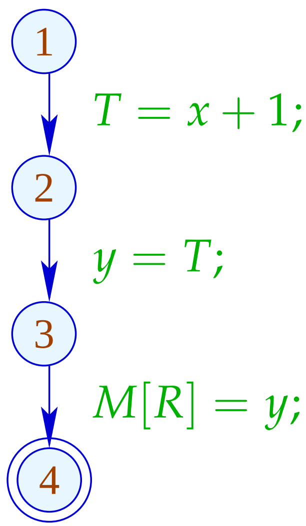
1.3 Beseitigung überflüssiger Umspeicherungen

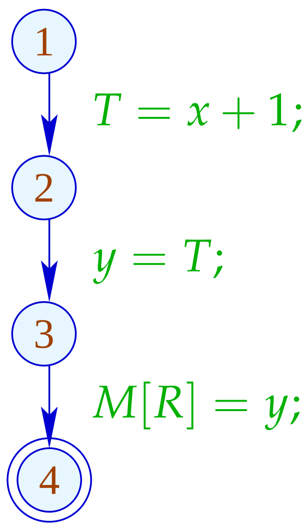
Beispiel:



Offenbar ist die Umspeicherung nutzlos :-(

1.3 Beseitigung überflüssiger Umspeicherungen

Beispiel:

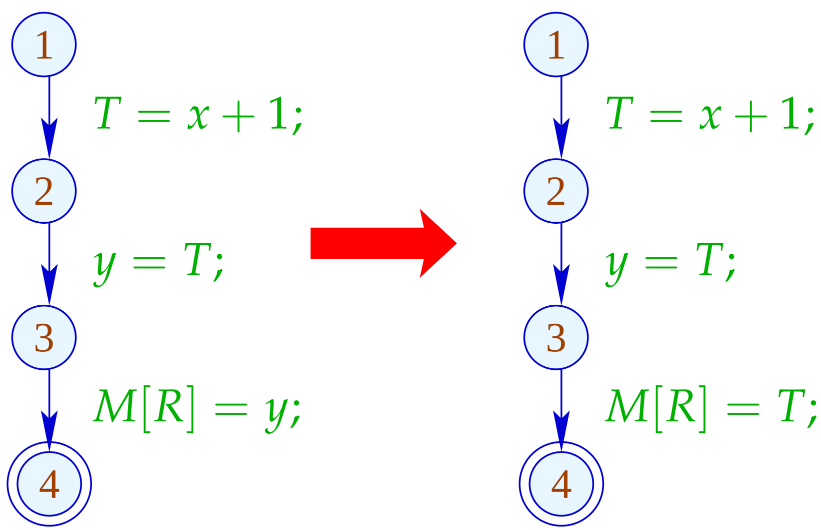


Offenbar ist die Umspeicherung nutzlos :-(

Statt y könnten wir auch T abspeichern :-)

1.3 Beseitigung überflüssiger Umspeicherungen

Beispiel:

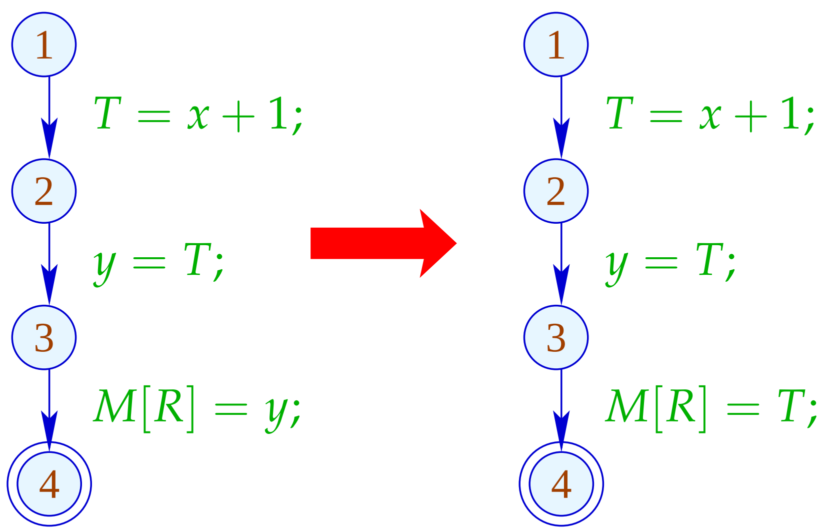


Offenbar ist die Umspeicherung nutzlos :-(

Statt y könnten wir auch T abspeichern :-)

1.3 Beseitigung überflüssiger Umspeicherungen

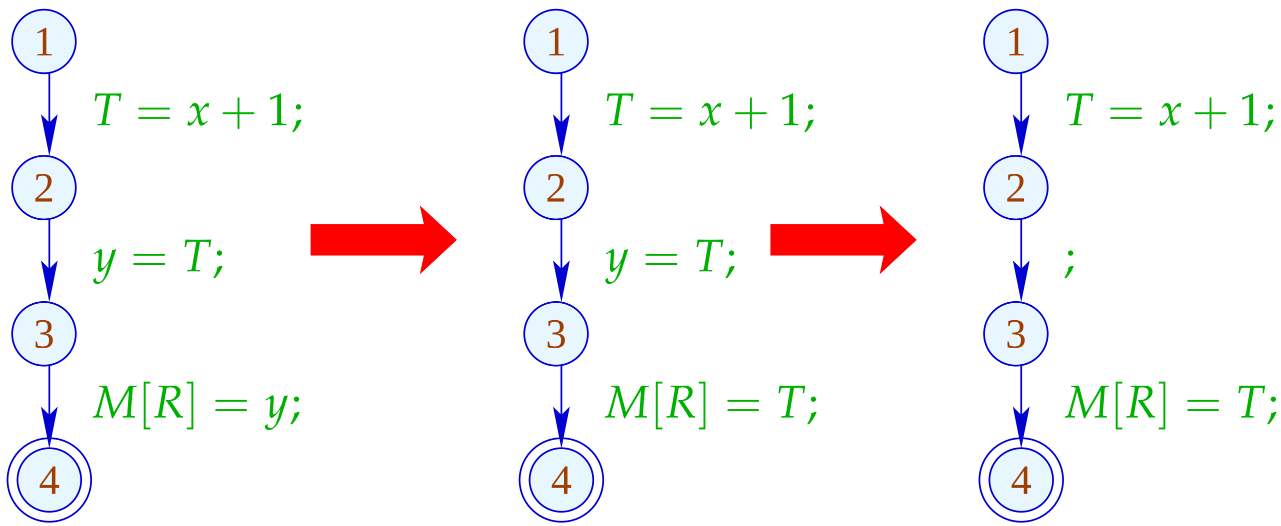
Beispiel:



Vorteil: Jetzt ist y tot :-))

1.3 Beseitigung überflüssiger Umspeicherungen

Beispiel:



Vorteil: Jetzt ist y tot :-))

Idee:

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

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

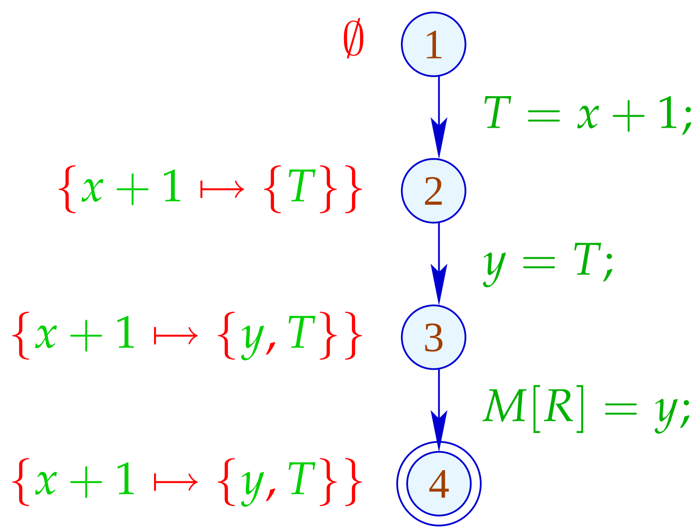
Idee:

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

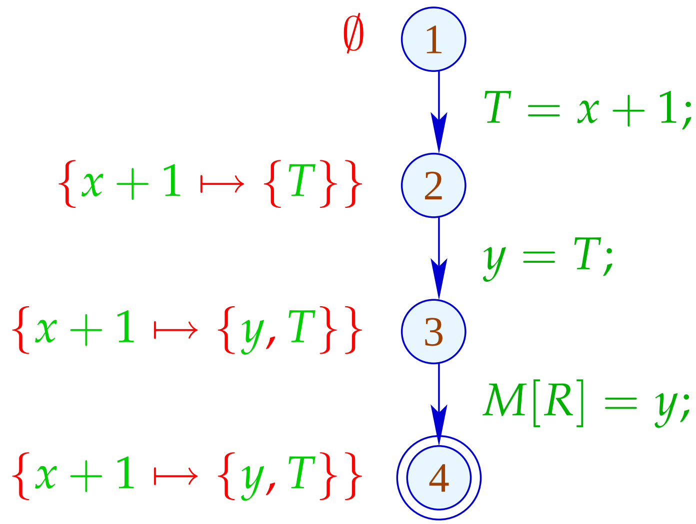
Wir benutzen: $\mathbb{V} = \mathit{Expr} \to 2^{\mathit{Vars}}$ und definieren:

$$
\begin{array}{lll}
[\![;]\!]^\sharp\, V & = & V \\
[\![\mathrm{Pos}(e)]\!]^\sharp\, V & = & [\![\mathrm{Neg}(e)]\!]^\sharp\, V \quad = \quad V \\
[\![x = e;]\!]^\sharp\, V\, e' & = & \begin{cases} \{x\} & \text{falls} \quad e' = e \\ (V\, e')\backslash\{x\} & \text{sonst} \end{cases} \\
[\![x = y;]\!]^\sharp\, V\, e & = & \begin{cases} (V\, e) \cup \{x\} & \text{falls} \quad y \in V\, e \\ (V\, e)\backslash\{x\} & \text{sonst} \end{cases} \\
[\![x = M[R];]\!]^\sharp\, V\, e & = & (V\, e)\backslash\{x\} \\
[\![M[R_1] = R_2;]\!]^\sharp\, V & = & V
\end{array}
$$

Im Beispiel:



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:

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

Beobachtung:

Die neuen Kanten-Effekte sind distributiv:

Dazu zeigen wir, dass die folgenden Funktionen distributiv sind:

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

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

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

Offenbar gilt:

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

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

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

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

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

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

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

(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

$\Longrightarrow$

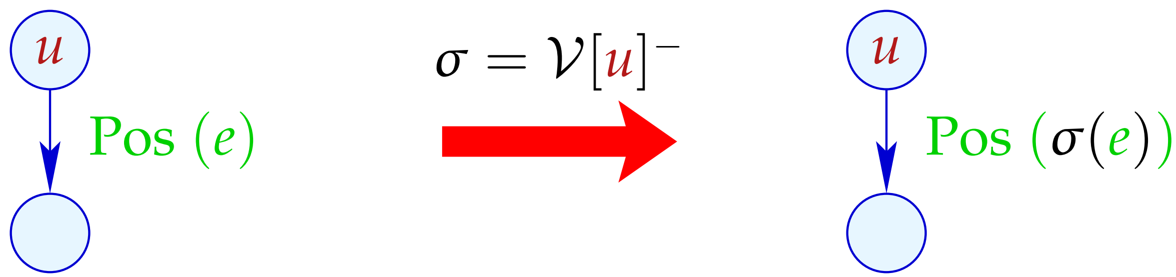
der Zugriff auf x kann durch Zugriff auf T_e ersetzt werden :-)

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

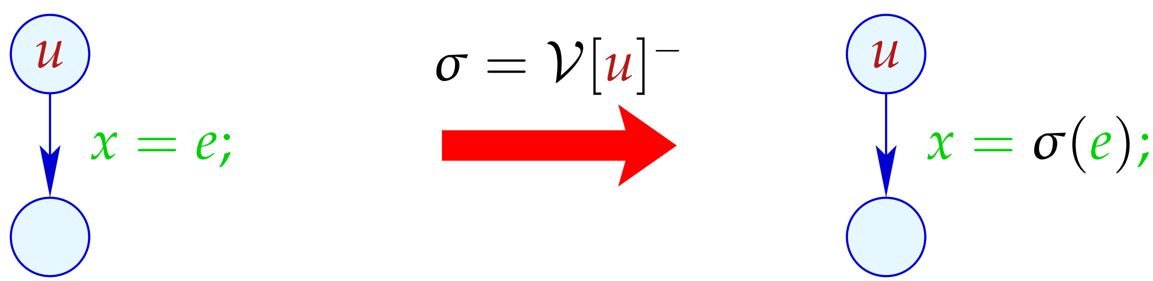
$$V^- \, x = \begin{cases} T_e & \text{falls } x \in V\, e \\ x & \text{sonst} \end{cases}$$

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

Transformation 4:



... analog für Kanten mit Neg (e)



