

IC637 Program Analysis

Lecture 3: Interval Domain

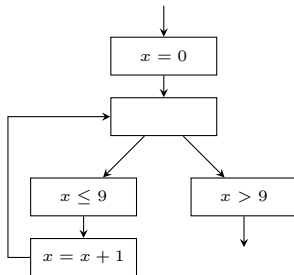
Minseok Jeon

2025 Fall

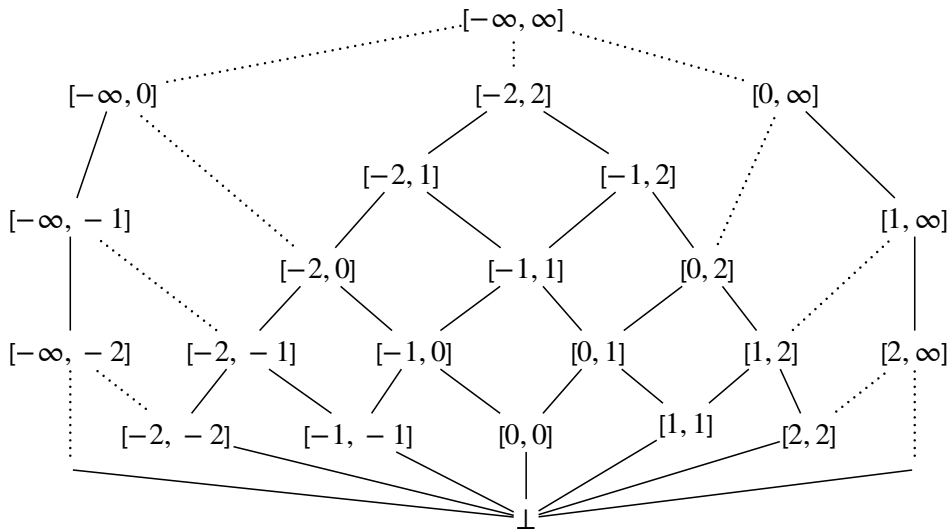
Example

- Can (extended) sign domain prove the assertion?
- How can we prove the assertion?

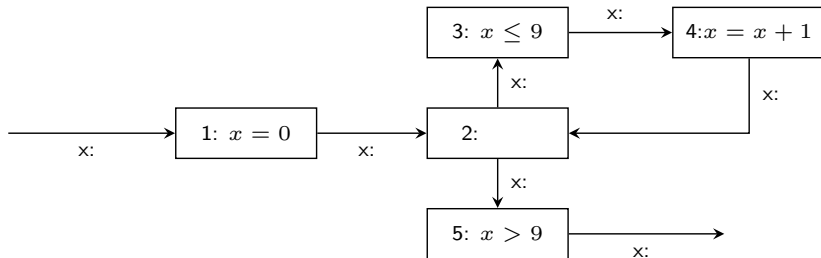
```
int main(){  
    int x = 0;  
    while(x <= 9){  
        x = x + 1;  
    }  
    assert(x == 10);  
    return 0;  
}
```



