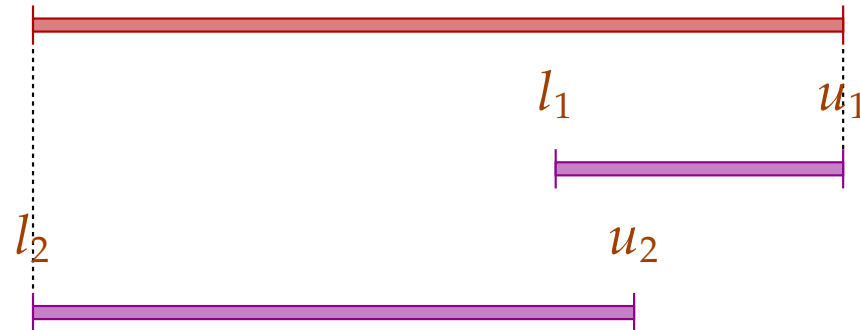


Damit:

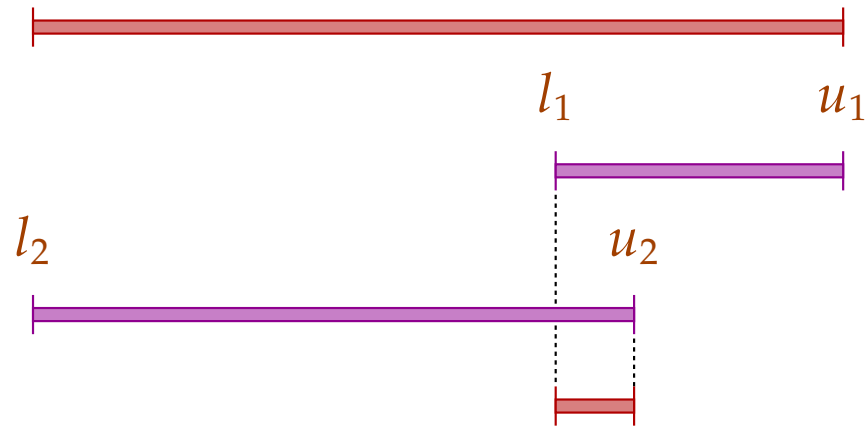
$$[l_1, u_1] \sqcup [l_2, u_2] = [l_1 \sqcap l_2, u_1 \sqcup u_2]$$



Damit:

$$[l_1, u_1] \sqcup [l_2, u_2] = [l_1 \sqcap l_2, u_1 \sqcup u_2]$$

$$[l_1, u_1] \sqcap [l_2, u_2] = [l_1 \sqcup l_2, u_1 \sqcap u_2] \quad \text{sofern } (l_1 \sqcup l_2) \leq (u_1 \sqcap u_2)$$



Achtung:

- \mathbb{I} ist kein vollständiger Verband :-)
- \mathbb{I} besitzt **unendliche aufsteigende Ketten**, z.B.

$$[0, 0] \sqsubset [0, 1] \sqsubset [-1, 1] \sqsubset [-1, 2] \sqsubset \dots$$

Achtung:

- \mathbb{I} ist kein vollständiger Verband :-)
- \mathbb{I} besitzt **unendliche aufsteigende Ketten**, z.B.

$$[0, 0] \sqsubset [0, 1] \sqsubset [-1, 1] \sqsubset [-1, 2] \sqsubset \dots$$

Beschreibungsrelation:

$$z \Delta [l, u] \quad \text{gdw.} \quad l \leq z \leq u$$

Konkretisierung:

$$\gamma [l, u] = \{z \in \mathbb{Z} \mid l \leq z \leq u\}$$

Beispiel:

$$\begin{aligned}\gamma[0,7] &= \{0, \dots, 7\} \\ \gamma[0, \infty] &= \{0, 1, 2, \dots, \}\end{aligned}$$

Rechnen mit Intervallen: Intervall-Arithmetik :-)

Addition:

$$\begin{aligned}[l_1, u_1] +^\# [l_2, u_2] &= [l_1 + l_2, u_1 + u_2] && \text{wobei} \\ -\infty +_- &= -\infty \\ +\infty +_- &= +\infty \\ // &-\infty + \infty \text{ kommt nicht vor} && :-)\end{aligned}$$

