Idee:

Für jeden Ausdruck merken wir uns die Variablen, die gegenwärtig seinen Wert enthalten :-)

Wir benutzen: $\mathbb{V} = \mathit{Expr} \to 2^{\mathit{Vars}}$ und definieren:

$$
\begin{array}{lllll}
[\![;]\!]^\sharp\, V & = & V & & \\
[\![\mathrm{Pos}(c)]\!]^\sharp\, V & = & [\![\mathrm{Neg}(c)]\!]^\sharp\, V & = & V \\
[\![\mathrm{Pos}(x)]\!]^\sharp\, V & = & [\![\mathrm{Neg}(x)]\!]^\sharp\, V & = & V \\
[\![\mathrm{Pos}(e)]\!]^\sharp\, V\, e' & = & [\![\mathrm{Neg}(e)]\!]^\sharp\, V\, e' & = & \left\{ \begin{array}{ll} \emptyset & \text{falls } e' = e \\ V\, e' & \text{sonst} \end{array} \right.
\end{array}
$$

$\ldots$

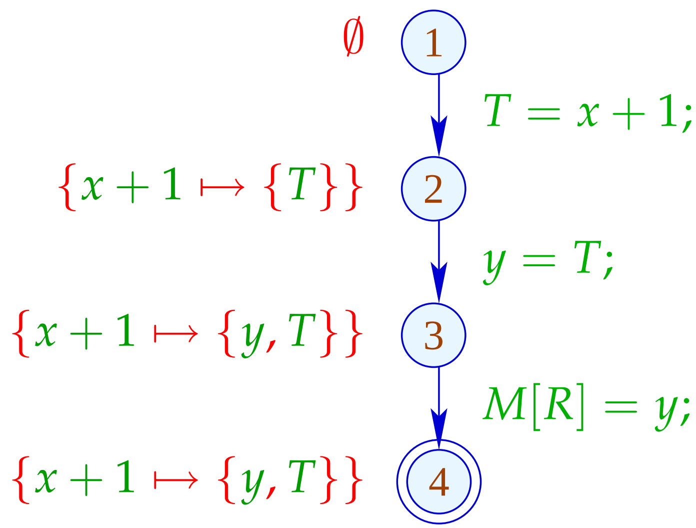
$$
\begin{array}{lll}
[\![x = c;]\!]^\sharp\, V\, e' & = & \begin{cases} (V\, c) \cup \{x\} & \text{falls} \quad e' = c \\ (V\, e') \backslash \{x\} & \text{sonst} \end{cases} \\
[\![x = y;]\!]^\sharp\, V\, e & = & \begin{cases} (V\, e) \cup \{x\} & \text{falls} \quad y \in V\, e \\ (V\, e) \backslash \{x\} & \text{sonst} \end{cases} \\
[\![x = e;]\!]^\sharp\, V\, e' & = & \begin{cases} \{x\} & \text{falls} \quad e' = e \\ (V\, e') \backslash \{x\} & \text{sonst} \end{cases} \\
[\![x = M[c];]\!]^\sharp\, V\, e' & = & (V\, e') \backslash \{x\} \\
[\![x = M[y];]\!]^\sharp\, V\, e' & = & (V\, e') \backslash \{x\} \\
[\![x = M[e];]\!]^\sharp\, V\, e' & = & \begin{cases} \emptyset & \text{falls } e' = e \\ (V\, e') \backslash \{x\} & \text{sonst} \end{cases}
\end{array}
$$

// analog für die verschiedenen Stores

Im Beispiel:

$$\begin{tikzcd} \emptyset & 1 \arrow[d, "T = x+1;"] \\ \{x+1 \mapsto \{T\}\} & 2 \arrow[d, "y = T;"] \\ \{x+1 \mapsto \{y, T\}\} & 3 \arrow[d, "M[R] = y;"] \\ \{x+1 \mapsto \{y, T\}\} & 4 \end{tikzcd}$$

Im Beispiel:



$\longrightarrow$ Wir propagieren die Information vorwärts :-)

An *start* haben wir $V_0\, e = \emptyset$ für alle e

$\longrightarrow$ $\sqsubseteq \subseteq \mathbb{V} \times \mathbb{V}$ definieren wir durch:

$$V_1 \sqsubseteq V_2 \quad \text{gdw.} \quad V_1\, e \supseteq V_2\, e \quad \text{für alle} \quad e$$

