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

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

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

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

Wir schließen:

$\rightarrow$ Lösen des Constraint-Systems 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

$\Longrightarrow$

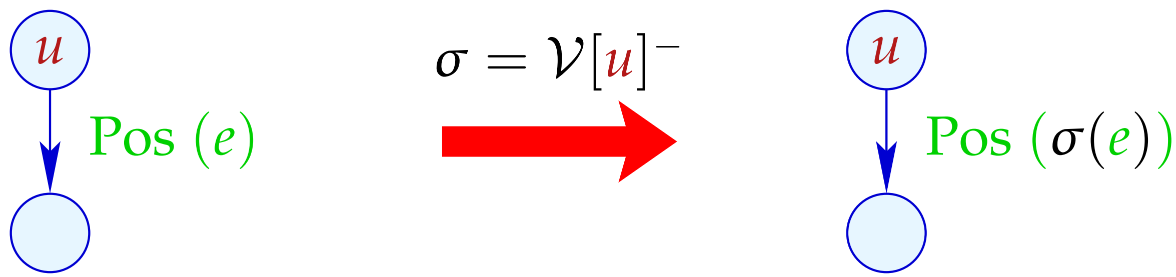
der Zugriff auf x kann durch Zugriff auf T_e ersetzt werden :-)

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

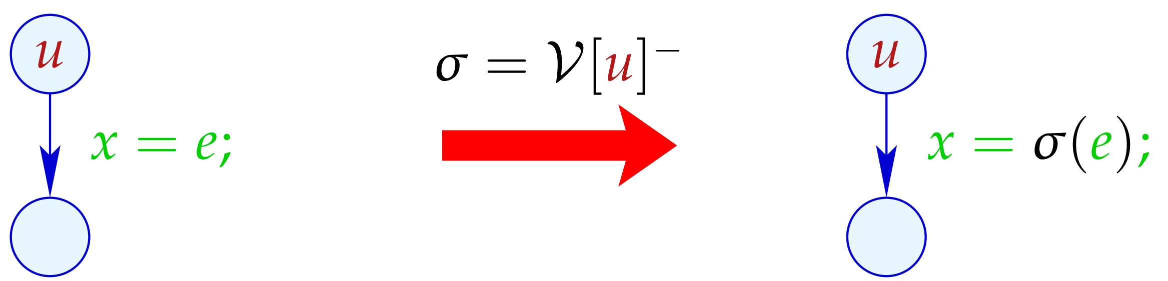
$$V^-\, x = \begin{cases} T_e & \text{falls } x \in V\, e \\ x & \text{sonst} \end{cases}$$

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

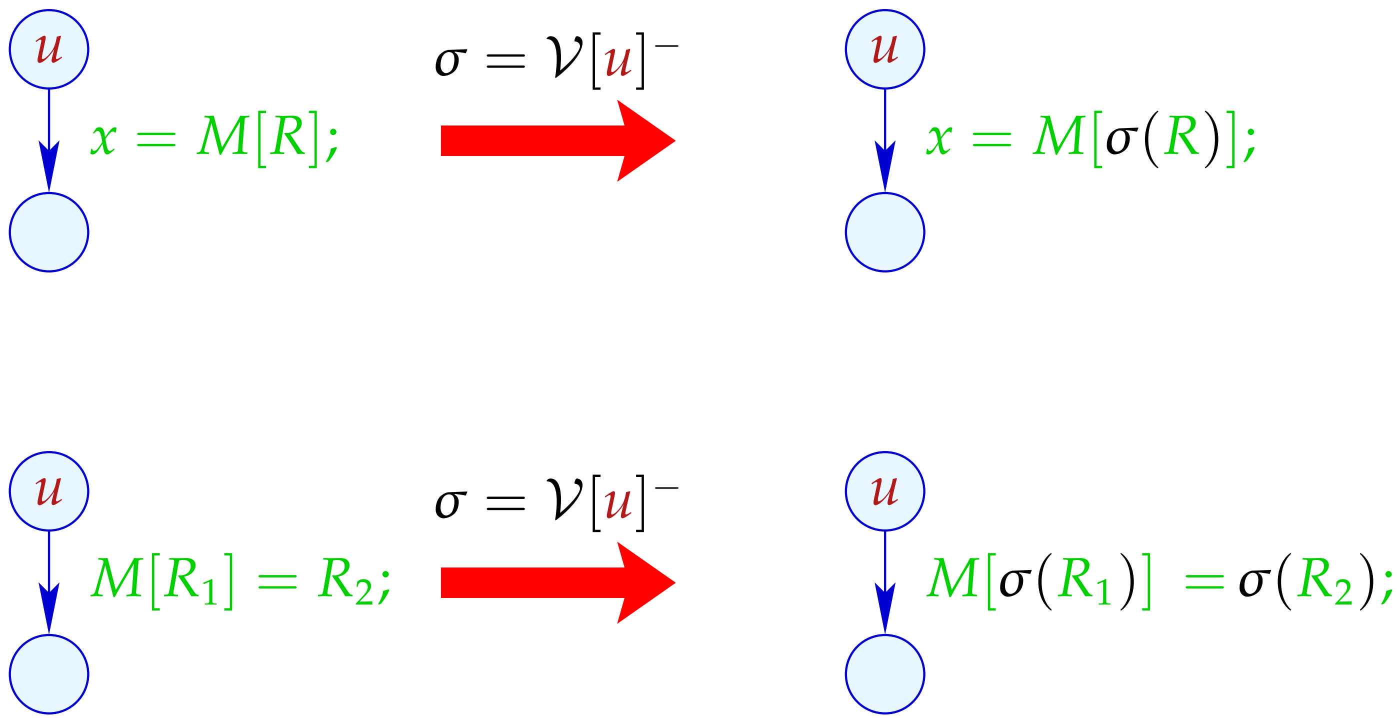
Transformation 4:



... analog für Kanten mit $\mathrm{Neg}\ (e)$



Transformation 4 (Forts.):



Vorgehen insgesamt:

(1) Verfügbarkeit von Ausdrücken: T1 + T2

- \+ verringert arithmetische Operationen
- – fügt überflüssige Umspeicherungen ein

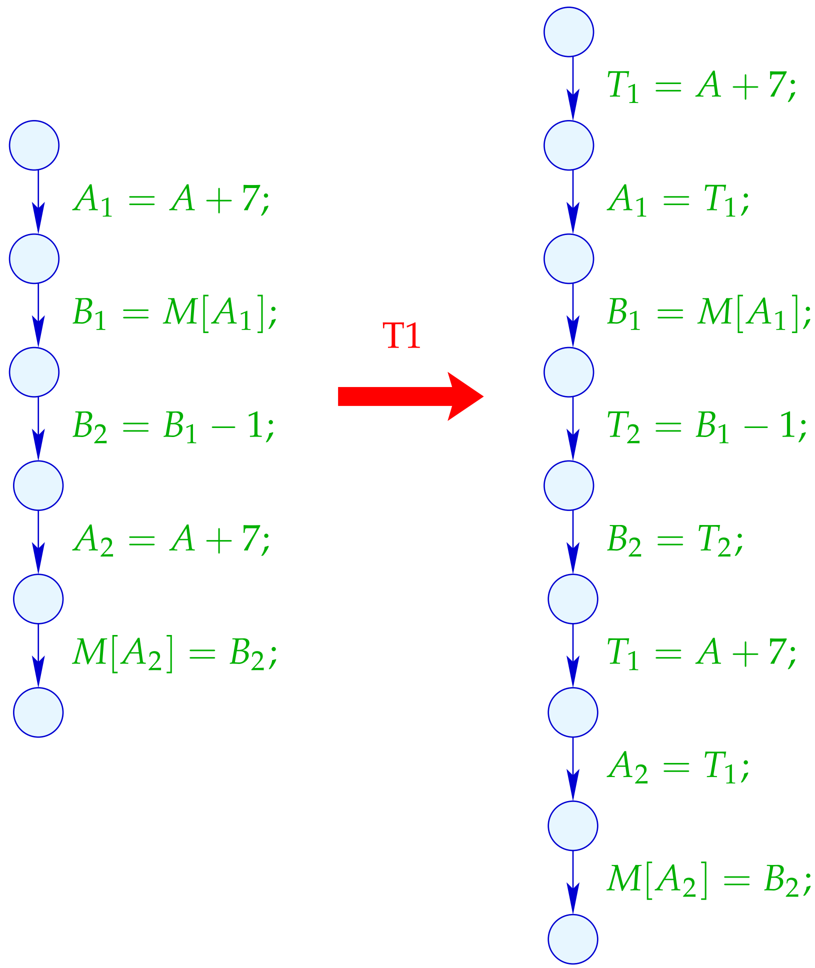
(2) Werte von Variablen: T4

- \+ erzeugt tote Variablen

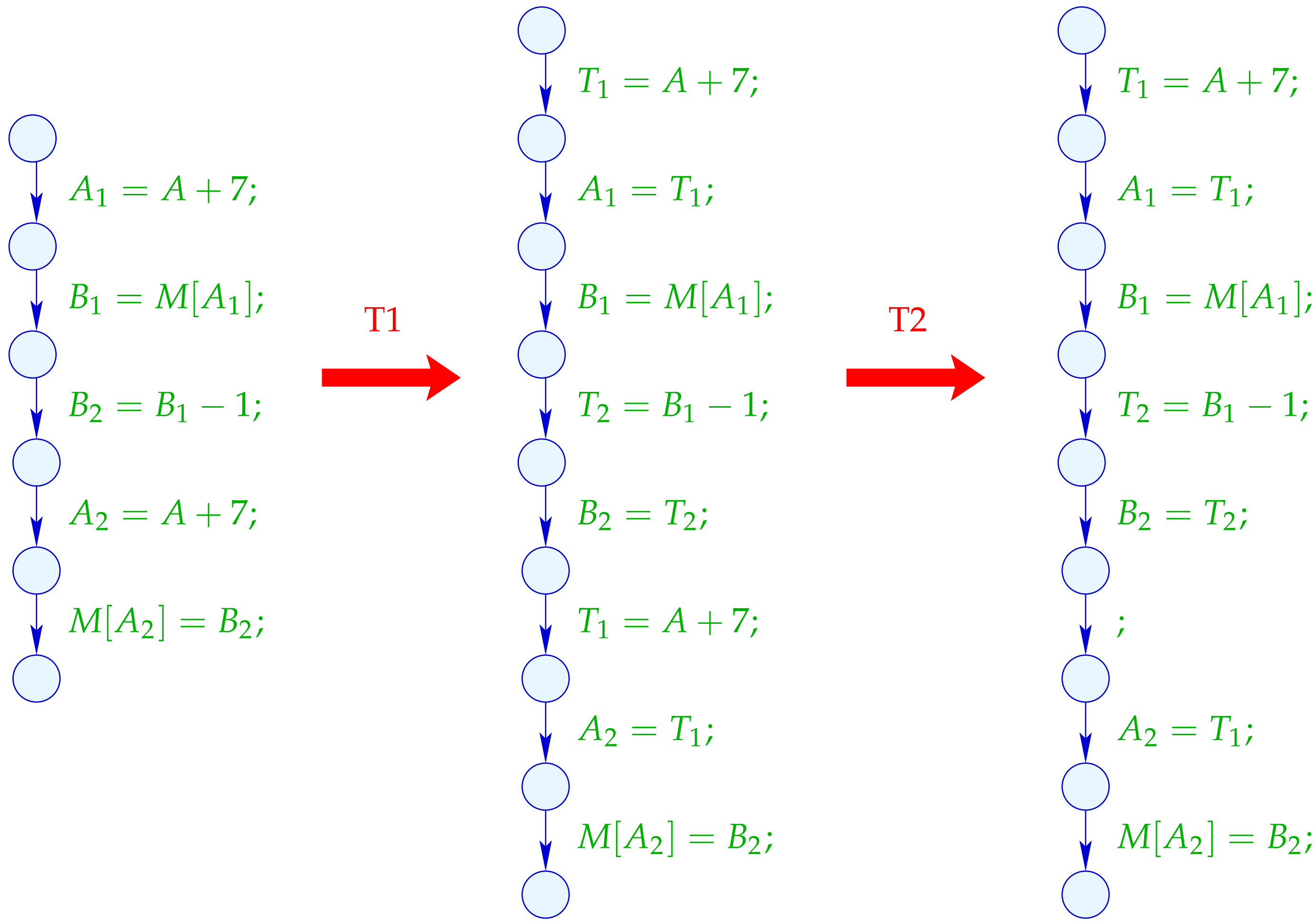
(3) (wahre) Lebendigkeit von Variablen: T3

- \+ beseitigt Zuweisungen an tote Variablen

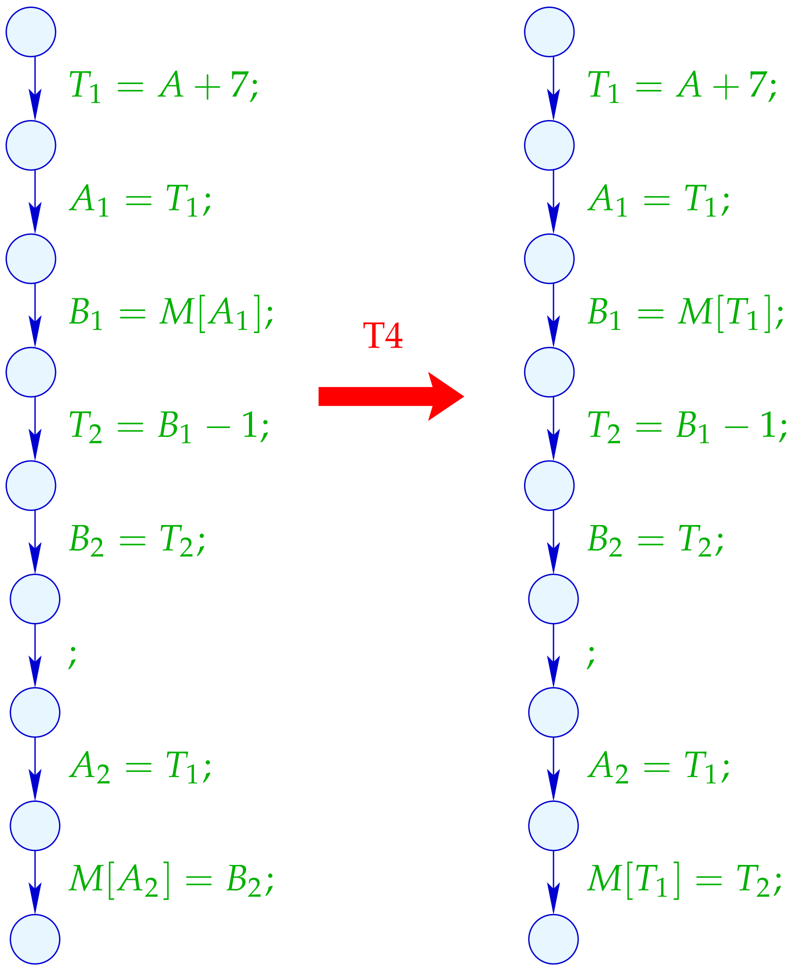
Beispiel: `a[7]--;`


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

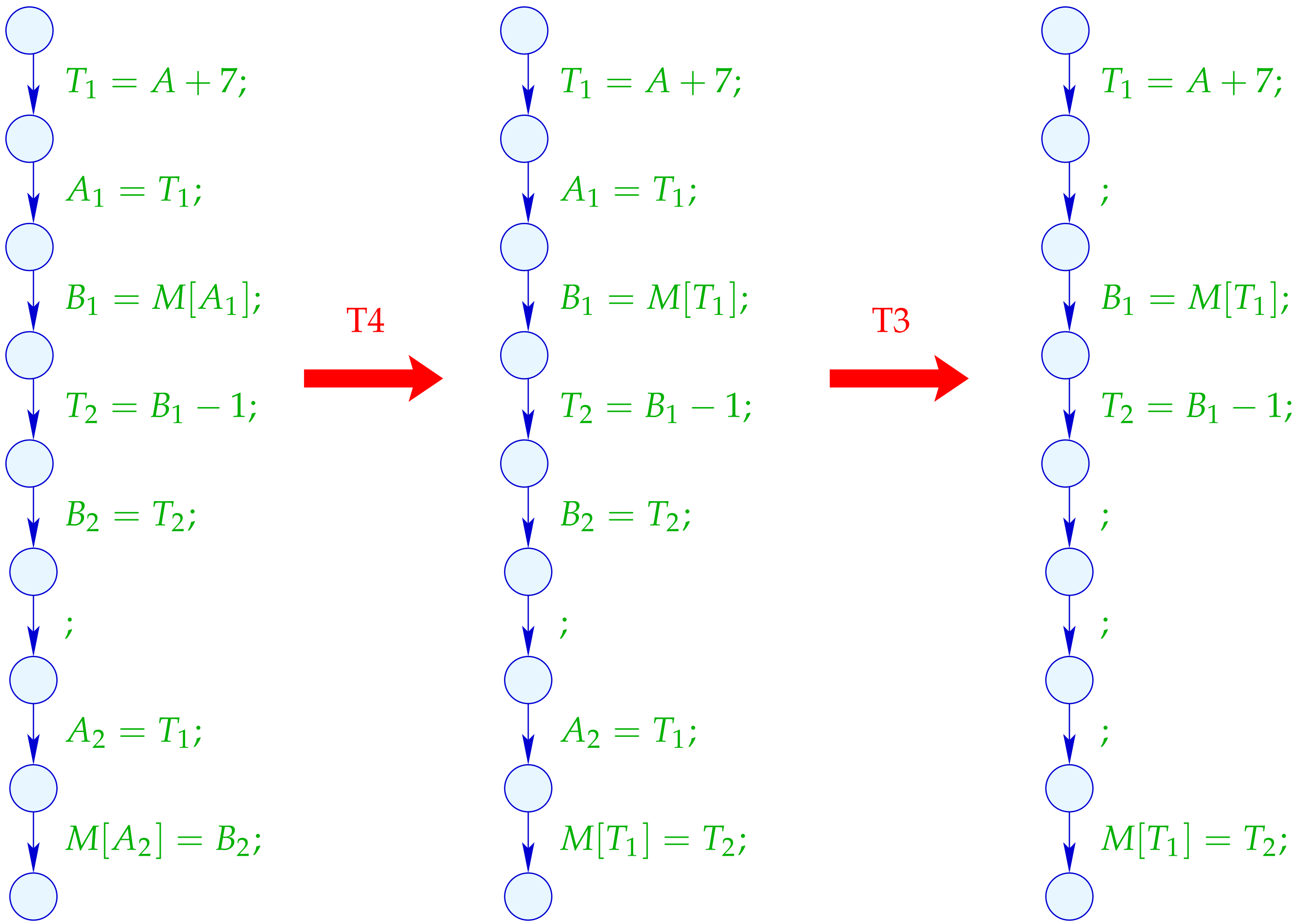
Beispiel: `a[7]--;`


$A_1 = A + 7;$
$B_1 = M[A_1];$
$B_2 = B_1 - 1;$
$A_2 = A + 7;$
$M[A_2] = B_2;$
T1
$T_1 = A + 7;$
$A_1 = T_1;$
$B_1 = M[A_1];$
$T_2 = B_1 - 1;$
$B_2 = T_2;$
$T_1 = A + 7;$
$A_2 = T_1;$
$M[A_2] = B_2;$
T2
$T_1 = A + 7;$
$A_1 = T_1;$
$B_1 = M[A_1];$
$T_2 = B_1 - 1;$
$B_2 = T_2;$
$;$
$A_2 = T_1;$
$M[A_2] = B_2;$

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



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



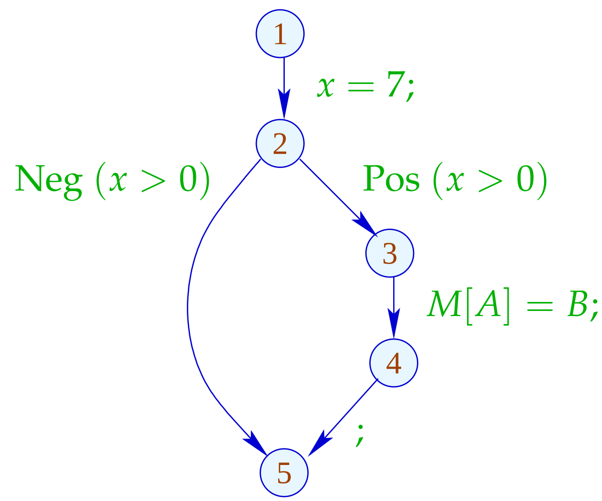
1.4 Konstanten-Propagation

Idee:

Führe möglichst große Teile des Codes bereits zur Compilezeit aus!

Beispiel:

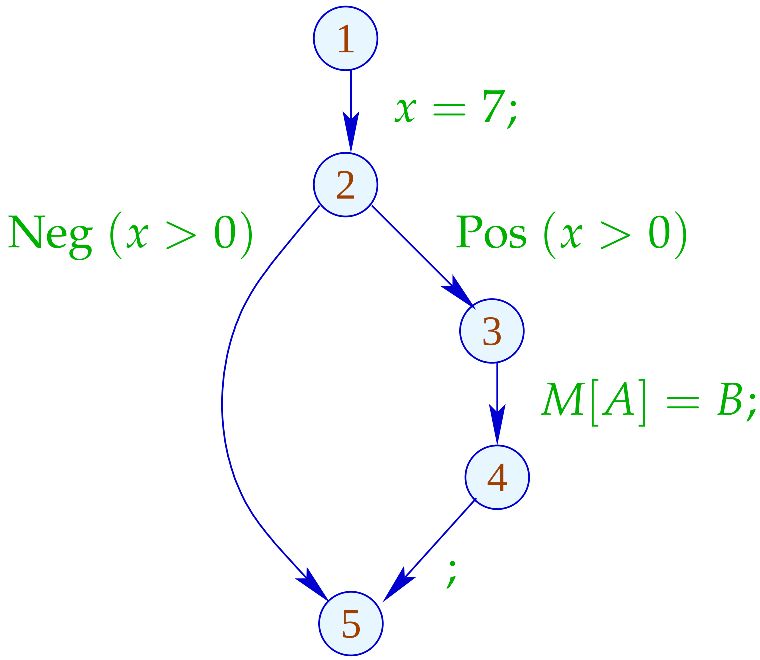
$$x = 7;$$
$$\text{if } (x > 0)$$
$$\quad M[A] = B;$$



Offenbar hat x stets den Wert 7 :-)

Deshalb wird stets der Speicherzugriff durchgeführt :-))

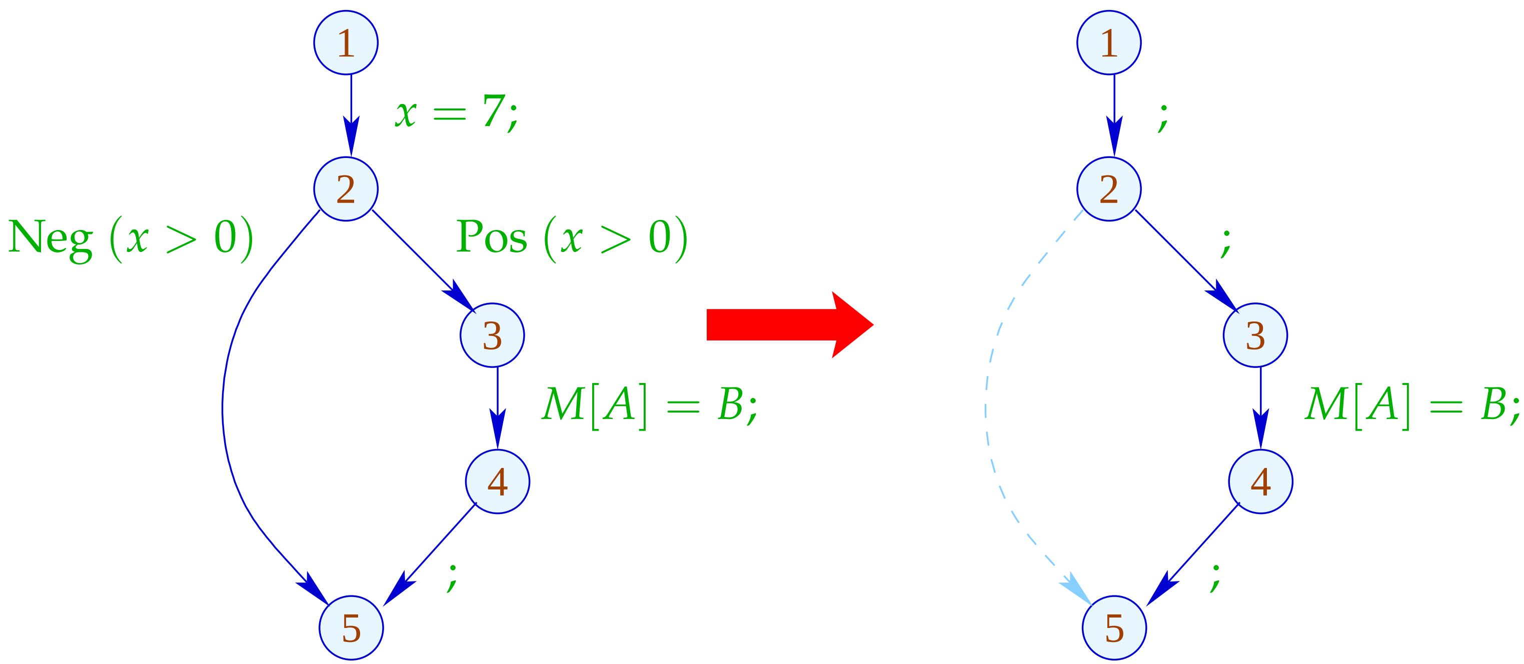
Ziel:



Offenbar hat x stets den Wert 7 :-)

Deshalb wird stets der Speicherzugriff durchgeführt :-))

Ziel:



Idee:

Entwerfe eine Analyse, die für jedes u

- die Werte ermittelt, die Variablen sicher haben;
- mitteilt, ob u überhaupt erreichbar ist :-)

Idee:

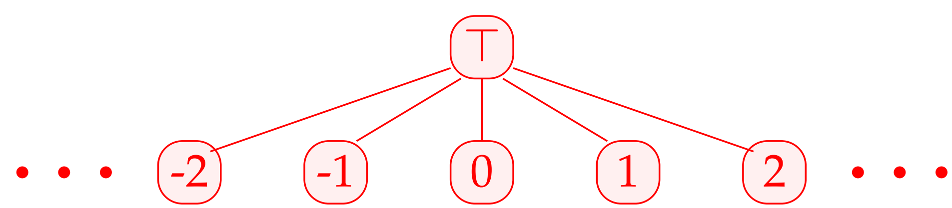
Entwerfe eine Analyse, die für jedes u

- die Werte ermittelt, die Variablen sicher haben;
- mitteilt, ob u überhaupt erreichbar ist :-)

Den vollständigen Verband konstruieren wir in zwei Schritten.

(1) Die möglichen Werte für Variablen:

$$\mathbb{Z}^\top = \mathbb{Z} \cup \{\top\} \qquad \text{mit} \quad x \sqsubseteq y \quad \text{gdw. } y = \top \text{ oder } x = y$$



Achtung: $\mathbb{Z}^\top$ ist selbst kein vollständiger Verband :-(

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

// $\bot$ heißt: “nicht erreichbar” :-))

mit $D_1 \sqsubseteq D_2$ gdw. $\bot = D_1$ oder

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

Bemerkung: $\mathbb{D}$ ist ein vollständiger Verband :-)

Achtung: $\mathbb{Z}^\top$ ist selbst kein vollständiger Verband :-(

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

// $\perp$ heißt: "nicht erreichbar" :-))

mit $D_1 \sqsubseteq D_2$ gdw. $\perp = D_1$ oder

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

Bemerkung: $\mathbb{D}$ ist ein vollständiger Verband :-)

Betrachte dazu $X \subseteq \mathbb{D}$. O.E. $\perp \notin X$.

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

Ist $X = \emptyset$, dann $\bigsqcup X = \perp \in \mathbb{D}$:-)

Ist $X \neq \emptyset$, dann ist $\bigsqcup X = D$ mit

$$D\, x = \bigsqcup\{f\, x \mid f \in X\}$$

$$= \begin{cases} z & \text{falls} \quad f\, x = z \quad (f \in X) \\ \top & \text{sonst} \end{cases}$$

:-))

Ist $X \neq \emptyset$, dann ist $\bigsqcup X = D$ mit

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

:-))

Zu jeder Kante $k = (\_, lab, \_)$ konstruieren wir eine Effekt-Funktion $[\![k]\!]^\sharp = [\![lab]\!]^\sharp : \mathbb{D} \to \mathbb{D}$, die die konkrete Berechnung simuliert.

Offenbar ist $[\![lab]\!]^\sharp \bot = \bot$ für alle lab :-)

Sei darum nun $\bot \neq D \in \mathit{Vars} \to \mathbb{Z}^\top$.

Idee:

- Wir benutzen D, um die Werte von Ausdrücken zu ermitteln.

Idee:

- Wir benutzen D, um die Werte von Ausdrücken zu ermitteln.
- Für manche Teilausdrücke erhalten wir $\top$:-)

Idee:

- Wir benutzen D, um die Werte von Ausdrücken zu ermitteln.
- Für manche Teilausdrücke erhalten wir $\top$:-)

$\Longrightarrow$

Wir müssen die konkreten Operatoren $\Box$ durch abstrakte Operatoren $\Box^\sharp$ ersetzen, die mit $\top$ umgehen können:

$$a \,\Box^\sharp\, b = \begin{cases} \top & \text{falls} \quad a = \top \text{ oder } b = \top \\ a \,\Box\, b & \text{sonst} \end{cases}$$

Idee:

- Wir benutzen D, um die Werte von Ausdrücken zu ermitteln.
- Für manche Teilausdrücke erhalten wir $\top$:-)

  $\Longrightarrow$

  Wir müssen die konkreten Operatoren $\Box$ durch abstrakte Operatoren $\Box^\sharp$ ersetzen, die mit $\top$ umgehen können:

$$a \,\Box^\sharp\, b = \begin{cases} \top & \text{falls} \quad a = \top \text{ oder } b = \top \\ a \,\Box\, b & \text{sonst} \end{cases}$$

- Mit den abstrakten Operatoren können wir eine abstrakte Ausdrucks-Auswertung definieren:

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

Abstrakte Ausdrucksauswertung ist wie konkrete Ausdrucksauswertung, aber mit abstrakten Werten und Operatoren. Hier:

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

... analog für unäre Operatoren :-)

Abstrakte Ausdrucksauswertung ist wie konkrete Ausdrucksauswertung, aber mit abstrakten Werten und Operatoren. Hier:

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

... analog für unäre Operatoren :-)

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

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

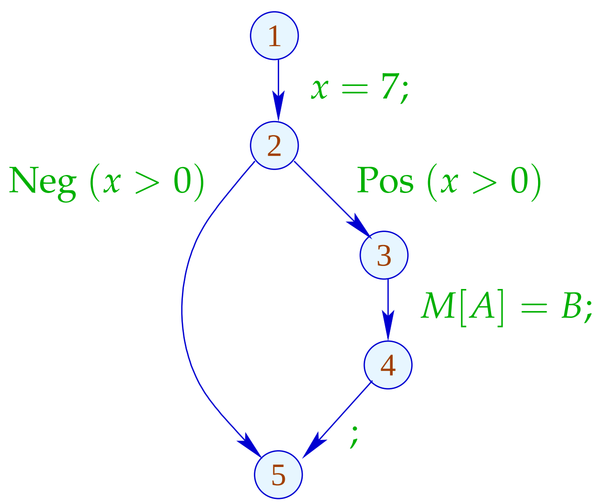
Damit erhalten wir für die Kanten-Effekte $[\![lab]\!]^\sharp$:

$$
\begin{array}{lll}
[\![;]\!]^\sharp\, D & = & D \\
[\![\text{Pos}\,(e)]\!]^\sharp\, D & = & \left\{ \begin{array}{ll} \bot & \text{falls} \quad 0 = [\![e]\!]^\sharp\, D \\ D & \text{sonst} \end{array} \right. \\
[\![\text{Neg}\,(e)]\!]^\sharp\, D & = & \left\{ \begin{array}{ll} D & \text{falls} \quad 0 \sqsubseteq [\![e]\!]^\sharp\, D \\ \bot & \text{sonst} \end{array} \right. \\
[\![x = e;]\!]^\sharp\, D & = & D \oplus \{x \mapsto [\![e]\!]^\sharp\, D\} \\
[\![x = M[R];]\!]^\sharp\, D & = & D \oplus \{x \mapsto \top\} \\
[\![M[R_1] = R_2;]\!]^\sharp\, D & = & D
\end{array}
$$

... sofern $D \neq \bot$:-)

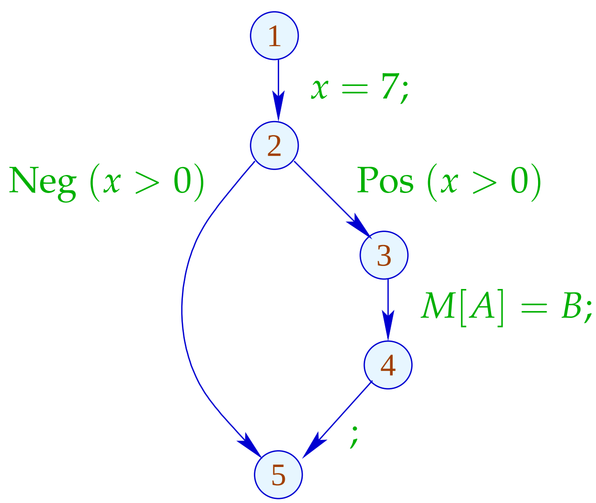
An *start* gilt $D_\perp = \{x \mapsto \top \mid x \in Vars\}$.

Beispiel:



An *start* gilt $D_\perp = \{x \mapsto \top \mid x \in \mathit{Vars}\}$.

Beispiel:



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

Die abstrakten Kanten-Effekte $[\![k]\!]^\sharp$ setzen wir wieder zu den Effekten von Pfaden $\pi = k_1 \ldots k_r$ zusammen durch:

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

Idee zur Korrektheit: Abstrakte Interpretation

Cousot, Cousot 1977

Patrick Cousot, ENS, Paris