Interval Domain

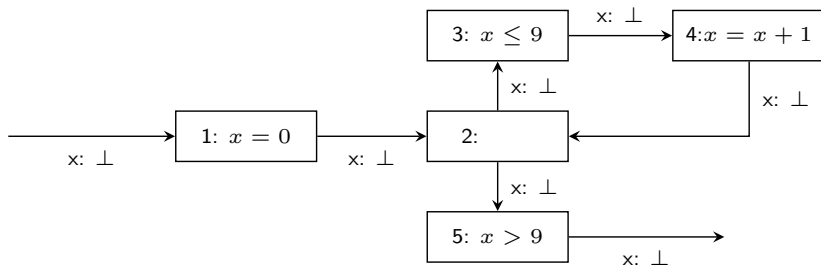


Fixed Point Computation



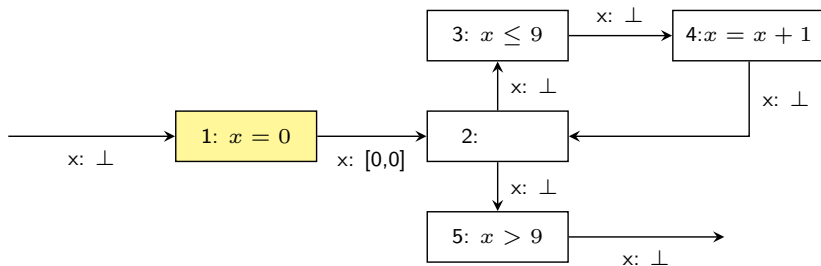
Worklist = $\{1, 2, 3, 4, 5\}$

Fixed Point Computation



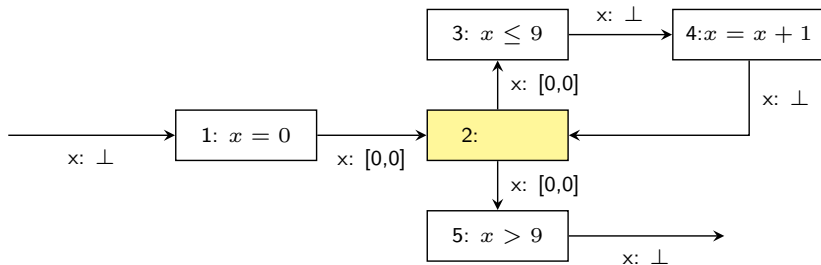
Worklist = $\{1, 2, 3, 4, 5\}$

Fixed Point Computation



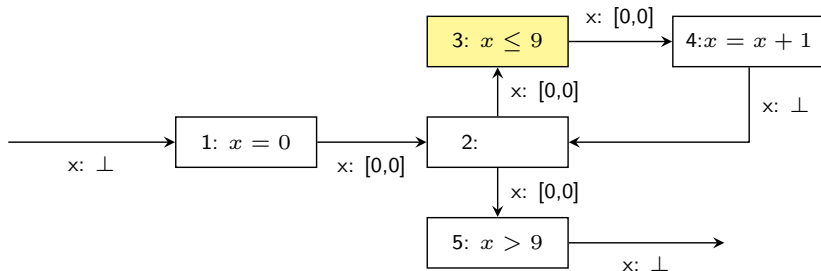
Worklist = $\{1, 2, 3, 4, 5\}$

Fixed Point Computation



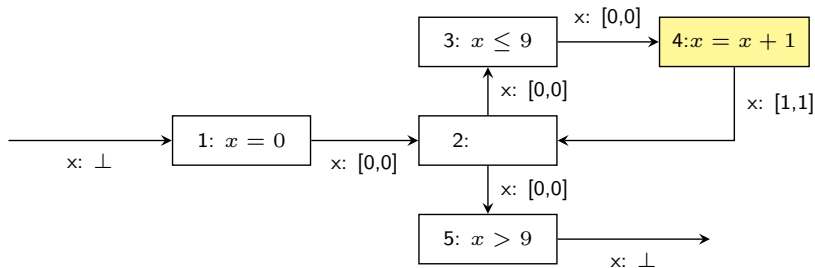
Worklist = $\{1, 2, 3, 4, 5\}$

Fixed Point Computation



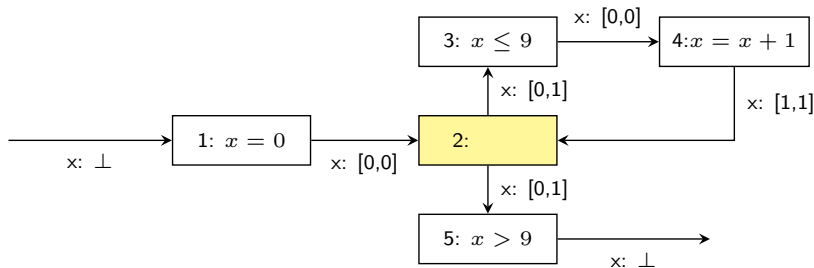
Worklist = $\{\cancel{1}, \cancel{2}, \cancel{3}, 4, 5\}$

Fixed Point Computation



Worklist = $\{\cancel{1}, 2, \cancel{3}, \cancel{4}, 5\}$

Fixed Point Computation

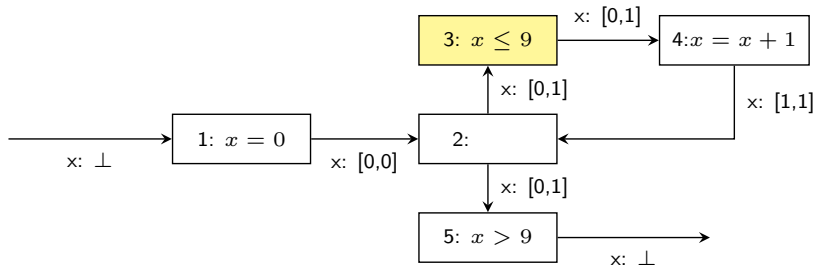


Input state : $[0, 0] \sqcup [1, 1] = [0, 1]$

(1st iteration of loop)

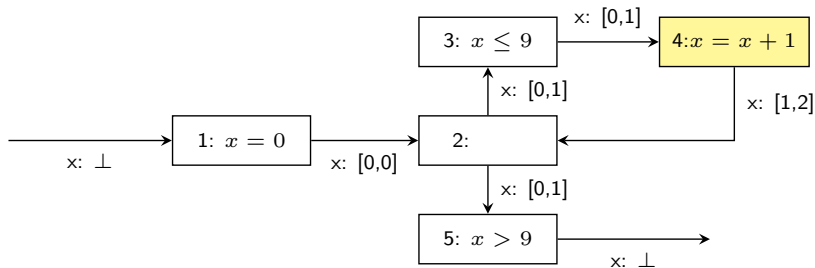
Worklist = $\{\cancel{1}, \cancel{2}, 3, \cancel{4}, 5\}$

Fixed Point Computation



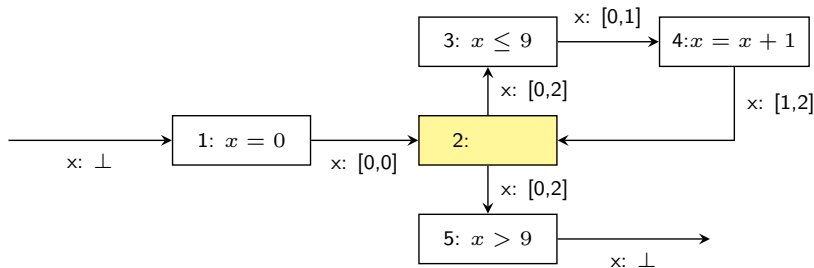
$$[0, 1] \sqcap [-\infty, 9] = [0, 1]$$
$$\text{Worklist} = \{\cancel{1}, \cancel{2}, \cancel{3}, 4, 5\}$$

Fixed Point Computation



Worklist = $\{\cancel{1}, 2, \cancel{3}, \cancel{4}, 5\}$

Fixed Point Computation

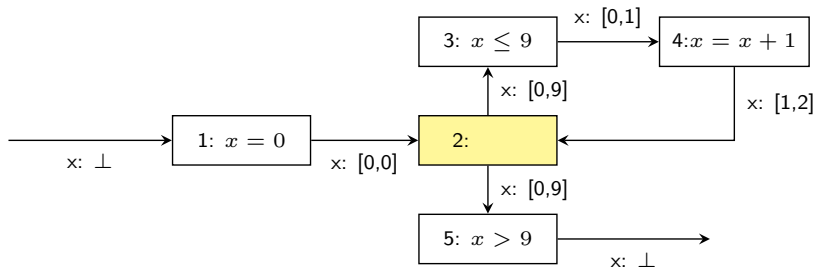


Input state : $[0, 0] \sqcup [1, 2] = [0, 2]$

(2nd iteration of loop)

Worklist = $\{\cancel{1}, \cancel{2}, 3, \cancel{4}, 5\}$

Fixed Point Computation

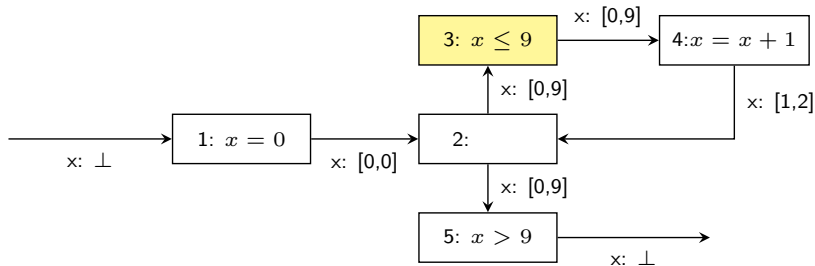


Input state : $[0, 0] \sqcup [1, 9] = [0, 9]$

(9th iteration of loop)