Beobachtung:

Die neuen Kanten-Effekte sind distributiv:

Dazu zeigen wir, dass die folgenden Funktionen distributiv sind:

(1) $f_1^x \, V \, e = (V \, e) \backslash \{x\}$

(2) $f_2^{e,a} \, V = V \oplus \{e \mapsto a\}\}$

(3) $f_3^{x,y} \, V \, e = (y \in V \, e) \, ? \, (V \, e \cup \{x\}) : ((V \, e) \backslash \{x\})$

Offenbar gilt:

$$\begin{array}{lll} [\![x = e;]\!]^\sharp & = & f_2^{e,\{x\}} \circ f_1^x \\ [\![x = y;]\!]^\sharp & = & f_3^{x,y} \\ [\![x = M[R];]\!]^\sharp & = & f_1^x \end{array}$$

Distributivität ist unter Komposition abgeschlossen. Damit folgt die Behauptung :-))

(1) Für $f\,V\,e = (V\,e)\backslash\{x\}$ gilt:

$$
\begin{array}{lll}
f\,(V_1 \sqcup V_2)\,e & = & ((V_1 \sqcup V_2)\,e)\backslash\{x\} \\
& = & ((V_1\,e) \cap (V_2\,e))\backslash\{x\} \\
& = & ((V_1\,e)\backslash\{x\}) \cap ((V_2\,e)\backslash\{x\}) \\
& = & (f\,V_1\,e) \cap (f\,V_2\,e) \\
& = & (f\,V_1 \sqcup f\,V_2)\,e \qquad \text{:-)}
\end{array}
$$

(2) Für $\quad f\,V = V \oplus \{e \mapsto a\} \quad$ gilt:

$$
\begin{aligned}
f\,(V_1 \sqcup V_2)\,e' &= ((V_1 \sqcup V_2) \oplus \{e \mapsto a\})\,e' \\
&= (V_1 \sqcup V_2)\,e' \\
&= (f\,V_1 \sqcup f\,V_2)\,e' \qquad \text{sofern} \quad e \neq e'
\end{aligned}
$$

$$
\begin{aligned}
f\,(V_1 \sqcup V_2)\,e &= ((V_1 \sqcup V_2) \oplus \{e \mapsto a\})\,e \\
&= a \\
&= ((V_1 \oplus \{e \mapsto a\})\,e) \cap ((V_2 \oplus \{e \mapsto a\})\,e) \\
&= (f\,V_1 \sqcup f\,V_2)\,e \qquad\qquad \text{:-)}
\end{aligned}
$$

(3) Für $f\,V\,e = (y \in V\,e)\,?\,(V\,e \cup \{x\}) : ((V\,e)\backslash\{x\})$ gilt:

$$
\begin{array}{lll}
f\,(V_1 \sqcup V_2)\,e & = & (((V_1 \sqcup V_2)\,e)\backslash\{x\}) \cup (y \in (V_1 \sqcup V_2)\,e)\,?\,\{x\} : \emptyset \\
 & = & ((V_1\,e \cap V_2\,e)\backslash\{x\}) \cup (y \in (V_1\,e \cap V_2\,e))\,?\,\{x\} : \emptyset \\
 & = & ((V_1\,e \cap V_2\,e)\backslash\{x\}) \cup \\
 & & ((y \in V_1\,e)\,?\,\{x\} : \emptyset) \cap ((y \in V_2\,e)\,?\,\{x\} : \emptyset) \\
 & = & (((V_1\,e)\backslash\{x\}) \cup (y \in V_1\,e)\,?\,\{x\} : \emptyset) \cap \\
 & & (((V_2\,e)\backslash\{x\}) \cup (y \in V_2\,e)\,?\,\{x\} : \emptyset) \\
 & = & (f\,V_1 \sqcup f\,V_2)\,e \qquad\qquad :\text{-})
\end{array}
$$

Wir schließen:

- $\rightarrow$ Lösen des Ungleichungssystems liefert die MOP-Lösung :-)
- $\rightarrow$ Sei $\mathcal{V}$ diese Lösung.

