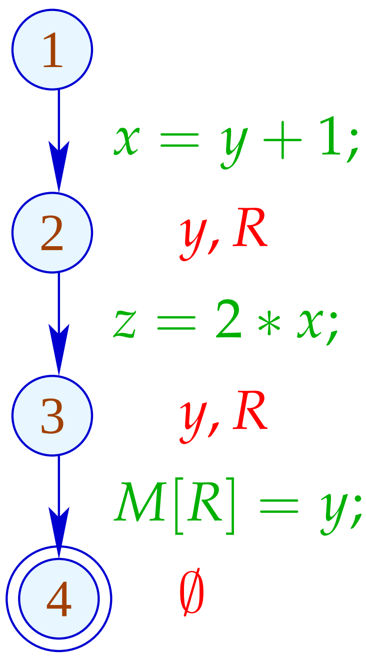
Beispiel:



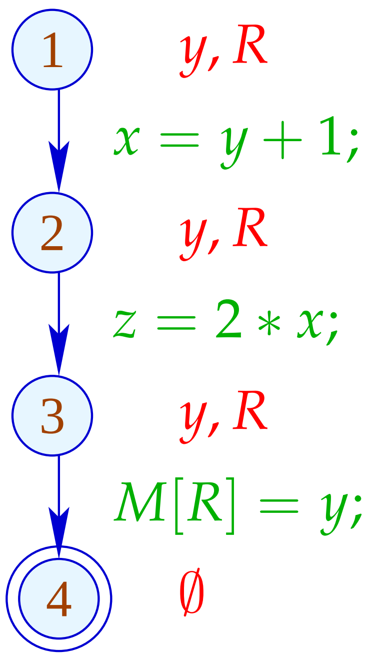
Beispiel:



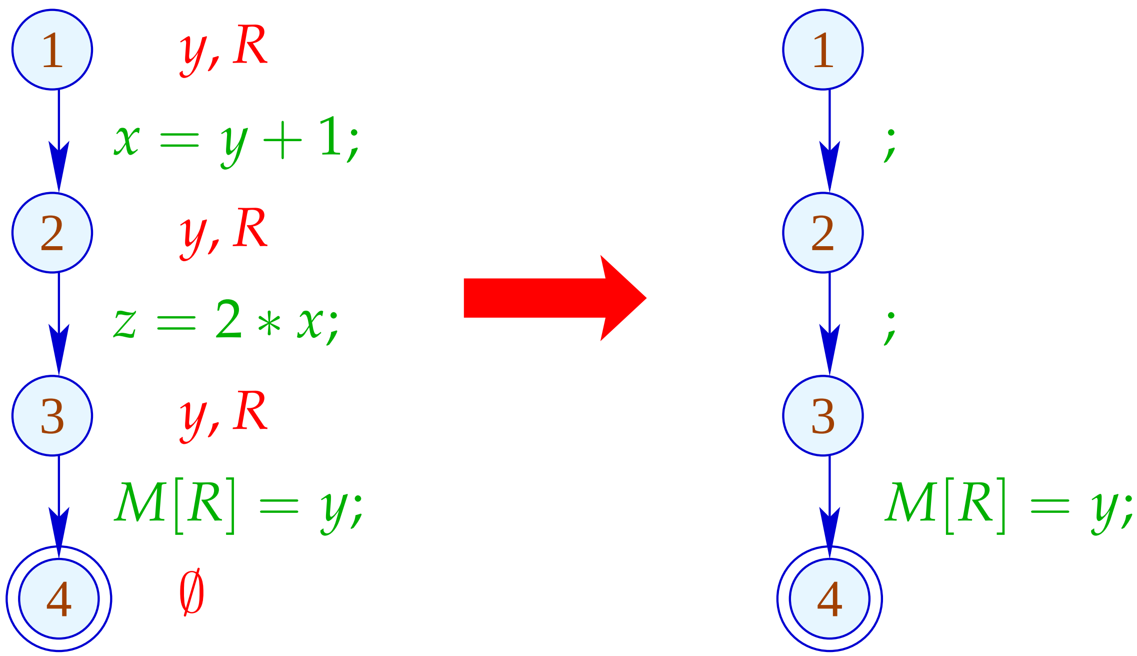
Beispiel:



Beispiel:



Beispiel:



Die Kanten-Effekte:

$$
\begin{array}{lllll}
[\![;]\!]^\sharp\, L & = & L & & \\
[\![\mathrm{Pos}(e)]\!]^\sharp\, L & = & [\![\mathrm{Neg}(e)]\!]^\sharp\, L & = & L \cup \mathit{Vars}(e) \\
[\![x = e;]\!]^\sharp\, L & = & (L \backslash \{x\}) \cup & & \mathit{Vars}(e) \\
[\![R_1 = M[R_2];]\!]^\sharp\, L & = & (L \backslash \{R_1\}) \cup & & \{R_2\} \\
[\![M[R_1] = R_2;]\!]^\sharp\, L & = & L \cup \{R_1, R_2\} & &
\end{array}
$$

Die Kanten-Effekte:

$$
\begin{array}{lll}
[\![;]\!]^\sharp\, L & = & L \\
[\![\mathrm{Pos}(e)]\!]^\sharp\, L & = & [\![\mathrm{Neg}(e)]\!]^\sharp\, L \quad = \quad L \cup \mathit{Vars}(e) \\
[\![x = e;]\!]^\sharp\, L & = & (L \backslash \{x\}) \cup \; (x \in L)\; ?\; \mathit{Vars}(e) :\; \emptyset \\
[\![R_1 = M[R_2];]\!]^\sharp\, L & = & (L \backslash \{R_1\}) \cup \; (R_1 \in L)\; ?\; \{R_2\} :\; \emptyset \\
[\![M[R_1] = R_2;]\!]^\sharp\, L & = & L \cup \{R_1, R_2\}
\end{array}
$$

Beachte:

- Die Kanten-Effekte für echt lebendige Variablen sind komplizierter als für lebendige Variablen :-)
- Sie sind aber immer noch distributiv !!

Beachte:

- Die Kanten-Effekte für echt lebendige Variablen sind komplizierter als für lebendige Variablen :-)
- Sie sind aber immer noch distributiv !!

  Dazu betrachten wir für $\mathbb{D} = 2^U$, $f\,y = (u \in y)\,?\,b : \emptyset$

  Wir überprüfen:

$$
\begin{array}{lll}
f\,(y_1 \cup y_2) & = & (u \in y_1 \cup y_2)\,?\,b : \emptyset \\
 & = & (u \in y_1 \vee u \in y_2)\,?\,b : \emptyset \\
 & = & (u \in y_1)\,?\,b : \emptyset \cup (u \in y_2)\,?\,b : \emptyset \\
 & = & f\,y_1 \cup f\,y_2
\end{array}
$$

Beachte:

- Die Kanten-Effekte für echt lebendige Variablen sind komplizierter als für lebendige Variablen :-)
- Sie sind aber immer noch distributiv !!

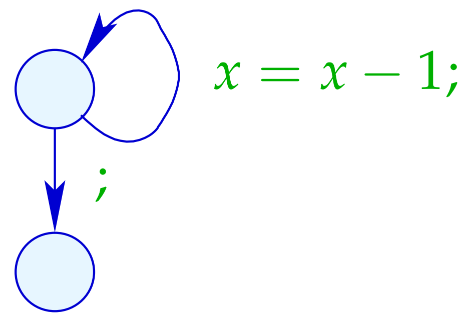
  Dazu betrachten wir für $\mathbb{D} = 2^U$, $f\,y = (u \in y) \,?\, b : \emptyset$

  Wir überprüfen:

$$\begin{aligned} f\,(y_1 \cup y_2) &= (u \in y_1 \cup y_2) \,?\, b : \emptyset \\ &= (u \in y_1 \vee u \in y_2) \,?\, b : \emptyset \\ &= (u \in y_1) \,?\, b : \emptyset \cup (u \in y_2) \,?\, b : \emptyset \\ &= f\,y_1 \cup f\,y_2 \end{aligned}$$

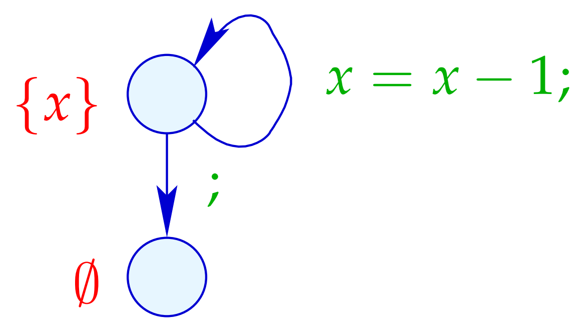
$\Longrightarrow$ Constraint-System liefert MOP :-))

- Echte Lebendigkeit findet mehr überflüssige Zuweisungen als wiederholte Lebendigkeit !!!



- Echte Lebendigkeit findet mehr überflüssige Zuweisungen als wiederholte Lebendigkeit !!!

Lebendigkeit:



- Echte Lebendigkeit findet mehr überflüssige Zuweisungen als wiederholte Lebendigkeit !!!

Echte Lebendigkeit: