

Betrachte das folgende while-Programm:

```
if  $[5x + 2 < 7y]^1$  then
|    $[y := 2y]^2$ 
end if
while  $[(5x + 2) + y < 3x + z]^3$  do
|   if  $[7y < 3x + z]^4$  then
|   |    $[z := z - 1]^5$ 
|   else
|   |    $[y := 2y]^6$ 
|   end if
end while
 $[x := 3x]^7$ 
```

Das Programm induziert u.a. die folgenden Elemente:

$\text{Var} := \{x, y, z\}$	$b_1 := [5x + 2 < 7y]^1$	$b_2 := [y := 2y]^2$
$B := \{b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$	$b_3 := [(5x + 2) + y < 3x + z]^3$	$b_4 := [7y < 3x + z]^4$
$b_5 := [z := z - 1]^5$	$b_6 := [y := 2y]^6$	$b_7 := [x := 3x]^7$

Die zusammengesetzten Ausdrücke im Programm lauten

$$\text{AExp} = \{3x, 5x, 2y, 7y, z - 1, 5x + 2, 5x + 2 + y, 3x + z\}.$$

Bemerkung: Die atomaren Ausdrücke, also Literale und Variablen, werden hier weggelassen.

Anmerkungen zur Tabelle auf der nächsten Seite:

1. In der Durchführung einer Kleene-Iteration kommt es ausschließlich auf sich ändernde Werte an. Es müssen nur die Blöcke neu berechnet werden, deren Vorgänger-Daten in der vorherigen Iteration verändert wurden.
2. Leere Zellen markieren, wo sich keine Vorgänger der letzten Iteration verändert haben.
3. Die Pfeile markieren, wo Änderungen aus einer vorigen Iteration zu neuen Berechnungen in der Nächsten führen. Sie folgen dem Gleichungssystem und dem Kontrollfluss-Graphen.

Available Expressions

Gegeben: Der Kontrollfluss-Graph eines while-Programms.

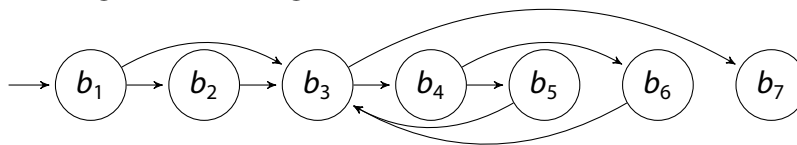
Gesucht: Für jeden Block die Menge der Ausdrücke, die in jedem Weg **vor** diesem Block schon mal berechnet werden **musste**, ohne dass zwischendurch eine Variable neu zugewiesen wird.

Betrachte das Datenflusssystem $\langle \langle B, E, F \rangle, \langle \mathcal{P}(\text{AExp}), \supseteq \rangle, \emptyset, TF \rangle$ (beachte die umgedrehte Ordnungsrelation, da dies eine **must**-Analyse ist) mit $TF := \{ f_b : \mathcal{P}(\text{AExp}) \rightarrow \mathcal{P}(\text{AExp}) \mid b \in B \}$ wobei die Transferfunktionen von folgender Form sind: $f_b(X) := (X \setminus \text{kill}(b)) \cup \text{gen}(b)$. Dabei sind $\text{kill}(b) \in D$ und $\text{gen}(b) \in D$ für jeden Block $b \in B$ für Available Expressions so definiert:

$$\text{kill}(b) := \begin{cases} \text{AExp} \cap \text{Vars}^{-1}(x) & \text{falls } b = [x := a]^I \\ \emptyset & \text{sonst} \end{cases} \quad \text{gen}(b) := \text{AExp}(b) \setminus \text{kill}(b)$$

Beachte die Verwendung von $\cap$. In Tabellen-Form werden nun sowohl die kill- und gen-Werte, die Gleichungen des induzierten Gleichungssystems, als auch die Kleene-Iteration dargestellt.

