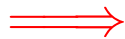


## 1.5 Interval Analysis

### Observation:

- Programmers often use global constants for switching debugging code on/off.



Constant propagation is useful :-)

- In general, precise values of variables will be unknown — perhaps, however, a tight **interval !!!**

## Example:

```
for ( $i = 0; i < 42; i++$ )  
    if ( $0 \leq i \wedge i < 42$ ) {  
         $A_1 = A + i;$   
         $M[A_1] = i;$   
    }  
// A start address of an array  
// if the array-bound check
```

Obviously, the inner check is superfluous :-)

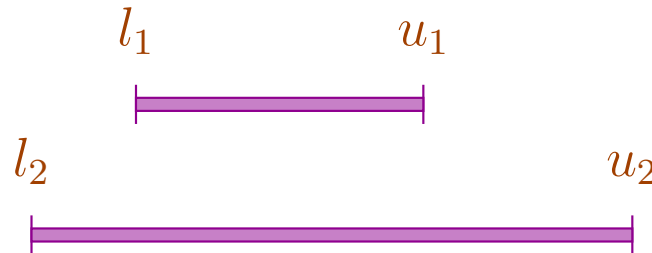
## Idea 1:

Determine for every variable  $x$  an (as tight as possible :-) interval of possible values:

$$\mathbb{I} = \{[l, u] \mid l \in \mathbb{Z} \cup \{-\infty\}, u \in \mathbb{Z} \cup \{+\infty\}, l \leq u\}$$

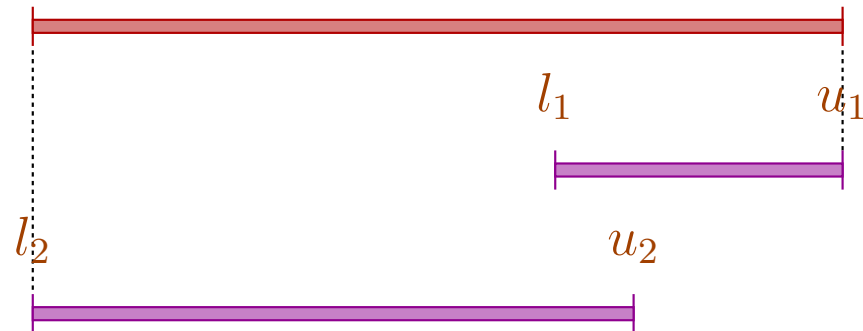
## Partial Ordering:

$$[l_1, u_1] \sqsubseteq [l_2, u_2] \quad \text{iff} \quad l_2 \leq l_1 \wedge u_1 \leq u_2$$



Thus:

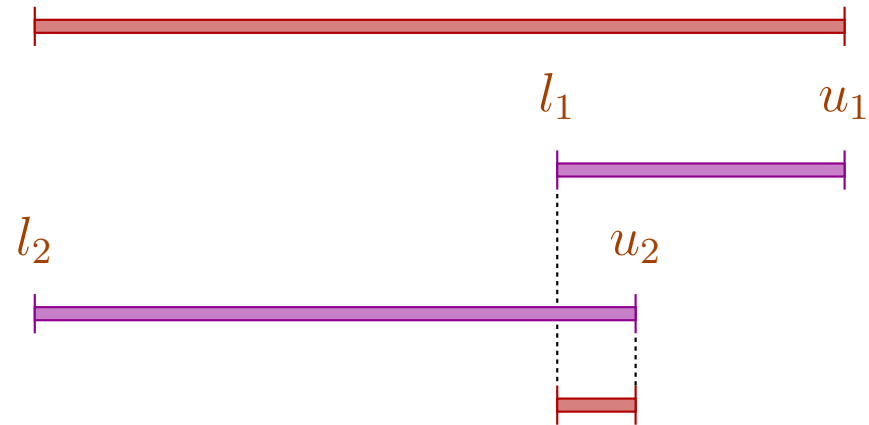
$$[l_1, u_1] \sqcup [l_2, u_2] = [l_1 \sqcap l_2, u_1 \sqcup u_2]$$



Thus:

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

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



## Caveat:

→  $\mathbb{I}$  is not a complete lattice :-)

→  $\mathbb{I}$  has infinite ascending chains, e.g.,

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

## Caveat:

→  $\mathbb{I}$  is not a complete lattice :-)

→  $\mathbb{I}$  has infinite ascending chains, e.g.,

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

## Description Relation:

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

## Concretization:

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

Example:

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

Computing with intervals:

Interval Arithmetic :-)

Addition:

$$[l_1, u_1] +^{\#} [l_2, u_2] = [l_1 + l_2, u_1 + u_2] \quad \text{where}$$

$$-\infty + _ = -\infty$$

$$+\infty + _ = +\infty$$

//  $-\infty + \infty$  cannot occur :-)

Negation:

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

Multiplication:

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

Example:

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

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

- If 0 is **not** contained in the interval of the denominator, then:

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

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

- If:  $l_2 \leq 0 \leq u_2$ , we define:

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

Equality:

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

Equality:

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

Example:

$$[42, 42] ==^\# [42, 42] = [1, 1]$$

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

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

Less:

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

Less:

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

Example:

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

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

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

By means of  $\mathbb{I}$  we construct the complete lattice:

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

Description Relation:

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

The **abstract evaluation** of expressions is defined analogously to constant propagation. We have:

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

## The Effects of Edges:

$$\begin{aligned}
 \llbracket ; \rrbracket^\# D &= D \\
 \llbracket x = e; \rrbracket^\# D &= D \oplus \{x \mapsto \llbracket e \rrbracket^\# D\} \\
 \llbracket x = M[e]; \rrbracket^\# D &= D \oplus \{x \mapsto \top\} \\
 \llbracket M[e_1] = e_2; \rrbracket^\# D &= D \\
 \llbracket \text{Pos}(e) \rrbracket^\# D &= \begin{cases} \perp & \text{if } [0, 0] = \llbracket e \rrbracket^\# D \\ D & \text{otherwise} \end{cases} \\
 \llbracket \text{Neg}(e) \rrbracket^\# D &= \begin{cases} D & \text{if } [0, 0] \sqsubseteq \llbracket e \rrbracket^\# D \\ \perp & \text{otherwise} \end{cases}
 \end{aligned}$$

... given that  $D \neq \perp$  :-)

## Better Exploitation of Conditions:

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

where :

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

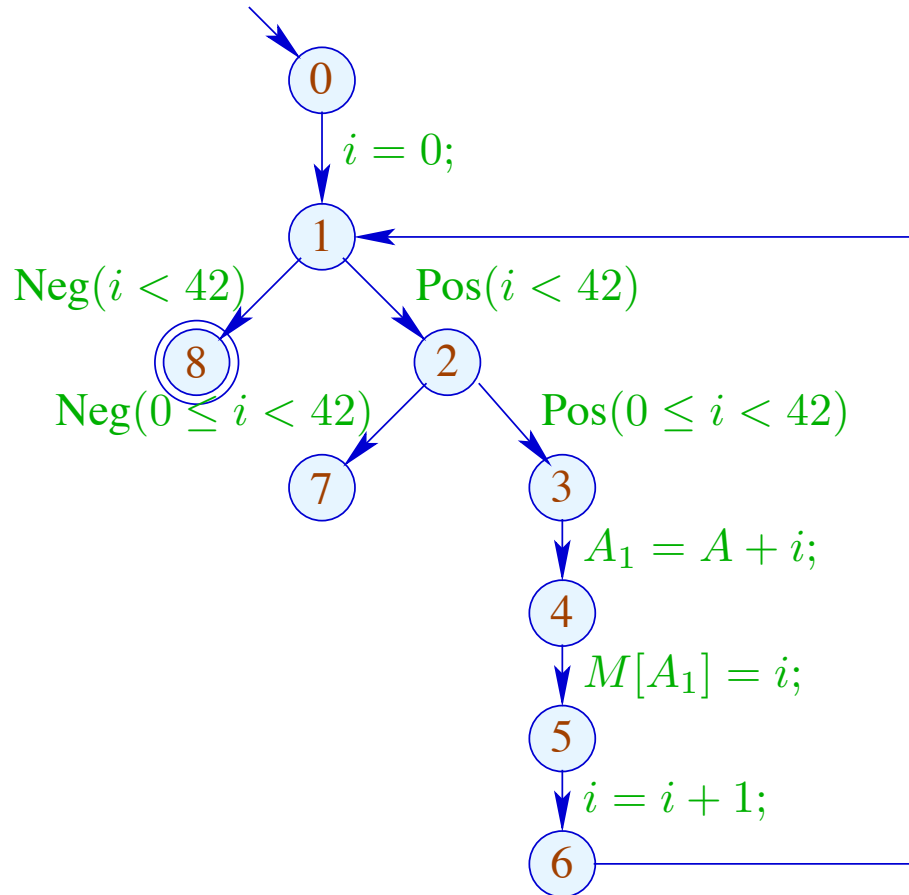
## Better Exploitation of Conditions (cont.):

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

where :

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

# Example:



	$i$	
	$l$	$u$
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:

- The solution can be computed with RR-iteration — after about 42 rounds :-)
- On some programs, iteration may **never** terminate :-((

## Idea 1: Widening

- Accelerate the iteration — at the **prize of imprecision** :-)
- Allow only a bounded number of modifications of values !!!

... in the Example:

- dis-allow updates of interval bounds in  $\mathbb{Z}$  ...

⇒ a maximal chain:

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

## Formalization of the Approach:

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

denote a system of constraints over  $\mathbb{D}$  where the  $f_i$  are **not necessarily** monotonic.

Nonetheless, an **accumulating** iteration can be defined. Consider the system of equations:

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

We obviously have:

(a)  $\underline{x}$  is a solution of (1) iff  $\underline{x}$  is a solution of (2).

(b) The function  $G : \mathbb{D}^n \rightarrow \mathbb{D}^n$  with

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

is **increasing**, i.e.,  $\underline{x} \sqsubseteq G \underline{x}$  for all  $\underline{x} \in \mathbb{D}^n$ .

(c) The sequence  $G^k \underline{\perp}$ ,  $k \geq 0$ , is an ascending chain:

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

(d) If  $G^k \underline{\perp} = G^{k+1} \underline{\perp} = \underline{y}$ , then  $\underline{y}$  is a solution of (1).

(e) If  $\mathbb{D}$  has infinite strictly ascending chains, then (d) is not yet sufficient ...

**but:** we could consider the modified system of equations:

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

for a binary operation **widening**:

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

(RR)-iteration for (3) still will compute a solution of (1) :-)

## ... for Interval Analysis:

- The complete lattice is:  $\mathbb{D}_{\mathbb{I}} = (\text{Vars} \rightarrow \mathbb{I})_{\perp}$
- the widening  $\sqcup$  is defined by:

$$\perp \sqcup D = D \sqcup \perp = D \quad \text{and for } D_1 \neq \perp \neq D_2:$$

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

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

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

$\implies \sqcup$  is **not commutative** !!!

## Example:

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

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

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

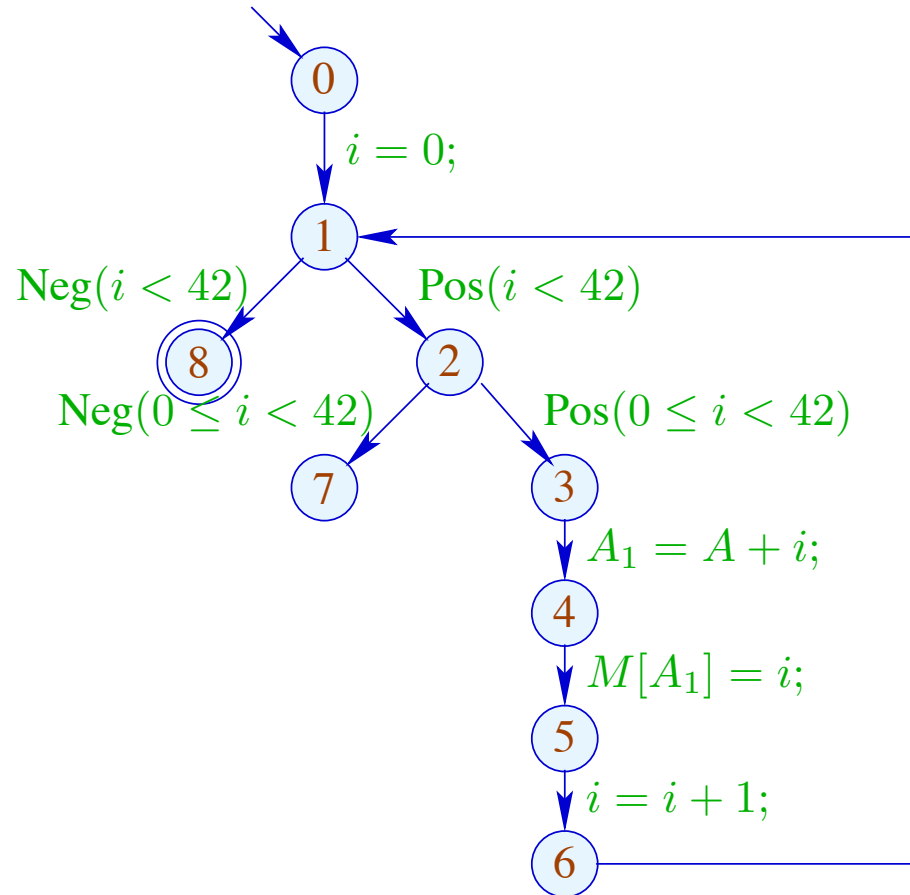
- Widening returns larger values **more quickly**.
- It should be constructed in such a way that termination of iteration is guaranteed :-)
- For interval analysis, widening bounds the number of iterations by:

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

## Conclusion:

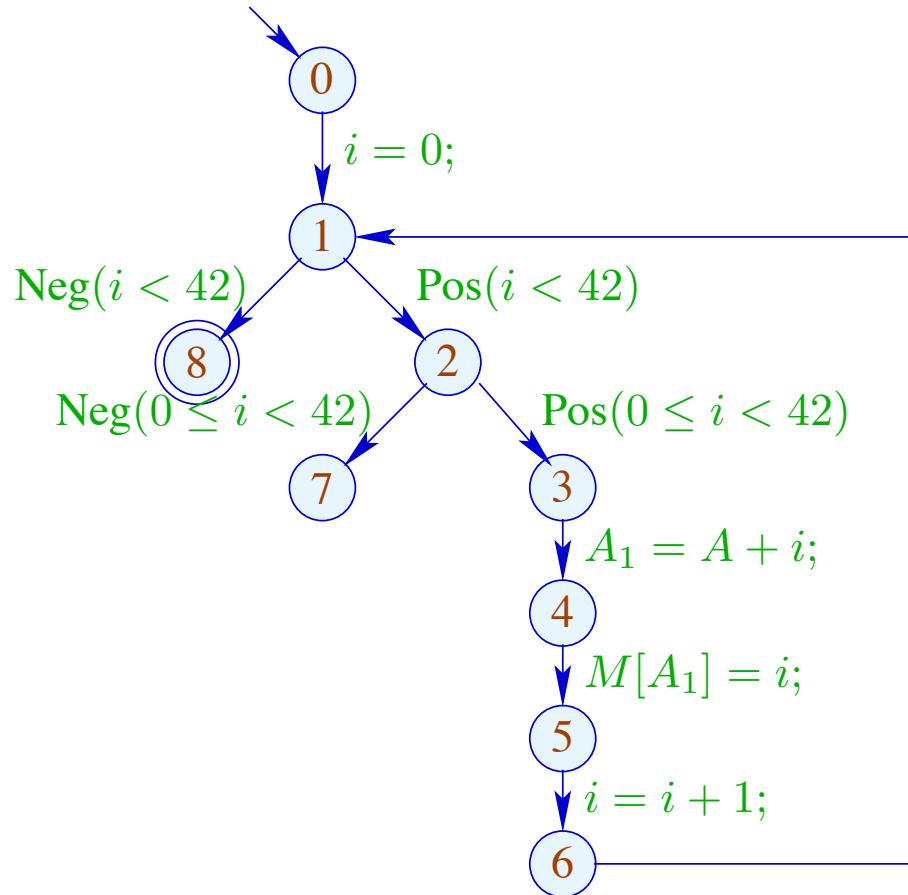
- In order to determine a solution of (1) over a complete lattice with infinite ascending chains, we define a suitable widening and then solve (3) :-)
- **Caveat:** The construction of suitable widenings is a **dark art !!!**  
Often  $\sqcup$  is chosen **dynamically** during iteration such that
  - the abstract values do not get too **complicated**;
  - the number of updates remains bounded ...

## Our Example:



	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	$\perp$	

## Our Example:



	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	$+\infty$		

... obviously, the result is disappointing :-)

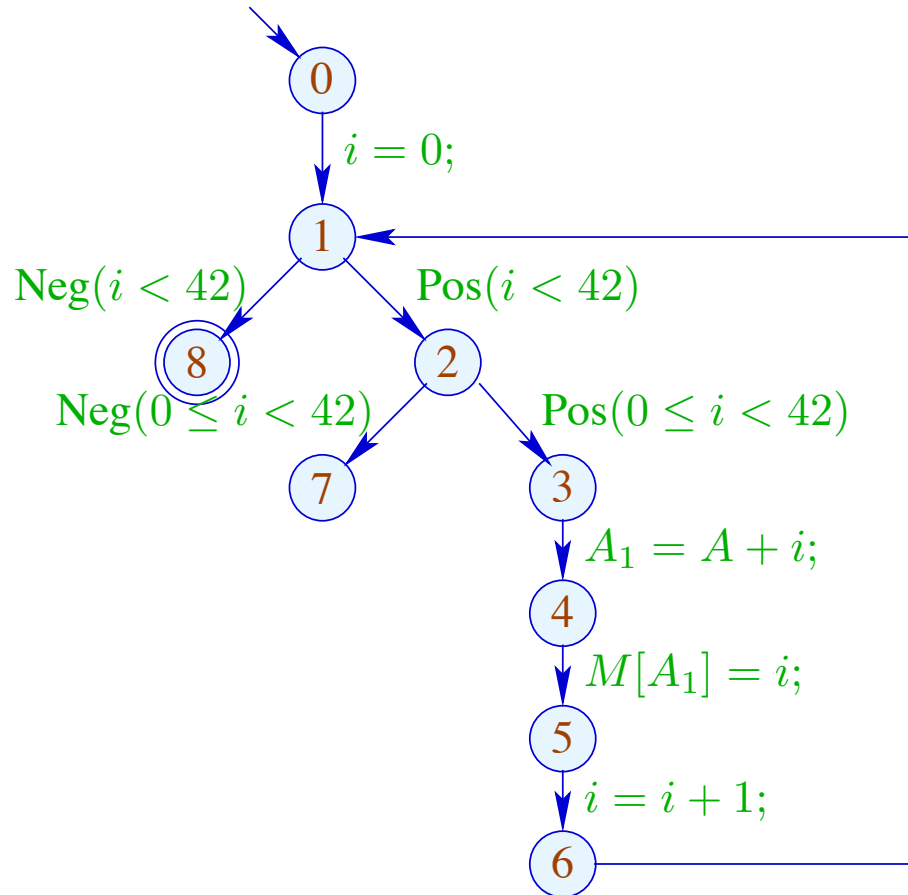
## Idea 2:

In fact, acceleration with  $\sqsubseteq$  need only be applied at **sufficiently many** places!

A set  $I$  is a **loop separator**, if every loop contains at least one point from  $I$  :-)

If we apply widening only at program points from such a set  $I$ , then RR-iteration still terminates !!!

In our Example:

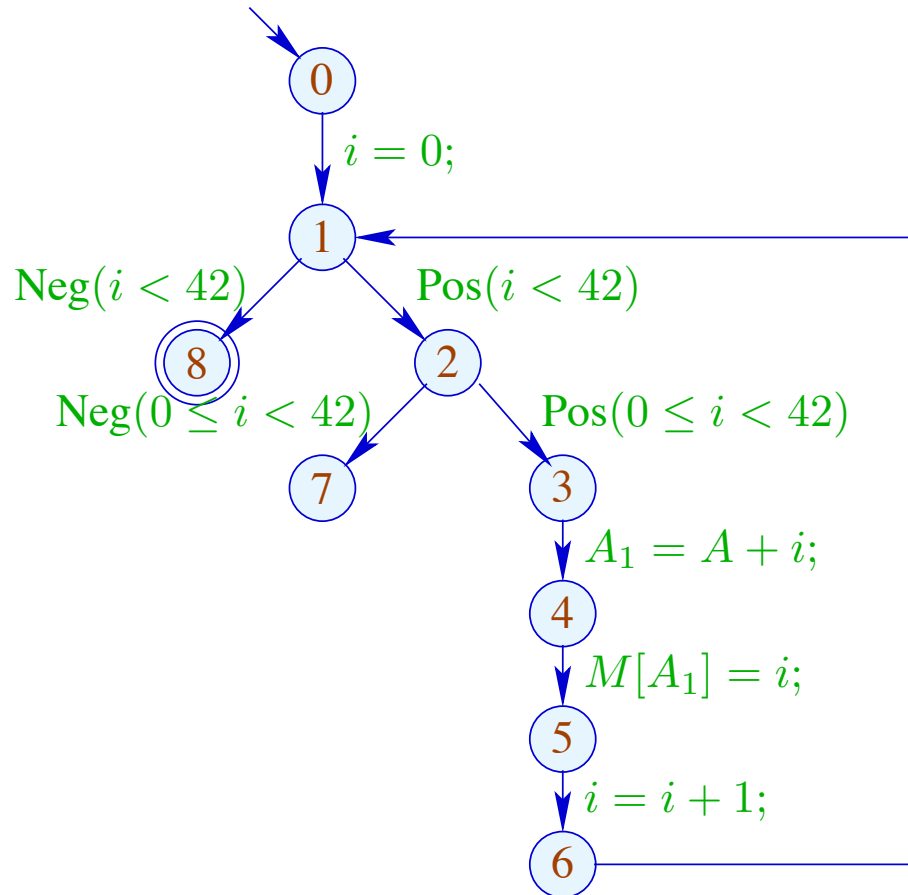


$I_1 = \{1\}$  or:

$I_2 = \{2\}$  or:

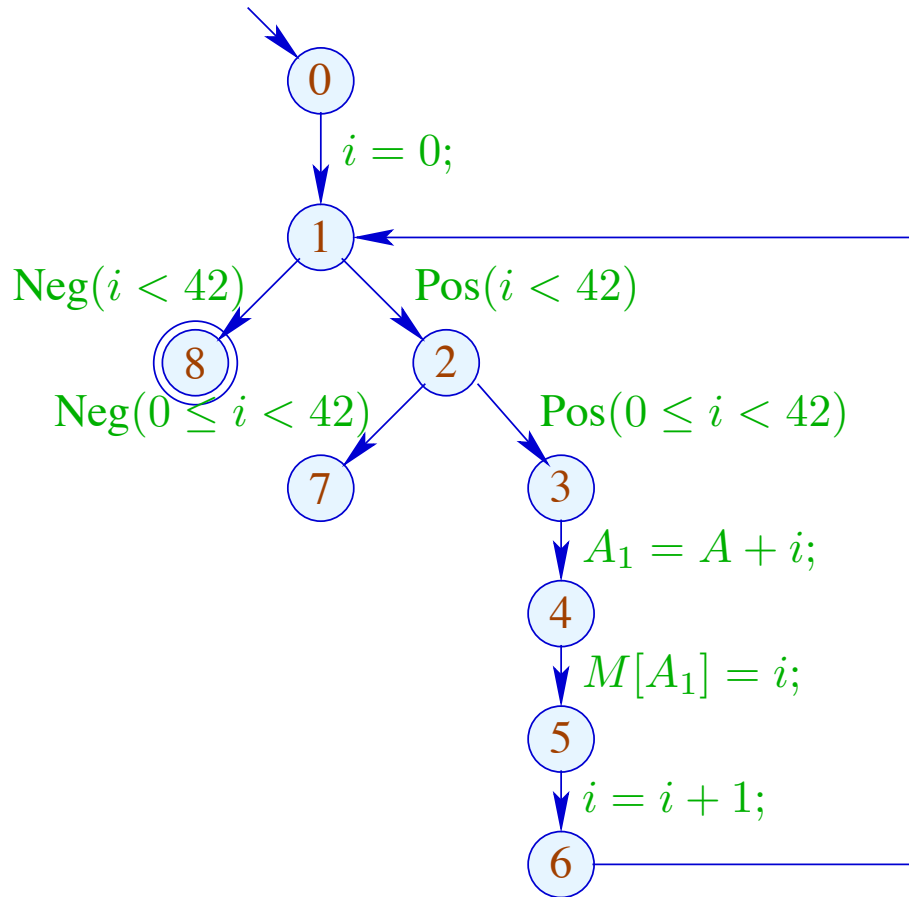
$I_3 = \{3\}$

The Analysis with  $I = \{1\}$  :



	1		2		3	
	$l$	$u$	$l$	$u$	$l$	$u$
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	$+\infty$		

# The Analysis with $I = \{2\}$ :



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

## Discussion:

- Both runs of the analysis determine interesting information :-)
- The run with  $I = \{2\}$  proves that always  $i = 42$  after leaving the loop.
- Only the run with  $I = \{1\}$  finds, however, that the outer check makes the inner check superfluous :-)

How can we find a suitable loop separator  $I$  ???

### Idea 3: Narrowing

Let  $\underline{x}$  denote any solution of (1), i.e.,

$$x_i \supseteq f_i \underline{x}, \quad i = 1, \dots, n$$

Then for monotonic  $f_i$ ,

$$\underline{x} \supseteq F \underline{x} \supseteq F^2 \underline{x} \supseteq \dots \supseteq F^k \underline{x} \supseteq \dots$$

// Narrowing Iteration

### Idea 3: Narrowing

Let  $\underline{x}$  denote any solution of (1), i.e.,

$$x_i \sqsupseteq f_i \underline{x}, \quad i = 1, \dots, n$$

Then for monotonic  $f_i$ ,

$$\underline{x} \sqsupseteq F \underline{x} \sqsupseteq F^2 \underline{x} \sqsupseteq \dots \sqsupseteq F^k \underline{x} \sqsupseteq \dots$$

// Narrowing Iteration

Every tuple  $F^k \underline{x}$  is a solution of (1) :-)

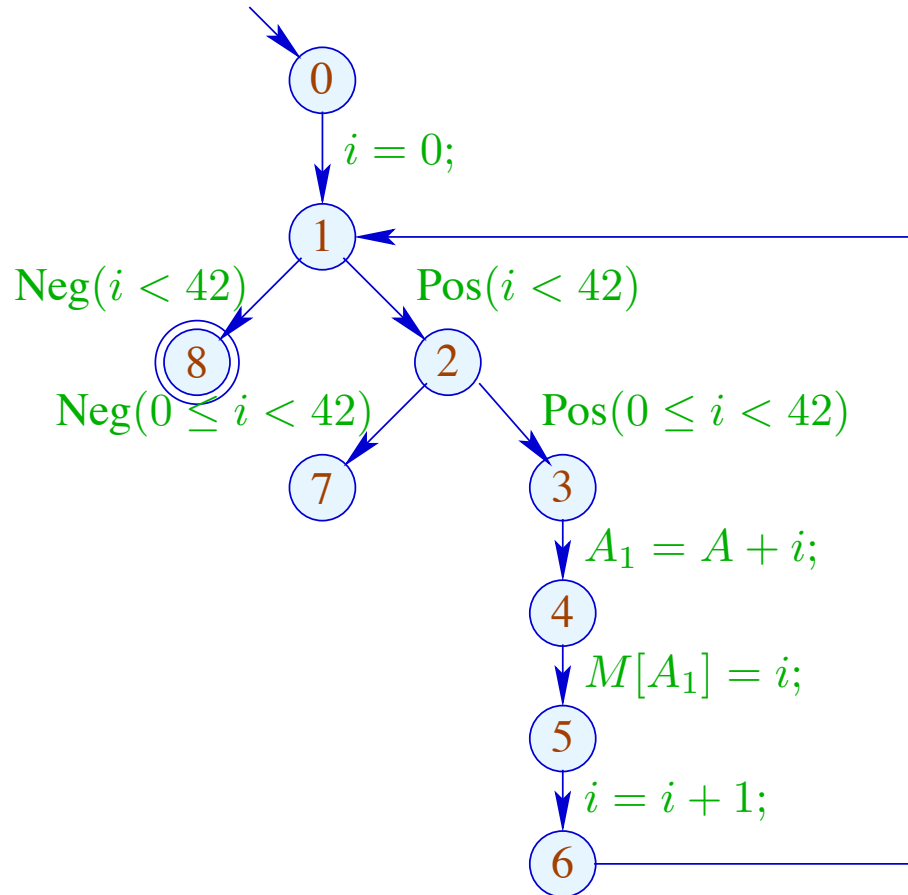


Termination is no problem anymore:

we stop whenever we want :-))

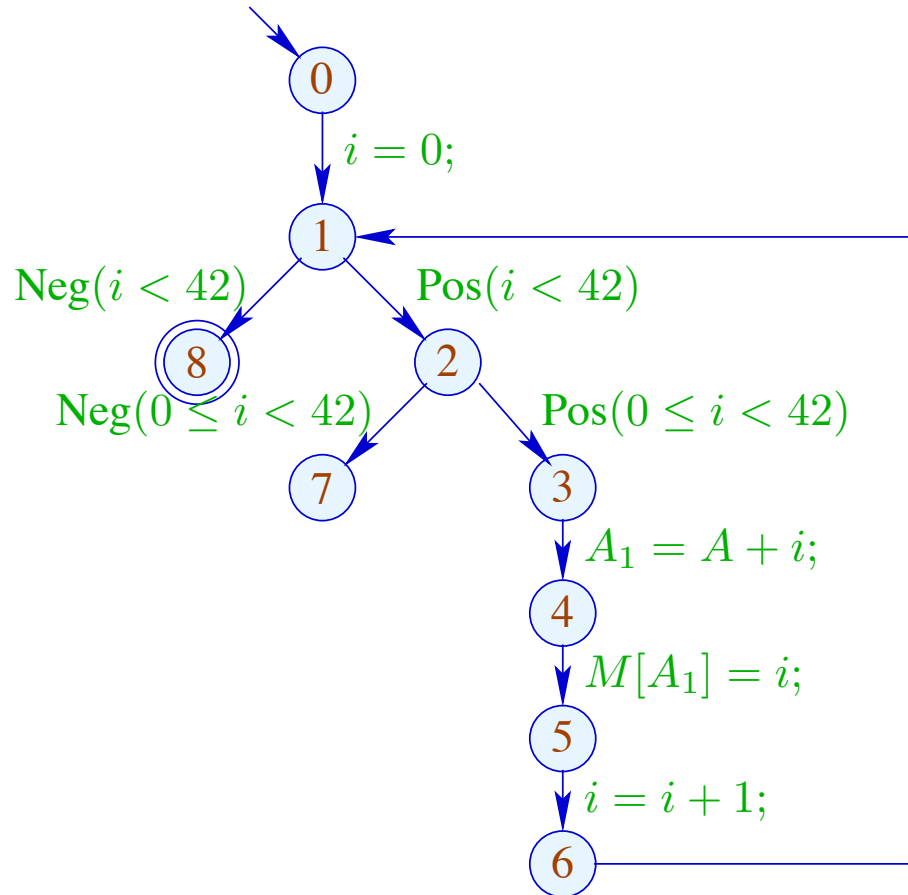
// The same also holds for RR-iteration.

## Narrowing Iteration in the Example:



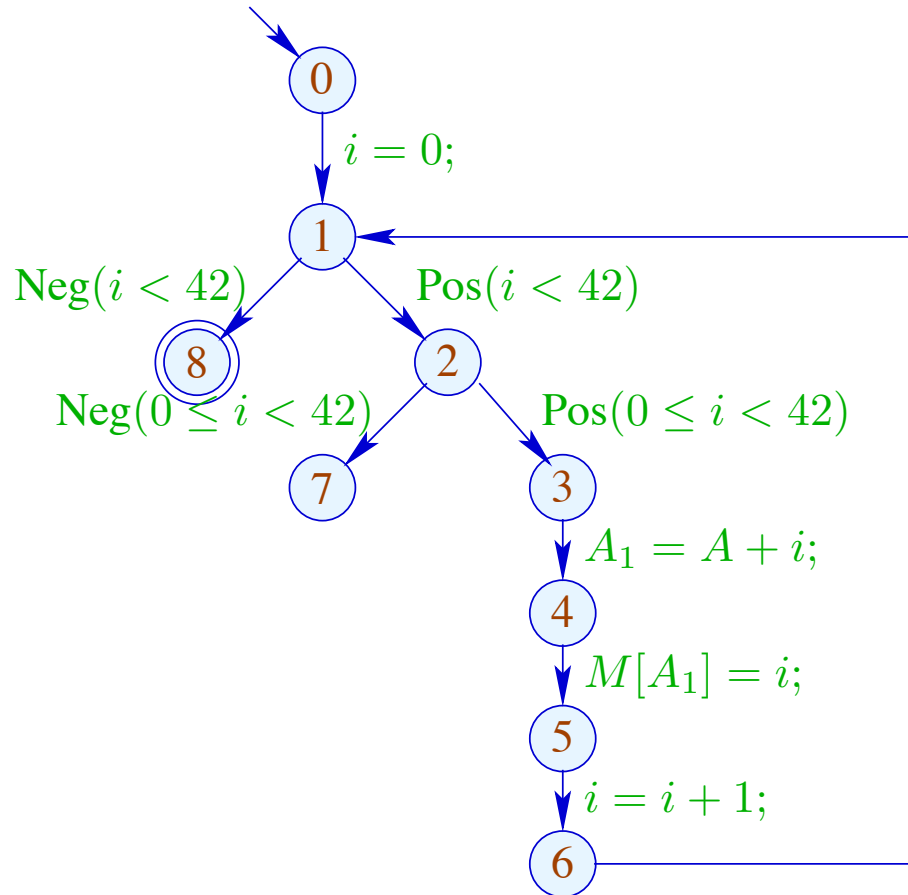
	0	
	$l$	$u$
0	$-\infty$	$+\infty$
1	0	$+\infty$
2	0	$+\infty$
3	0	$+\infty$
4	0	$+\infty$
5	0	$+\infty$
6	1	$+\infty$
7	42	$+\infty$
8	42	$+\infty$

## Narrowing Iteration in the Example:



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

## Narrowing Iteration in the Example:



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

## Discussion:

- We start with a safe approximation.
- We find that the inner check is redundant :-)
- We find that at exit from the loop, always  $i = 42$  :-))
- It was not necessary to construct an optimal loop separator :-)))

## Last Question:

Do we have to accept that narrowing may not terminate ???

