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 ≤ i ∧ 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:

$$[l_1, u_1] \subseteq [l_2, u_2] \iff l_2 \leq l_1 \land u_1 \leq u_2$$
Thus:

\[ [l_1, u_1] \sqcup [l_2, u_2] = [l_1 \sqcap l_2, u_1 \sqcup u_2] \]
Thus:

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

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

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.,

$$[0, 0] \sqsubseteq [0, 1] \sqsubseteq [-1, 1] \sqsubseteq [-1, 2] \sqsubseteq \ldots$$
Caveat:

→ \( \mathbb{I} \) is not a complete lattice  

\( \rightarrow \) \( \mathbb{I} \) has infinite ascending chains, e.g.,

\[
[0, 0] \sqsubset [0, 1] \sqsubset [-1, 1] \sqsubset [-1, 2] \sqsubset \ldots
\]

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:

\[ \gamma [0, 7] = \{0, \ldots, 7\} \]
\[ \gamma [0, \infty] = \{0, 1, 2, \ldots, \} \]

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:
\[[l_1, u_1] \ast^\# [l_2, u_2] = [a, b] \quad \text{where} \]
\[a = l_1 l_2 \sqcap l_1 u_2 \sqcap u_1 l_2 \sqcap u_1 u_2\]
\[b = l_1 l_2 \sqcup l_1 u_2 \sqcup u_1 l_2 \sqcup u_1 u_2\]

Example:
\[[0, 2] \ast^\# [3, 4] = [0, 8]\]
\[[{-1, 2}] \ast^\# [3, 4] = [{-4, 8}]\]
\[[{-1, 2}] \ast^\# [{-3, 4}] = [{-6, 8}]\]
\[[{-1, 2}] \ast^\# [{-4, -3}] = [{-8, 4}]\]
Division: $[l_1, u_1] / [l_2, u_2] = [a, b]$

- If 0 is not contained in the interval of the denominator, then:
  \[
  a = \frac{l_1}{l_2} \cap \frac{l_1}{u_2} \cap \frac{u_1}{l_2} \cap \frac{u_1}{u_2}
  \]
  \[
  b = \frac{l_1}{l_2} \cup \frac{l_1}{u_2} \cup \frac{u_1}{l_2} \cup \frac{u_1}{u_2}
  \]

- If: $l_2 \leq 0 \leq u_2$, we define:
  \[
  [a, b] = [-\infty, +\infty]
  \]
Equality:

\[ [l_1, u_1] =\# [l_2, u_2] = \begin{cases} [1, 1] & \text{if } l_1 = u_1 = l_2 = u_2 \\ [0, 0] & \text{if } u_1 < l_2 \lor u_2 < l_1 \\ [0, 1] & \text{otherwise} \end{cases} \]
Equality:

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

Example:

\[
[42, 42] =\# [42, 42] = [1, 1] \\
[0, 7] =\# [0, 7] = [0, 1] \\
[1, 2] =\# [3, 4] = [0, 0]
\]
Less:

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

\[
\begin{align*}
[1, 2] <^# [9, 42] &= [1, 1] \\
[0, 7] <^# [0, 7] &= [0, 1] \\
[3, 4] <^# [1, 2] &= [0, 0]
\end{align*}
\]
By means of \( \mathbb{I} \) we construct the complete lattice:

\[
\mathbb{D}_\mathbb{I} = (\text{Vars} \rightarrow \mathbb{I})_\bot
\]

**Description Relation:**

\[
\rho \triangle D \quad \text{iff} \quad D \neq \bot \land \forall x \in \text{Vars} : (\rho x) \triangle (D x)
\]

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

\[
([e] \rho) \triangle ([e] \downarrow D) \quad \text{whenever} \quad \rho \triangle D
\]
The Effects of Edges:

\[
\begin{align*}
[x]# D &= D \\
[x = e;]# D &= D \oplus \{ x \mapsto [e]# D \} \\
[x = M[e];]# D &= D \oplus \{ x \mapsto \top \} \\
[M[e_1] = e_2;]# D &= D \\
[\text{Pos}(e)]# D &= \begin{cases} \\
\bot & \text{if } [0, 0] = [e]# D \\
D & \text{otherwise} \\
\end{cases} \\
[\text{Neg}(e)]# D &= \begin{cases} \\
D & \text{if } [0, 0] \sqsubseteq [e]# D \\
\bot & \text{otherwise} \\
\end{cases}
\end{align*}
\]

... given that \( D \neq \bot \) :-)}
Better Exploitation of Conditions:

\[ \lbrack \text{Pos}(e) \rbrack^\# D = \begin{cases} \bot & \text{if } [0, 0] = \lbrack e \rbrack^\# D \\ D_1 & \text{otherwise} \end{cases} \]

where:

\[ D_1 = \begin{cases} D \oplus \{ x \mapsto (D x) \cap ( \lbrack e_1 \rbrack^\# D ) \} & \text{if } e \equiv x = e_1 \\ D \oplus \{ x \mapsto (D x) \cap [ -\infty, u ] \} & \text{if } e \equiv x \leq e_1, \lbrack e_1 \rbrack^\# D = [ -, u ] \\ D \oplus \{ x \mapsto (D x) \cap [ l, \infty ] \} & \text{if } e \equiv x \geq e_1, \lbrack e_1 \rbrack^\# D = [ l, - ] \end{cases} \]
Better Exploitation of Conditions (cont.):

\[ [\text{Neg}(e)]^\# D = \begin{cases} \bot & \text{if } \ [0, 0] \not\subseteq [e]^\# D \\ D_1 & \text{otherwise} \end{cases} \]

where:

\[ D_1 = \begin{cases} D \oplus \{x \mapsto (D \times) \cap ([e_1]^\# D)\} & \text{if } e \equiv x \neq e_1 \\ D \oplus \{x \mapsto (D \times) \cap [-\infty, u]\} & \text{if } e \equiv x > e_1, [e_1]^\# D = [\_, u] \\ D \oplus \{x \mapsto (D \times) \cap [l, \infty]\} & \text{if } e \equiv x < e_1, [e_1]^\# D = [l, \_] \end{cases} \]
Example:

State transitions:

- $i = 0$
- $i = i + 1$
- $A_1 = A + i$
- $M[A_1] = i$

Table:

<table>
<thead>
<tr>
<th>i</th>
<th>l</th>
<th>u</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$-\infty$</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>42</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>41</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>41</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>41</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>41</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>42</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>42</td>
<td>42</td>
</tr>
</tbody>
</table>
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 \( \mathbb{Z} \) ...

\[ [3, 17] \sqsubseteq [3, +\infty] \sqsubseteq [-\infty, +\infty] \]
Formalization of the Approach:

Let \( x_i \supseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \) \hspace{1cm} (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:

\[
x_i = x_i \sqcup f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \hspace{1cm} (2)
\]

We obviously have:

(a) \( x \) is a solution of (1) iff \( x \) 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., \( x \subseteq Gx \) for all \( x \in \mathbb{D}^n \).
(c) The sequence $G^k \bot, \; k \geq 0,$ is an ascending chain:

\[ \bot \subseteq G \subseteq \ldots \subseteq G^k \subseteq \ldots \]

(d) If $G^k \bot = G^{k+1} \bot = y,$ then $y$ 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:

\[ x_i = x_i \sqcup f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \]  \hspace{1cm} (3)

for a binary operation **widening**:

\[ \sqcup : \mathbb{D}^2 \rightarrow \mathbb{D} \]  \hspace{1cm} with \hspace{1cm} $v_1 \sqcup v_2 \sqsubseteq v_1 \sqcup v_2$

(RR)-iteration for (3) still will compute a solution of (1) :-)}
... for Interval Analysis:

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

$$\bot \sqcup D = D \sqcup \bot = D$$

and for $D_1 \neq \bot \neq D_2$:

$$(D_1 \sqcup D_2) x = (D_1 x) \sqcup (D_2 x)$$

where

$$[l_1, u_1] \sqcup [l_2, u_2] = [l, u]$$

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:

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

→ 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 ⊔ is chosen dynamically during iteration such that
  → the abstract values do not get too complicated;
  → the number of updates remains bounded ...
Our Example:

\[ i = 0; \]

\[ \text{Neg}(i < 42) \]

\[ \text{Pos}(i < 42) \]

\[ \text{Neg}(0 \leq i < 42) \]

\[ \text{Pos}(0 \leq i < 42) \]

\[ A_1 = A + i; \]

\[ M[A_1] = i; \]

\[ i = i + 1; \]

\[ l \quad u \]

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-\infty</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>\bot</td>
</tr>
<tr>
<td>8</td>
<td>\bot</td>
</tr>
</tbody>
</table>
Our Example:

\[ i = 0; \]

\[ \text{Neg}(i < 42) \]

\[ \text{Pos}(i < 42) \]

\[ \text{Neg}(0 \leq i < 42) \]

\[ \text{Pos}(0 \leq i < 42) \]

\[ A_1 = A + i; \]

\[ M[A_1] = i; \]

\[ i = i + 1; \]

\[ l \quad u \quad l \quad u \quad l \quad u \]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th></th>
<th>2</th>
<th></th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$-\infty$</td>
<td>$+\infty$</td>
<td>$-\infty$</td>
<td>$+\infty$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$+\infty$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$+\infty$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$+\infty$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$+\infty$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$+\infty$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$+\infty$</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$\perp$</td>
<td>42</td>
<td>$+\infty$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$\perp$</td>
<td>42</td>
<td>$+\infty$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

dito
... 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:

\[ i = 0; \]

\[ \text{Neg}(i < 42) \]

\[ \text{Neg}(0 \leq i < 42) \]

\[ \text{Pos}(0 \leq i < 42) \]

\[ \text{Pos}(i < 42) \]

\[ I_1 = \{1\} \quad \text{or:} \]

\[ I_2 = \{2\} \quad \text{or:} \]

\[ I_3 = \{3\} \]
The Analysis with \( I = \{1\} \):

\[
i = 0;
\]

\[
\text{Neg}(i < 42) \quad \text{Pos}(i < 42)
\]

\[
\text{Neg}(0 \leq i < 42) \quad \text{Pos}(0 \leq i < 42)
\]

\[
\begin{align*}
A_1 &= A + i; \\
M[A_1] &= i; \\
i &= i + 1;
\end{align*}
\]

\[
\begin{array}{c|c|c|c|c|c|c|c}
 & 1 & 2 & 3 \\
\hline
l & l & l & l \\
u & u & u & u \\
\hline
0 & -\infty & +\infty & -\infty & +\infty \\
1 & 0 & 0 & 0 & +\infty \\
2 & 0 & 0 & 0 & 41 \\
3 & 0 & 0 & 0 & 41 \\
4 & 0 & 0 & 0 & 41 \\
5 & 0 & 0 & 0 & 41 \\
6 & 1 & 1 & 1 & 42 \\
7 & \bot & \bot & & \\
8 & \bot & 42 & +\infty & \\
\end{array}
\]
The Analysis with  \( I = \{2\} \):
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 \( x \) denote any solution of (1), i.e.,
\[
x_i \supseteq f_i x , \quad i = 1, \ldots, n
\]
Then for monotonic \( f_i \),
\[
x \supseteq F x \supseteq F^2 x \supseteq \ldots \supseteq F^k x \supseteq \ldots
\]
// Narrowing Iteration
Idea 3: Narrowing

Let $x$ denote any solution of (1), i.e.,

$$x_i \supseteq f_i x,$$  

$i = 1, \ldots, n$

Then for monotonic $f_i$,

$$x \supseteq F x \supseteq F^2 x \supseteq \ldots \supseteq F^k x \supseteq \ldots$$

// Narrowing Iteration

Every tuple $F^k x$ 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:

\[ i = 0; \]

Neg\((i < 42)\)

Pos\((i < 42)\)

