Script generated by TTT

Title: Seidl: Programmoptimierung (19.11.2015)

Date: Thu Nov 19 08:38:36 CET 2015

Duration: 84:34 min

Pages: 40

Less:

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

$x_1 < x_2$

Example

$$\begin{array}{rcl} [1,2] <^\sharp [9,42] & = & [1,1] \\ [0,7] <^\sharp [0,7] & = & [0,1] \\ [3,4] <^\sharp [1,2] & = & [0,0] \end{array}$$

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

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

Description Relation:

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

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

$$([\![e]\!]\, \rho) \;\Delta\; ([\![e]\!]^\sharp\, D) \quad \text{whenever} \quad \rho \;\Delta\; D$$

The Effects of Edges:

$$\begin{array}{lcl} [\![;]\!]^\sharp\, D & = & D \\ [\![x = e;]\!]^\sharp\, D & = & D \oplus \{x \mapsto [\![e]\!]^\sharp\, D\} \\ [\![x = M[e];]\!]^\sharp\, D & = & D \oplus \{x \mapsto \top\} \\ [\![M[e_1] = e_2;]\!]^\sharp\, D & = & D \\ [\![\mathrm{Pos}\,(e)]\!]^\sharp\, D & = & \begin{cases} \perp & \text{if} \quad [0,0] = [\![e]\!]^\sharp\, D \\ D & \text{otherwise} \end{cases} \\ [\![\mathrm{Neg}\,(e)]\!]^\sharp\, D & = & \begin{cases} D & \text{if} \quad [0,0] \sqsubseteq [\![e]\!]^\sharp\, D \\ \perp & \text{otherwise} \end{cases} \end{array}$$

... given that $D \neq \perp$.

Better Exploitation of Conditions (cont.)

$x == e$

$$[\![\mathrm{Neg}\,(e)]\!]^\sharp\, D \;=\; \begin{cases} \bot & \text{if} \quad [0,0] \not\sqsubseteq [\![e]\!]^\sharp\, D \\ D_1 & \text{otherwise} \end{cases}$$

where :

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

Better Exploitation of Conditions

$[0, 5]$

$$[\![\mathrm{Pos}\,(e)]\!]^\sharp\, D \;=\; \begin{cases} \bot & \text{if} \quad [0,0] = [\![e]\!]^\sharp\, D \\ D_1 & \text{otherwise} \end{cases}$$

where :

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

$D\, x \mapsto D\,x \sqcap D\,y$ $\quad x == y$

$D\, y \mapsto D\,x \sqcap D\,y$

Better Exploitation of Conditions (cont.)

$[0, 5]$

$$[\![\mathrm{Neg}\,(e)]\!]^\sharp\, D \;=\; \begin{cases} \bot & \text{if} \quad [0,0] \not\sqsubseteq [\![e]\!]^\sharp\, D \\ D_1 & \text{otherwise} \end{cases}$$

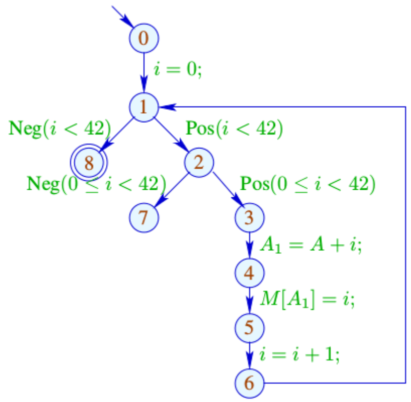
where :

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

$D\, x \mapsto D\,x \sqcap D\,y$ $\quad x == y$

$D\, y \mapsto D\,x \sqcap D\,y$

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	$\bot$	
8	42	42

$[0, 42]$

$[1, 42]$

$[0, 42] \sqcap [42, \infty] =$

$[42, 42]$

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 $\mathbb{Z}$ ...

$\Longrightarrow$ a maximal chain:

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

Formalization of the Approach

Let $x_i \sqsupseteq f_i(x_1, \ldots, x_n)\,, \quad i = 1, \ldots, n$ (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:

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

We obviously have:

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

(b) The function $G : \mathbb{D}^n \to \mathbb{D}^n$ with $G(x_1, \ldots, x_n) = (y_1, \ldots, y_n)\,, \quad y_i = x_i \sqcup f_i(x_1, \ldots, x_n)$ is increasing, i.e., $\underline{x} \sqsubseteq G\,\underline{x}$ for all $\underline{x} \in \mathbb{D}^n$.

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

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

(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:

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

for a binary operation widening:

$$\sqcup\!\!\!\!\sqcup : \mathbb{D}^2 \to \mathbb{D} \qquad \text{with} \qquad v_1 \sqcup v_2 \sqsubseteq v_1 \sqcup\!\!\!\!\sqcup v_2$$

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

... for Interval Analysis:

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

$\perp \sqcup\!\!\!\!\sqcup D = D \sqcup\!\!\!\!\sqcup \perp = D$ and for $D_1 \neq \perp \neq D_2$:

$$\begin{array}{rcll} (D_1 \sqcup\!\!\!\!\sqcup D_2)\, x & = & (D_1\, x) \sqcup\!\!\!\!\sqcup (D_2\, x) & \text{where} \\ [l_1, u_1] \sqcup\!\!\!\!\sqcup [l_2, u_2] & = & [l, u] & \text{with} \\ l & = & \begin{cases} l_1 & \text{if } l_1 \leq l_2 \\ -\infty & \text{otherwise} \end{cases} & \\ u & = & \begin{cases} u_1 & \text{if } u_1 \geq u_2 \\ +\infty & \text{otherwise} \end{cases} & \end{array}$$

$\Longrightarrow$ $\sqcup\!\!\!\!\sqcup$ is not commutative !!!

Example

$$
\begin{aligned}
[0,2] \sqcup [1,2] &= [0,2] \\
[1,2] \sqcup [0,2] &= [-\infty,2] \\
[1,5] \sqcup [3,7] &= [1,+\infty]
\end{aligned}
$$

→ 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).

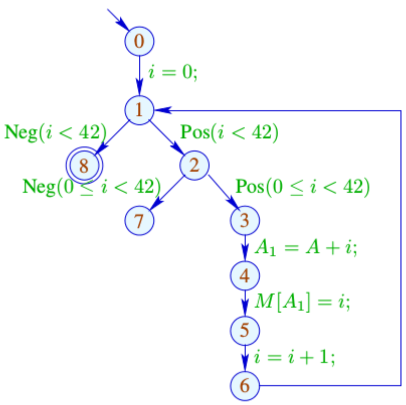
- Caveat The construction of suitable widenings is a dark art !!!

  Often $\sqcup$ is chosen dynamically during iteration such that

  → the abstract values do not get too complicated;

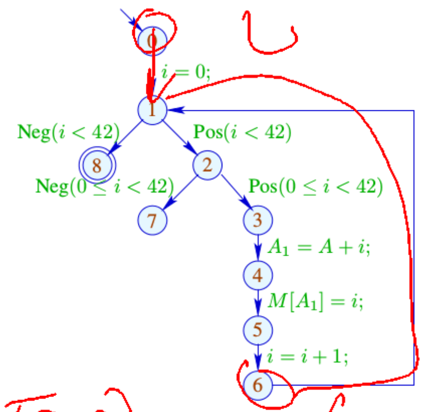
  → the number of updates remains bounded ...

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



$\sqcup$

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

$[0,0] \sqcup ([0,7] \sqsubseteq [0,\infty])$

... obviously, the result is disappointing !

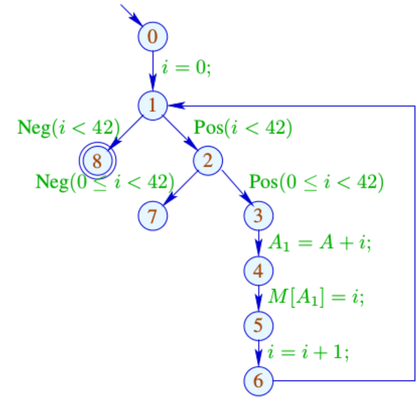
Idea 2

In fact, acceleration with $\sqcup$ 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 !!!

Our Example



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

... obviously, the result is disappointing !

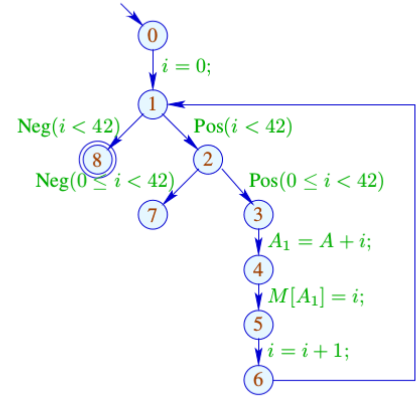
Idea 2

In fact, acceleration with $\sqcup$ 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

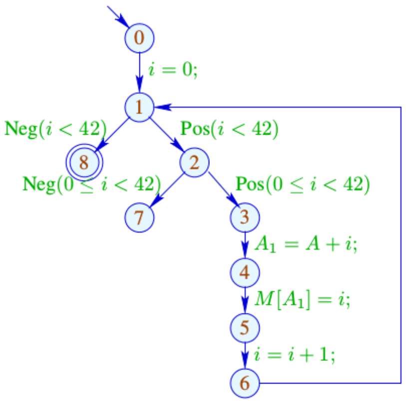


$I_1 = \{1\}$ or:

$I_2 = \{2\}$ or:

$I_3 = \{3\}$

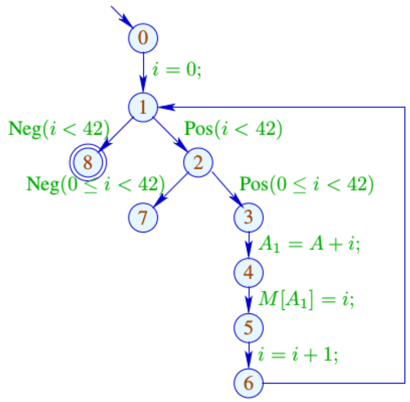
The Analysis with $I = \{1\}$:



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

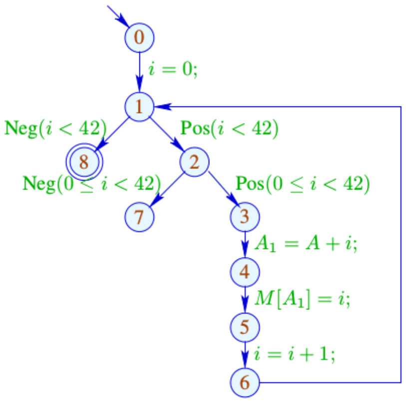
$[0, +\infty] \sqcap [42, \infty] = [42, \infty]$

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



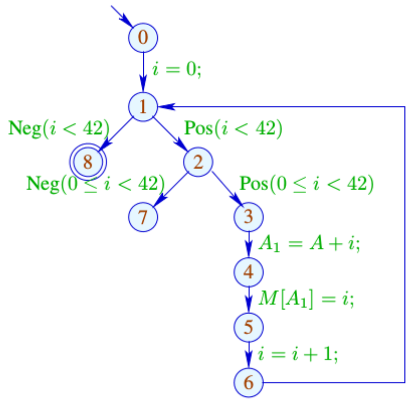
	1		2		3		4
	l	u	l	u	l	u	
0	$-\infty$	$+\infty$	$-\infty$	$+\infty$	$-\infty$	$+\infty$	ditto
1	0	0	0	1	0	42	
2	0	0	0	$+\infty$	0	$+\infty$	
3	0	0	0	41	0	41	
4	0	0	0	41	0	41	
5	0	0	0	41	0	41	
6	1	1	1	42	1	42	
7	$\perp$		42	$+\infty$	42	$+\infty$	
8	$\perp$		$\perp$		42	42	

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



	1		2		3	
	l	u	l	u	l	u
0	$-\infty$	$+\infty$	$-\infty$	$+\infty$	ditto	
1	0	0	0	$+\infty$		
2	0	0	0	41		
3	0	0	0	41		
4	0	0	0	41		
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	$-\infty$	$+\infty$	$-\infty$	$+\infty$	$-\infty$	$+\infty$	ditto
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	
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.,

$$x_i \sqsupseteq f_i\,\underline{x}\ , \qquad i = 1, \ldots, n$$

Then for monotonic f_i ,

$$\underline{x} \sqsupseteq F\,\underline{x} \sqsupseteq F^2\,\underline{x} \sqsupseteq \ldots \sqsupseteq F^k\,\underline{x} \sqsupseteq \ldots$$

// Narrowing Iteration

Idea 3: Narrowing

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

$$x_i \sqsupseteq f_i\,\underline{x}\ , \qquad i = 1, \ldots, n$$

Then for monotonic f_i ,

$$\underline{x} \sqsupseteq F\,\underline{x} \sqsupseteq F^2\,\underline{x} \sqsupseteq \ldots \sqsupseteq F^k\,\underline{x} \sqsupseteq \ldots$$

// Narrowing Iteration

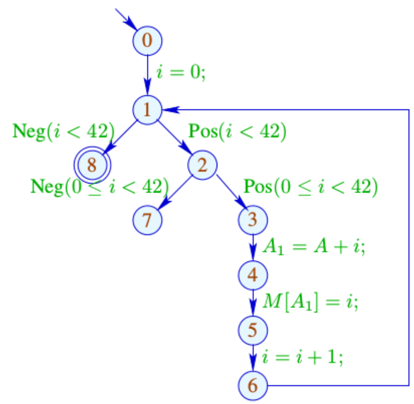
Every tuple $F^k\,\underline{x}$ is a solution of (1).

$\Longrightarrow$

Termination is no problem anymore:
we stop whenever we want !

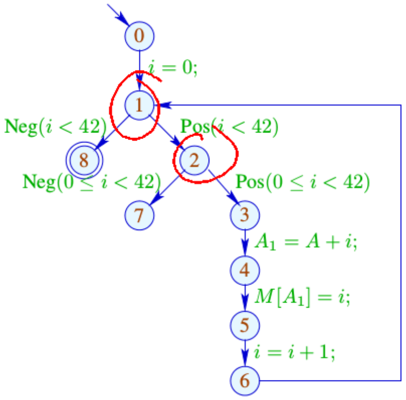
// The same also holds for RR-iteration.

Narrowing Iteration in the Example



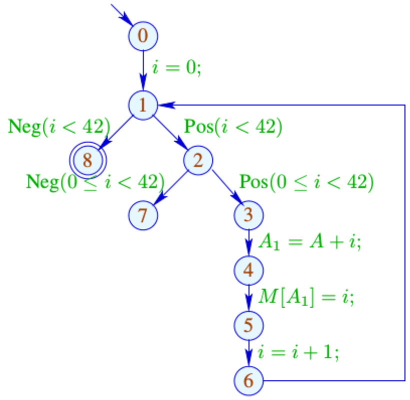
	0	
	l	u
0	$-\infty$	$+\infty$
1	0	$+\infty$
2	0	$+\infty$
3	0	$+\infty$
4	0	$+\infty$
5	0	$+\infty$
6	1	$+\infty$
7	42	$+\infty$
8	42	$+\infty$

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 !

Final Question

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

4. Idea: Accelerated Narrowing

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

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

Then consider the system of equations:

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

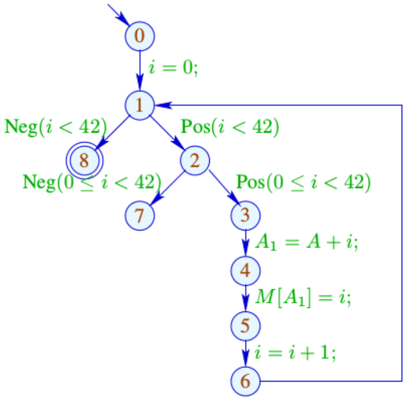
Obviously, we have for monotonic f_i : $H^k \underline{x} = F^k \underline{x}$.

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

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

$$a_1 \sqcap a_2 \sqsubseteq a_1 \overline{\sqcap} a_2 \sqsubseteq a_1$$

Accelerated Narrowing 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

$$[1,\infty] \sqcap [2,20]$$
$$= [1,20] \sqsupset [2,20]$$

... for Interval Analysis

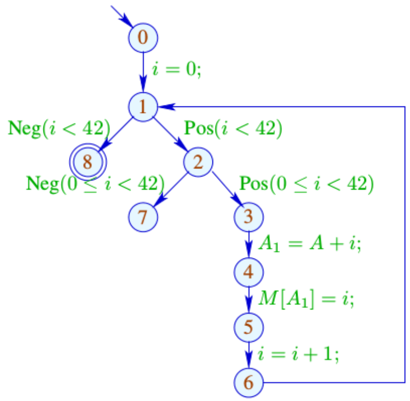
We preserve finite interval bounds.

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

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

$\Longrightarrow$ $\sqcap$ is not commutative !!!

Accelerated Narrowing 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

- 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:

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

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:

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

354

$i < n$

1.6 Pointer Analysis

$n - i$

Questions

- Are two addresses possibly equal?
- Are two addresses definitively equal?

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:

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

354

1.6 Pointer Analysis

Questions

→ Are two addresses possibly equal?

→ Are two addresses definitively equal?