Transformation 4 (Forts.):

$$\begin{tikzcd} u \arrow[d, "x = M[R];"] \\ {} \end{tikzcd} \quad \xrightarrow{\sigma = \mathcal{V}[u]^-} \quad \begin{tikzcd} u \arrow[d, "x = M[\sigma(R)];"] \\ {} \end{tikzcd}$$

$$\begin{tikzcd} u \arrow[d, "M[R_1] = R_2;"] \\ {} \end{tikzcd} \quad \xrightarrow{\sigma = \mathcal{V}[u]^-} \quad \begin{tikzcd} u \arrow[d, "M[\sigma(R_1)] = \sigma(R_2);"] \\ {} \end{tikzcd}$$

Vorgehen insgesamt:

(1) Verfügbarkeit von Ausdrücken: T1 + T2

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

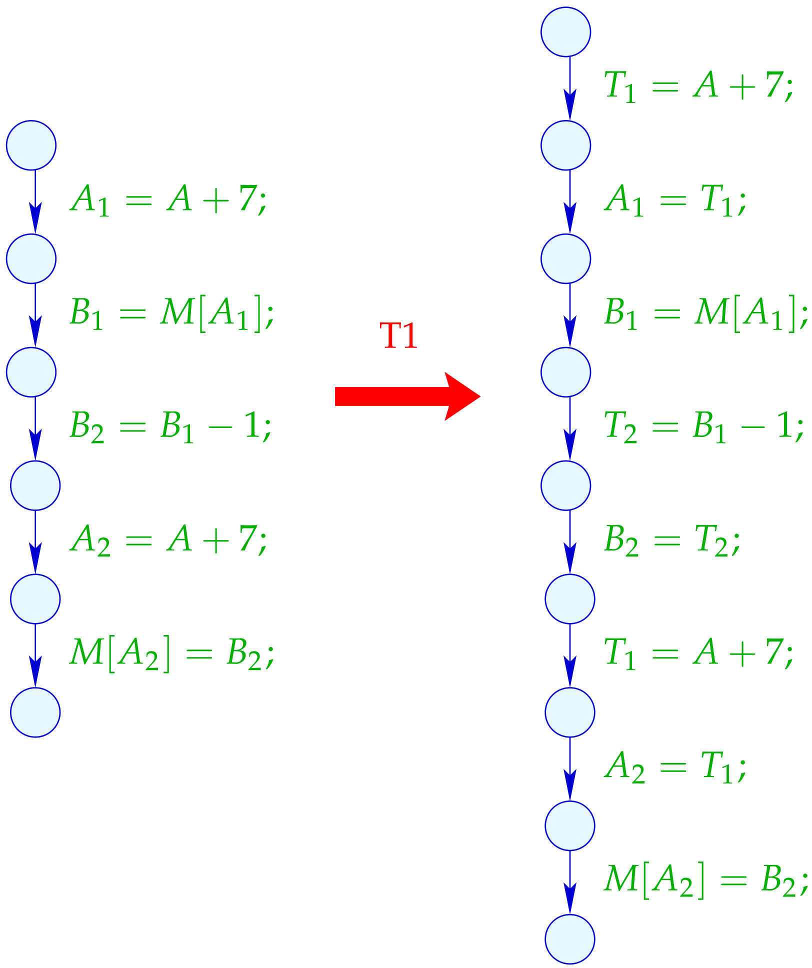
(2) Werte von Variablen: T4

- \+ erzeugt tote Variablen

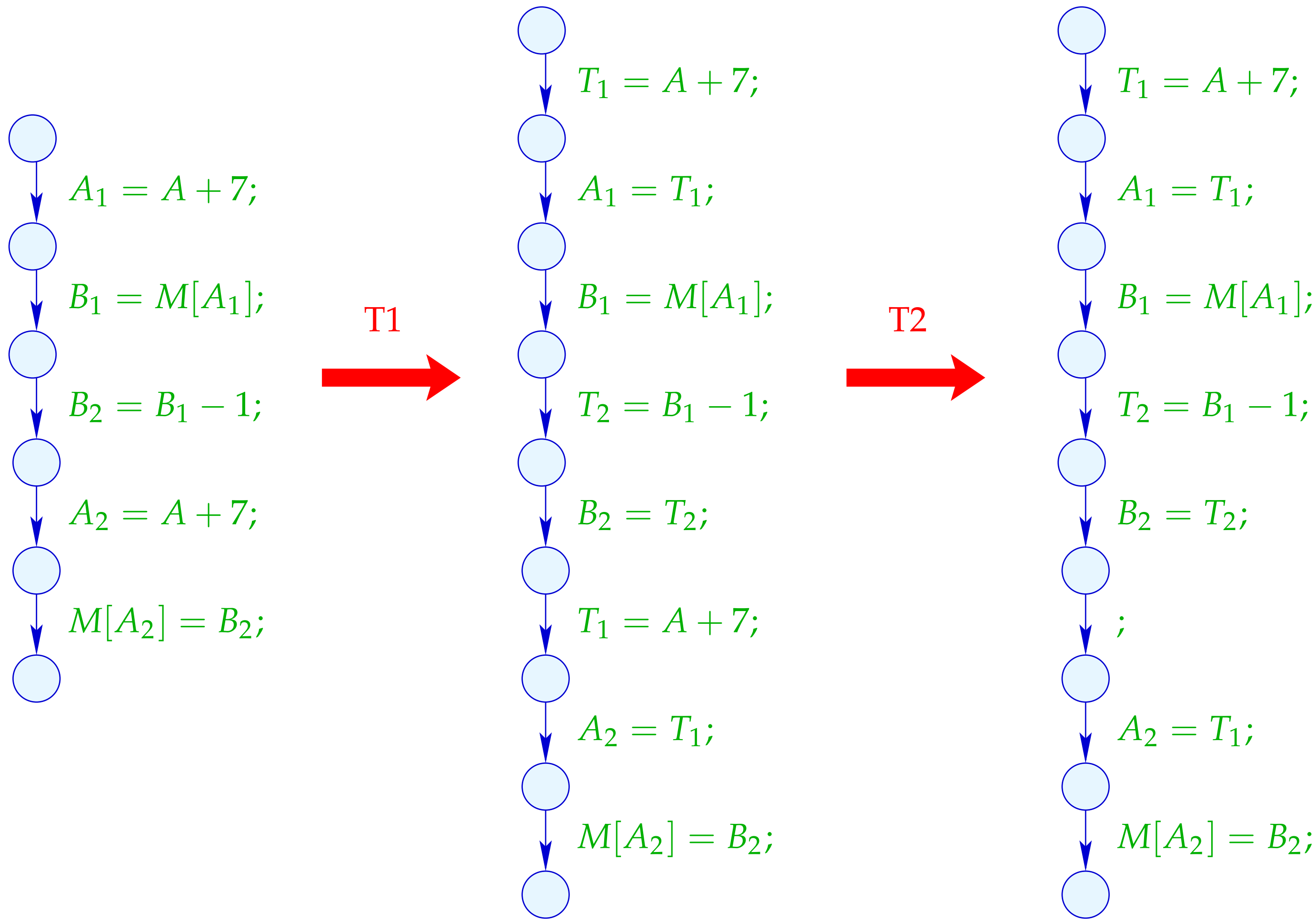
(3) (wahre) Lebendigkeit von Variablen: T3

- \+ beseitigt Zuweisungen an tote Variablen

Beispiel: `a[7]--;`

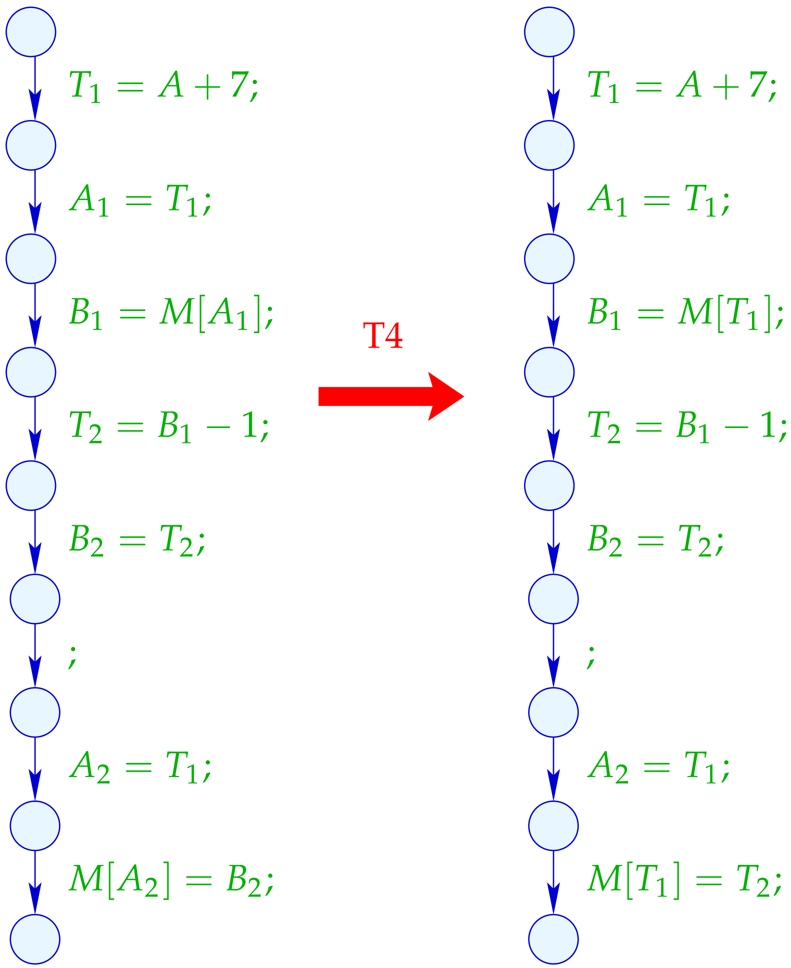

$A_1 = A + 7;$
$B_1 = M[A_1];$
$B_2 = B_1 - 1;$
$A_2 = A + 7;$
$M[A_2] = B_2;$
T1
$T_1 = A + 7;$
$A_1 = T_1;$
$B_1 = M[A_1];$
$T_2 = B_1 - 1;$
$B_2 = T_2;$
$T_1 = A + 7;$
$A_2 = T_1;$
$M[A_2] = B_2;$