Negation:

$$-\# [l, u] = [-u, -l]$$

Multiplikation:

$$\begin{aligned} [l_1, u_1] *^\# [l_2, u_2] &= [a, b] \quad \text{wobei} \\ a &= l_1 l_2 \sqcap l_1 u_2 \sqcap u_1 l_2 \sqcap u_1 u_2 \\ b &= l_1 l_2 \sqcup l_1 u_2 \sqcup u_1 l_2 \sqcup u_1 u_2 \end{aligned}$$

Beispiel:

$$\begin{aligned} [0, 2] *^\# [3, 4] &= [0, 8] \\ [-1, 2] *^\# [3, 4] &= [-4, 8] \\ [-1, 2] *^\# [-3, 4] &= [-6, 8] \\ [-1, 2] *^\# [-4, -3] &= [-8, 4] \end{aligned}$$

Division: $[l_1, u_1] /^\# [l_2, u_2] = [a, b]$

- Ist 0 **nicht** im Nenner-Intervall enthalten, sei:

$$a = l_1/l_2 \sqcap l_1/u_2 \sqcap u_1/l_2 \sqcap u_1/u_2$$

$$b = l_1/l_2 \sqcup l_1/u_2 \sqcup u_1/l_2 \sqcup u_1/u_2$$

- Gilt: $l_2 \leq 0 \leq u_2$, setzen wir:

$$[a, b] = [-\infty, +\infty]$$

Gleichheit:

$$[l_1, u_1] ==^\# [l_2, u_2] = \begin{cases} [1, 1] & \text{falls } l_1 = u_1 = l_2 = u_2 \\ [0, 0] & \text{falls } u_1 < l_2 \vee u_2 < l_1 \\ [0, 1] & \text{sonst} \end{cases}$$

Gleichheit:

$$[l_1, u_1] ==^\# [l_2, u_2] = \begin{cases} [1, 1] & \text{falls } l_1 = u_1 = l_2 = u_2 \\ [0, 0] & \text{falls } u_1 < l_2 \vee u_2 < l_1 \\ [0, 1] & \text{sonst} \end{cases}$$

Beispiel:

$$\begin{aligned} [42, 42] ==^\# [42, 42] &= [1, 1] \\ [0, 7] ==^\# [0, 7] &= [0, 1] \\ [1, 2] ==^\# [3, 4] &= [0, 0] \end{aligned}$$

Kleiner:

$$[l_1, u_1] <^\# [l_2, u_2] = \begin{cases} [1, 1] & \text{falls } u_1 < l_2 \\ [0, 0] & \text{falls } u_2 \leq l_1 \\ [0, 1] & \text{sonst} \end{cases}$$

Kleiner:

$$[l_1, u_1] <^\# [l_2, u_2] = \begin{cases} [1, 1] & \text{falls } u_1 < l_2 \\ [0, 0] & \text{falls } u_2 \leq l_1 \\ [0, 1] & \text{sonst} \end{cases}$$

Beispiel:

$$[1, 2] <^\# [9, 42] = [1, 1]$$

$$[0, 7] <^\# [0, 7] = [0, 1]$$

$$[3, 4] <^\# [1, 2] = [0, 0]$$

Mithilfe von \mathbb{I} konstruieren wir den vollständigen Verband:

$$\mathbb{D}_{\mathbb{I}} = (\text{Vars} \rightarrow \mathbb{I})_{\perp}$$

Beschreibungsrelation:

$$\rho \Delta D \quad \text{gdw.} \quad D \neq \perp \quad \wedge \quad \forall x \in \text{Vars} : (\rho x) \Delta (D x)$$

Die **abstrakte Ausdrucksauswertung** definieren wir analog Konstantenpropagation. Wir finden:

$$(\llbracket e \rrbracket \rho) \Delta (\llbracket e \rrbracket^{\#} D) \quad \text{sofern} \quad \rho \Delta D$$

Die Kanteneffekte:

$$\begin{aligned} \llbracket ; \rrbracket^\# D &= D \\ \llbracket x = e; \rrbracket^\# D &= D \oplus \{x \mapsto \llbracket e \rrbracket^\# D\} \\ \llbracket x = M[R]; \rrbracket^\# D &= D \oplus \{x \mapsto \top\} \\ \llbracket M[R_1] = R_2; \rrbracket^\# D &= D \\ \llbracket \text{Pos}(e) \rrbracket^\# D &= \begin{cases} \perp & \text{falls } [0, 0] = \llbracket e \rrbracket^\# D \\ D & \text{sonst} \end{cases} \\ \llbracket \text{Neg}(e) \rrbracket^\# D &= \begin{cases} D & \text{falls } [0, 0] \sqsubseteq \llbracket e \rrbracket^\# D \\ \perp & \text{sonst} \end{cases} \end{aligned}$$

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

Bessere Ausnutzung von Bedingungen:

$$\llbracket \text{Pos}(e) \rrbracket^\# D = \begin{cases} \perp & \text{falls } [0, 0] = \llbracket e \rrbracket^\# D \\ D_1 & \text{sonst} \end{cases}$$

wobei :

$$D_1 = \begin{cases} D \oplus \{x \mapsto (D x) \sqcap (\llbracket e_1 \rrbracket^\# D)\} & \text{falls } e \equiv x == e_1 \\ D \oplus \{x \mapsto (D x) \sqcap [-\infty, u]\} & \text{falls } e \equiv x \leq e_1, \llbracket e_1 \rrbracket^\# D = [-, u] \\ D \oplus \{x \mapsto (D x) \sqcap [l, \infty]\} & \text{falls } e \equiv x \geq e_1, \llbracket e_1 \rrbracket^\# D = [l, -] \end{cases}$$

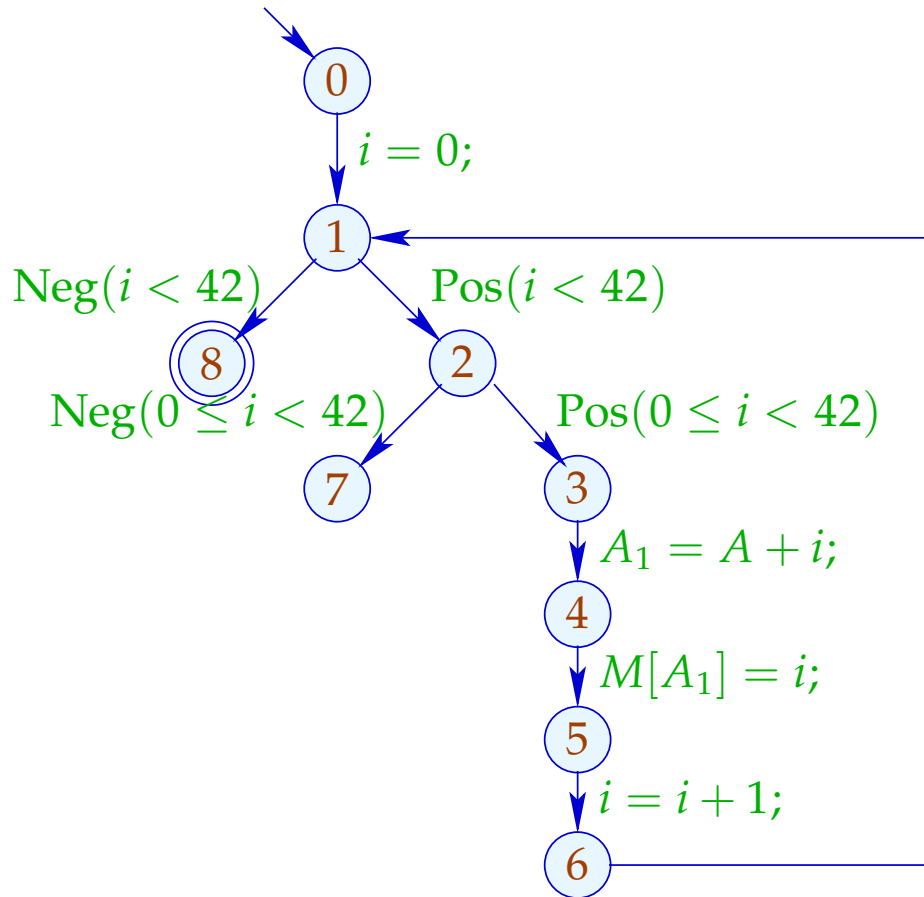
Bessere Ausnutzung von Bedingungen (Forts.):

$$\llbracket \text{Neg}(e) \rrbracket^\# D = \begin{cases} \perp & \text{falls } [0, 0] \not\subseteq \llbracket e \rrbracket^\# D \\ D_1 & \text{sonst} \end{cases}$$

wobei :

$$D_1 = \begin{cases} D \oplus \{x \mapsto (D x) \sqcap (\llbracket e_1 \rrbracket^\# D)\} & \text{falls } e \equiv x \neq e_1 \\ D \oplus \{x \mapsto (D x) \sqcap [-\infty, u]\} & \text{falls } e \equiv x > e_1, \llbracket e_1 \rrbracket^\# D = [-, u] \\ D \oplus \{x \mapsto (D x) \sqcap [l, \infty]\} & \text{falls } e \equiv x < e_1, \llbracket e_1 \rrbracket^\# D = [l, -] \end{cases}$$

Beispiel:



	<i>i</i>	
	<i>l</i>	<i>u</i>
0	$-\infty$	$+\infty$
1	0	42
2	0	41
3	0	41
4	0	41
5	0	41
6	1	42
7	\perp	
8	42	42

Problem:

- Die Lösung lässt sich mit RR-Iteration berechnen — nach ca. 42 Runden :-)
- Auf manchen Programmen terminiert die Iteration nie :-((

Idee 1: Widening

- Iteriere beschleunigt — unter Preisgabe von Präzision :-)
- Erlaube nur beschränkt oft die Modifikation eines Werts !!!

... im Beispiel:

- verbiete Updates von Intervall-Grenzen in $\mathbb{Z} \dots$

⇒ eine maximale Kette:

$$[3, 17] \sqsubset [3, +\infty] \sqsubset [-\infty, +\infty]$$

Formalisierung dieses Vorgehens:

$$\text{Sei } x_i \sqsupseteq f_i(x_1, \dots, x_n), \quad i = 1, \dots, n \quad (1)$$

ein Ungleichungssystem über \mathbb{D} , wobei die f_i nicht notwendigerweise monoton sind.

Trotzdem können wir eine **akkumulierende** Iteration definieren.

Betrachte das Gleichungssystem:

$$x_i = x_i \sqcup f_i(x_1, \dots, x_n), \quad i = 1, \dots, n \quad (2)$$

Offenbar gilt:

(a) \underline{x} ist Lösung von (1) gdw. \underline{x} Lösung von (2) ist.

(b) Die Funktion $G : \mathbb{D}^n \rightarrow \mathbb{D}^n$ mit

$$G(x_1, \dots, x_n) = (y_1, \dots, y_n), \quad y_i = x_i \sqcup f_i(x_1, \dots, x_n)$$

ist **vergrößernd**, d.h. $\underline{x} \sqsubseteq G \underline{x}$ für alle $\underline{x} \in \mathbb{D}^n$.

(c) Die Folge $G^k \underline{x}$, $k \geq 0$, ist eine aufsteigende Kette:

$$\underline{x} \sqsubseteq G \underline{x} \sqsubseteq \dots \sqsubseteq G^k \underline{x} \sqsubseteq \dots$$

(d) Gilt $G^k \underline{x} = G^{k+1} \underline{x} = \underline{y}$ ist \underline{y} eine Lösung von (1).

(e) Hat \mathbb{D} unendliche aufsteigende Ketten, ist uns mit (d) noch nicht viel gedient ...

aber: wir könnten statt Gleichungssystem (2) ein Gleichungssystem:

$$x_i = x_i \sqcup f_i(x_1, \dots, x_n), \quad i = 1, \dots, n \quad (3)$$

betrachten für eine binäre Operation **Widening**:

$$\sqcup : \mathbb{D}^2 \rightarrow \mathbb{D} \quad \text{mit} \quad v_1 \sqcup v_2 \sqsubseteq v_1 \sqcup v_2$$

Dann berechnet (RR)-Iteration für (3) immer noch eine Lösung von (1) :-)

... für die Intervall-Analyse:

- Der vollständige Verband ist: $\mathbb{D}_{\mathbb{I}} = (\text{Vars} \rightarrow \mathbb{I})_{\perp}$
- Das Widening \sqcup definieren wir als:

$$\perp \sqcup D = D \sqcup \perp = D \quad \text{und für } D_1 \neq \perp \neq D_2:$$

$$(D_1 \sqcup D_2) \mathbf{x} = (D_1 \mathbf{x}) \sqcup (D_2 \mathbf{x}) \quad \text{wobei}$$

$$[l_1, u_1] \sqcup [l_2, u_2] = [l, u] \quad \text{mit}$$

$$l = \begin{cases} l_1 & \text{falls } l_1 \leq l_2 \\ -\infty & \text{sonst} \end{cases}$$
$$u = \begin{cases} u_1 & \text{falls } u_1 \geq u_2 \\ +\infty & \text{sonst} \end{cases}$$

$\implies \sqcup$ ist nicht kommutativ !!!

Beispiel:

$$[0, 2] \sqcup [1, 2] = [0, 2]$$

$$[1, 2] \sqcup [0, 2] = [-\infty, 2]$$

$$[1, 5] \sqcup [3, 7] = [1, +\infty]$$

- Widening liefert **schneller** größere Werte.
- Es sollte so gewählt werden, dass es die Terminierung der Iteration garantiert :-)
- Bei Intervall-Analyse begrenzt es die Anzahl der Iterationen auf:

$$\#Punkte \cdot (1 + 2 \cdot \#Vars)$$

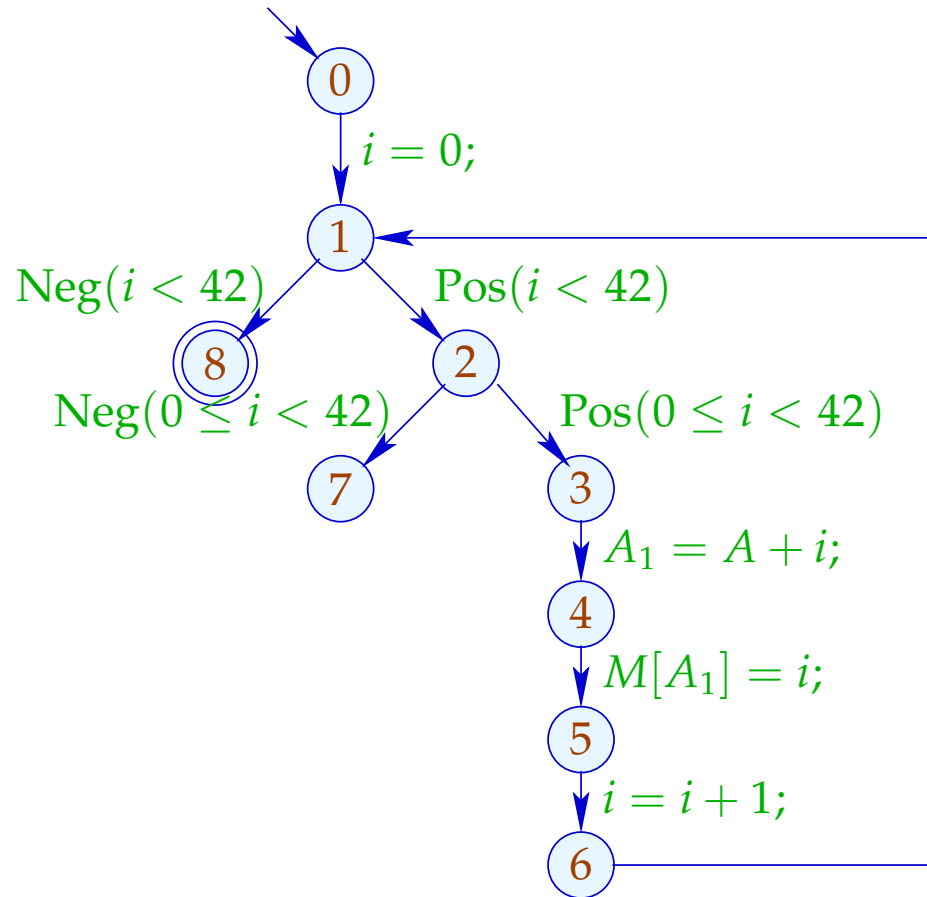
Fazit:

- Um eine Lösung von (1) über einem vollständigen Verband mit unendlichen aufsteigenden Ketten zu bestimmen, definieren wir ein geeignetes Widening und lösen dann (3) :-)
- **Achtung:** Die Konstruktion geeigneter Widenings ist eine schwarze Kunst !!!

Oft wählt man \sqsubseteq ganz pragmatisch **dynamisch** während der Iteration, so dass

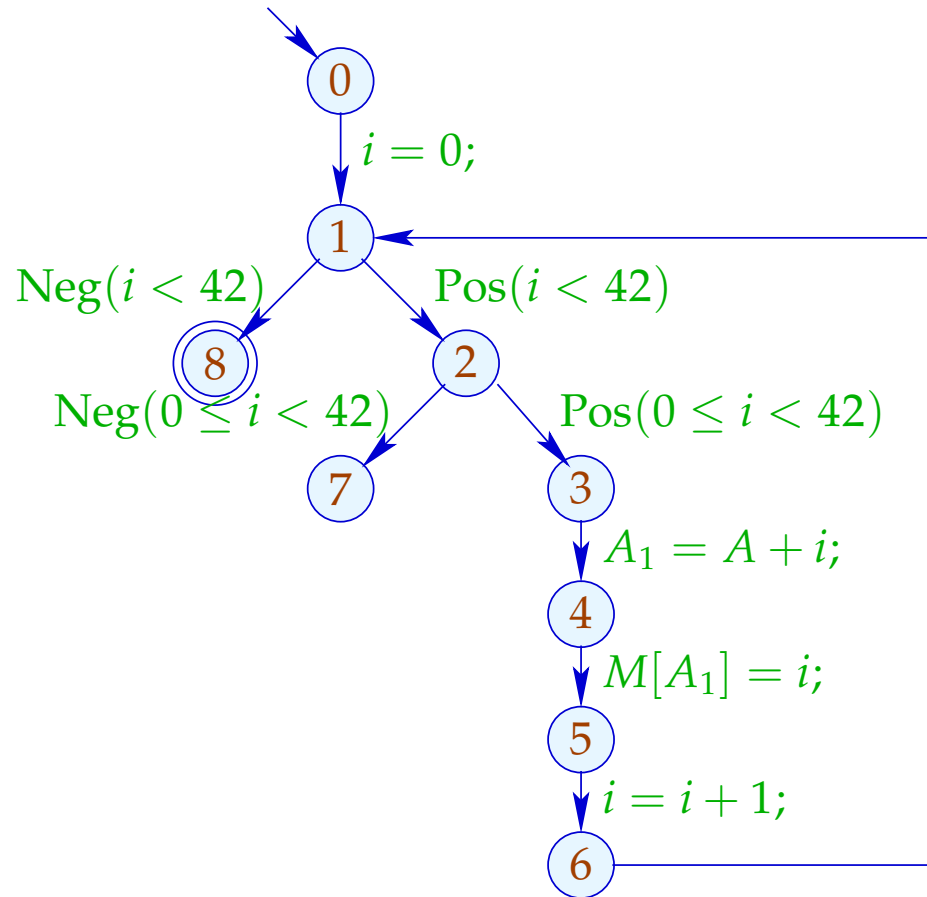
- die abstrakten Werte nicht zu **kompliziert** werden;
- die Anzahl der Updates fest beschränkt bleibt ...

Unser Beispiel:



	1	
	l	u
0	$-\infty$	$+\infty$
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	1	1
7	\perp	
8	\perp	

Unser Beispiel:



	1		2		3	
	<i>l</i>	<i>u</i>	<i>l</i>	<i>u</i>	<i>l</i>	<i>u</i>
0	$-\infty$	$+\infty$	$-\infty$	$+\infty$		
1	0	0	0	$+\infty$		
2	0	0	0	$+\infty$		
3	0	0	0	$+\infty$		
4	0	0	0	$+\infty$	dito	
5	0	0	0	$+\infty$		
6	1	1	1	$+\infty$		
7		\perp	42	$+\infty$		
8		\perp	42	$+\infty$		

... offenbar ist das Ergebnis enttäuschend :-)

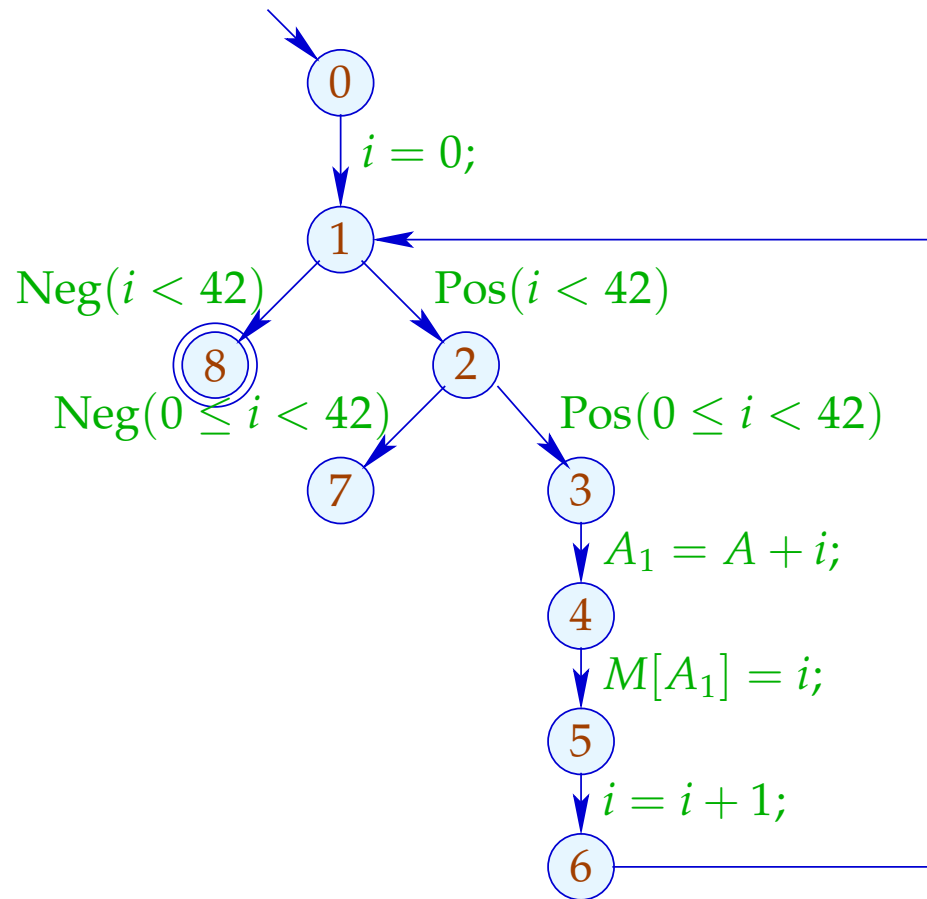
Idee 2:

Eigentlich reicht es, die Beschleunigung mittels \sqsubseteq nur an **genügend vielen** Stellen anzuwenden!

Eine Menge I heißt **Loop Separator** (Kreis-Trenner), falls jeder Kreis mindestens einen Punkt aus I enthält :-)

Wenden wir Widening nur an den Punkten aus einer solchen Menge I , terminiert RR-Iteration immer noch !!!

In unserem Beispiel:

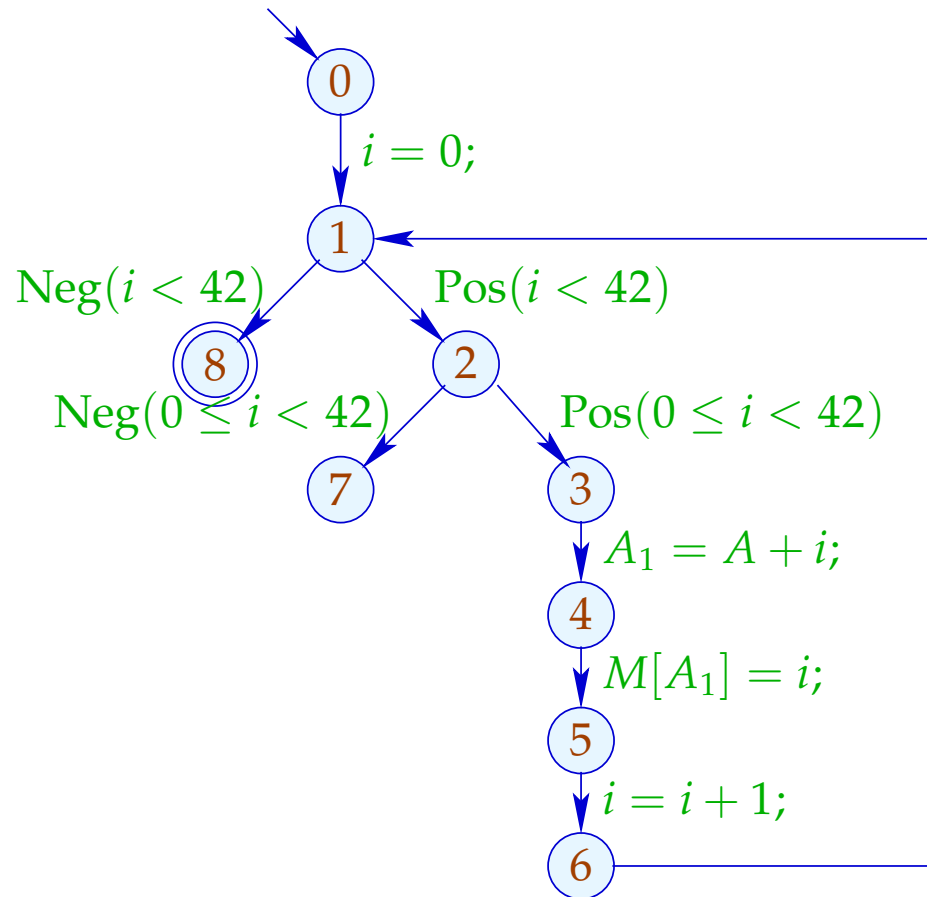


$I_1 = \{1\}$ oder auch:

$I_2 = \{2\}$ oder auch:

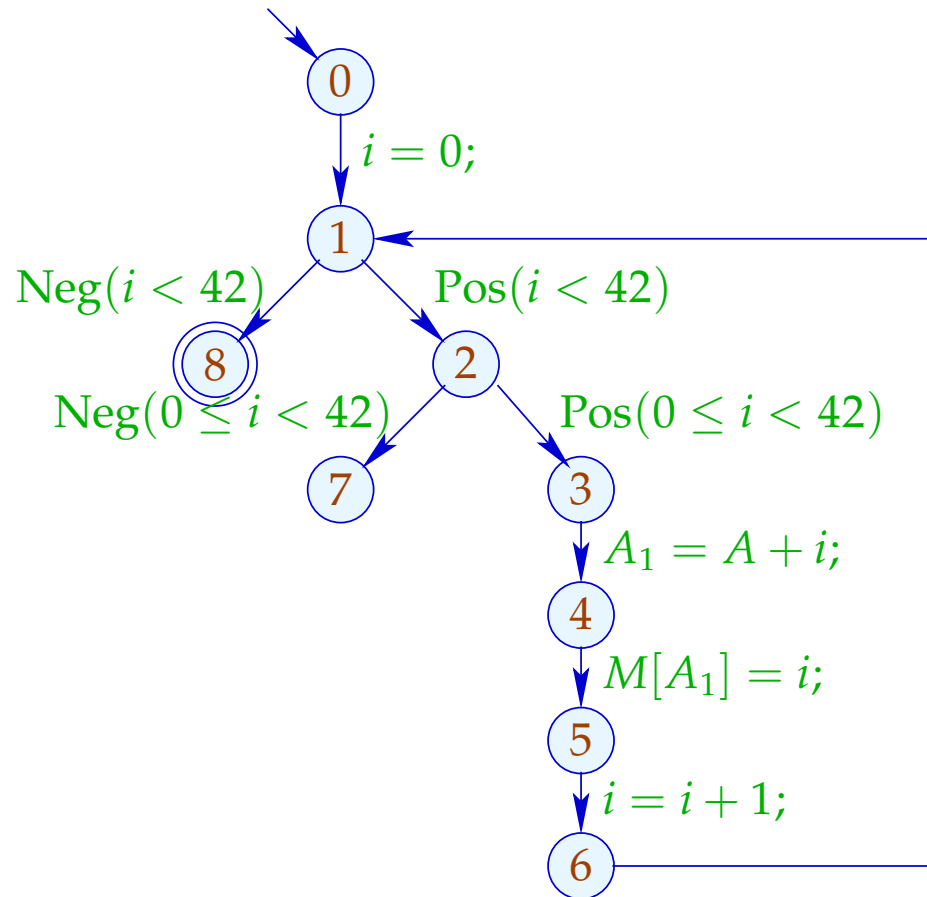
$I_3 = \{3\}$

Die Analyse mit $I = \{1\}$:



	1		2		3	
	<i>l</i>	<i>u</i>	<i>l</i>	<i>u</i>	<i>l</i>	<i>u</i>
0	$-\infty$	$+\infty$	$-\infty$	$+\infty$		
1	0	0	0	$+\infty$		
2	0	0	0	41		
3	0	0	0	41		
4	0	0	0	41	dito	
5	0	0	0	41		
6	1	1	1	42		
7	\perp		\perp			
8	\perp		42	$+\infty$		

Die Analyse mit $I = \{2\}$:



	1		2		3	
	l	u	l	u	l	u
0	$-\infty$	$+\infty$	$-\infty$	$+\infty$		
1	0	0	0	42		
2	0	0	0	$+\infty$		
3	0	0	0	41		
4	0	0	0	41	dito	
5	0	0	0	41		
6	1	1	1	42		
7		\perp	42	$+\infty$		
8		\perp	42	42		

Diskussion:

- Beide Analysen-Läufe berechnen interessante Informationen :-)
- Der Lauf mit $I = \{2\}$ belegt, dass nach Verlassen der Schleife stets $i = 42$ gilt.
- Nur der Lauf mit $I = \{1\}$ belegt aber, dass der äußere Test den inneren überflüssig macht :-)

Wie findet man einen geeigneten Loop Separator $I ???$