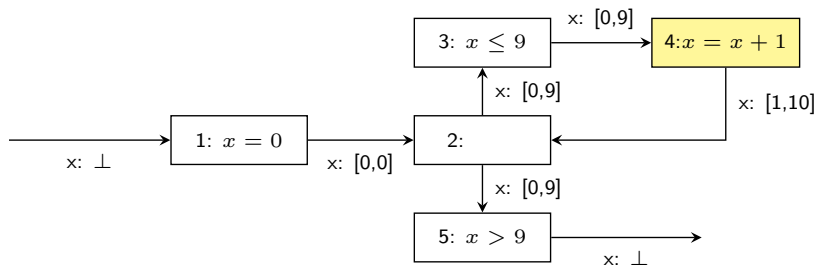
Worklist = $\{\cancel{1}, \cancel{2}, 3, \cancel{4}, 5\}$

Fixed Point Computation



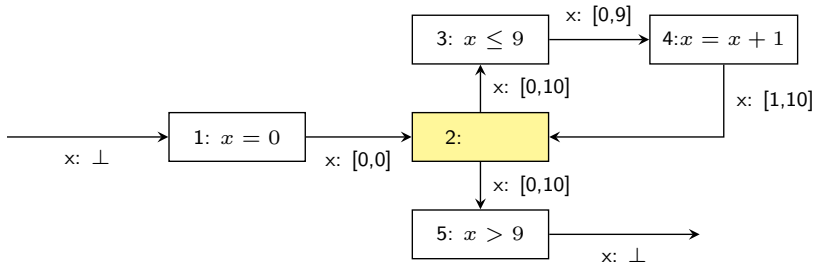
$$[0, 9] \sqcap [-\infty, 9] = [0, 9]$$
$$\text{Worklist} = \{\cancel{1}, \cancel{2}, \cancel{3}, 4, 5\}$$

Fixed Point Computation



Worklist = $\{\cancel{1}, 2, \cancel{3}, \cancel{4}, 5\}$

Fixed Point Computation

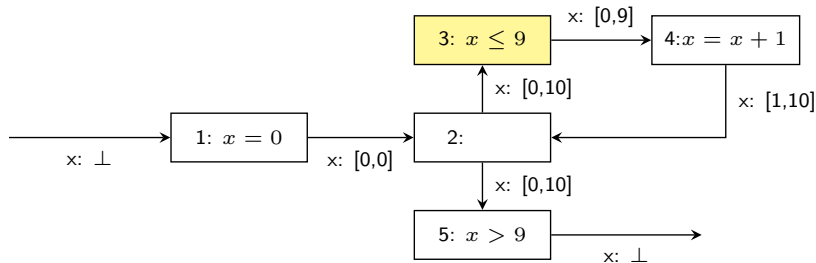


Input state : $[0, 0] \sqcup [1, 10] = [0, 10]$

(10th iteration of loop)

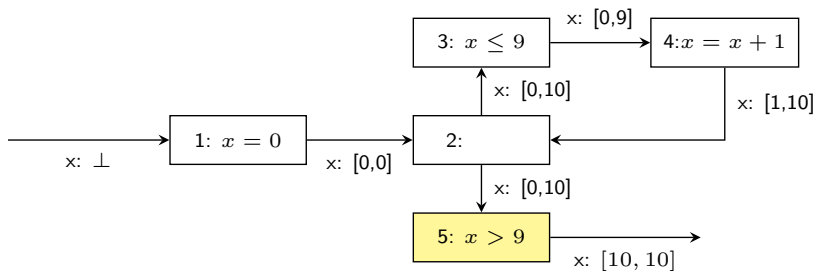
Worklist = $\{\cancel{1}, \cancel{2}, 3, \cancel{4}, 5\}$

Fixed Point Computation



$$[0, 10] \sqcap [-\infty, 9] = [0, 9]$$
$$\text{Worklist} = \{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, 5\}$$

Fixed Point Computation



$$[0, 10] \sqcap [10, \infty] = [10, 10]$$

$$\text{Worklist} = \{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, \cancel{5}\}$$

Interval Domain

- The set of intervals:

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

- Partial order:

$$\perp \sqsubseteq \hat{z} \quad (\text{for any } \hat{z} \in \hat{\mathbb{Z}})$$

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

Interval Domain

- Addition / Subtraction / Multiplication:

$$[l_1, u_1] \hat{+} [l_2, u_2] =$$

$$[l_1, u_1] \hat{-} [l_2, u_2] =$$

$$[l_1, u_1] \hat{\times} [l_2, u_2] =$$

- Equality (=) produces $\top$ except for the cases:

$$[l_1, u_1] \hat{=} [l_2, u_2] = \text{true} \quad (\text{if} \quad)$$

$$[l_1, u_1] \hat{=} [l_2, u_2] = \text{false} \quad (\text{if} \quad)$$

- “Less than” (<) produces $\top$ except for the cases:

$$[l_1, u_1] \hat{<} [l_2, u_2] = \text{true} \quad (\text{if} \quad)$$

$$[l_1, u_1] \hat{<} [l_2, u_2] = \text{false} \quad (\text{if} \quad)$$

Interval Domain

- Join:

$$\begin{aligned}\perp \sqcup \hat{z} &= \hat{z} \sqcup \perp = \hat{z} \quad (\text{for any } \hat{z} \in \hat{\mathbb{Z}}) \\ [l_1, u_1] \sqcup [l_2, u_2] &= [\min(l_1, l_2), \max(u_1, u_2)]\end{aligned}$$

- Meet:

$$\begin{aligned}\perp \sqcap \hat{z} &= \hat{z} \sqcap \perp = \perp \quad (\text{for any } \hat{z} \in \hat{\mathbb{Z}}) \\ [l_1, u_1] \sqcap [l_2, u_2] &= \begin{cases} [\max(l_1, l_2), \min(u_1, u_2)] & \text{if } \max(l_1, l_2) \leq \min(u_1, u_2) \\ \perp & \text{otherwise} \end{cases}\end{aligned}$$

Abstract Memory

$$\hat{\mathbb{M}} = \mathbf{Var} \rightarrow \hat{\mathbb{Z}}$$

$$m_1 \sqsubseteq m_2 \iff \forall x \in \mathbf{Var}, m_1(x) \sqsubseteq m_2(x)$$

$$m_1 \sqcup m_2 = \lambda x. m_1(x) \sqcup m_2(x)$$

$$m_1 \sqcap m_2 = \lambda x. m_1(x) \sqcap m_2(x)$$

$$m_1 \nabla m_2 = \lambda x. m_1(x) \nabla m_2(x)$$

$$m_1 \triangle m_2 = \lambda x. m_1(x) \triangle m_2(x)$$

Question

- Give an example that the fixed point computation does not terminate.

Widening and Narrowing

- Widening:

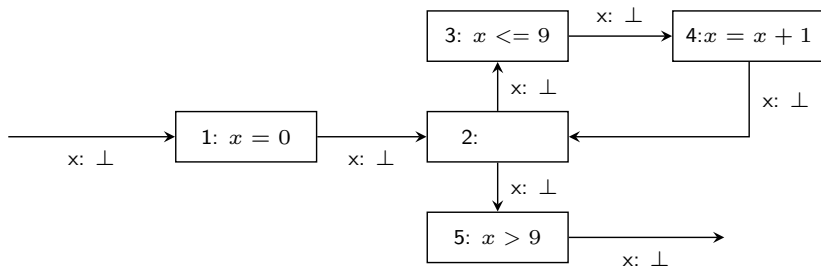
$$\perp \nabla \hat{z} = \hat{z} \nabla \perp = \hat{z} \quad (\text{for any } \hat{z} \in \hat{\mathbb{Z}})$$

$$[l_1, u_1] \nabla [l_2, u_2] = [l', u'] \text{ where}$$

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

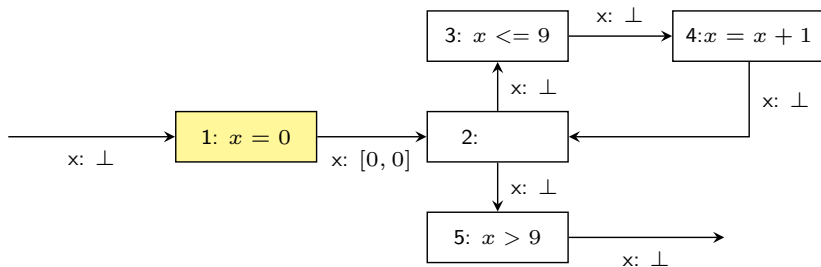
$$u' = \begin{cases} u_1 & \text{if } u_2 \leq u_1 \\ +\infty & \text{otherwise} \end{cases}$$

Fixed Point Computation with Widening



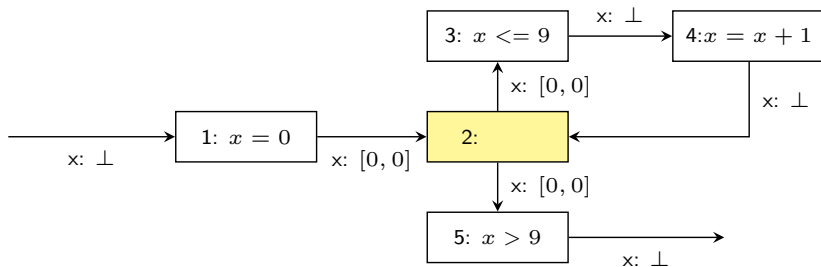
Worklist = $\{1, 2, 3, 4, 5\}$

Fixed Point Computation with Widening



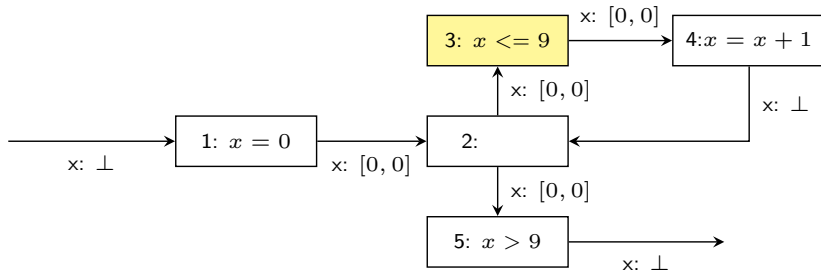
Worklist = $\{1, 2, 3, 4, 5\}$

Fixed Point Computation with Widening



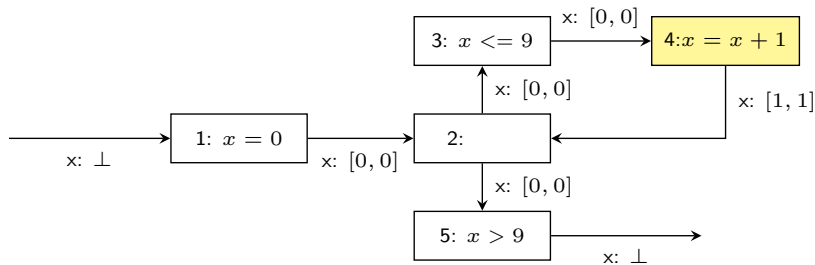
Worklist = $\{\cancel{1}, \cancel{2}, 3, 4, 5\}$

Fixed Point Computation with Widening



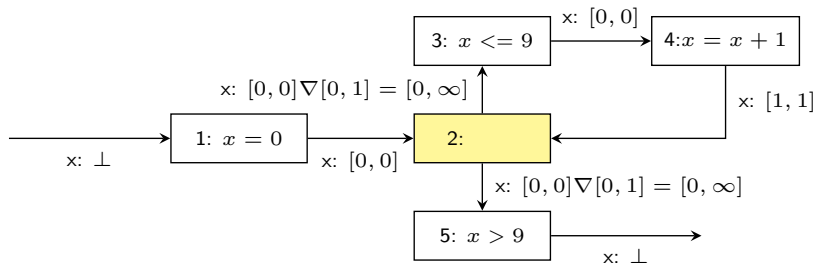
Worklist = $\{\cancel{1}, \cancel{2}, \cancel{3}, 4, 5\}$

Fixed Point Computation with Widening



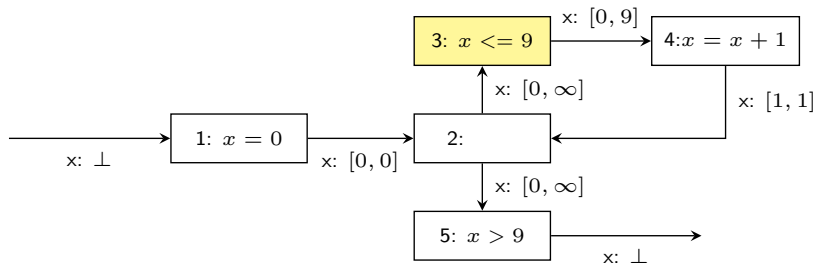
Worklist = $\{\cancel{1}, 2, \cancel{3}, \cancel{4}, 5\}$

Fixed Point Computation with Widening



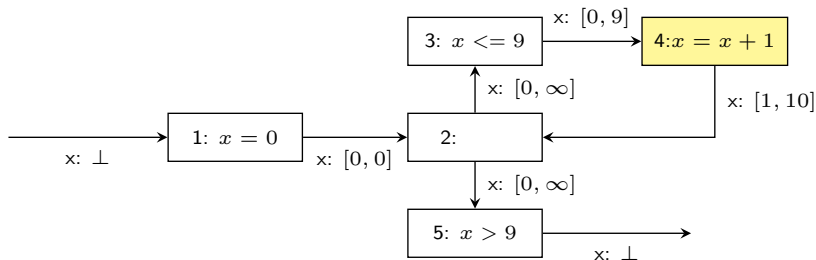
Worklist = $\{\cancel{1}, \cancel{2}, 3, \cancel{4}, 5\}$

Fixed Point Computation with Widening



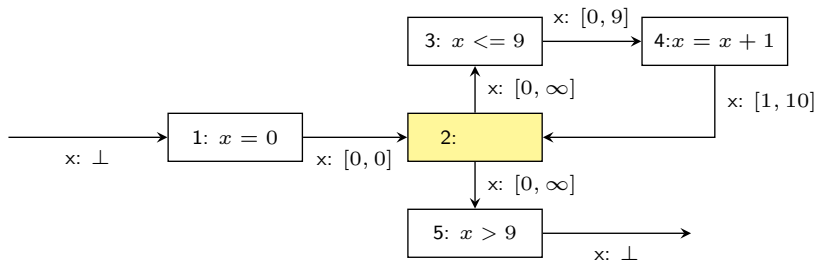
Worklist = $\{1, 2, 3, 4, 5\}$

Fixed Point Computation with Widening



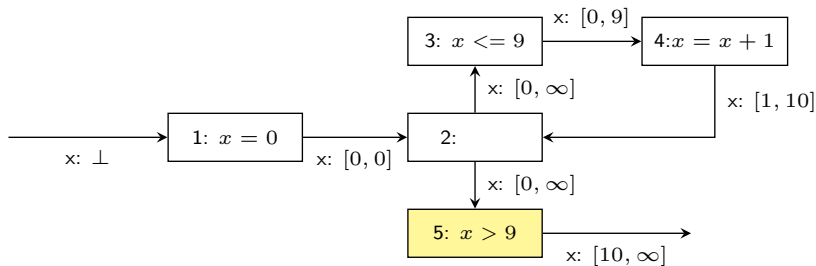
Worklist = $\{\cancel{1}, 2, \cancel{3}, \cancel{4}, 5\}$

Fixed Point Computation with Widening



Worklist = $\{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, 5\}$

Fixed Point Computation with Widening



Worklist = $\{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, \cancel{5}\}$

Widening and Narrowing

- Narrowing:

$$\perp \triangle \hat{z} = \perp \quad (\text{for any } \hat{z} \in \hat{\mathbb{Z}})$$

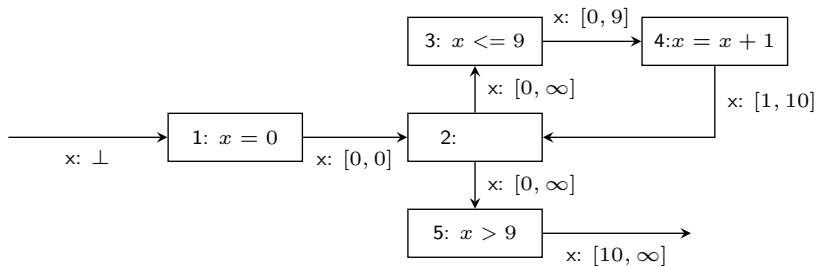
$$\hat{z} \triangle \perp = \perp \quad (\text{for any } \hat{z} \in \hat{\mathbb{Z}})$$

$$[l_1, u_1] \triangle [l_2, u_2] = [l', u'] \text{ where}$$

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

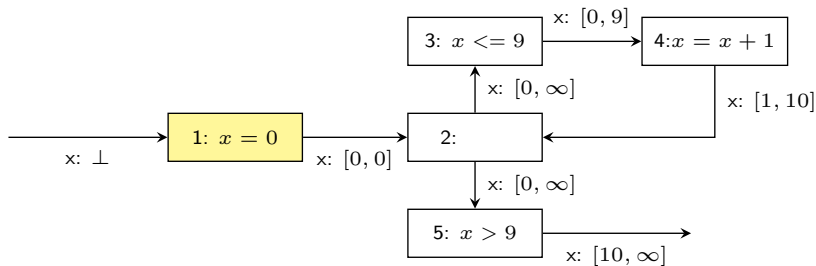
$$u' = \begin{cases} u_2 & \text{if } u_1 = +\infty \\ u_1 & \text{otherwise} \end{cases}$$

Fixed Point Computation with Narrowing



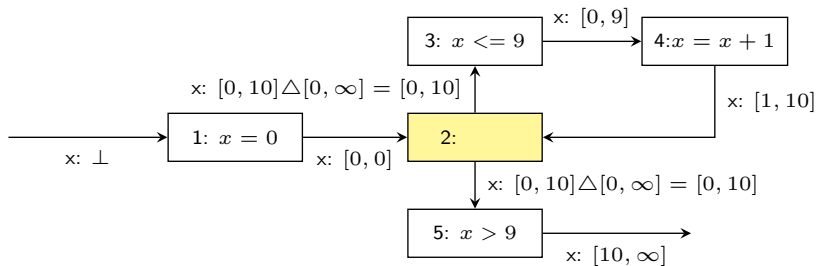
Worklist = $\{1, 2, 3, 4, 5\}$

Fixed Point Computation with Narrowing



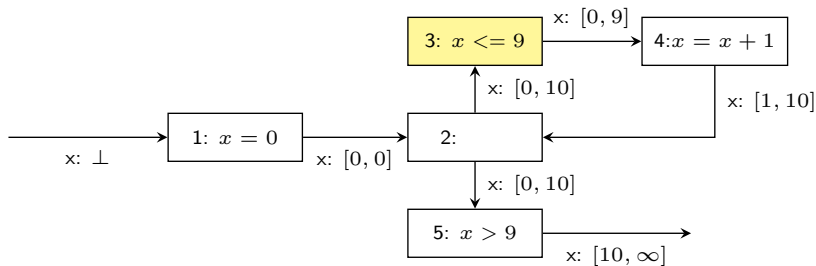
Worklist = $\{1, 2, 3, 4, 5\}$

Fixed Point Computation with Narrowing



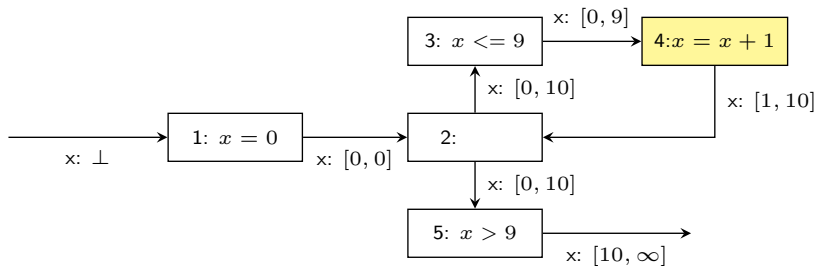
Worklist = $\{\cancel{1}, \cancel{2}, 3, 4, 5\}$

Fixed Point Computation with Narrowing



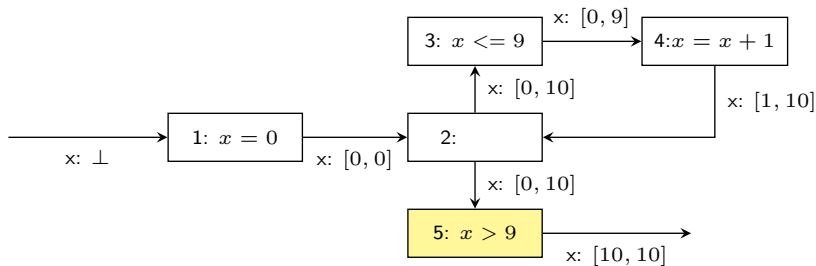
Worklist = $\{\cancel{1}, \cancel{2}, \cancel{3}, 4, 5\}$

Fixed Point Computation with Narrowing



Worklist = $\{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, 5\}$

Fixed Point Computation with Narrowing



Worklist = $\{\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, \cancel{5}\}$

Worklist Algorithm

Fixpoint comp. with Widening

$W := \mathbf{Node}$

$T := \lambda n. \perp_{\hat{\mathbb{M}}}$

while $W \neq \emptyset$

$n := \text{choose}(W)$

$W := W \setminus \{n\}$

$in := \text{inputof}(n, T)$

$out := \text{analyze}(n, in)$

 if $out \not\sqsubseteq T(n)$ then

 if widening is needed

$T(n) := T(n) \nabla out$

 else

$T(n) := T(n) \sqcup out$

$W := W \cup \text{succ}(n)$

Fixpoint comp. with narrowing

$W := \mathbf{Node}$

while $W \neq \emptyset$

$n := \text{choose}(W)$

$W := W \setminus \{n\}$

$in := \text{inputof}(n, T)$

$out := \text{analyze}(n, in)$

 if $T(n) \not\sqsubseteq out$ then

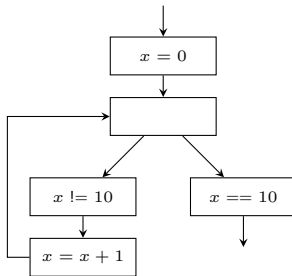
$T(n) := T(n) \triangle out$

$W := W \cup \text{succ}(n)$

Example

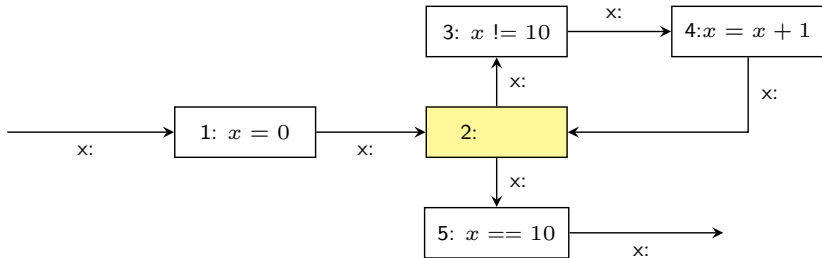
- Describe the result of the interval analysis:
 - without widening
 - with widening/narrowing

```
int main(){  
    int x = 0;  
    while (x != 10) {  
        x = x + 1;  
    }  
    assert(x == 10);  
    return 0;  
}
```



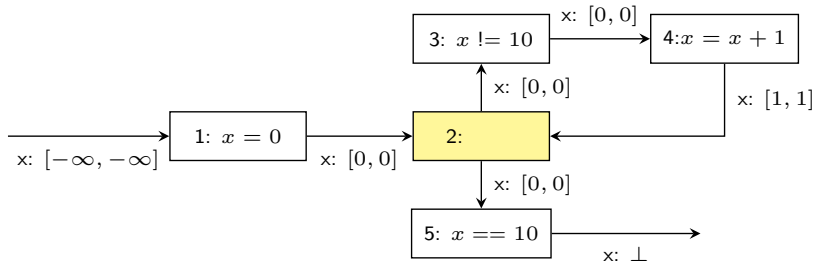
Widening with Thresholds

- Assume a set of thresholds $\theta = \{5, 10\}$ is given before the analysis



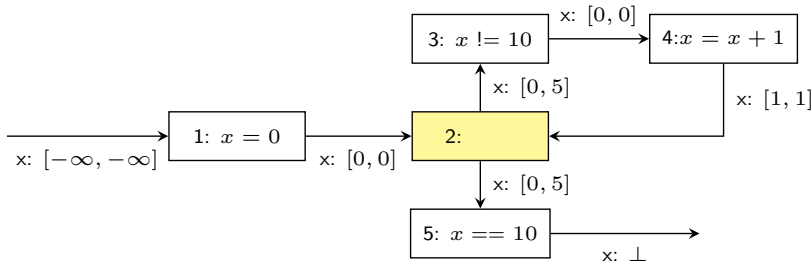
Widening with Thresholds

- Compute output by joining inputs : $[0, 0] \sqcup [1, 1] = [0, 1]$



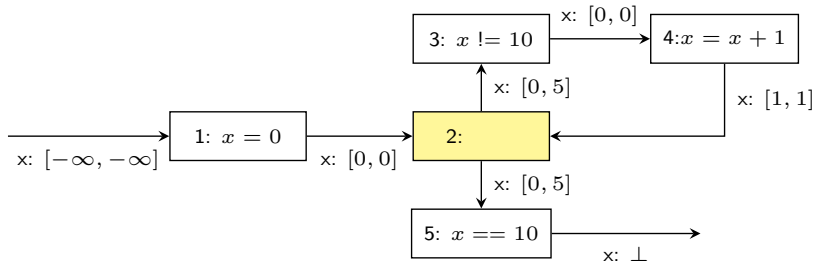
Widening with Thresholds

- Given $\theta = \{5, 10\}$, when applying widening: $[0, 0] \nabla [0, 1] = [0, 5]$

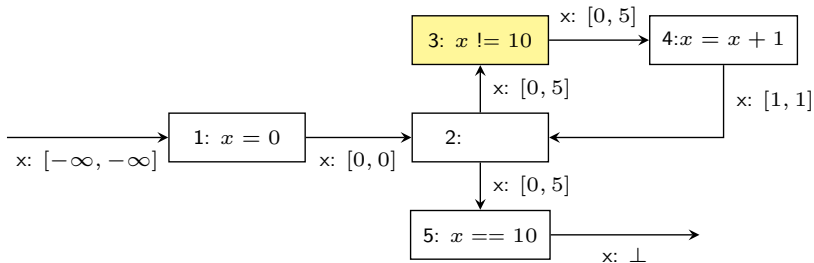


Widening with Thresholds

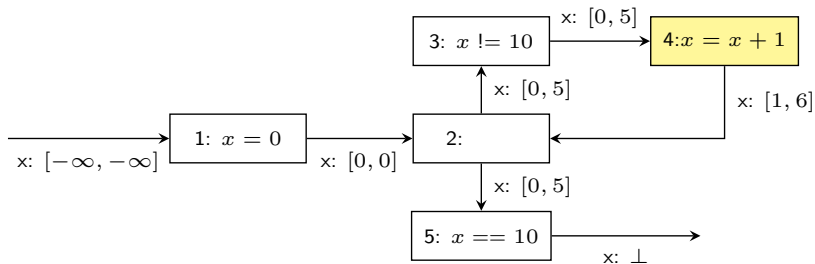
- Check fixed point is reached : $[0, 5] \not\subseteq [0, 0]$



Widening with Thresholds

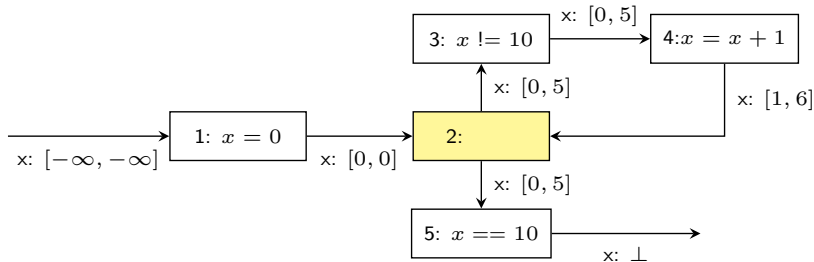


Widening with Thresholds



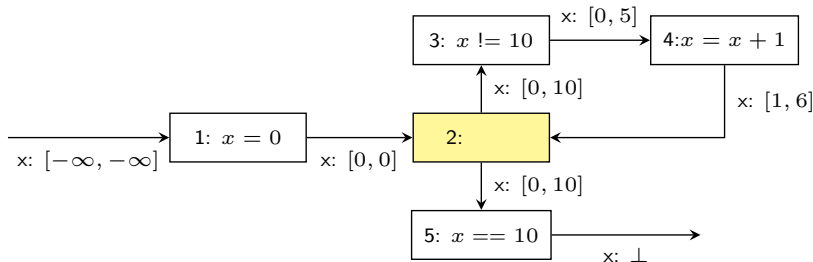
Widening with Thresholds

- Compute output by joining inputs: $[1, 6] \sqcup [0, 0] = [0, 6]$



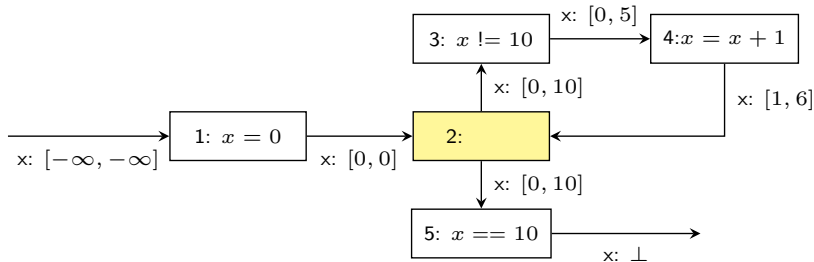
Widening with Thresholds

- Given $\text{theta} = \{5, 10\}$, use 10 as threshold when applying widening:
 $[0, 5] \nabla [0, 6] = [0, 10]$

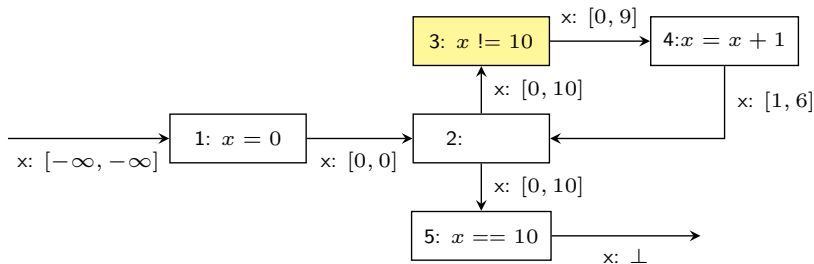


Widening with Thresholds

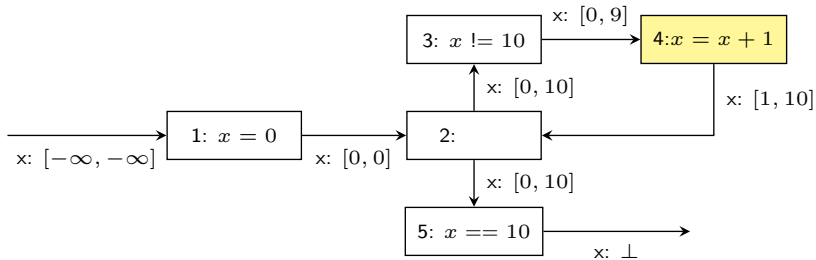
- Check if fixed point is reached: $[0, 10] \not\subseteq [0, 5]$



Widening with Thresholds



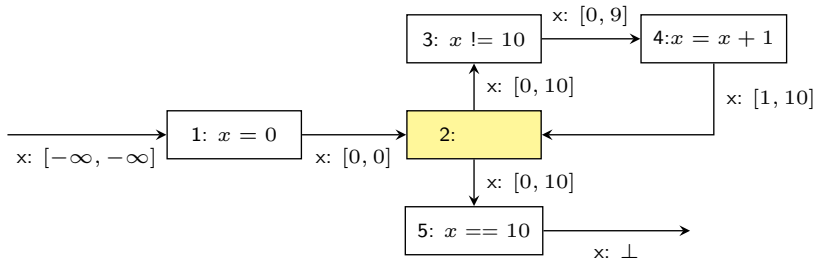
Widening with Thresholds



Widening with Thresholds

- Compute output by joining inputs:

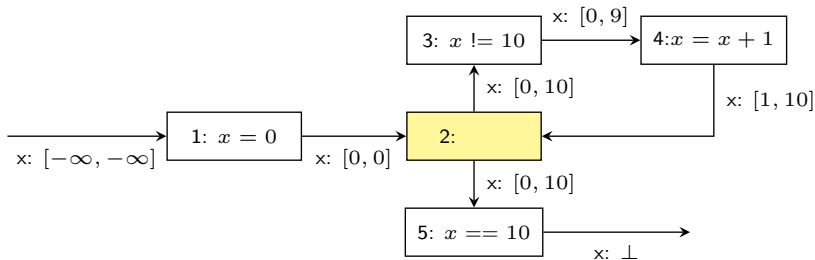
$$[0, 0] \sqcup [1, 10] = [0, 10]$$



Widening with Thresholds

- Apply widening:

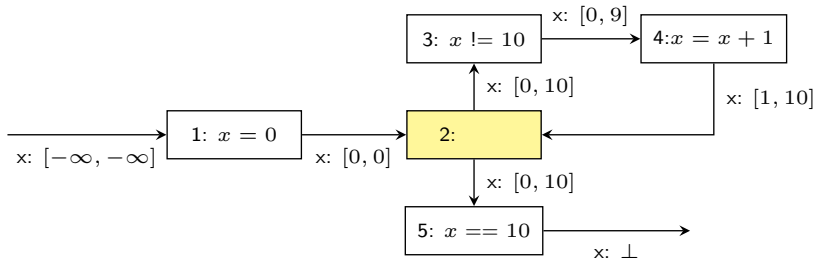
$$[0, 10] \nabla [0, 10] = [0, 10]$$



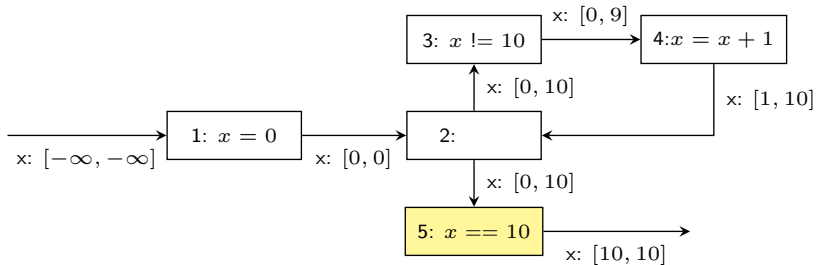
Widening with Thresholds

- Check if fixed point is reached:

$$[0, 10] \sqsubseteq [0, 10]$$



Widening with Thresholds



Widening with Thresholds

- A threshold set $\theta \subseteq \mathbb{Z}$ is given.

$$\perp \nabla_{\theta} \hat{z} = \hat{z}$$

$$\hat{z} \Delta_{\theta} \perp = \hat{z}$$

$$[l_1, u_1] \nabla_{\theta} [l_2, u_2] = [l_3, u_3] \text{ where}$$

$$l_3 = \begin{cases} glb(\theta, l_2) & \text{if } l_1 > l_2 \\ l_1 & \text{otherwise} \end{cases}$$

$$u_3 = \begin{cases} lub(\theta, u_2) & \text{if } u_1 < u_2 \\ u_1 & \text{otherwise} \end{cases}$$

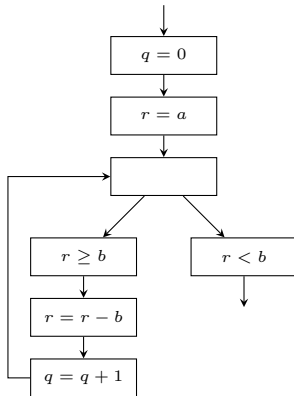
$$glb(\theta, n) = \max\{t \in \theta \mid t \leq n\}$$

$$lub(\theta, n) = \min\{t \in \theta \mid t \geq n\}$$

Exercise

- Describe the result of the interval analysis with widening and narrowing

```
//a >= 0, b >= 0
q = 0;
r = a;
while(r >= b){
    r = r - b;
    q = q + 1;
}
assert(q >= 0);
assert(r >= 0);
```



Summary

- **Interval Domain:** Abstract domain $\hat{\mathbb{Z}} = \{\perp\} \cup \{[l, u] \mid l, u \in \mathbb{Z} \cup \{-\infty, \infty\}, l \leq u\}$
 - Partial order: $\perp \sqsubseteq x$, $[l_1, u_1] \sqsubseteq [l_2, u_2]$ iff $l_2 \leq l_1$ and $u_1 \leq u_2$
 - Join: $[l_1, u_1] \sqcup [l_2, u_2] = [\min(l_1, l_2), \max(u_1, u_2)]$
 - Meet: $[l_1, u_1] \sqcap [l_2, u_2] = [\max(l_1, l_2), \min(u_1, u_2)]$ if overlap, $\perp$ otherwise
- **Worklist Algorithm:** Iterative fixed-point computation
 - Widening phase: Apply widening at loop headers until convergence
 - Narrowing phase: Apply narrowing to refine results
- **Widening & Narrowing:** Essential for termination in infinite domains
 - Widening (∇): Extrapolates to infinity to ensure convergence
 - Narrowing ($\triangle$): Refines over-approximations from widening
 - Widening with thresholds: Uses predefined values to improve precision
- **Key Insight:** Balance between precision and scalability (e.g., termination)