Beispiel: `a[7]--;`

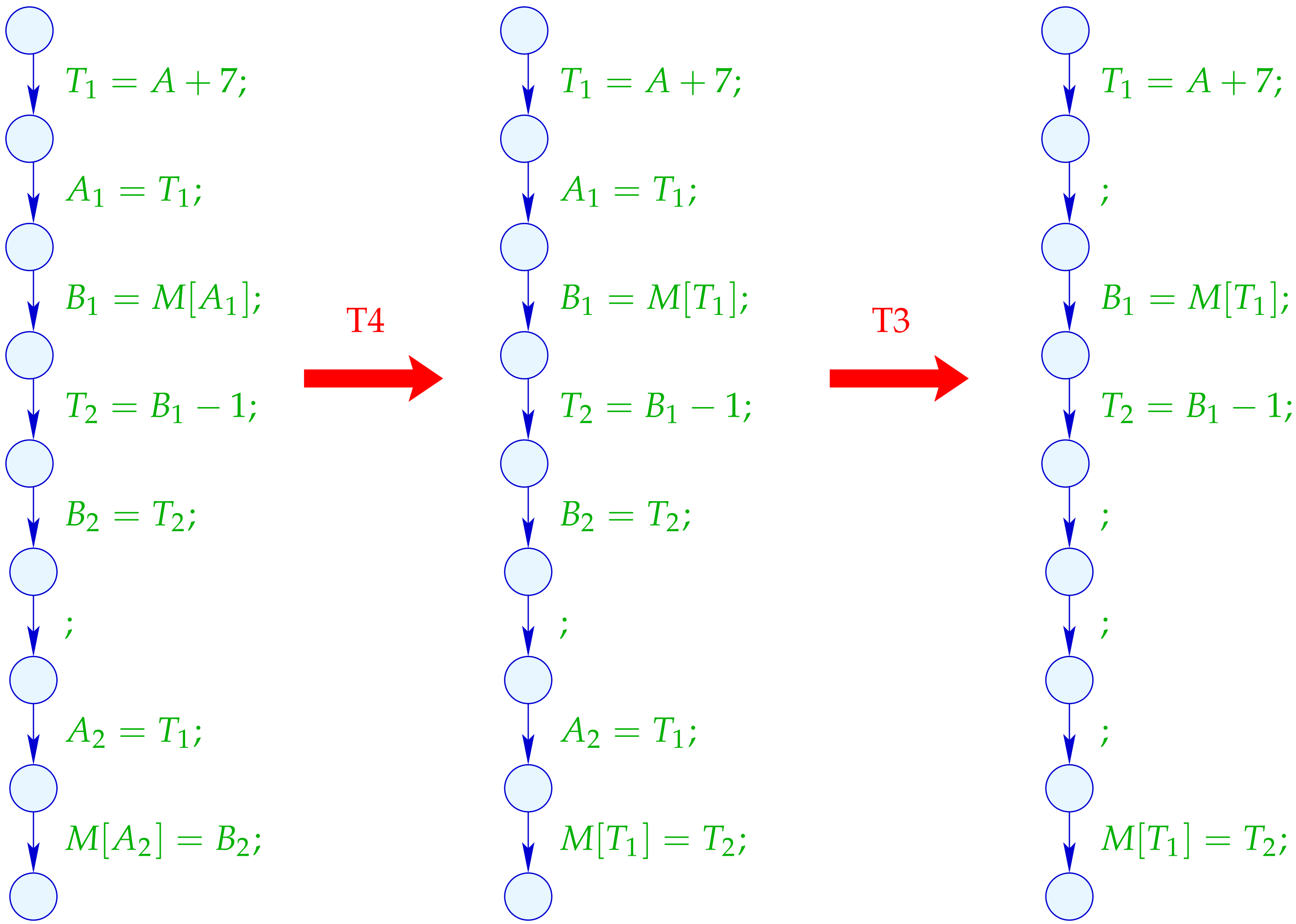


$A_1 = A + 7;$
$B_1 = M[A_1];$
$B_2 = B_1 - 1;$
$A_2 = A + 7;$
$M[A_2] = B_2;$
T1
$T_1 = A + 7;$
$A_1 = T_1;$
$B_1 = M[A_1];$
$T_2 = B_1 - 1;$
$B_2 = T_2;$
$T_1 = A + 7;$
$A_2 = T_1;$
$M[A_2] = B_2;$
T2
$T_1 = A + 7;$
$A_1 = T_1;$
$B_1 = M[A_1];$
$T_2 = B_1 - 1;$
$B_2 = T_2;$
$;$
$A_2 = T_1;$
$M[A_2]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

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



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



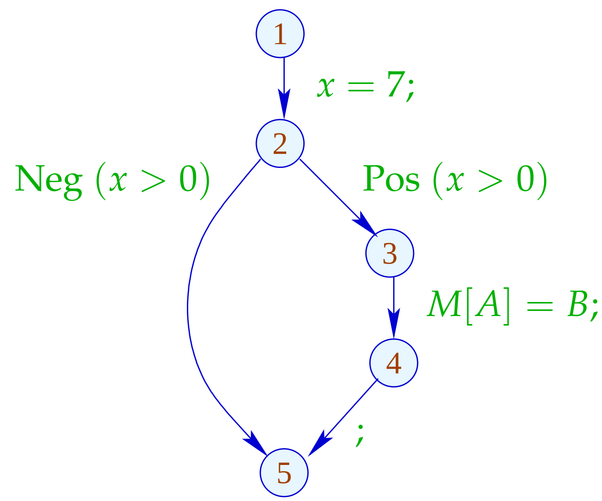
1.4 Konstanten-Propagation

Idee:

Führe möglichst große Teile des Codes bereits zur Compilezeit aus!

Beispiel:

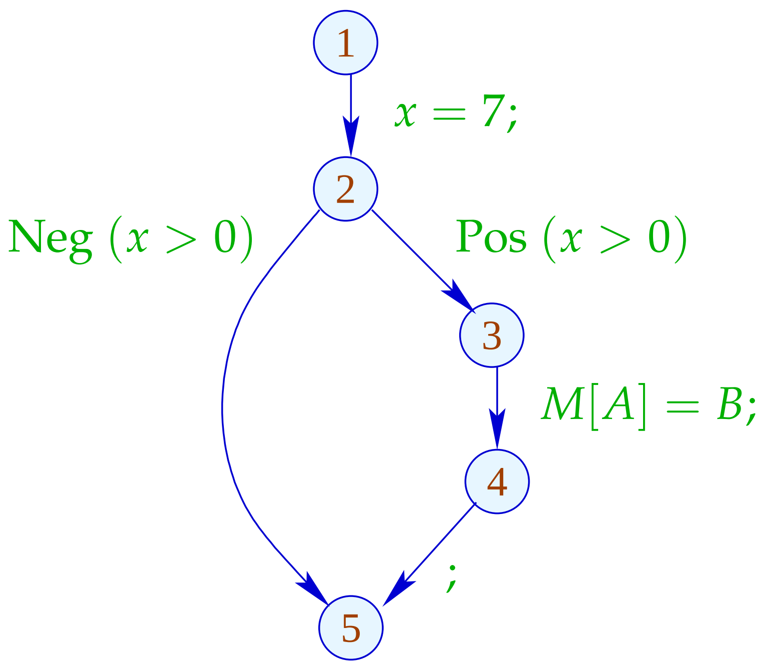
$$x = 7;$$
$$\text{if } (x > 0)$$
$$\qquad M[A] = B;$$



Offenbar hat x stets den Wert 7 :-)

Deshalb wird stets der Speicherzugriff durchgeführt :-))

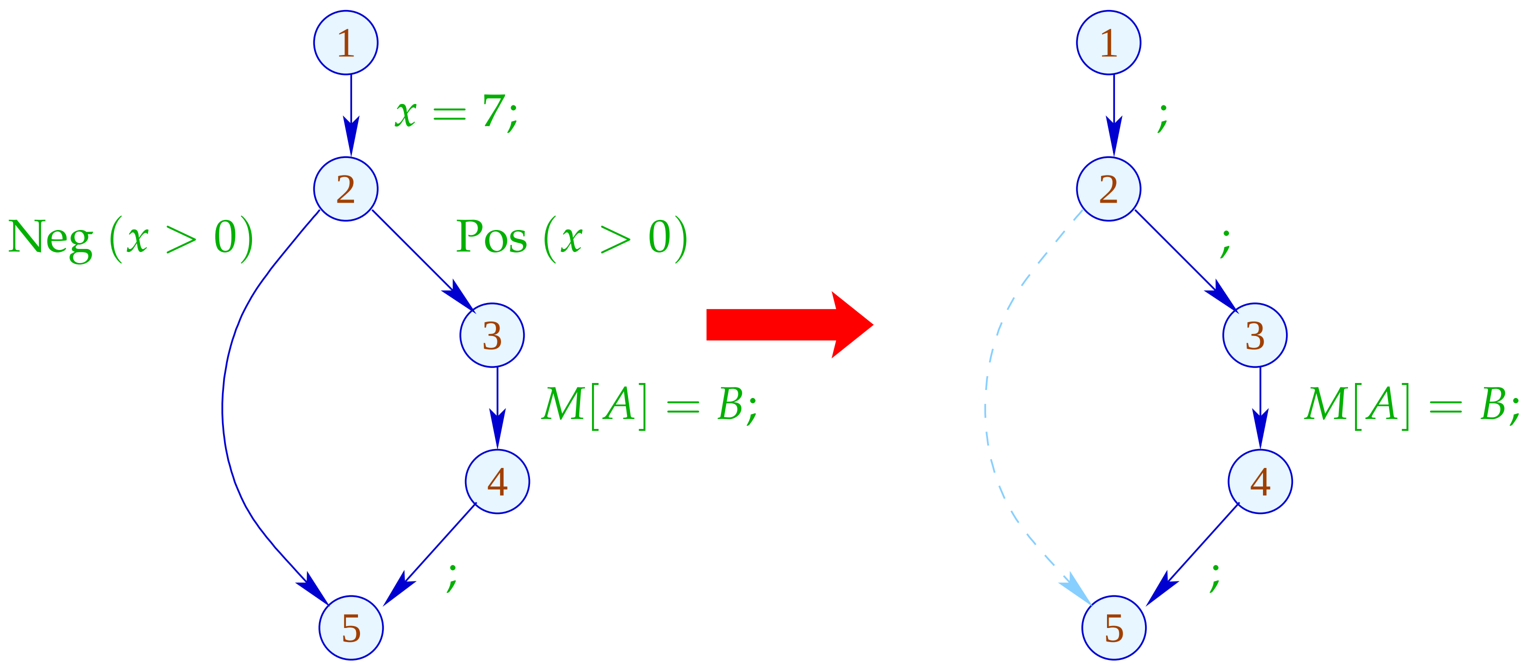
Ziel:



Offenbar hat x stets den Wert 7 :-)

Deshalb wird stets der Speicherzugriff durchgeführt :-))

Ziel:



Verallgemeinerung: Partielle Auswertung

Neil D. Jones, DIKU, Kopenhagen

Idee:

Entwerfe eine Analyse, die für jedes u

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

Idee:

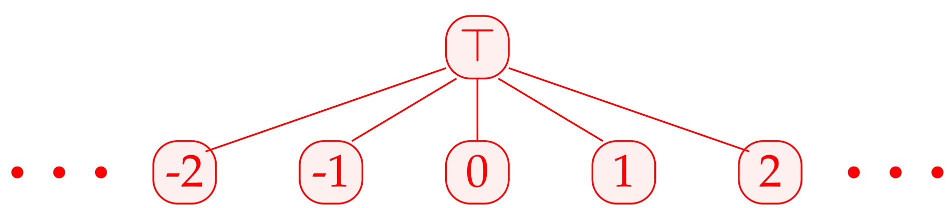
Entwerfe eine Analyse, die für jedes u

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

// $\perp$ 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 :-)

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

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

:-))

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

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

:-))

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

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

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

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:

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

... analog für unäre Operatoren :-)

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

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

... analog für unäre Operatoren :-)

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

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