Script generated by TTT

Title: Seidl: Programmoptimierung (18.11.2013)

Date: Mon Nov 18 14:16:07 CET 2013

Duration: 89:28 min

Pages: 42

We conclude: The assertion $(*)$ is true :-))

The MOP-Solution

$$\mathcal{D}^*[v] \;=\; \bigsqcup \{ [\![\pi]\!]^\sharp \, D_\top \mid \pi : start \to^* v \}$$

where $D_\top \, x = \top \qquad (x \in Vars)$.

By $(*)$, we have for all initial states s and all program executions π which reach v :

$$([\![\pi]\!]\, s) \;\;\Delta\;\; (\mathcal{D}^*[v])$$

In order to approximate the MOP, we use our constraint system :-))

We conclude: The assertion $(*)$ is true :-))

The MOP-Solution

$$\mathcal{D}^*[v] \;=\; \bigsqcup \{ [\![\pi]\!]^\sharp \, D_\top \mid \pi : start \to^* v \}$$

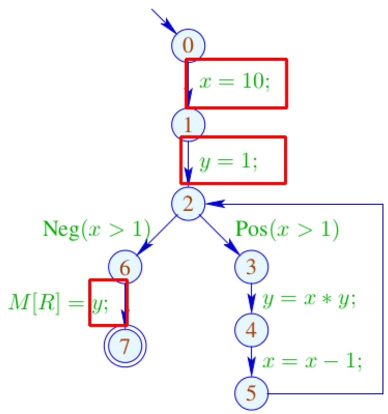
where $D_\top \, x = \top \qquad (x \in Vars)$.

By $(*)$, we have for all initial states s and all program executions π which reach v :

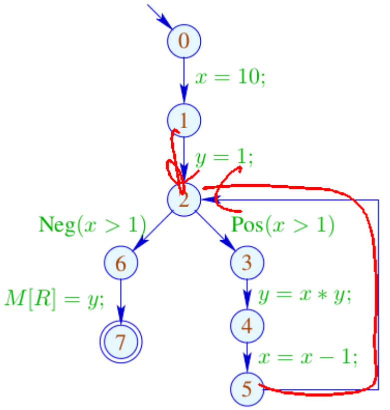
$$([\![\pi]\!]\, s) \;\;\Delta\;\; (\mathcal{D}^*[v])$$

In order to approximate the MOP, we use our constraint system :-))

Example:



Example:

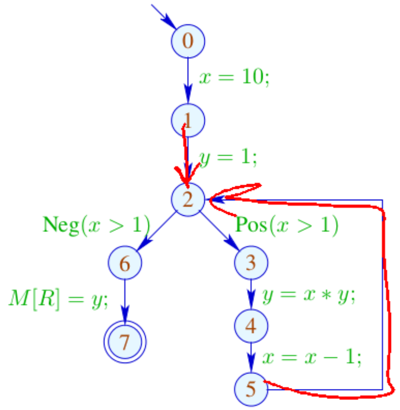


	1	
	x	y
0	⊤	⊤
1	10	⊤
2	10	1
3	10	1
4	10	10
5	9	10
6	⊥	
7	⊥	

$\perp$

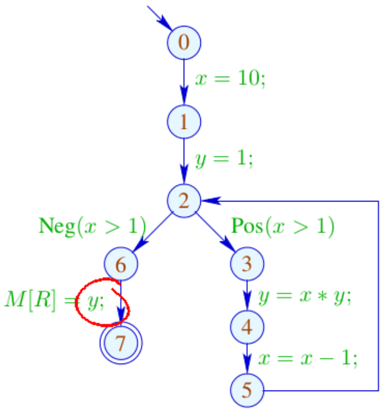
$(10,1) \sqcup (9,10) = (\top,\top)$

Example:



	1		2	
	x	y	x	y
0	⊤	⊤	⊤	⊤
1	10	⊤	10	⊤
2	10	1	⊤	⊤
3	10	1	⊤	⊤
4	10	10	⊤	⊤
5	9	10	⊤	⊤
6	⊥		⊤	⊤
7	⊥		⊤	⊤

Example:



	1		2		3	
	x	y	x	y	x	y
0	⊤	⊤	⊤	⊤		
1	10	⊤	10	⊤		
2	10	1	⊤	⊤		
3	10	1	⊤	⊤		
4	10	10	⊤	⊤	dito	
5	9	10	⊤	⊤		
6	⊥		⊤	⊤		
7	⊥		⊤	⊤		

Conclusion:

Although we compute with concrete values, we fail to compute everything :-(

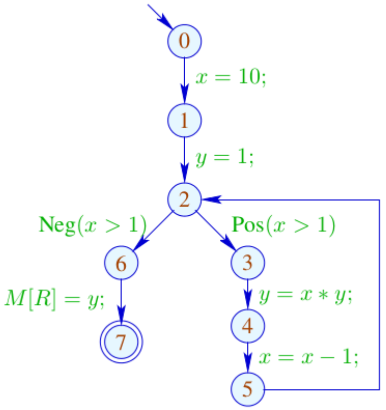
The fixpoint iteration, at least, is guaranteed to terminate:

For n program points and m variables, we maximally need: $n \cdot (m + 1)$ rounds :-)

Caveat:

The effects of edge are not distributive !!!

Example:



	1		2		3	
	x	y	x	y	x	y
0	$\top$	$\top$	$\top$	$\top$		
1	10	$\top$	10	$\top$		
2	10	1	$\top$	$\top$		
3	10	1	$\top$	$\top$		
4	10	10	$\top$	$\top$	dito	
5	9	10	$\top$	$\top$		
6	$\bot$		$\top$	$\top$		
7	$\bot$		$\top$	$\top$		

Conclusion:

Although we compute with concrete values, we fail to compute everything :-(

The fixpoint iteration, at least, is guaranteed to terminate:

For n program points and m variables, we maximally need: $n \cdot (m + 1)$ rounds :-)

Caveat:

The effects of edge are not distributive !!!

Counter Example: $f = [\![x = x + y;]\!]^\sharp$

$$\begin{array}{lrcl}
\text{Let} & D_1 & = & \{x \mapsto 2, y \mapsto 3\} \\
 & D_2 & = & \{x \mapsto 3, y \mapsto 2\} \\[1em]
\text{Dann} & f\,D_1 \sqcup f\,D_2 & = & \{x \mapsto 5, y \mapsto 3\} \sqcup \{x \mapsto 5, y \mapsto 2\} \\
 & & = & \{x \mapsto 5, y \mapsto \top\} \\
 & & \neq & \{x \mapsto \top, y \mapsto \top\} \\
 & & = & f\,\{x \mapsto \top, y \mapsto \top\} \\
 & & = & f\,(D_1 \sqcup D_2)
\end{array}$$

:-((

We conclude:

The least solution $\mathcal{D}$ of the constraint system in general yields only an upper approximation of the MOP, i.e.,

$$\mathcal{D}^*[v] \quad \sqsubseteq \quad \mathcal{D}[v]$$

We conclude:

The least solution $\mathcal{D}$ of the constraint system in general yields only an upper approximation of the MOP, i.e.,

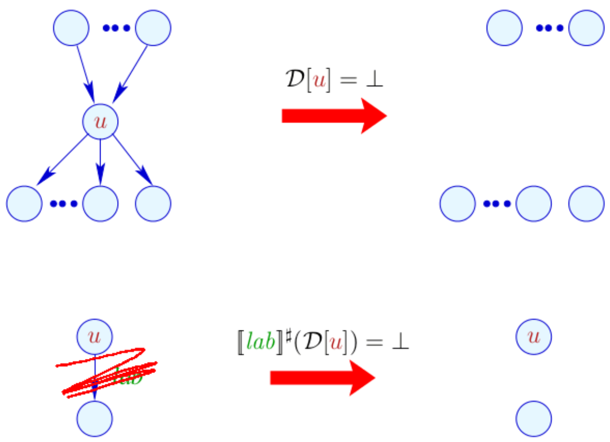
$$\mathcal{D}^*[v] \quad \sqsubseteq \quad \mathcal{D}[v]$$

As an upper approximation, $\mathcal{D}[v]$ nonetheless describes the result of every program execution π which reaches v :