## 4. Idea: Accelerated Narrowing

Assume that we have a solution  $\underline{x} = (x_1, \dots, x_n)$  of the system of constraints:

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

Then consider the system of equations:

$$x_i = x_i \sqcap f_i(x_1, \dots, x_n), \quad i = 1, \dots, n \quad (4)$$

Obviously, we have for monotonic  $f_i : H^k \underline{x} = F^k \underline{x} \quad :-)$

where  $H(x_1, \dots, x_n) = (y_1, \dots, y_n), \quad y_i = x_i \sqcap f_i(x_1, \dots, x_n)$ .

In (4), we replace  $\sqcap$  durch by the novel operator  $\sqbar{\cap}$  where:

$$a_1 \sqcap a_2 \sqsubseteq a_1 \sqbar{\cap} a_2 \sqsubseteq a_1$$

... for Interval Analysis:

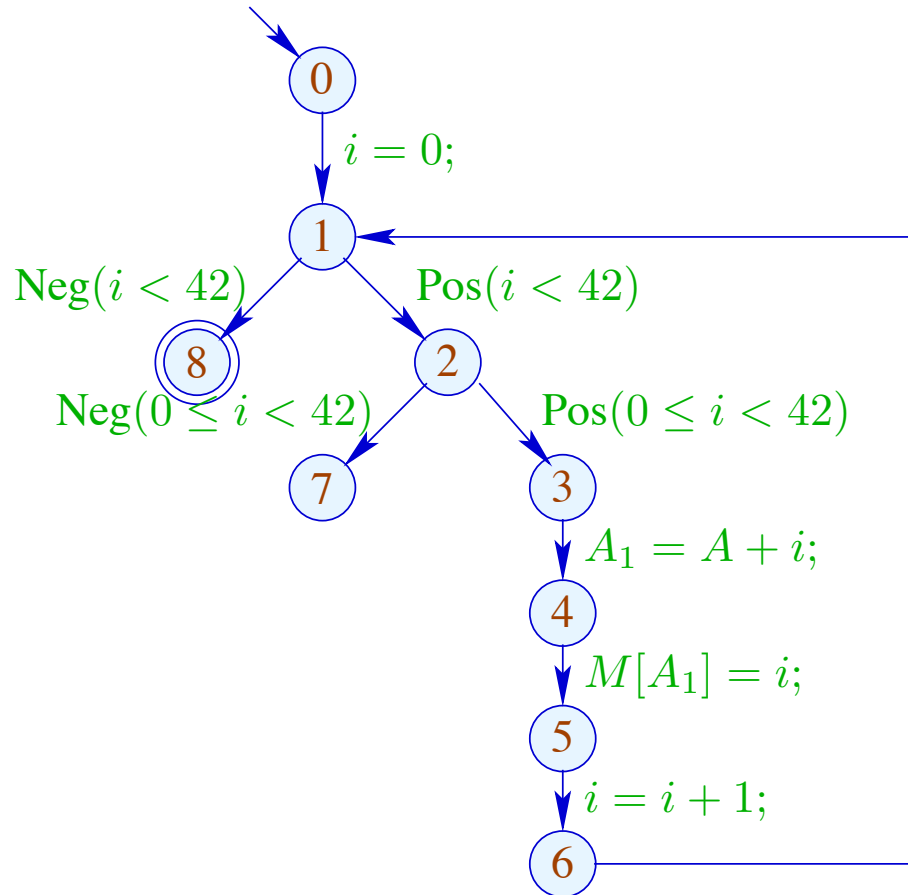
We preserve finite interval bounds :-)

Therefore,  $\perp \sqcap D = D \sqcap \perp = \perp$  and for  $D_1 \neq \perp \neq D_2$ :

$$(D_1 \sqcap D_2) x = (D_1 x) \sqcap (D_2 x) \quad \text{where}$$
$$[l_1, u_1] \sqcap [l_2, u_2] = [l, u] \quad \text{with}$$
$$l = \begin{cases} l_2 & \text{if } l_1 = -\infty \\ l_1 & \text{otherwise} \end{cases}$$
$$u = \begin{cases} u_2 & \text{if } u_1 = \infty \\ u_1 & \text{otherwise} \end{cases}$$

$\implies \sqcap$  is not commutative !!!

## Accelerated Narrowing in the Example:



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

## Discussion:

- **Caveat:** Widening also returns for non-monotonic  $f_i$  a solution. Narrowing is only applicable to monotonic  $f_i$  !!
- In the example, accelerated narrowing already returns the optimal result :-)
- If the operator  $\sqcap$  only allows for finitely many improvements of values, we may execute narrowing until stabilization.
- In case of interval analysis these are at most:

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