  Gilt $x \in \mathcal{V}[u]\, e$, enthält x an u den Wert von e — welchen wir in T_e abgespeichert haben

  $\Longrightarrow$

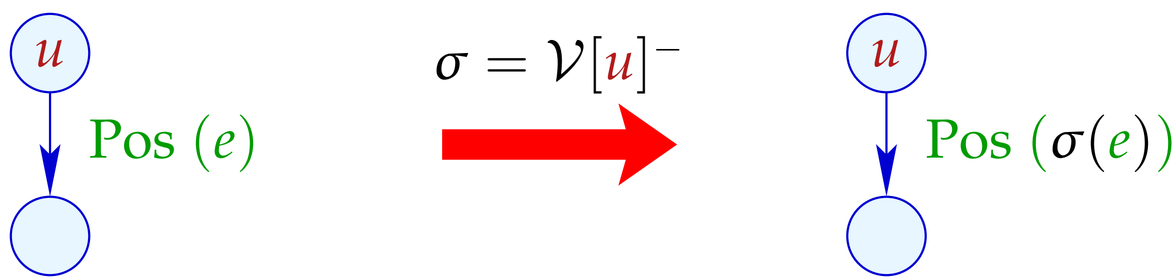
  der Zugriff auf x kann durch Zugriff auf T_e ersetzt werden :-)

Für $V \in \mathbb{V}$ sei V^- die Variablen-Substitution mit:

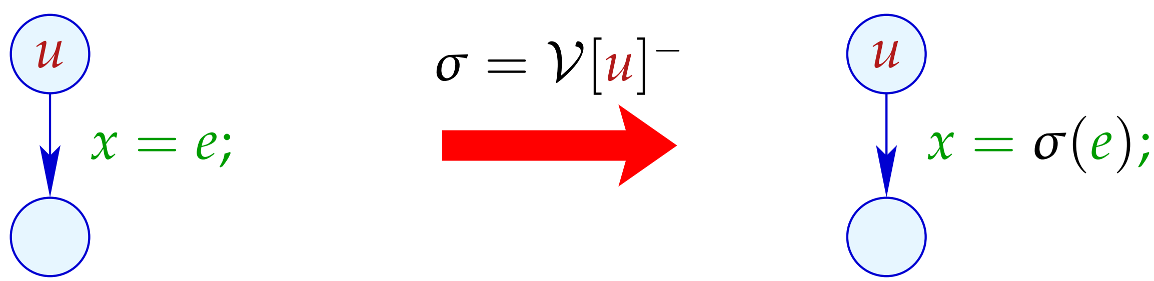
$$V^- x = \begin{cases} T_e & \text{falls } x \in V\, e \\ x & \text{sonst} \end{cases}$$

falls $V\, e \cap V\, e' = \emptyset$ für $e \neq e'$. Andernfalls: $V^- x = x$:-)

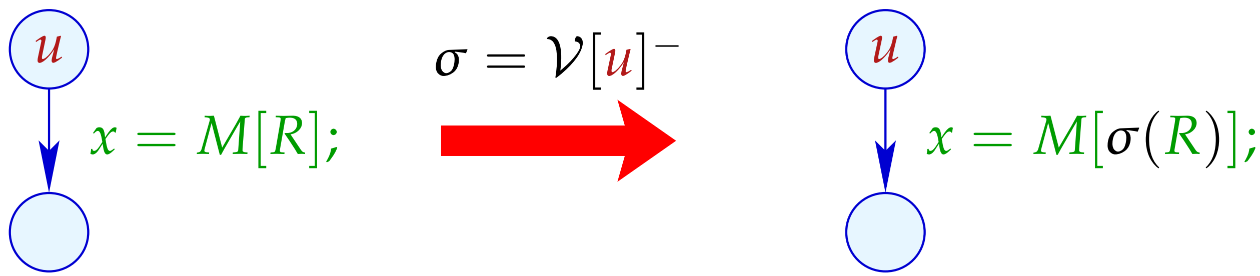
Transformation 3:

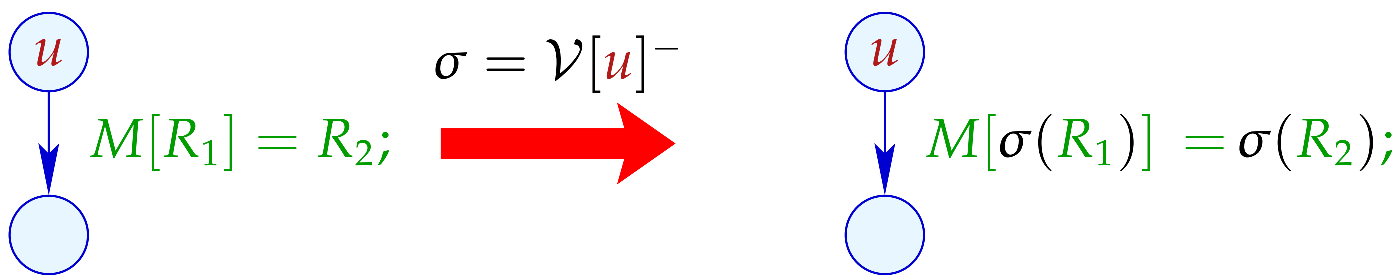


... analog für Kanten mit Neg (e)



Transformation 3 (Forts.):





Vorgehen insgesamt:

(1) Verfügbarkeit von Ausdrücken: T1

+ verringert arithmetische Operationen
– fügt überflüssige Umspeicherungen ein

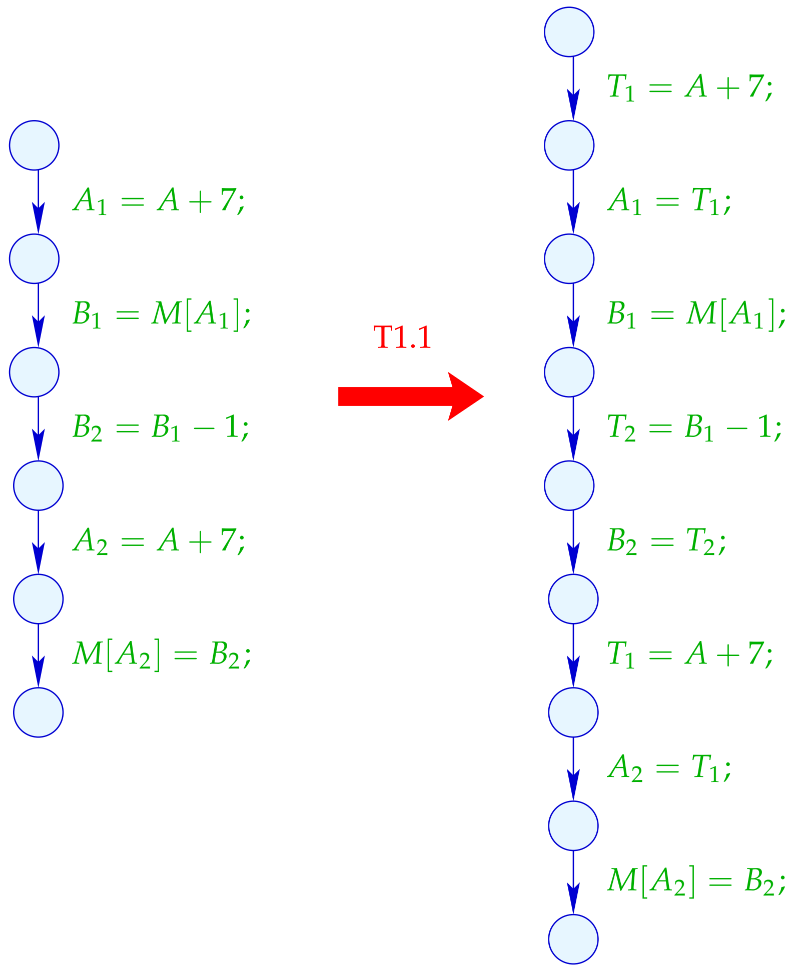
(2) Werte von Variablen: T3

+ erzeugt tote Variablen

(3) (wahre) Lebendigkeit von Variablen: T2

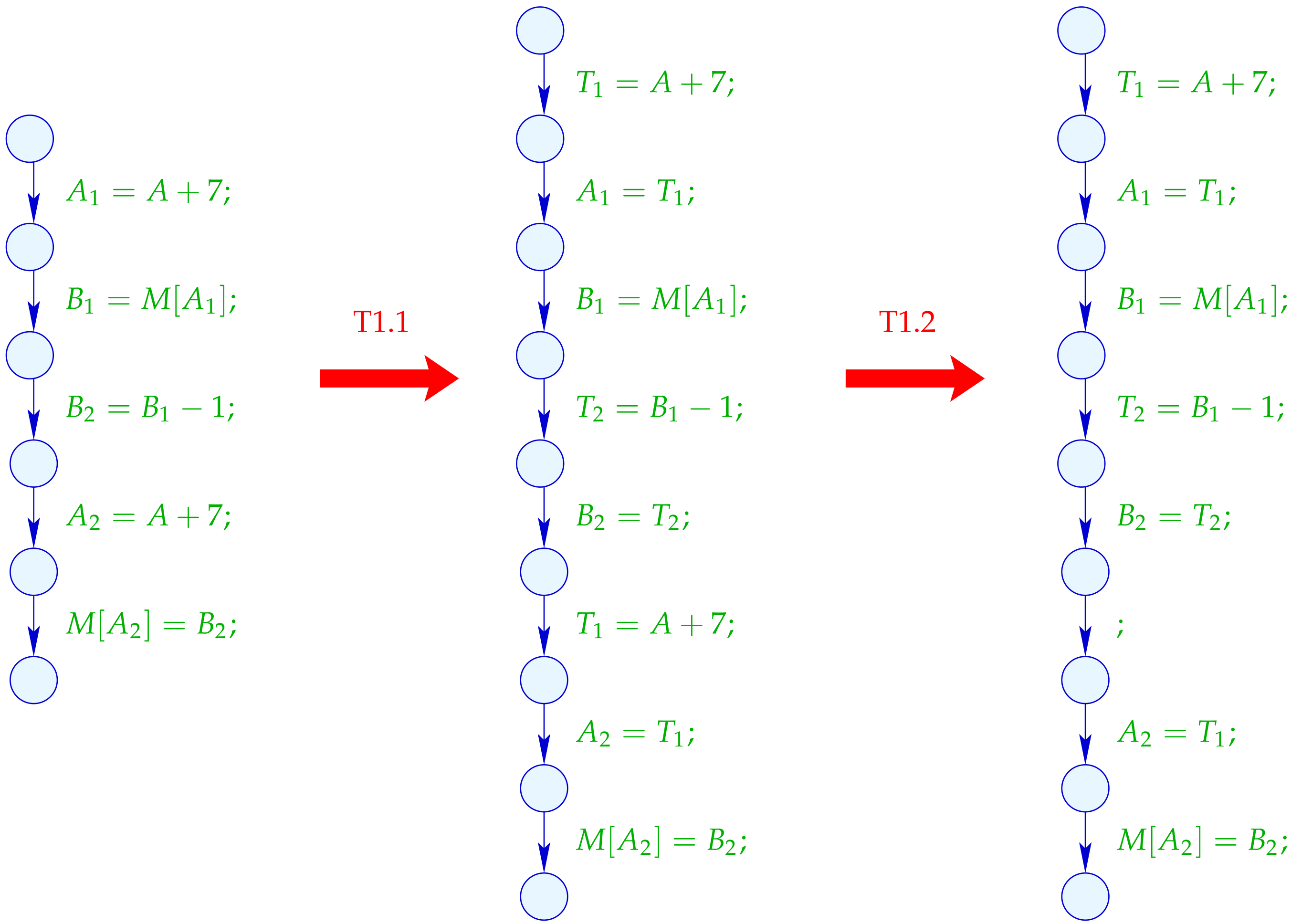
+ beseitigt Zuweisungen an tote Variablen

Beispiel: `a[7]--;`

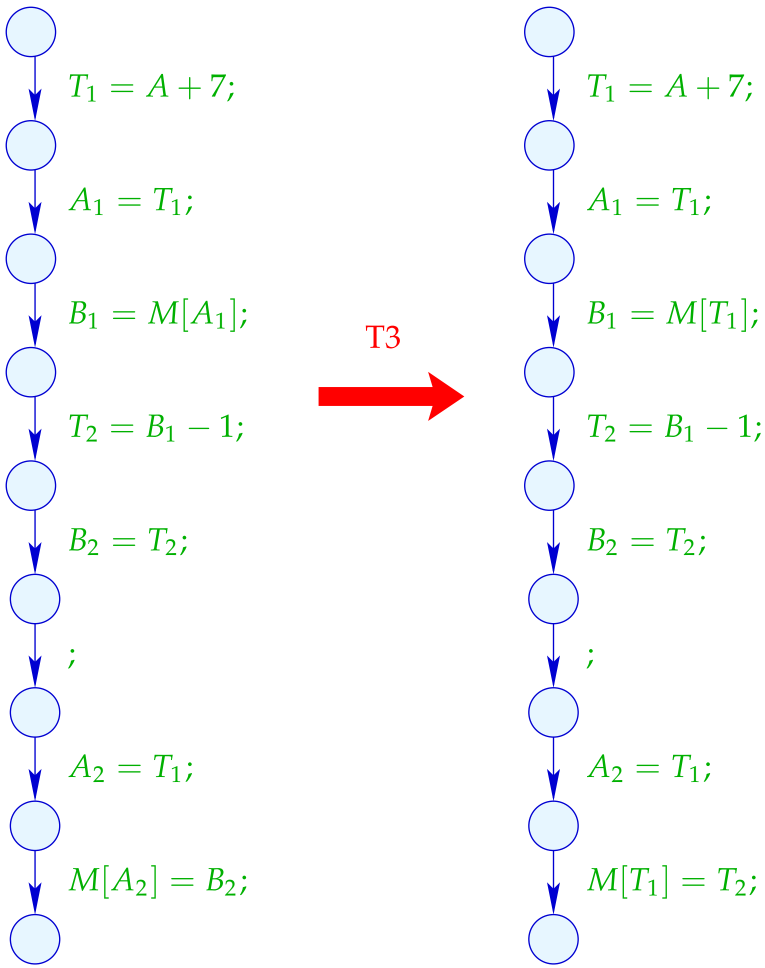


$A_1 = A + 7;$
$B_1 = M[A_1];$
$B_2 = B_1 - 1;$
$A_2 = A + 7;$
$M[A_2] = B_2;$
T1.1
$T_1 = A + 7;$
$A_1 = T_1;$
$B_1 = M[A_1];$
$T_2 = B_1 - 1;$
$B_2 = T_2;$
$T_1 = A + 7;$
$A_2 = T_1;$
$M[A_2] = B_2;$

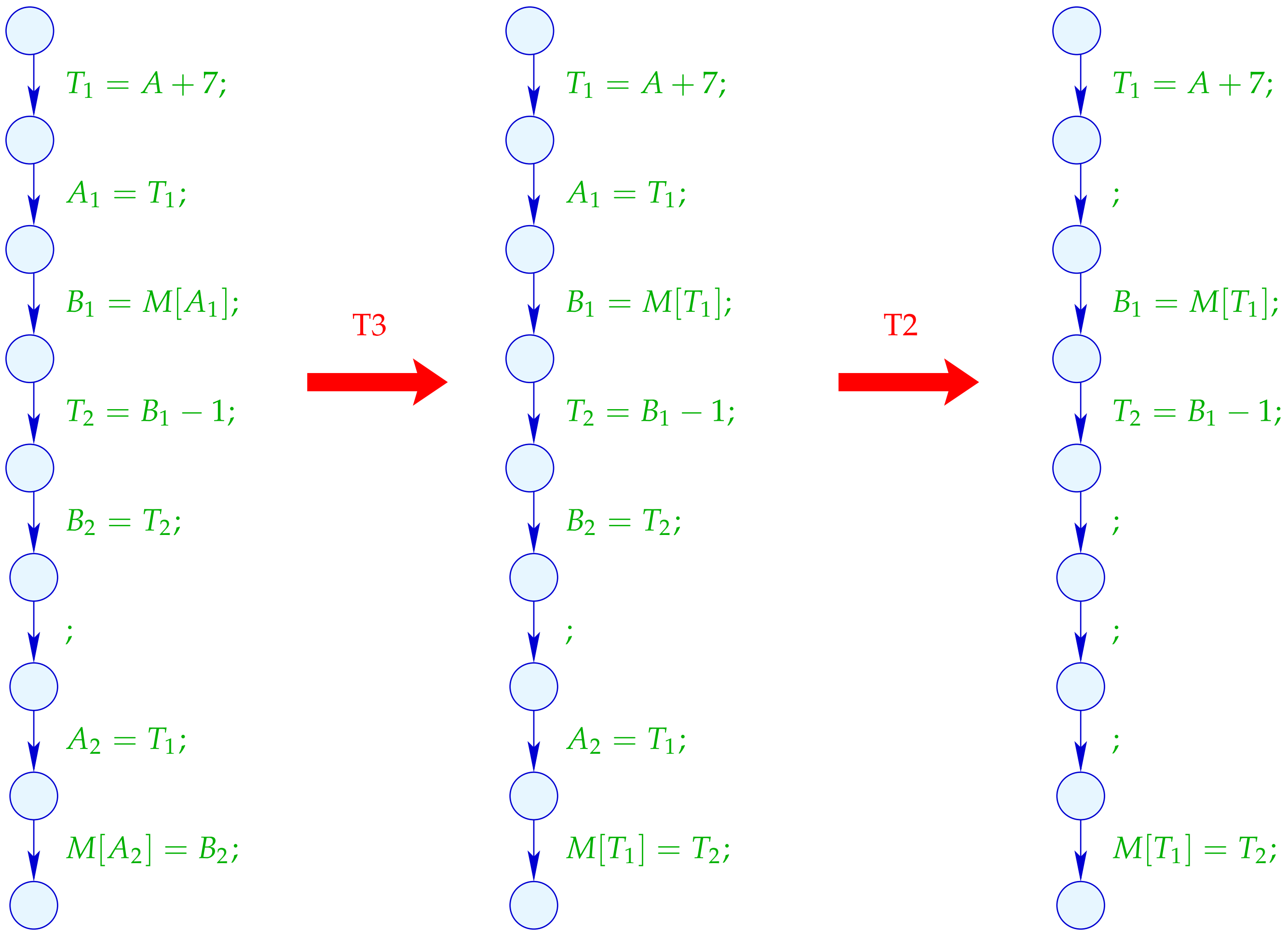
Beispiel: `a[7]--;`


$A_1 = A + 7;$
$B_1 = M[A_1];$
$B_2 = B_1 - 1;$
$A_2 = A + 7;$
$M[A_2] = B_2;$
T1.1
$T_1 = A + 7;$
$A_1 = T_1;$
$B_1 = M[A_1];$
$T_2 = B_1 - 1;$
$B_2 = T_2;$
$T_1 = A + 7;$
$A_2 = T_1;$
$M[A_2] = B_2;$
T1.2
$T_1 = A + 7;$
$A_1 = T_1;$
$B_1 = M[A_1];$
$T_2 = B_1 - 1;$
$B_2 = T_2;$
$;$
$A_2 = T_1;$
$M[A_2] = B_2;$

Beispiel (Forts.): `a[7]--;`



Beispiel (Forts.): `a[7]--;`



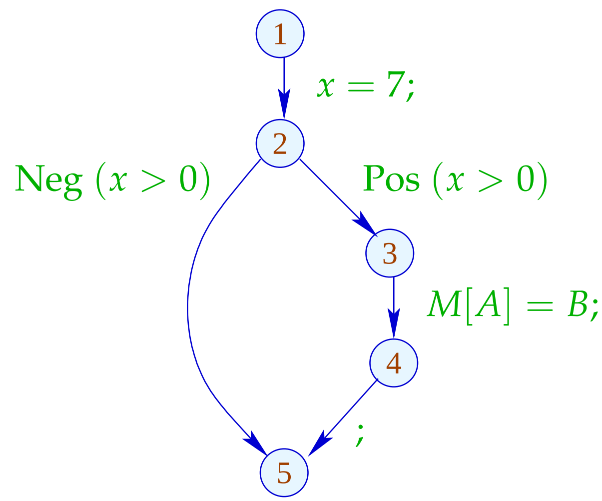
1.4 Konstanten-Propagation

Idee:

Führe möglichst große Teile des Codes bereits zur Compilezeit aus!

Beispiel:

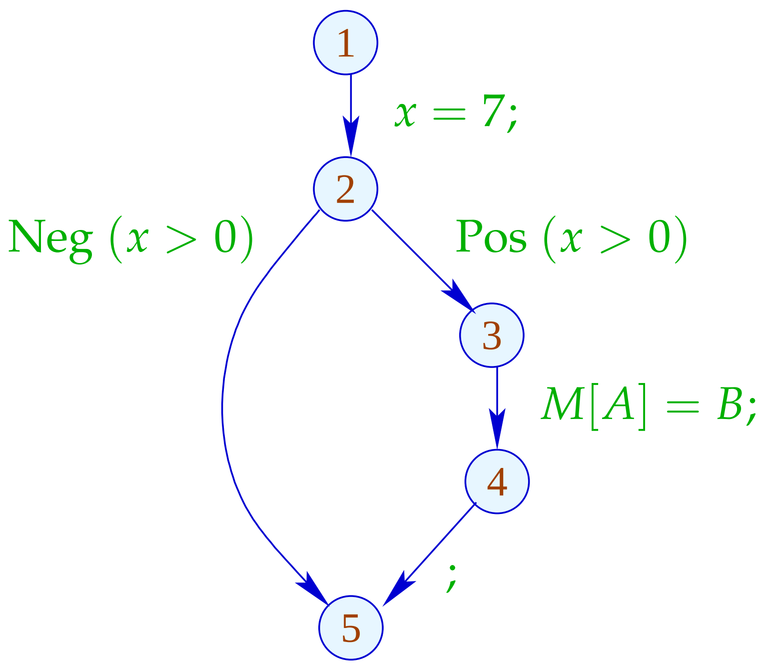
$$x = 7;$$
$$\text{if } (x > 0)$$
$$\quad M[A] = B;$$



Offenbar hat x stets den Wert 7 :-)

Deshalb wird stets der Speicherzugriff durchgeführt :-))

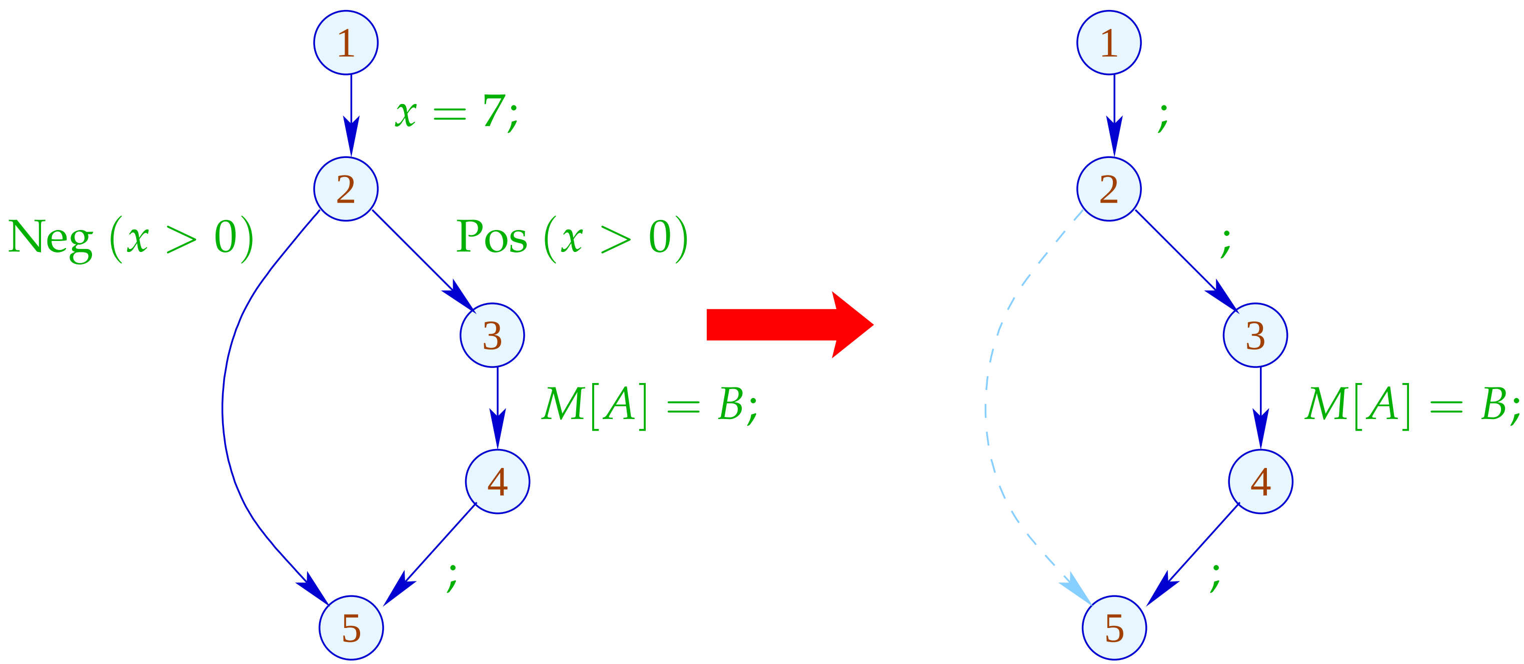
Ziel:



Offenbar hat x stets den Wert 7 :-)

Deshalb wird stets der Speicherzugriff durchgeführt :-))

Ziel:



Verallgemeinerung: Partielle Auswertung

Neil D. Jones, DIKU, Kopenhagen