Neg\((0 \leq i < 42)\)

Pos\((0 \leq i < 42)\)

\[ A_1 = A + i; \]

\[ M[A_1] = i; \]

\[ i = i + 1; \]

\begin{array}{|c|c|c|}
\hline
l & u \\
\hline
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 \\
\hline
\end{array}
Narrowing Iteration in the Example:

0

\[ i = 0; \]

1

\[ \text{Pos}(i < 42) \]

3

\[ A_1 = A + i; \]

4

\[ M[A_1] = i; \]

5

\[ i = i + 1; \]

2

\[ \text{Pos}(0 \leq i < 42) \]

8

\[ \text{Neg}(i < 42) \]

Neg(0 \leq i < 42)
Narrowing Iteration in the Example:

\[ i = 0; \]

\[ \text{Neg}(i < 42) \quad \text{Pos}(i < 42) \]

\[ \text{Neg}(0 \leq i < 42) \quad \text{Pos}(0 \leq i < 42) \]

\[ A_1 = A + i; \]

\[ M[A_1] = i; \]

\[ i = i + 1; \]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>l</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
</tr>
<tr>
<td>u</td>
<td>$+\infty$</td>
<td>$+\infty$</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>l</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>u</td>
<td>$+\infty$</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>l</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>u</td>
<td>$+\infty$</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>l</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>u</td>
<td>$+\infty$</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>l</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>u</td>
<td>$+\infty$</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>l</td>
<td>42</td>
<td>$+\infty$</td>
<td>$\perp$</td>
</tr>
<tr>
<td>u</td>
<td>42</td>
<td>$+\infty$</td>
<td>$\perp$</td>
</tr>
</tbody>
</table>
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 \( \mathbf{x} = (x_1, \ldots, x_n) \) of the system of constraints:

\[
x_i \supseteq f_i(x_1, \ldots, x_n) , \quad i = 1, \ldots, n
\]  

(1)

Then consider the system of equations:

\[
x_i = x_i \sqcap f_i(x_1, \ldots, x_n) , \quad i = 1, \ldots, n
\]  

(4)

Obviously, we have for monotonic \( f_i \):

\[
H^k \mathbf{x} = F^k \mathbf{x} \quad :-)
\]

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

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

\[
a_1 \sqcap a_2 \subseteq a_1 \sqcap a_2 \subseteq a_1
\]
... for Interval Analysis:

We preserve finite interval bounds $:-)$

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

$$(D_1 \sqcap D_2) x = (D_1 x) \sqcap (D_2 x)$$

where

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

$$l = \begin{cases} l_2 & \text{if } l_1 = -\infty \\ l_1 & \text{otherwise} \end{cases}$$

$$u = \begin{cases} u_2 & \text{if } u_1 = \infty \\ u_1 & \text{otherwise} \end{cases}$$

$\implies \sqcap$ is not commutative $!!!$
Accelerated Narrowing in the Example:

\[ i = 0; \]

\[ i < 42 \]

\[ i \leq 42 \]

\[ A_1 = A + i; \]

\[ M[A_1] = i; \]

\[ i = i + 1; \]

\[ i < 42 \]

\[ i \leq 42 \]

\[ 0 \leq i < 42 \]

\[ l \quad u \]
\[ \begin{array}{c|c|c|c|c}
0 & -\infty & +\infty & -\infty & +\infty & -\infty & +\infty \\
1 & 0 & +\infty & 0 & +\infty & 0 & 42 \\
2 & 0 & +\infty & 0 & 41 & 0 & 41 \\
3 & 0 & +\infty & 0 & 41 & 0 & 41 \\
4 & 0 & +\infty & 0 & 41 & 0 & 41 \\
5 & 0 & +\infty & 0 & 41 & 0 & 41 \\
6 & 1 & +\infty & 1 & 42 & 1 & 42 \\
7 & 42 & +\infty & \bot & \bot & \ Bot & \bot \\
8 & 42 & +\infty & 42 & +\infty & 42 & 42 \\
\end{array} \]
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:

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