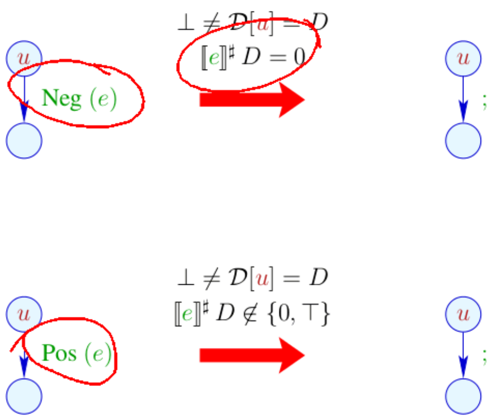
$$([\![\pi]\!]\,(\rho, \mu)) \quad \Delta \quad (\mathcal{D}[v])$$

whenever $[\![\pi]\!]\,(\rho, \mu)$ is defined ;-))

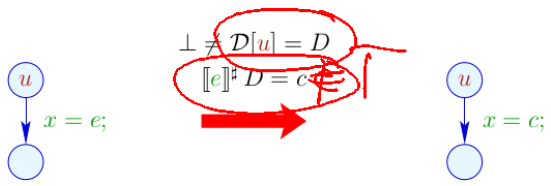
Transformation 4: Removal of Dead Code



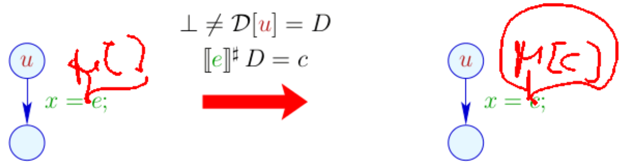
Transformation 4 (cont.): Removal of Dead Code



Transformation 4 (cont.): Simplified Expressions



Transformation 4 (cont.): Simplified Expressions



Extensions:

- Instead of complete right-hand sides, also subexpressions could be simplified:

$$x + (3 * y) \quad \xrightarrow{\{x \mapsto \top, y \mapsto 5\}} \quad x + 15$$

... and further simplifications be applied, e.g.:

$$\begin{array}{lcl} x * 0 & \Longrightarrow & 0 \\ x * 1 & \Longrightarrow & x \\ x + 0 & \Longrightarrow & x \\ x - 0 & \Longrightarrow & x \\ & \ldots & \end{array}$$

- So far, the information of conditions has not yet be optimally exploited:

$\top \sqcap 7 \quad = 7$

$$\begin{array}{l} \text{if } (x == 7) \\ \quad y = x + 3; \end{array}$$

Even if the value of x before the if statement is unknown, we at least know that x definitely has the value 7 — whenever the then-part is entered :-)

Therefore, we can define:

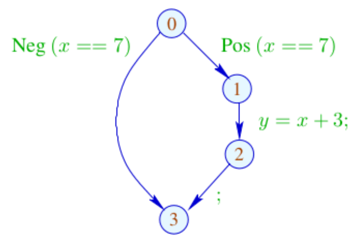
$$[\![\text{Pos}\,(x == e)]\!]^\sharp\, D = \begin{cases} D & \text{if} \quad [\![x == e]\!]^\sharp\, D = 1 \\ \bot & \text{if} \quad [\![x == e]\!]^\sharp\, D = 0 \\ D_1 & \text{otherwise} \end{cases}$$

where

$$D_1 = D \oplus \{x \mapsto (D\,x \sqcap [\![e]\!]^\sharp\, D)\}$$

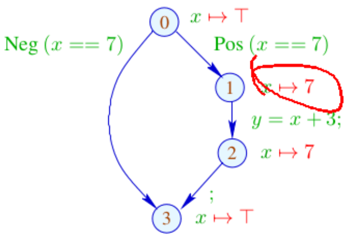
The effect of an edge labeled $\text{Neg}\,(x \neq e)$ is analogous :-)

Our Example:



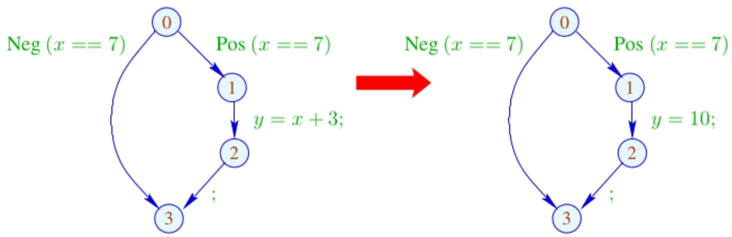
The effect of an edge labeled Neg $(x \neq e)$ is analogous :-)

Our Example:



The effect of an edge labeled Neg $(x \neq e)$ is analogous :-)

Our Example:



1.5 Interval Analysis

Observation:

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

  $\Longrightarrow$

  Constant propagation is useful :-)

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

Example:

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

Obviously, the inner check is superfluous :-)

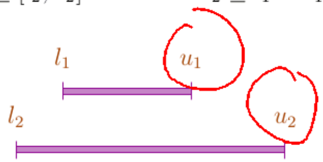
Idea 1:

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

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

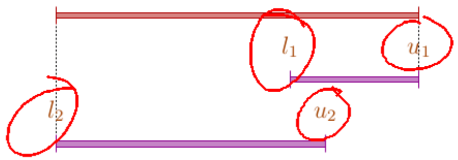
Partial Ordering:

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



Thus:

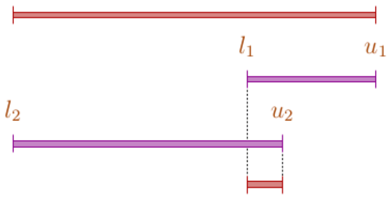
$$[l_1, u_1] \sqcup [l_2, u_2] \quad = \quad [l_1 \sqcap l_2, u_1 \sqcup u_2]$$



Thus:

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

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



Caveat:

- → $\mathbb{I}$ is not a complete lattice :-)
- → $\mathbb{I}$ has infinite ascending chains, e.g.,

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

Caveat:

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

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

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

Description Relation:

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

Concretization:

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

Example:

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

Computing with intervals: Interval Arithmetic :-)

Addition:

$$\begin{aligned} [l_1,u_1] +^\sharp [l_2,u_2] &= [l_1+l_2, u_1+u_2] \quad \text{where} \\ -\infty + \_ &= -\infty \\ +\infty + \_ &= +\infty \\ // \quad -\infty + \infty & \text{ cannot occur :-)} \end{aligned}$$

Negation:

$$-^\sharp\,[l,u] = [-u,-l]$$

Multiplication:

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

Example:

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

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

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

$$\begin{aligned} a &= l_1/l_2 \sqcap l_1/u_2 \sqcap u_1/l_2 \sqcap u_1/u_2 \\ b &= l_1/l_2 \sqcup l_1/u_2 \sqcup u_1/l_2 \sqcup u_1/u_2 \end{aligned}$$

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

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

Equality:

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

$$[1,2] ==^\sharp [1,2]$$

$$[1,2] ==^\sharp [2,3]$$

Equality:

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

Example:

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

Less:

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

Less:

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

Example:

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

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

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

Description Relation:

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

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

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

ρ

The Effects of Edges:

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

$[15, \infty]$

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

Better Exploitation of Conditions:

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

where :

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

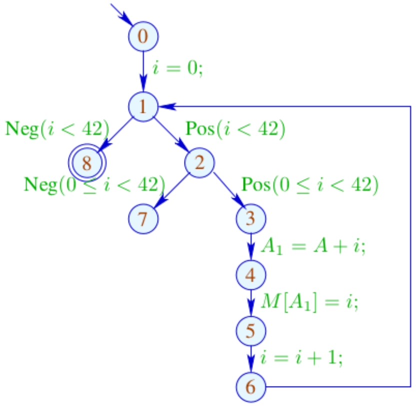
Better Exploitation of Conditions (cont.):

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

where :

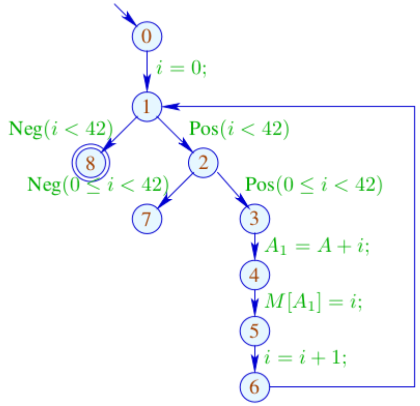
$$D_1 = \begin{cases} D \oplus \{x \mapsto (D\,x) \sqcap ([\![e_1]\!]^\sharp\, D)\} & \text{if } e \equiv x \neq e_1 \\ D \oplus \{x \mapsto (D\,x) \sqcap [-\infty, u]\} & \text{if } e \equiv x > e_1, [\![e_1]\!]^\sharp\, D = [\_, u] \\ D \oplus \{x \mapsto (D\,x) \sqcap [l, \infty]\} & \text{if } e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

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

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

335

Problem:

$\rightarrow$ The solution can be computed with RR-iteration — after about 42 rounds :-(

$\rightarrow$ 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:

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