Kontrollflussgraph in Programmrichtung:



b	b_1	b_2	b_3	b_4	b_5	b_6	b_7
$\text{kill}(b)$		$2y, 7y, 5x + 2 + y$			$z - 1, 3x + z$	$2y, 7y, 5x + 2 + y$	$3x, 3x + z, 5x, 5x + 2, 5x + 2 + y$
$\text{gen}(b)$	$5x, 7y, 5x + 2$		$5x, 5x + 2, 5x + 2 + y, 3x, 3x + z$	$7y, 3x, 3x + z$			
X_b	$(i =)\emptyset$	$f_1(X_1)$	$f_1(X_1) \cap f_2(X_2) \cap f_5(X_5) \cap f_6(X_6)$	$f_3(X_3)$	$f_4(X_4)$	$f_4(X_4)$	$f_3(X_3)$
$\perp$	AExp	AExp	AExp	AExp	AExp	AExp	AExp
$g_S(\perp)$	$\emptyset$		$3x, 5x, 5x+2$				
$g_S^2(\perp)$		$5x, 5x+2, 7y$	$5x, 5x+2$	$5x, 5x+2, 3x, 3x+z, 5x+2+y$			$5x, 5x+2, 3x, 3x+z, 5x+2+y$
$g_S^3(\perp)$			...	...	$5x, 5x+2, 3x, 3x+z, 5x+2+y, 7y$	$5x, 5x+2, 3x, 3x+z, 5x+2+y, 7y$	...
$g_S^4(\perp)$			...				

Die Folge stabilisiert sich nach 3 Iterationen bei $\text{lfp}(g_S) = g_S^3(\perp)$:

Very Busy Expressions

Gegeben: Der Kontrollfluss-Graph eines while-Programms.

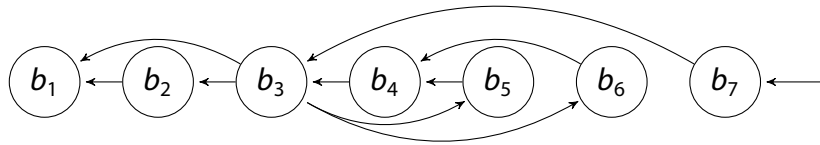
Gesucht: Für jeden Block, die Menge der Ausdrücke, die in jedem Weg **nach** diesem Block berechnet werden **müssen**, bevor eine der darin benutzten Variablen überschrieben wird.

Datenflusssystem $S = \langle \langle B, E, F \rangle, \langle \mathcal{P}(\text{AExp}), \supseteq \rangle, \emptyset, TF \rangle$ mit $TF := \{ f_b : \mathcal{P}(\text{AExp}) \rightarrow \mathcal{P}(\text{AExp}) \mid b \in B \}$ wieder von der Form $f_b(X) := (X \setminus \text{kill}(b)) \cup \text{gen}(b)$ mit

$$\text{kill}(b) := \begin{cases} \text{AExp} \cap \text{Vars}^{-1}(x) & \text{falls } b = [x := a]^i \\ \emptyset & \text{sonst} \end{cases} \quad \text{gen}(b) := \text{AExp}(b).$$

Bemerkung:

1. kill bleibt gleich zu Available Expressions, aber gen ist nicht mehr eingeschränkt.
2. Wir betrachten den Kontrollfluss-Graphen in entgegengesetzter Richtung ($\rightarrow X_b$).



b	b_1	b_2	b_3	b_4	b_5	b_6	b_7
$\text{kill}(b)$		$2y, 7y, 5x+2+y$			$z-1, 3x+z$	$2y, 7y, 5x+2+y$	$3x, 3x+z, 5x, 5x+2, 5x+2+y$
$\text{gen}(b)$	$5x, 5x+2, 7y$	$2y$	$5x, 5x+2, 5x+2+y, 3x, 3x+z$	$7y, 3x, 3x+z$	$z-1$	$2y$	$3x$
X_b	$f_2(X_2) \cap f_3(X_3)$	$f_3(X_3)$	$f_4(X_4) \cap f_7(X_7)$	$f_5(X_5) \cap f_6(X_6)$	$f_3(X_3)$	$f_3(X_3)$	$\emptyset$
$\perp$	AExp	AExp	AExp	AExp	AExp	AExp	AExp
$g_s(\perp)$	$5x, 5x+2, 3x, 3x+z, z-1$	...	$3x, 2y, 7y, z-1$	$5x, 5x+2, 3x, 2y, z-1$	...	...	$\emptyset$
$g_s^2(\perp)$	...	...	$3x$		...	...	
$g_s^3(\perp)$	$5x, 5x+2, 3x, 3x+z,$	$5x, 5x+2, 5x+2+y, 3x, 3x+z,$			$5x, 5x+2, 5x+2+y, 3x, 3x+z,$	$5x, 5x+2, 5x+2+y, 3x, 3x+z,$	
$g_s^4(\perp)$				$5x, 5x+2, 3x$			
$g_s^5(\perp)$			...				

Die Iteration stabilisiert sich nach 4 Schritten bei $\text{lfp}(g_s) = g_s^4(\perp)$: