The set of possible states state of the program is

$$\mathcal{S} = \mathsf{Vars} \to \mathbb{Z}$$

The evaluation of an arithmetic expression e under state $\rho \in \mathcal{S}$ is denoted

$$[\![e]\!]\ \rho : \mathbb{Z}$$

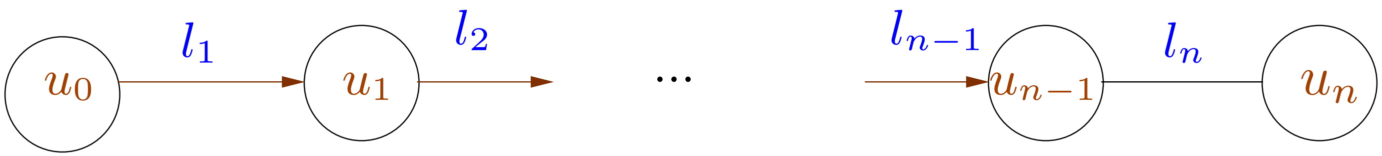
An edge $k = (u, l, v)$ induces a partial transformation on program states. The transformation depends only on the label l.

$$[\![k]\!]\ \rho = [\![l]\!]\ \rho$$

$$\text{where } [\![l]\!]\ : \mathcal{S} \to \mathcal{S}$$

$$\begin{array}{lll} [\![;]\!]\ \rho & = \rho; & \\ [\![x = e;]\!]\ \rho & = \rho \oplus \{x \mapsto [\![e]\!]\ \rho\} & \text{//i.e. } \rho \text{ modified at point } x \\ [\![e_1 \geq e_2]\!]\ \rho & = \rho & \text{if } [\![e_1]\!]\ \rho \geq [\![e_2]\!]\ \rho \end{array}$$

A path π is a sequence of consequetive edges in the CFG.



$$\pi = k_1, \ldots, k_n \text{ where each } k_i \text{ is of the form } (u_{i-1}, l_i, u_i).$$

$$\text{We write } \pi : u_0 \longrightarrow^* u_n$$

The transformation induced by a path is the composition of the transformations induced by the edges.

$$[\![\pi]\!] \; = [\![k_n]\!] \circ \ldots \circ [\![k_1]\!]$$

Each node can be reached through possibly infinitely many paths, leading to infinitely many different states at each program point.

We are interested in the set of all such states at each program point.

Suppose we know that a set V of states is possible at a node u.

By following an edge $k = (u, l, v)$, a new set of states becomes possible at node v. This set is denoted $[\![k]\!]^\sharp\ V = [\![l]\!]^\sharp\ V : 2^{\mathcal{S}} \to 2^{\mathcal{S}}$.

We define abstract transformation

$$[\![l]\!]^\sharp\ V = \{[\![l]\!]\ \rho \mid \rho \in V \text{ and } [\![l]\!] \text{ is defined for } \rho\}.$$

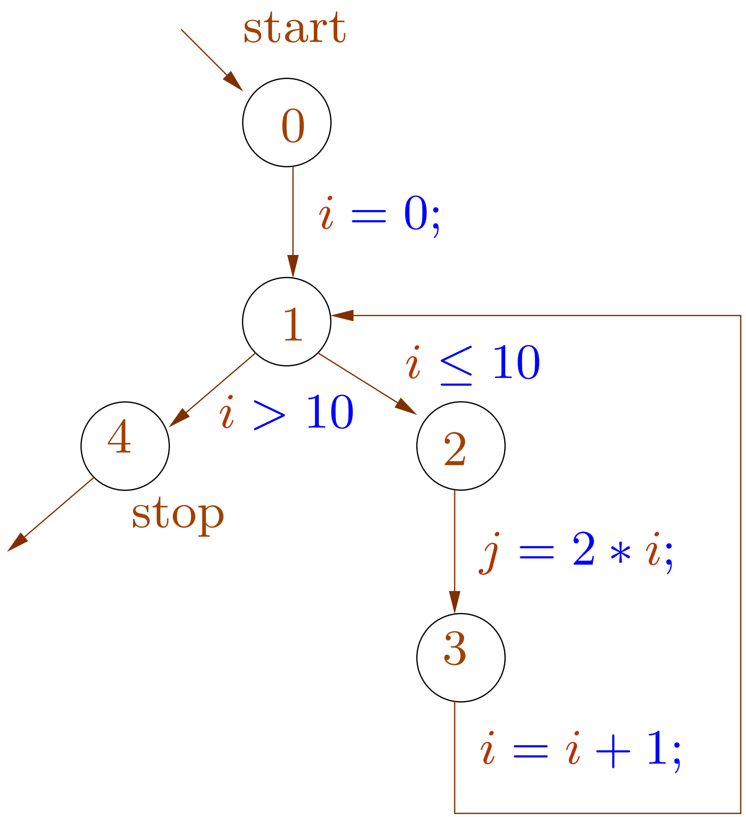
As before, $[\![k_1, \ldots, k_n]\!]^\sharp\ V = ([\![k_n]\!]^\sharp \circ \ldots \circ [\![k_1]\!]^\sharp\ )V$.

At the *start* node, all states are possible.

For each node v we want to compute the set

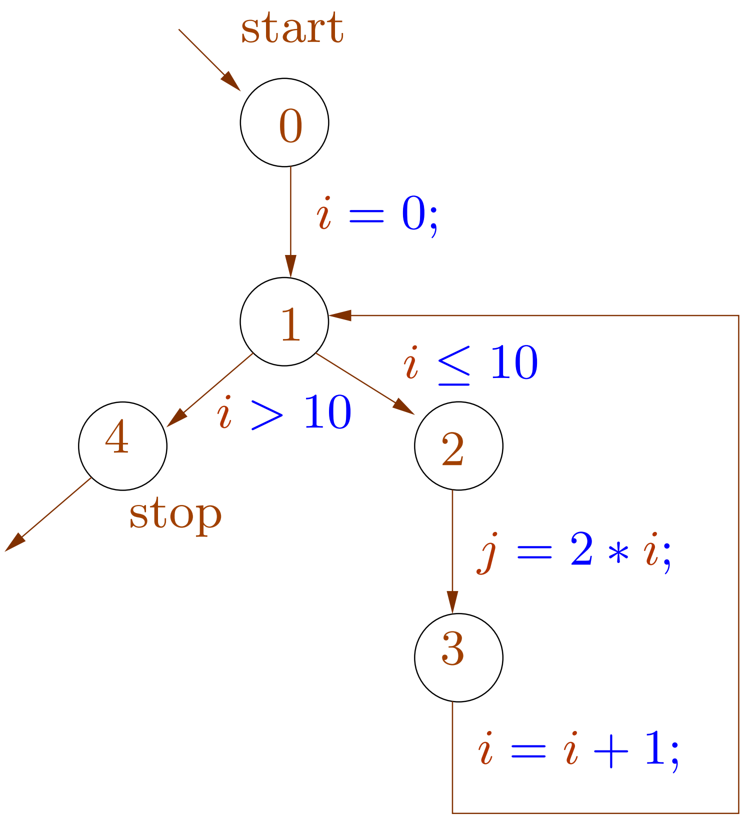
$$\mathcal{V}^*[v] = \bigcup\{[\![\pi]\!]^\sharp\ \mathcal{S} \mid \pi : \mathit{start} \to^* v\}$$

Example



u	$\mathcal{V}^*[u]$
0	$-\infty < i, j < \infty$
1	$i = 0 \wedge -\infty < j < \infty$ $\vee 1 \leq i \leq 11 \wedge j = 2i{-}2$
2	$i = 0 \wedge -\infty < j < \infty$ $\vee 1 \leq i \leq 10 \wedge j = 2i{-}2$
3	$i = 0 \wedge -\infty < j < \infty$ $\vee 1 \leq i \leq 10 \wedge j = 2i$
4	$i = 11 \wedge j = 20$

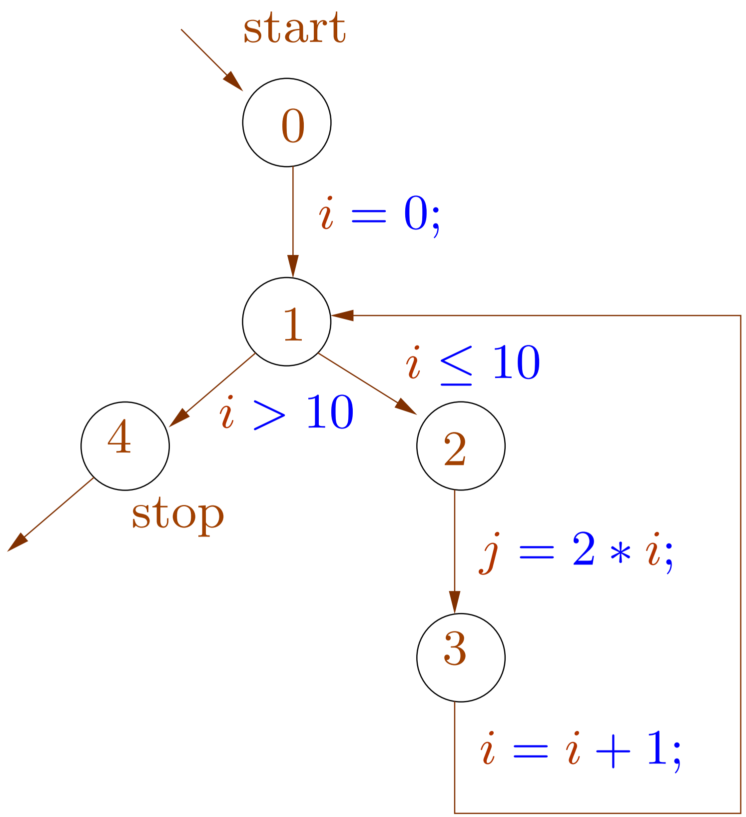
Example



u	$\mathcal{V}^*[u]$
0	$-\infty < i, j < \infty$
1	$i = 0 \land -\infty < j < \infty$ $\lor 1 \le i \le 11 \land j = 2i{-}2$
2	$i = 0 \land -\infty < j < \infty$ $\lor 1 \le i \le 10 \land j = 2i{-}2$
3	$i = 0 \land -\infty < j < \infty$ $\lor 1 \le i \le 10 \land j = 2i$
4	$i = 11 \land j = 20$

How to compute the sets $\mathcal{V}^*[v]$ in general?

Example

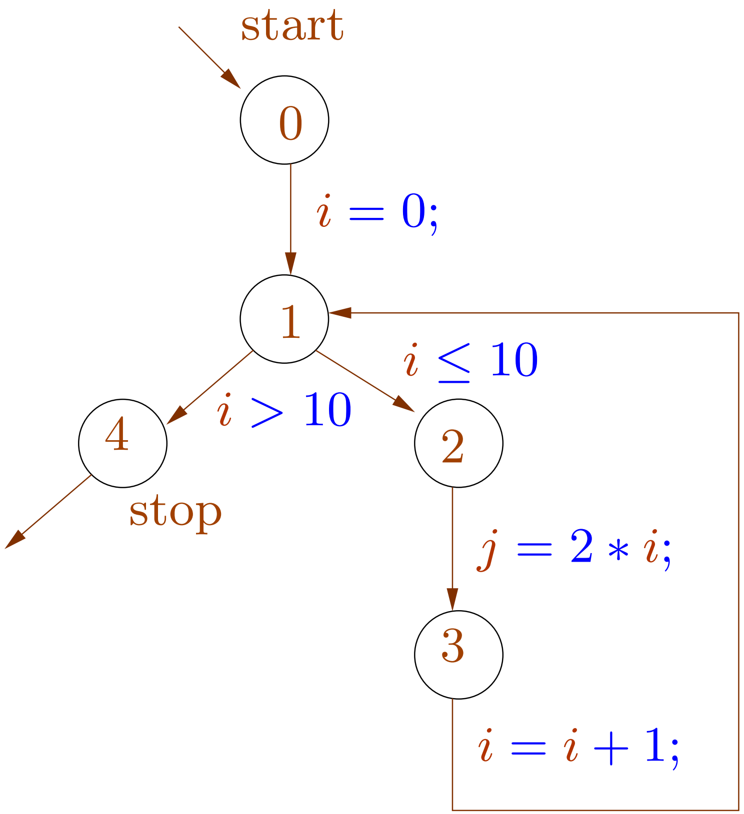


u	$\mathcal{V}^*[u]$
0	$-\infty < i, j < \infty$
1	$i = 0 \wedge -\infty < j < \infty$ $\vee 1 \leq i \leq 11 \wedge j = 2i - 2$
2	$i = 0 \wedge -\infty < j < \infty$ $\vee 1 \leq i \leq 10 \wedge j = 2i - 2$
3	$i = 0 \wedge -\infty < j < \infty$ $\vee 1 \leq i \leq 10 \wedge j = 2i$
4	$i = 11 \wedge j = 20$

How to compute the sets $\mathcal{V}^*[v]$ in general?

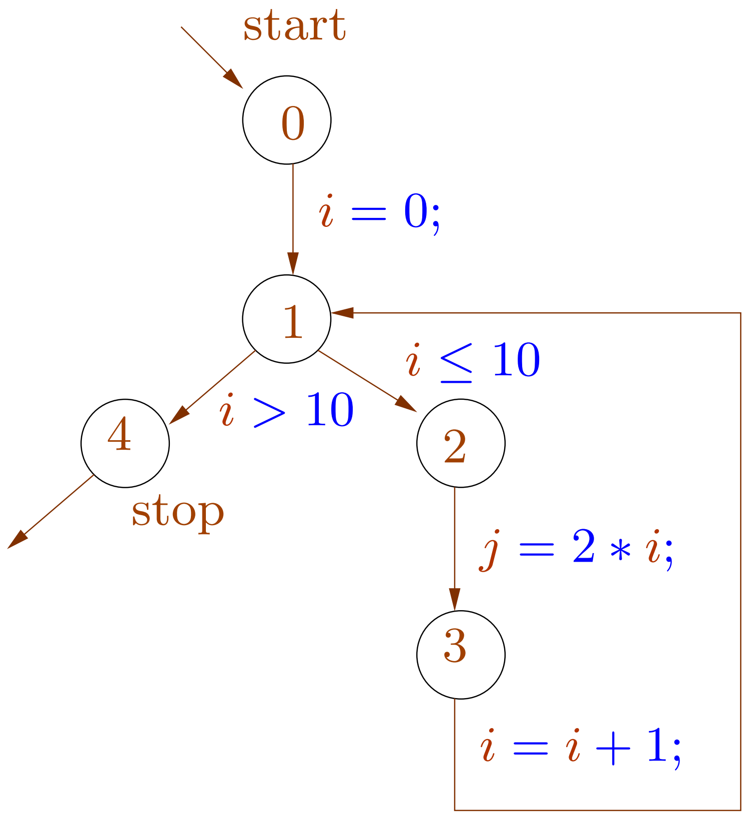
In general they are not computable!

We set up a constraint system.



$$
\begin{aligned}
\mathcal{V}[0] &\supseteq \quad \mathcal{S} \\
\mathcal{V}[1] &\supseteq \quad [\![i = 0;]\!] \; \mathcal{V}[0] \\
\mathcal{V}[1] &\supseteq \quad [\![i = i+1;]\!] \; \mathcal{V}[3] \\
\mathcal{V}[2] &\supseteq \quad [\![i \leq 10]\!] \; \mathcal{V}[1] \\
\mathcal{V}[3] &\supseteq \quad [\![j = 2*i;]\!] \; \mathcal{V}[2] \\
\mathcal{V}[4] &\supseteq \quad [\![i > 10]\!] \; \mathcal{V}[1]
\end{aligned}
$$

We set up a constraint system.



$$
\begin{aligned}
\mathcal{V}[0] &\supseteq \quad \mathcal{S} \\
\mathcal{V}[1] &\supseteq \quad [\![i = 0;]\!]\ \mathcal{V}[0] \\
\mathcal{V}[1] &\supseteq \quad [\![i = i{+}1;]\!]\ \mathcal{V}[3] \\
\mathcal{V}[2] &\supseteq \quad [\![i \leq 10]\!]\ \mathcal{V}[1] \\
\mathcal{V}[3] &\supseteq \quad [\![j = 2{*}i;]\!]\ \mathcal{V}[2] \\
\mathcal{V}[4] &\supseteq \quad [\![i > 10]\!]\ \mathcal{V}[1]
\end{aligned}
$$

The least solution (wrt $\subseteq$) of the constraints is exactly $\mathcal{V}^*$.

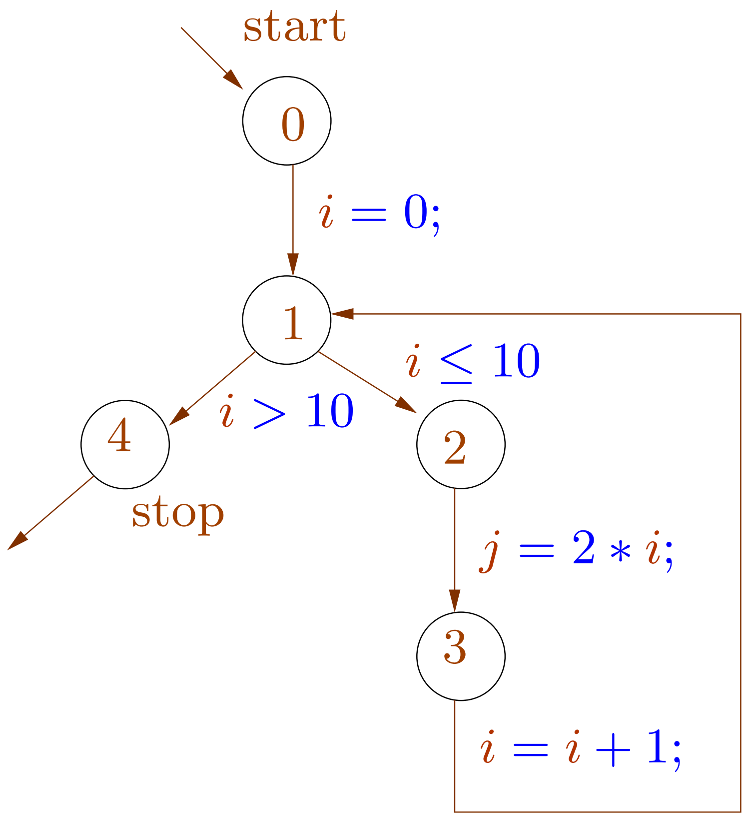
The least solution (wrt $\subseteq$) of the constraints is exactly $\mathcal{V}^*$.

Is this always true?

Does such a constraint system always have a least solution?

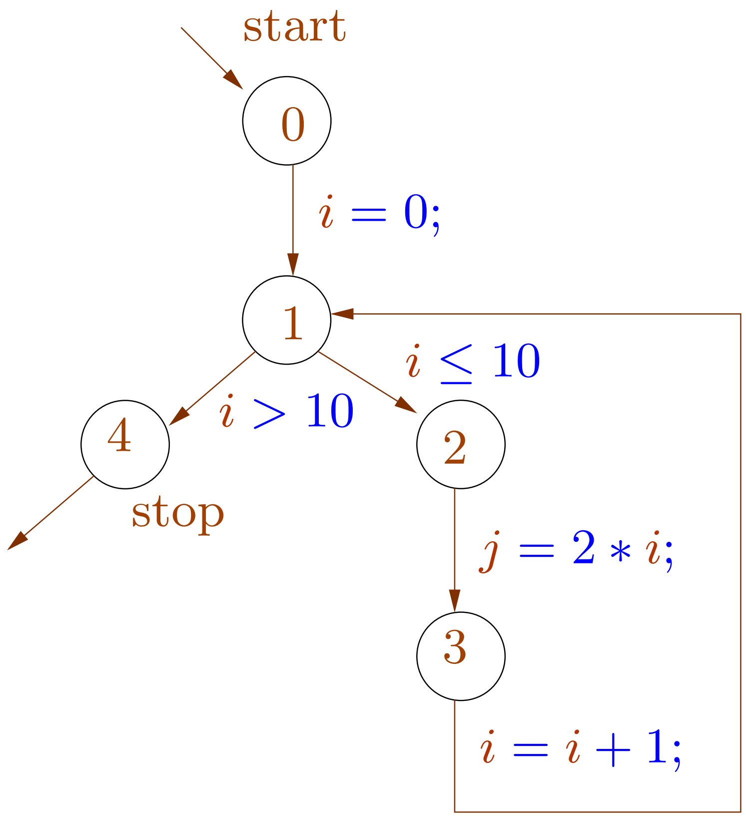
Is it computable? Efficiently?

An idea: do iterative computation of reachable states.



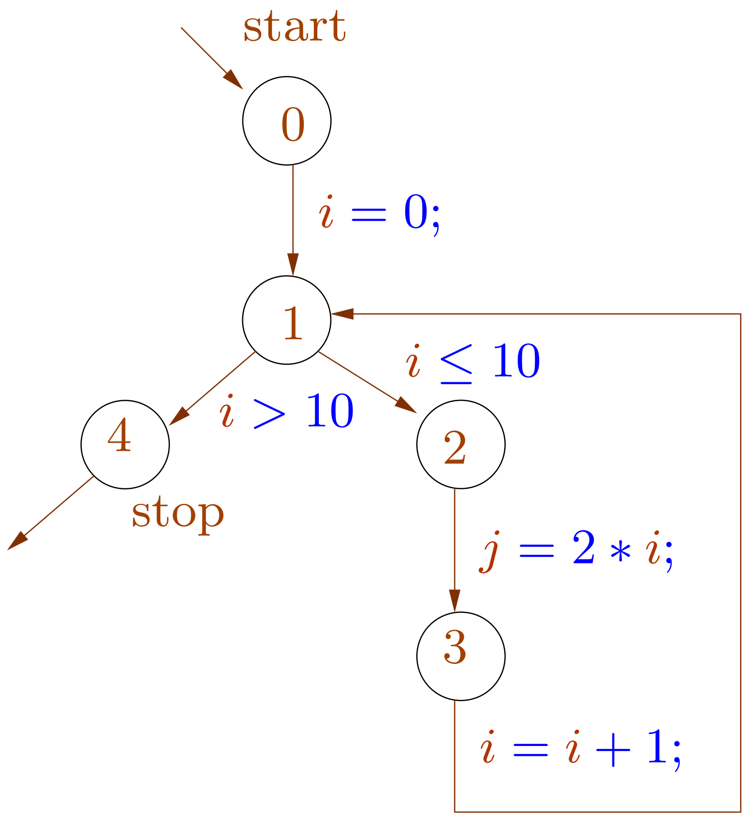
$\mathcal{V}[0]$	$\emptyset$
$\mathcal{V}[1]$	$\emptyset$
$\mathcal{V}[2]$	$\emptyset$
$\mathcal{V}[3]$	$\emptyset$
$\mathcal{V}[4]$	$\emptyset$

An idea: do iterative computation of reachable states.



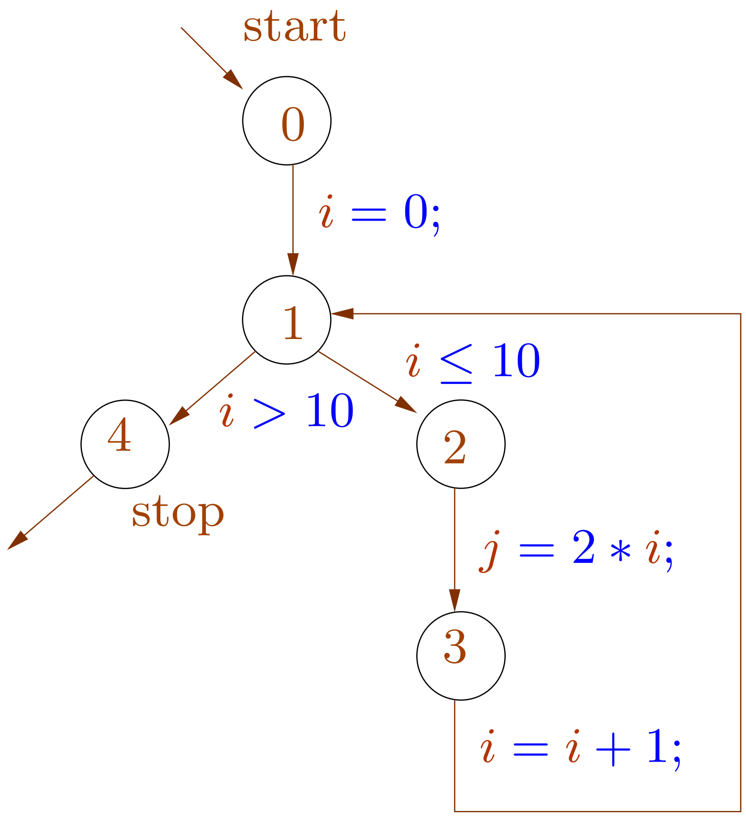
$\mathcal{V}[0]$	$\emptyset$	$\mathbb{Z} \times \mathbb{Z}$
$\mathcal{V}[1]$	$\emptyset$	
$\mathcal{V}[2]$	$\emptyset$	
$\mathcal{V}[3]$	$\emptyset$	
$\mathcal{V}[4]$	$\emptyset$	

An idea: do iterative computation of reachable states.



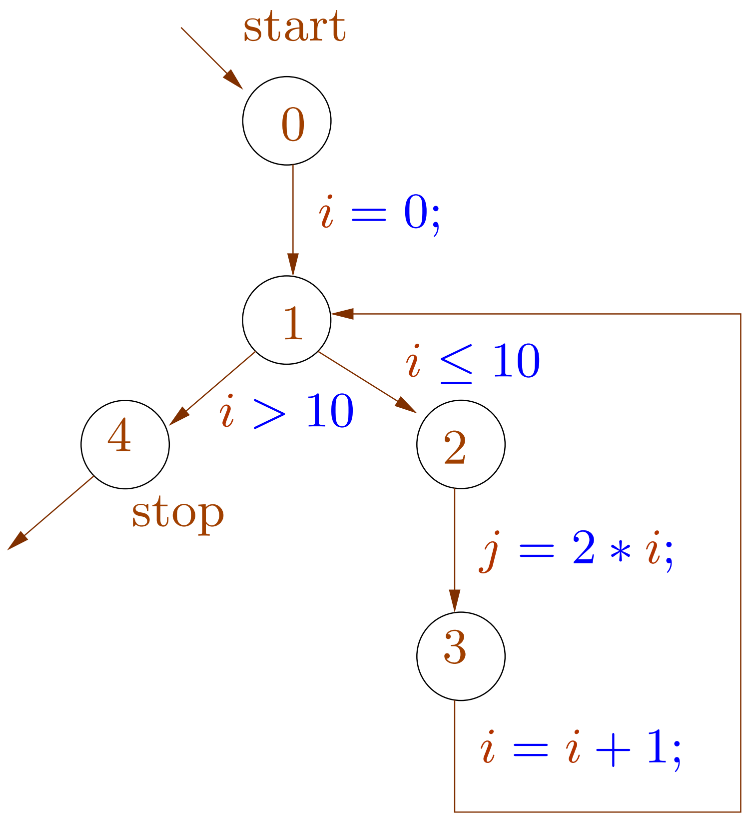
$\mathcal{V}[0]$	$\emptyset$	$\mathbb{Z} \times \mathbb{Z}$
$\mathcal{V}[1]$	$\emptyset$	$\{0\} \times \mathbb{Z}$
$\mathcal{V}[2]$	$\emptyset$	
$\mathcal{V}[3]$	$\emptyset$	
$\mathcal{V}[4]$	$\emptyset$	

An idea: do iterative computation of reachable states.



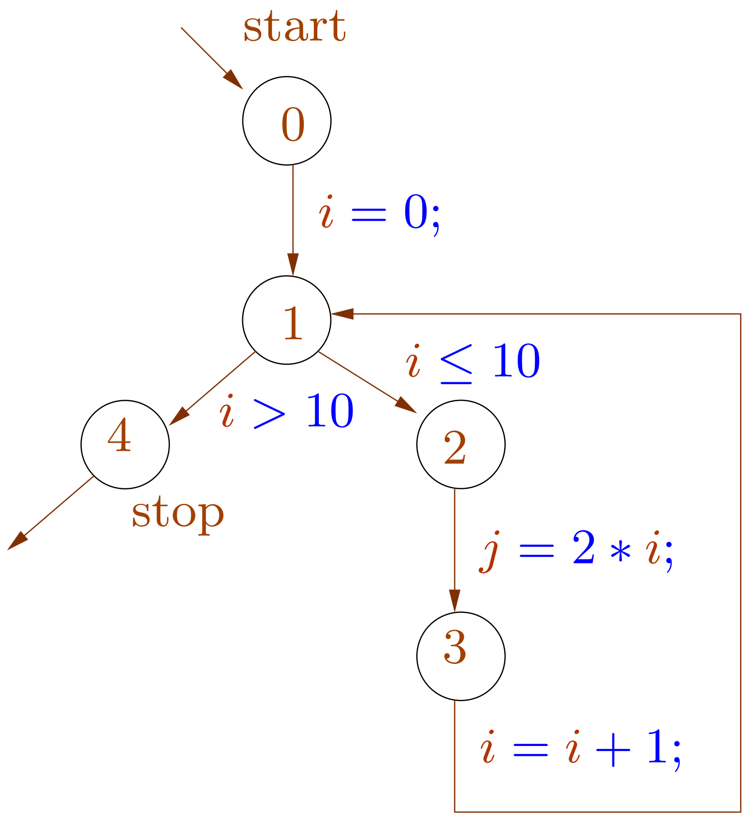
$\mathcal{V}[0]$	$\emptyset$	$\mathbb{Z} \times \mathbb{Z}$
$\mathcal{V}[1]$	$\emptyset$	$\{0\} \times \mathbb{Z}$
$\mathcal{V}[2]$	$\emptyset$	$\{0\} \times \mathbb{Z}$
$\mathcal{V}[3]$	$\emptyset$	
$\mathcal{V}[4]$	$\emptyset$	

An idea: do iterative computation of reachable states.



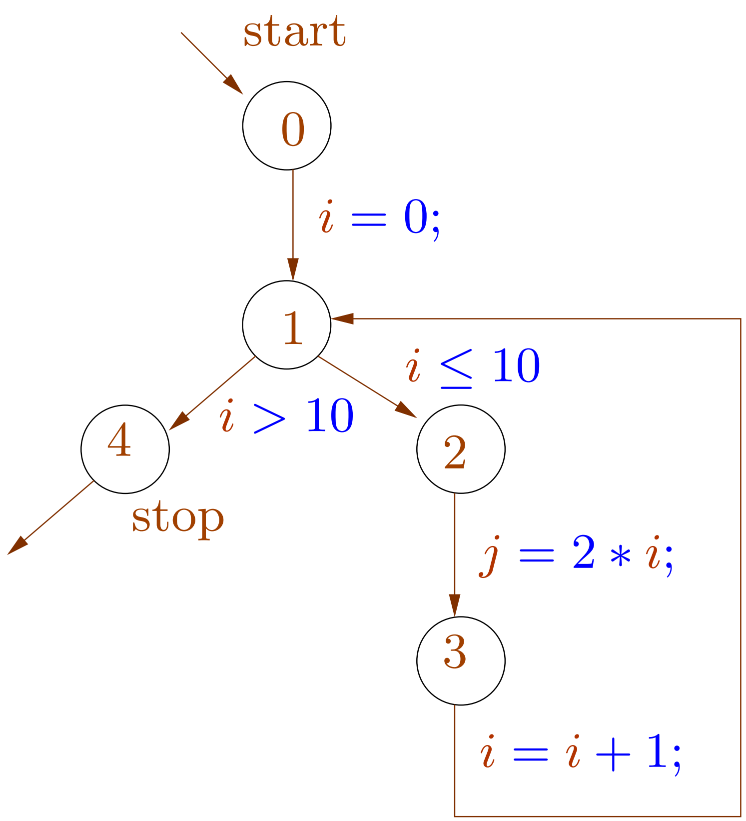
$\mathcal{V}[0]$	$\emptyset$	$\mathbb{Z} \times \mathbb{Z}$
$\mathcal{V}[1]$	$\emptyset$	$\{0\} \times \mathbb{Z}$
$\mathcal{V}[2]$	$\emptyset$	$\{0\} \times \mathbb{Z}$
$\mathcal{V}[3]$	$\emptyset$	$\{(0,0)\}$
$\mathcal{V}[4]$	$\emptyset$	

An idea: do iterative computation of reachable states.



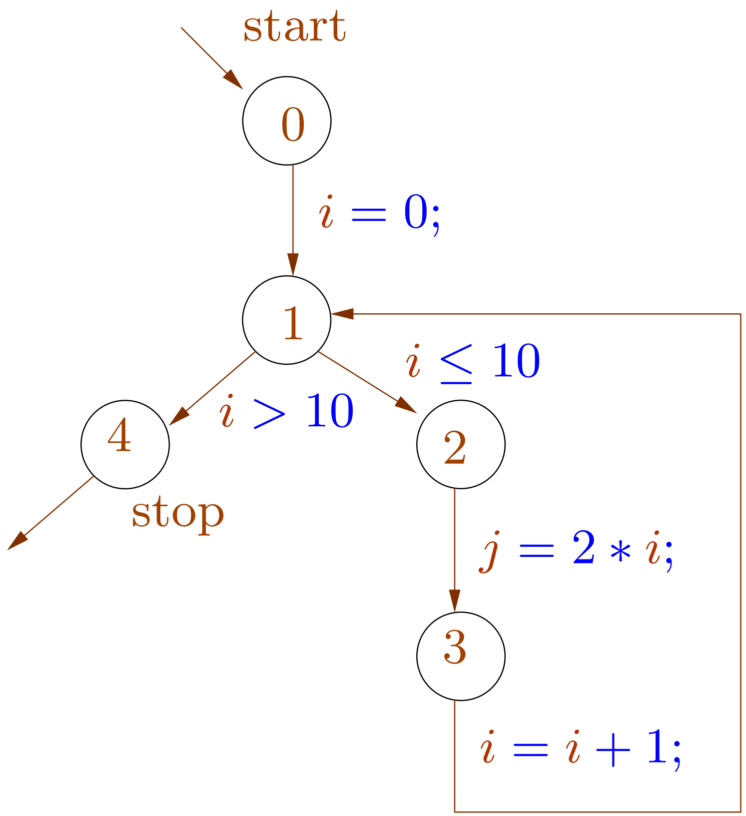
$\mathcal{V}[0]$	$\emptyset$	$\mathbb{Z} \times \mathbb{Z}$	
$\mathcal{V}[1]$	$\emptyset$	$\{0\} \times \mathbb{Z}$	$\{0\} \times \mathbb{Z} \cup \{(1, 0)\}$
$\mathcal{V}[2]$	$\emptyset$	$\{0\} \times \mathbb{Z}$	
$\mathcal{V}[3]$	$\emptyset$	$\{(0, 0)\}$	
$\mathcal{V}[4]$	$\emptyset$		

An idea: do iterative computation of reachable states.



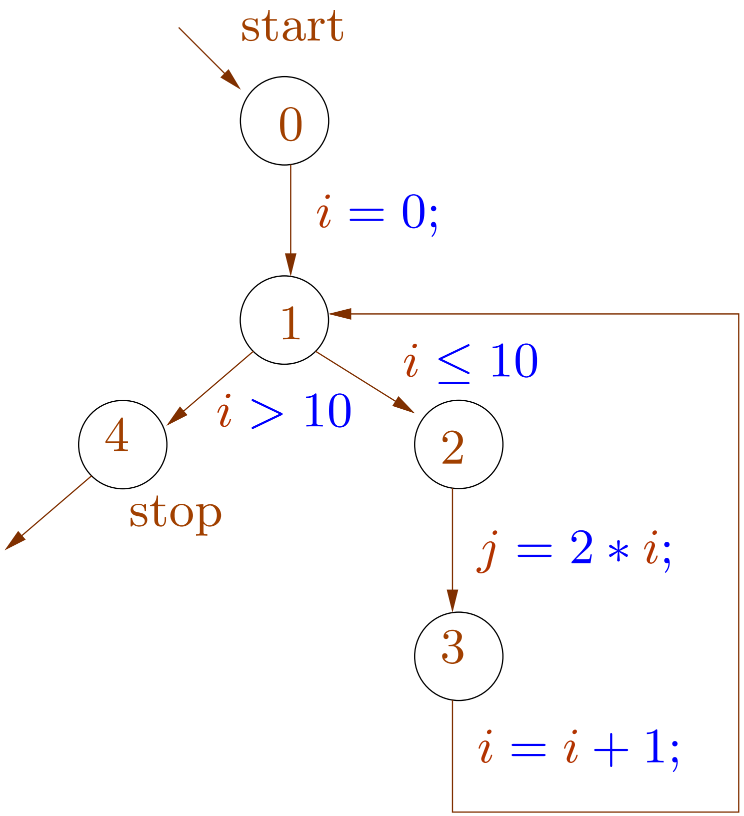
$\mathcal{V}[0]$	$\emptyset$	$\mathbb{Z} \times \mathbb{Z}$	
$\mathcal{V}[1]$	$\emptyset$	$\{0\} \times \mathbb{Z}$	$\{0\} \times \mathbb{Z} \cup \{(1,0)\}$
$\mathcal{V}[2]$	$\emptyset$	$\{0\} \times \mathbb{Z}$	$\{0\} \times \mathbb{Z} \cup \{(1,0)\}$
$\mathcal{V}[3]$	$\emptyset$	$\{(0,0)\}$	
$\mathcal{V}[4]$	$\emptyset$		

An idea: do iterative computation of reachable states.



$\mathcal{V}[0]$	$\emptyset$	$\mathbb{Z} \times \mathbb{Z}$	
$\mathcal{V}[1]$	$\emptyset$	$\{0\} \times \mathbb{Z}$	$\{0\} \times \mathbb{Z} \cup \{(1,0)\}$
$\mathcal{V}[2]$	$\emptyset$	$\{0\} \times \mathbb{Z}$	$\{0\} \times \mathbb{Z} \cup \{(1,0)\}$
$\mathcal{V}[3]$	$\emptyset$	$\{(0,0)\}$	$\{(0,0),(1,2)\}$
$\mathcal{V}[4]$	$\emptyset$		

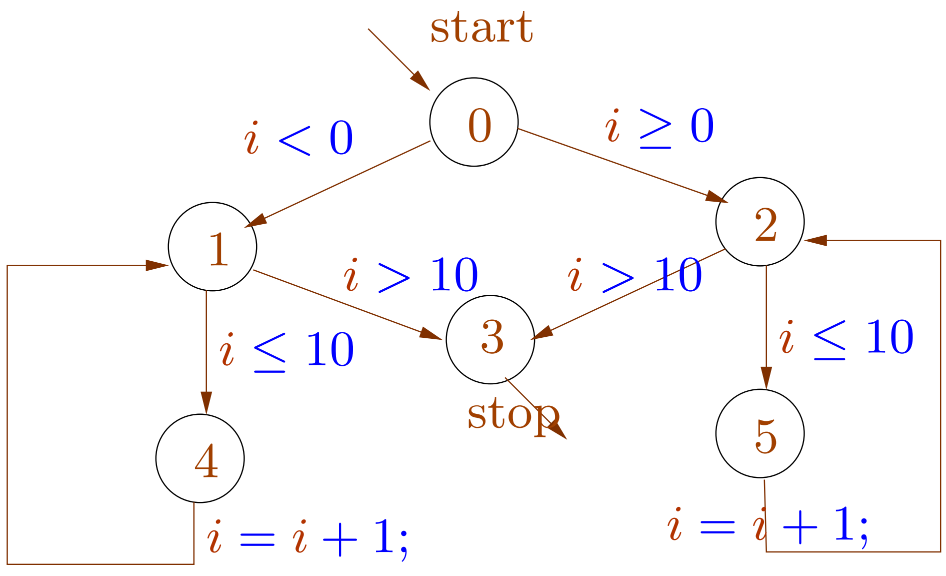
An idea: do iterative computation of reachable states.



$\mathcal{V}[0]$	$\emptyset$	$\mathbb{Z} \times \mathbb{Z}$		
$\mathcal{V}[1]$	$\emptyset$	$\{0\} \times \mathbb{Z}$	$\{0\} \times \mathbb{Z} \cup \{(1, 0)\}$	
$\mathcal{V}[2]$	$\emptyset$	$\{0\} \times \mathbb{Z}$	$\{0\} \times \mathbb{Z} \cup \{(1, 0)\}$	$\ldots$
$\mathcal{V}[3]$	$\emptyset$	$\{(0, 0)\}$	$\{(0, 0), (1, 2)\}$	
$\mathcal{V}[4]$	$\emptyset$			

Problem: too many iterations, infinite loops.

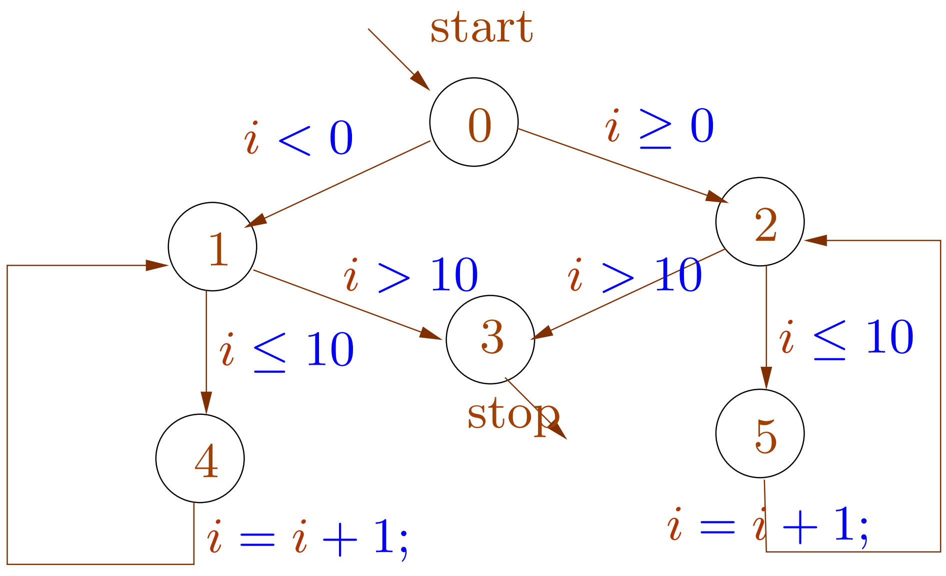
Solution: approximate computation of possible states.



0	$\emptyset$	$\mathbb{Z}$	$\mathbb{Z}$
1	$\emptyset$	$\mathbb{Z}^-$	$\mathbb{Z}$
2	$\emptyset$	$\mathbb{Z}^+$	$\mathbb{Z}^+$
3	$\emptyset$	$\mathbb{Z}^+$	$\mathbb{Z}^+$
4	$\emptyset$	$\mathbb{Z}^-$	$\mathbb{Z}$
5	$\emptyset$	$\mathbb{Z}^+$	$\mathbb{Z}^+$

Problem: too many iterations, infinite loops.

Solution: approximate computation of possible states.



0	$\emptyset$	$\mathbb{Z}$	$\mathbb{Z}$
1	$\emptyset$	$\mathbb{Z}^-$	$\mathbb{Z}$
2	$\emptyset$	$\mathbb{Z}^+$	$\mathbb{Z}^+$
3	$\emptyset$	$\mathbb{Z}^+$	$\mathbb{Z}^+$
4	$\emptyset$	$\mathbb{Z}^-$	$\mathbb{Z}$
5	$\emptyset$	$\mathbb{Z}^+$	$\mathbb{Z}^+$

Interpretation of our result:

the values of i at node 1 is included in $\mathbb{Z}$

the values of i at node 2 is included in $\mathbb{Z}^+$

This information we obtain is accurate.

In general we have some domain $\mathbb{D}$.

Examples: $2^{\mathcal{S}}$, $2^{\mathbb{Z}}$, $\{\emptyset, \mathbb{Z}^-, \mathbb{Z}^+, \mathbb{Z}\}$, the set of intervals over $\mathbb{Z}$.

In general we have some domain $\mathbb{D}$.

Examples: $2^{\mathcal{S}}$, $2^{\mathbb{Z}}$, $\{\emptyset, \mathbb{Z}^-, \mathbb{Z}^+, \mathbb{Z}\}$, the set of intervals over $\mathbb{Z}$.

We require an ordering $\sqsubseteq$ on the elements of this domain.

$\emptyset \sqsubseteq \mathbb{Z}^- \qquad \emptyset \sqsubseteq \mathbb{Z}^+ \qquad \mathbb{Z}^- \sqsubseteq \mathbb{Z} \qquad \mathbb{Z}^+ \sqsubseteq \mathbb{Z}$

Read $x \sqsubseteq y$ as "y is imprecise information compared to x".

In general we have some domain $\mathbb{D}$.

Examples: $2^{\mathcal{S}}$, $2^{\mathbb{Z}}$, $\{\emptyset, \mathbb{Z}^-, \mathbb{Z}^+, \mathbb{Z}\}$, the set of intervals over $\mathbb{Z}$.

We require an ordering $\sqsubseteq$ on the elements of this domain.

$\emptyset \sqsubseteq \mathbb{Z}^-$ $\quad$ $\emptyset \sqsubseteq \mathbb{Z}^+$ $\quad$ $\mathbb{Z}^- \sqsubseteq \mathbb{Z}$ $\quad$ $\mathbb{Z}^+ \sqsubseteq \mathbb{Z}$

Read $x \sqsubseteq y$ as "y is imprecise information compared to x".

We further require operations like least upper bounds.

$$\mathbb{Z}^- \sqcup \mathbb{Z}^+ = \mathbb{Z}$$

A brief discussion of complete lattices

Recall: a set $\mathbb{D}$ with relation $\sqsubseteq$ is a partial order if the following conditions hold for all $x, y, z \in \mathbb{D}$.

- Reflexivity: $x \sqsubseteq x$.
- Antisymmetry: $x \sqsubseteq y$ and $y \sqsubseteq x$ then $x = y$.
- Transitivity: if $x \sqsubseteq y$ and $y \sqsubseteq z$ then $x \sqsubseteq z$.

An element $d \in \mathbb{D}$ is called an upper bound of a set $X \subseteq \mathbb{D}$ if $x \sqsubseteq d$ for all $x \in X$.

$d \in \mathbb{D}$ is called least upper bound of $X \subseteq \mathbb{D}$ if

- d is an upper bound of X
- $d \sqsubseteq d'$ for every upper bound d' of X

An element $d \in \mathbb{D}$ is called an upper bound of a set $X \subseteq \mathbb{D}$ if $x \sqsubseteq d$ for all $x \in X$.

$d \in \mathbb{D}$ is called least upper bound of $X \subseteq \mathbb{D}$ if

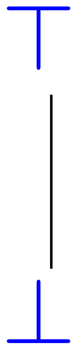
- d is an upper bound of X
- $d \sqsubseteq d'$ for every upper bound d' of X

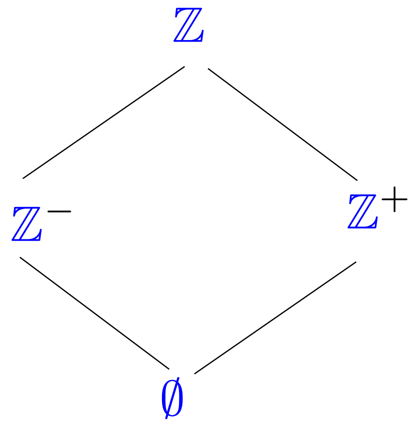
A partial order $(\mathbb{D}, \sqsubseteq)$ is called a complete lattice if every $X \subseteq \mathbb{D}$ has a least upper bound $\bigsqcup X$.

We write $x \sqcup y$ for $\bigsqcup\{x, y\}$.

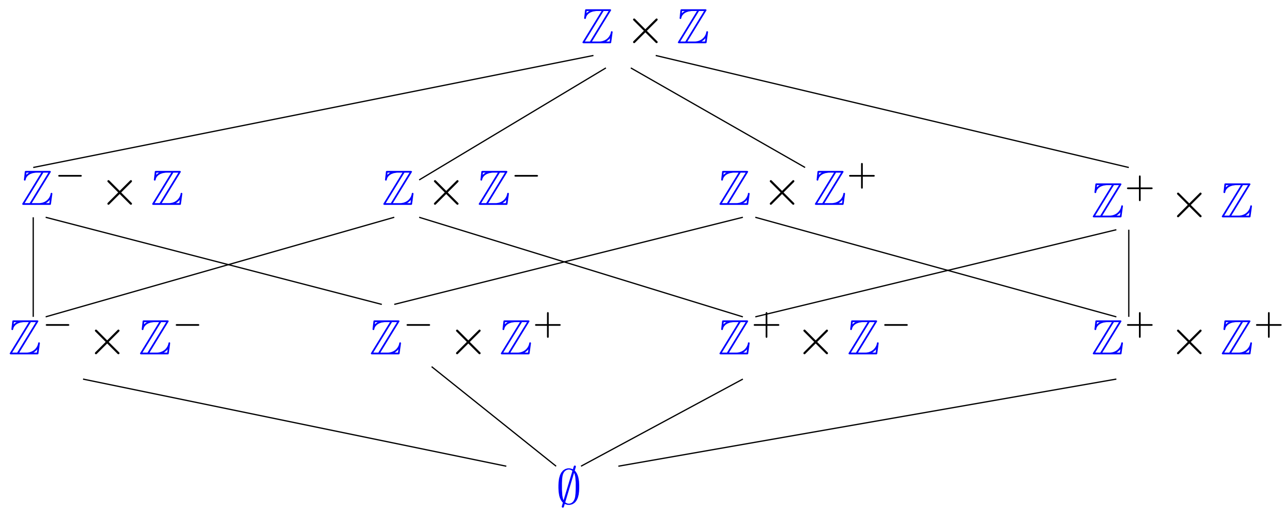
For $(2^{\mathcal{S}}, \subseteq)$ we have $\bigsqcup X = \bigcup X$.

Some complete lattices.

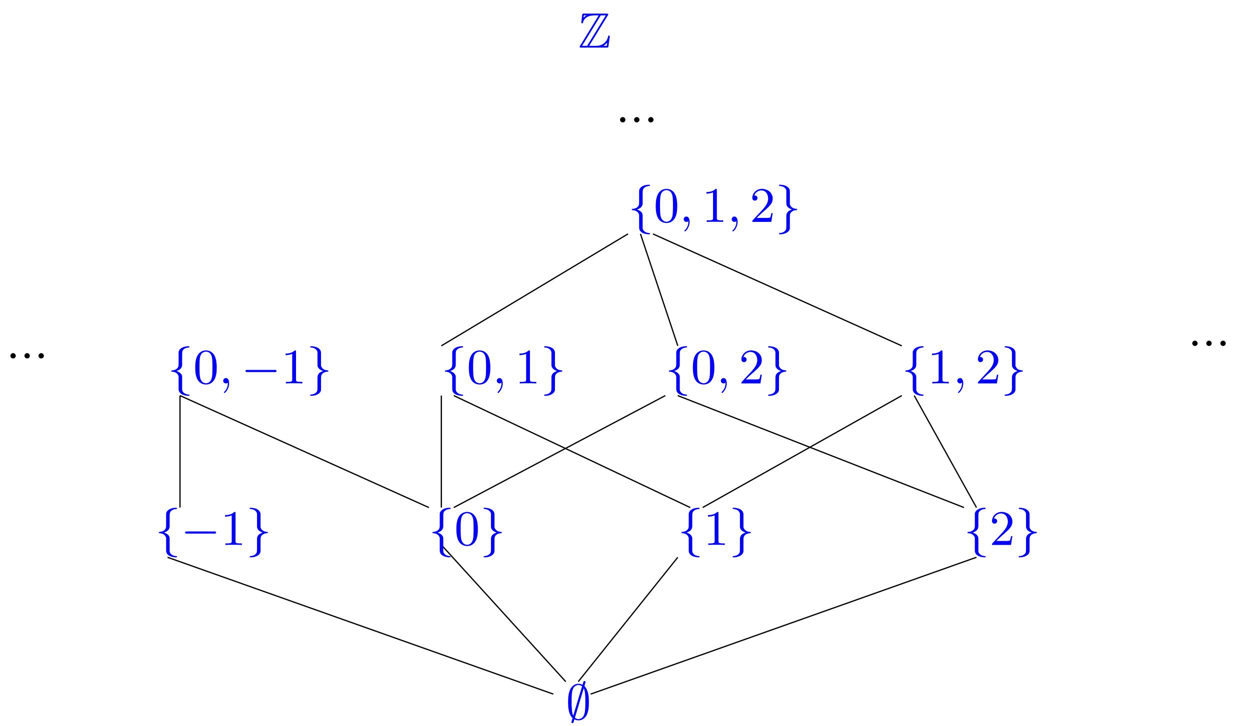




$$\mathbb{Z}^- = \{x \in \mathbb{Z} \mid x < 0\}$$
$$\mathbb{Z}^+ = \{x \in \mathbb{Z} \mid x \geq 0\}$$



An infinite complete lattice : $(2^{\mathbb{Z}}, \subseteq)$.



Every complete lattice has

- a top element: $\top = \bigsqcup \mathbb{D}$
- a bottom element: $\bot = \bigsqcup \emptyset$

Further every $X \subseteq \mathbb{D}$ has a greatest lower bound $\bigsqcap X$.

Every complete lattice has

- a top element: $\top = \bigsqcup \mathbb{D}$
- a bottom element: $\bot = \bigsqcup \emptyset$

Further every $X \subseteq \mathbb{D}$ has a greatest lower bound $\bigsqcap X$.

Proof: exercise.

A function $f : \mathbb{D}_1 \longrightarrow \mathbb{D}_2$ is called monotone if:

$$f(x) \sqsubseteq f(y) \text{ whenever } x \sqsubseteq y$$

A function $f : \mathbb{D}_1 \longrightarrow \mathbb{D}_2$ is called monotone if:

$$f(x) \sqsubseteq f(y) \text{ whenever } x \sqsubseteq y$$

The function $f : \mathbb{Z} \to \mathbb{Z}$ defined as $f(x) = x + 1$ is monotone.

Note: $(\mathbb{Z}, \leq)$ is not a complete lattice.

A function $f : \mathbb{D}_1 \to \mathbb{D}_2$ is called monotone if:

$$f(x) \sqsubseteq f(y) \text{ whenever } x \sqsubseteq y$$

The function $f : \mathbb{Z} \to \mathbb{Z}$ defined as $f(x) = x + 1$ is monotone.

Note: $(\mathbb{Z}, \leq)$ is not a complete lattice.

The transformations induced by the program edges are monotone:

$$\text{Recall: } [\![l]\!]^\sharp \; : 2^{\mathcal{S}} \to 2^{\mathcal{S}}$$

$$[\![l]\!]^\sharp \; V = \{[\![l]\!] \; \rho \mid \rho \in V \text{ and } [\![l]\!] \; \text{ is defined for } \rho\}.$$

$$\text{Hence if } V_1 \subseteq V_2 \text{ then } [\![l]\!]^\sharp \; V_1 \subseteq [\![l]\!]^\sharp \; V_2.$$

Some facts:

If $f : \mathbb{D}_1 \longrightarrow \mathbb{D}_2$ and $g : \mathbb{D}_2 \longrightarrow \mathbb{D}_3$ are monotone then the composition $g \circ f : \mathbb{D}_1 \longrightarrow \mathbb{D}_3$ is monotone.

Some facts:

If $f : \mathbb{D}_1 \longrightarrow \mathbb{D}_2$ and $g : \mathbb{D}_2 \longrightarrow \mathbb{D}_3$ are monotone then the composition $g \circ f : \mathbb{D}_1 \longrightarrow \mathbb{D}_3$ is monotone.

If $\mathbb{D}_2$ is a complete lattice then the set $[\mathbb{D}_1 \rightarrow \mathbb{D}_2]$ of monotone functions $f : \mathbb{D}_1 \longrightarrow \mathbb{D}_2$ is a complete lattice,

where $f \sqsubseteq g$ iff $f(x) \sqsubseteq g(x)$ for all $x \in \mathbb{D}_1$.

For $F \subseteq [\mathbb{D}_1 \rightarrow \mathbb{D}_2]$ we have

$$\bigsqcup F = f \text{ with } f(x) = \bigsqcup \{g(x) \mid g \in F\} \qquad .$$

For our program analysis problem, we want the least solution of the constraint system

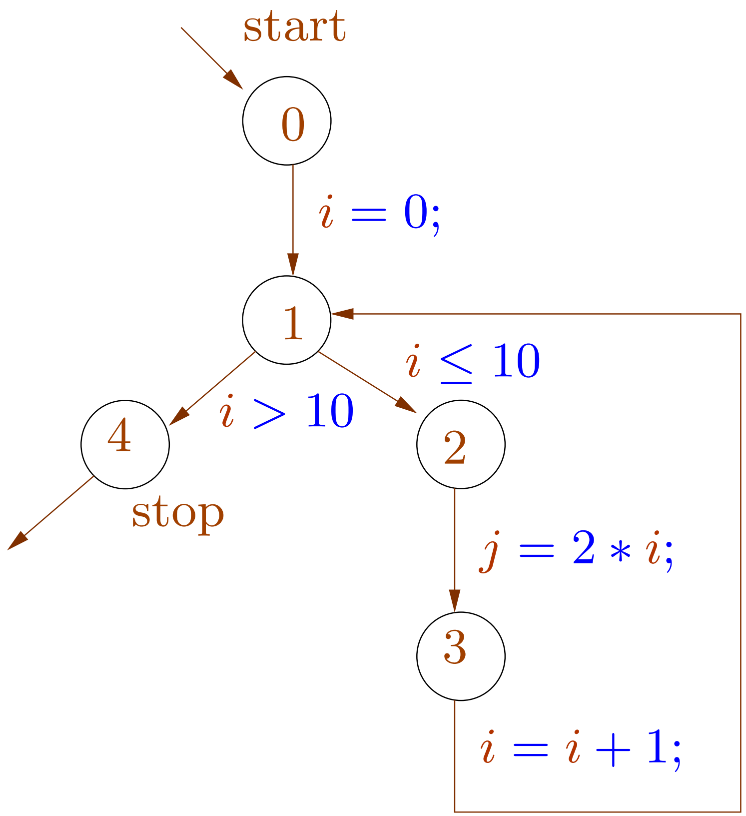
$$\begin{array}{lll} \mathcal{V}[0] \supseteq \mathcal{S} & & (0 \text{ is the } \textit{start} \text{ node}) \\ \mathcal{V}[v] \supseteq [\![l]\!]^\sharp\ \mathcal{V}[u] & & \text{for every edge } (u, l, v). \end{array}$$

We have the domain $\mathbb{D} = 2^{\mathcal{S}}$. Choose a variable for each set $\mathcal{V}[v]$.

We obtain a constraint system of the form

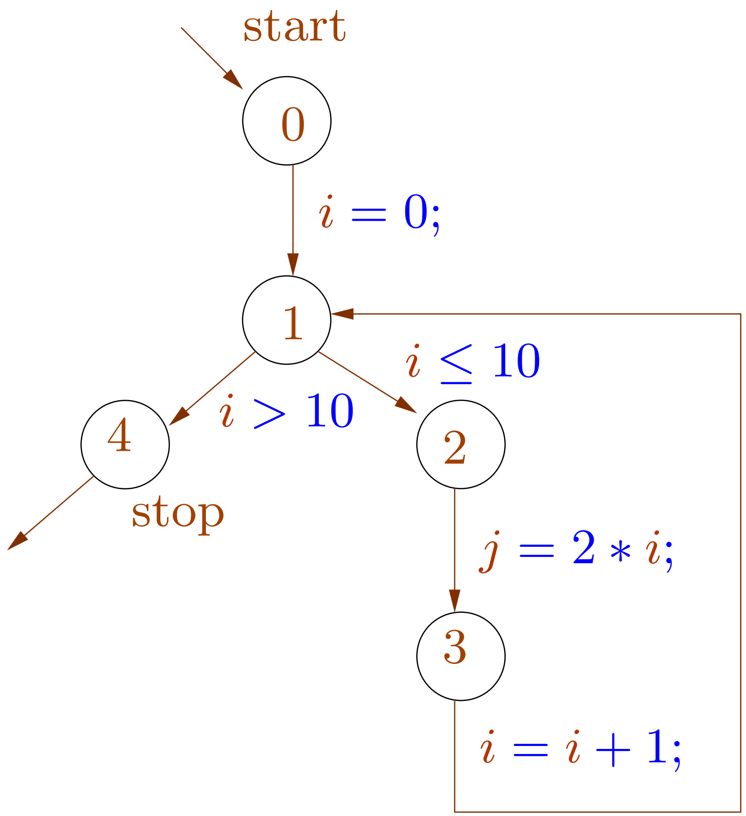
$$x_i \sqsupseteq f_i(x_1, \ldots, x_n) \qquad (1 \leq i \leq n)$$

Example



$$
\begin{aligned}
\mathcal{V}[0] &\supseteq \quad \mathcal{S} \\
\mathcal{V}[1] &\supseteq \quad [\![i = 0;]\!]\ \mathcal{V}[0] \\
\mathcal{V}[1] &\supseteq \quad [\![i = i+1;]\!]\ \mathcal{V}[3] \\
\mathcal{V}[2] &\supseteq \quad [\![i \leq 10]\!]\ \mathcal{V}[1] \\
\mathcal{V}[3] &\supseteq \quad [\![j = 2*i;]\!]\ \mathcal{V}[2] \\
\mathcal{V}[4] &\supseteq \quad [\![i > 10]\!]\ \mathcal{V}[1]
\end{aligned}
$$

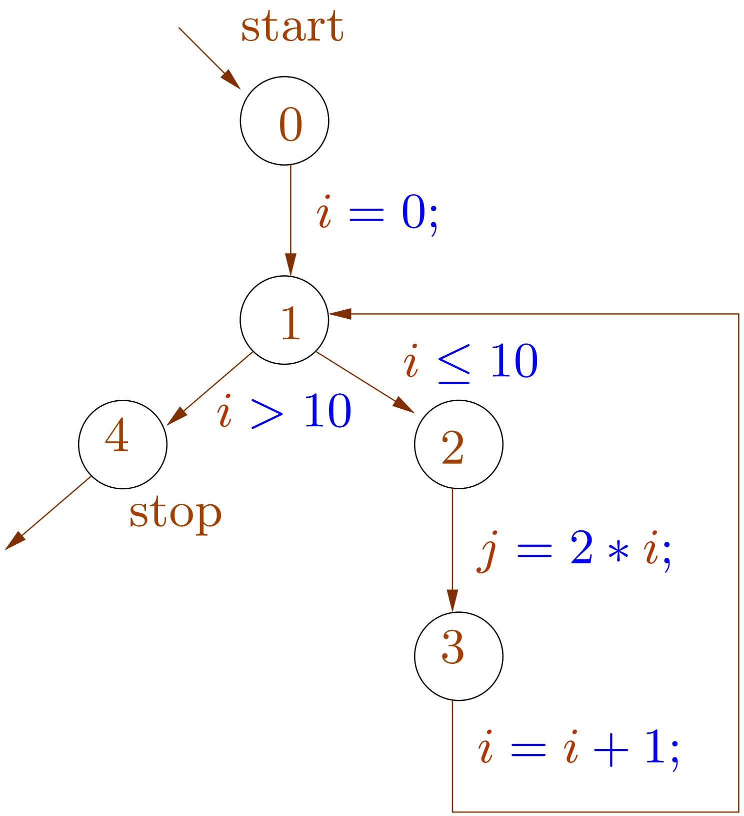
Example



$$
\begin{aligned}
\mathcal{V}[0] &\supseteq \quad \mathcal{S} \\
\mathcal{V}[1] &\supseteq \quad [\![i = 0;]\!]\ \mathcal{V}[0] \\
\mathcal{V}[1] &\supseteq \quad [\![i = i{+}1;]\!]\ \mathcal{V}[3] \\
\mathcal{V}[2] &\supseteq \quad [\![i \leq 10]\!]\ \mathcal{V}[1] \\
\mathcal{V}[3] &\supseteq \quad [\![j = 2{*}i;]\!]\ \mathcal{V}[2] \\
\mathcal{V}[4] &\supseteq \quad [\![i > 10]\!]\ \mathcal{V}[1]
\end{aligned}
$$

Transforms to ...

Example



$$
\begin{aligned}
\mathcal{V}[0] &\supseteq \quad \mathcal{S} \\
\mathcal{V}[1] &\supseteq \quad ([\![i = 0;]\!]\ \mathcal{V}[0] \\
&\qquad \cup [\![i = i{+}1;]\!]\ \mathcal{V}[3]) \\
\mathcal{V}[2] &\supseteq \quad [\![i \leq 10]\!]\ \mathcal{V}[1] \\
\mathcal{V}[3] &\supseteq \quad [\![j = 2{*}i;]\!]\ \mathcal{V}[2] \\
\mathcal{V}[4] &\supseteq \quad [\![i > 10]\!]\ \mathcal{V}[1]
\end{aligned}
$$

Since $\mathbb{D}$ is a lattice, $\mathbb{D}^n$ is also a lattice where

$$(d_1, \ldots, d_n) \sqsubseteq (d'_1, \ldots, d'_n) \text{ iff } d_i \sqsubseteq d'_i \text{ for } 1 \leq i \leq n$$

The functions $f_i : \mathbb{D}^n \rightarrow \mathbb{D}$ are monotone.

Define $F : \mathbb{D}^n \rightarrow \mathbb{D}^n$ as

$$F(y) = (f_1(y), \ldots, f_n(y)) \text{ where } y = (x_1, \ldots, x_n)$$

F is also monotone.

We need least solution of $y \sqsupseteq F(y)$.

Idea: use iteration

Start with the least element $\perp$ and compute the sequence

$\perp, F(\perp), F^2(\perp), F^3(\perp), \ldots$.

Do we always reach the least solution in this way?

Example: the complete lattice of Booleans: $\mathbb{D} = \{\bot, \top\}$.

Constraint system:

$$x \sqsupseteq y \vee z$$
$$y \sqsupseteq x \wedge y \wedge z$$
$$z \sqsupseteq \top$$

The iteration:

x	$\bot$			
y	$\bot$			
z	$\bot$			

We have $F^2(\bot) = F^3(\bot)$.

Example: the complete lattice of Booleans: $\mathbb{D} = \{\bot, \top\}$.

Constraint system:

$$x \sqsupseteq y \vee z$$
$$y \sqsupseteq x \wedge y \wedge z$$
$$z \sqsupseteq \top$$

The iteration:

x	$\bot$	$\bot$		
y	$\bot$	$\bot$		
z	$\bot$	$\top$		

We have $F^2(\bot) = F^3(\bot)$.

Example: the complete lattice of Booleans: $\mathbb{D} = \{\perp, \top\}$.

Constraint system:

$$x \sqsupseteq y \vee z$$

$$y \sqsupseteq x \wedge y \wedge z$$

$$z \sqsupseteq \top$$

The iteration:

x	$\perp$	$\perp$	$\top$	
y	$\perp$	$\perp$	$\perp$	
z	$\perp$	$\top$	$\top$	

We have $F^2(\perp) = F^3(\perp)$.

Example: the complete lattice of Booleans: $\mathbb{D} = \{\bot, \top\}$.

Constraint system:

$$x \sqsupseteq y \vee z$$
$$y \sqsupseteq x \wedge y \wedge z$$
$$z \sqsupseteq \top$$

The iteration:

x	$\bot$	$\bot$	$\top$	$\top$
y	$\bot$	$\bot$	$\bot$	$\bot$
z	$\bot$	$\top$	$\top$	$\top$

We have $F^2(\bot) = F^3(\bot)$.

Such an iteration produces an ascending chain

$$\perp \sqsubseteq F(\perp) \sqsubseteq F^2(\perp) \sqsubseteq F^3(\perp) \ldots$$

Such an iteration produces an ascending chain

$$\perp \sqsubseteq F(\perp) \sqsubseteq F^2(\perp) \sqsubseteq F^3(\perp) \ldots$$

Further if $F^k(\perp) = F^{k+1}(\perp)$ for some k

then clearly $F^k(\perp)$ is some solution of the constraint $F(x) \sqsubseteq x$.

Such an iteration produces an ascending chain

$$\perp \sqsubseteq F(\perp) \sqsubseteq F^2(\perp) \sqsubseteq F^3(\perp) \ldots$$

Further if $F^k(\perp) = F^{k+1}(\perp)$ for some k
then clearly $F^k(\perp)$ is some solution of the constraint $F(x) \sqsubseteq x$.

In fact it also the least solution of $F(x) \sqsubseteq x$.

Such an iteration produces an ascending chain

$$\perp \sqsubseteq F(\perp) \sqsubseteq F^2(\perp) \sqsubseteq F^3(\perp) \ldots$$

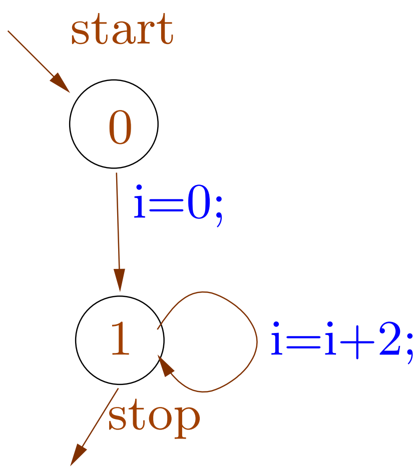
Further if $F^k(\perp) = F^{k+1}(\perp)$ for some k

then clearly $F^k(\perp)$ is some solution of the constraint $F(x) \sqsubseteq x$.

In fact it also the least solution of $F(x) \sqsubseteq x$.

Such a k always exists if the lattice is finite.

What in case of infinite lattices?



Constraint system:

$$\mathcal{V}[0] \supseteq \mathbb{Z}$$

$$\mathcal{V}[1] \supseteq \{0\} \cup \{x+2 \mid x \in \mathcal{V}[1]\}$$

The least solution:

$$\mathcal{V}[0] = \mathbb{Z} \text{ and } \mathcal{V}[1] = \{2n \mid n \geq 0\}.$$

Iteration doesn't terminate:

	$\perp$	$F(\perp)$	$F^2(\perp)$	$F^3(\perp)$	
$\mathcal{V}[0]$	$\emptyset$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\dots$
$\mathcal{V}[1]$	$\emptyset$	$\{0\}$	$\{0, 2\}$	$\{0, 2, 4\}$	

Existence of least solutions: Knaster-Tarski

Fact: In a complete lattice $\mathbb{D}$, every monotone function $f : \mathbb{D} \to \mathbb{D}$ has a least fixpoint a.

> Fixpoint: an element x such that $f(x) = x$.
> Prefixpoint: an element x such that $f(x) \sqsubseteq x$.

Existence of least solutions: Knaster-Tarski

Fact: In a complete lattice $\mathbb{D}$, every monotone function $f : \mathbb{D} \to \mathbb{D}$ has a least fixpoint a.

> Fixpoint: an element x such that $f(x) = x$.
>
> Prefixpoint: an element x such that $f(x) \sqsubseteq x$.

Let $P = \{x \in \mathbb{D} \mid f(x) \sqsubseteq x\}$ (the set of prefixpoints).

$\bigsqcap P$ is the least prefixpoint as well as the least fixpoint of f.

Example 1: Consider partial order $\mathbb{D}_1 = \mathbb{N}$ with $0 \sqsubseteq 1 \sqsubseteq 2 \sqsubseteq \ldots$.

The function $f(x) = x+1$ is monotonic.

However it has no fixpoint.

Actually $\mathbb{D}_1$ is not a complete lattice.

Example 1: Consider partial order $\mathbb{D}_1 = \mathbb{N}$ with $0 \sqsubseteq 1 \sqsubseteq 2 \sqsubseteq \ldots$.

The function $f(x) = x{+}1$ is monotonic.

However it has no fixpoint.

Actually $\mathbb{D}_1$ is not a complete lattice.

Example 2: Now we consider $\mathbb{D}_2 = \mathbb{N} \cup \{\infty\}$.

This is a complete lattice.

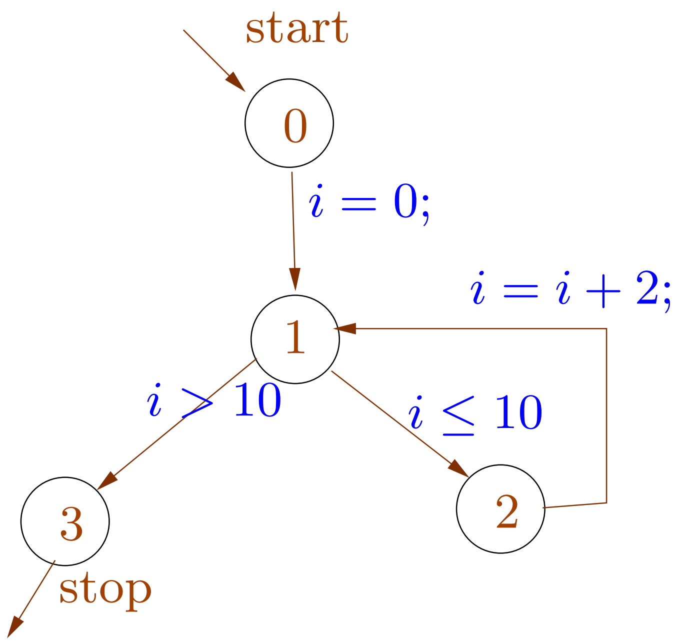
The function $f(x) = x{+}1$ is again monotonic.

The only fixpoint is ∞: $\infty{+}1 = \infty$.

Abstract Interpretation: Cousot, Cousot 1977

We use a suitable complete lattice as the domain of abstract values.

Example: intervals as abstract values:



$\mathcal{I}[0]$	$\bot$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-\infty, \infty]$
$\mathcal{I}[1]$	$\bot$	$[0, 0]$	$[0, 2]$	$[0, 12]$	$[0, 12]$
$\mathcal{I}[2]$	$\bot$	$[0, 0]$	$[0, 2] \ldots$	$[0, 10]$	$[0, 10]$
$\mathcal{I}[3]$	$\bot$	$\bot$	$\bot$	$[11, 12]$	$[11, 12]$

The analysis guarantees e.g. that at node 1 the value of i is always in the interval $[0, 12]$.

We have the set of concrete states $\mathcal{S} = (\mathsf{Vars} \to \mathbb{Z})$.

We choose a complete lattice $\mathbb{D}$ of abstract states.

We define an abstraction relation

$$\Delta \quad : \mathcal{S} \times \mathbb{D}$$

with the condition that

$$\rho \ \Delta \ a \quad \wedge \quad a \sqsubseteq b \quad \Longrightarrow \quad \rho \ \Delta \ b$$

$$\begin{tikzcd}
a \arrow[rr, "\sqsubseteq"] & & b \\
\rho \arrow[u, "\Delta"] \arrow[urr, dashed, "\Delta"'] & &
\end{tikzcd}$$

The concretization function: $\qquad \gamma(a) = \{\rho \mid \rho \ \Delta \ a\}$.

Example: For a program on two integer variables, $\mathsf{Vars} = \{x, y\}$.

The concrete states are from the set $\mathcal{S} = (\mathsf{Vars} \to \mathbb{Z})$ (or equivalently $\mathbb{Z}^2$).

For interval analysis, we choose the complete lattice

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

where $\mathbb{I} = \{[l, u] \mid l \in \mathbb{Z} \cup \{-\infty\}, u \in \mathbb{Z} \cup \{\infty\}, l \leq u\}$ is the set of intervals.

Partial order on $\mathbb{I}$: $[l_1, u_1] \sqsubseteq [l_2, u_2]$ iff $l_1 \geq l_2$ and $u_1 \leq u_2$

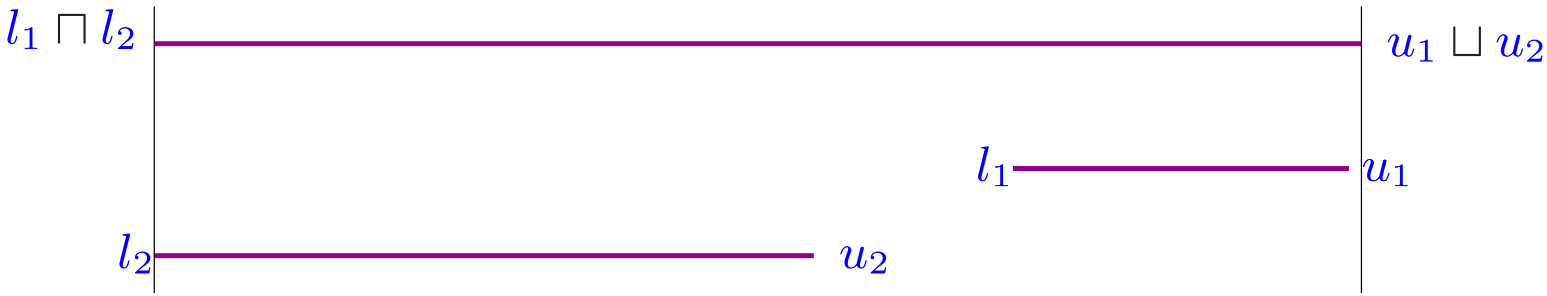
(As usual, $-\infty \leq n \leq \infty$ for all $n \in \mathbb{Z}$.)

Partial order on $\mathsf{Vars} \to \mathbb{I}$: $\quad D_1 \sqsubseteq D_2 \quad$ iff $D_1(x) \sqsubseteq D_2(x)$.

Extension to $(\mathsf{Vars} \to \mathbb{I})_\perp$: $\quad \perp \sqsubseteq D \quad$ for all D.

$(\mathsf{Vars} \to \mathbb{I})_\perp$ is a complete lattice. $(\mathsf{Vars} \to \mathbb{I})$ is not.

In particular we define $[l_1, u_1] \sqcup [l_2, u_2] = [l_1 \sqcap l_2, u_1 \sqcup u_2]$.



$\perp$ represents the "unreachable state": maps every variable to the "empty interval".

The abstraction relation:

$$\rho \;\Delta\; D \quad \text{iff} \quad D \neq \perp \;\text{ and }\; \rho(x) \;\Delta\; D(x) \text{ for each } x.$$

where $n \;\Delta\; [l, u]$ iff $l \leq n \leq u$.

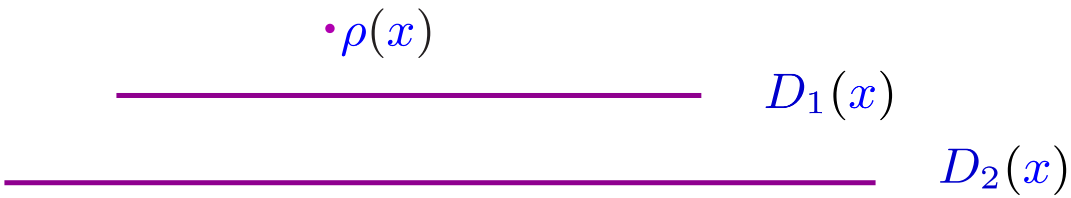
The abstraction relation:

$$\rho \;\Delta\; D \quad \text{iff} \quad D \neq \bot \;\text{ and }\; \rho(x) \;\Delta\; D(x) \text{ for each } x.$$

where $n \;\Delta\; [l, u]$ iff $l \leq n \leq u$.

This satisfies the required condition:

Suppose	$\rho \;\Delta\; D_1$ and $D_1 \sqsubseteq D_2$.
$\Longrightarrow$	$D_1 \neq \bot$ and $D_2 \neq \bot$.
	$\rho(x) \;\Delta\; D_1(x)$ and $D_1(x) \sqsubseteq D_2(x)$ for each x.
$\Longrightarrow$	$\rho(x) \;\Delta\; D_1(x)$ for each x.



The concretization function:

$$\gamma(\bot) = \{\}$$

$$\gamma(D) = \{\rho \mid \rho(x) \ \ \Delta \ \ D(x)\}, \qquad \text{for } D \neq \bot$$

$$\begin{aligned}\gamma(\{x \mapsto [3,5], y \mapsto [0,7]\}) = \qquad & \{\{x \mapsto 3, y \mapsto 0\}, \{x \mapsto 3, y \mapsto 1\}, \\ & \ldots \{x \mapsto 3, y \mapsto 7\} \\ & \ldots \{x \mapsto 5, y \mapsto 0\} \ldots \{x \mapsto 5, y \mapsto 7\}\}\end{aligned}$$

Abstraction of the partial transformation induced by edges.

Recall the edges $k = (u, l, v)$ induce a partial transformation on concrete states:

$$[\![k]\!] \ = [\![l]\!] \ : \mathcal{S} \to \mathcal{S}$$

Now on our chosen domain $\mathbb{D}$ we define a monotonic abstract transformation:

$$[\![k]\!]^\sharp \ = [\![l]\!]^\sharp \ : \mathbb{D} \to \mathbb{D}$$

The abstract transformation should simulate the concrete transformation:

$$\text{if} \quad \rho \ \Delta \ a \quad \text{and} \quad [\![l]\!] \ \rho \text{ is defined} \quad \text{then} \quad [\![l]\!] \ \rho \ \Delta \ [\![l]\!]^\sharp \ a.$$

$$\begin{tikzcd}
a \arrow[r, "{[\![k]\!]^\sharp}"] & {} \\
\rho \arrow[u, "\Delta"] \arrow[r, "{[\![k]\!]}"'] & {} \arrow[u, dashed, "\Delta"']
\end{tikzcd}$$

Abstract transformation for interval analysis.

For concrete operators $\Box$ we define monotonic abstract operators $\Box^\sharp$ such that

$$x_1 \;\; \Delta \;\; a_1 \wedge \ldots \wedge x_n \;\; \Delta \;\; a_n \Longrightarrow \Box(x_1, \ldots, x_n) \quad \Delta \quad \Box^\sharp(a_1, \ldots, a_n)$$

addition:

$$
\begin{array}{rcll}
[l_1, u_1] & +^\sharp & [l_2, u_2] & = [l_1 + l_2, u_1 + u_2]. \\
\_ & + & \infty & = \infty \\
\_ & + & -\infty & = -\infty \\
 & & & // \; \infty + -\infty \text{ is undefined.}
\end{array}
$$

substraction:

$$
\begin{array}{rcl}
-^\sharp & [l, u] & = [-u, -l]
\end{array}
$$

Multiplication: $[l_1, u_1] \quad *^\sharp \quad [l_2, u_2] = [m, n]$ where

$$\begin{aligned} m &= l_1 l_2 \sqcap l_1 u_2 \sqcap u_1 l_2 \sqcap u_1 u_2 \\ n &= l_1 l_2 \sqcup l_1 u_2 \sqcup u_1 l_2 \sqcup u_1 u_2 \end{aligned}$$

Example:

$$\begin{aligned} [1, 3] \quad *^\sharp \quad [5, 8] &= [5, 24] \\ [-1, 3] \quad *^\sharp \quad [5, 8] &= [-8, 24] \\ [-1, 3] \quad *^\sharp \quad [-5, 8] &= [-15, 24] \\ [-1, 3] \quad *^\sharp \quad [-5, -8] &= [-24, 5] \end{aligned}$$

Equality test:

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

Example:

$$\begin{aligned} [7,7] \quad &==^\sharp \quad [7,7] &= [1,1] \\ [1,7] \quad &==^\sharp \quad [9,12] &= [0,0] \\ [1,7] \quad &==^\sharp \quad [1,7] &= [0,1] \end{aligned}$$

Inequality test:

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

Example:

$$\begin{aligned} [1,7] \quad &<^\sharp \quad [9,12] \quad = [1,1] \\ [9,12] \quad &<^\sharp \quad [1,7] \quad = [0,0] \\ [1,7] \quad &<^\sharp \quad [6,8] \quad = [0,1] \end{aligned}$$

Monotonic abstract evaluation of expressions

For $D \neq \bot$,

$$
\begin{aligned}
[\![x]\!]^\sharp\ D &= D(x) \\
[\![n]\!]^\sharp\ D &= [n, n] \\
[\![\Box(e_1, \ldots, e_n)]\!]^\sharp\ D &= \Box^\sharp([\![e_1]\!]^\sharp\ D, \ldots, [\![e_n]\!]^\sharp\ D)
\end{aligned}
$$

Monotonic abstract evaluation of expressions

For $D \neq \bot$,

$$\begin{aligned} [\![x]\!]^\sharp\ D &= D(x) \\ [\![n]\!]^\sharp\ D &= [n, n] \\ [\![\Box(e_1, \ldots, e_n)]\!]^\sharp\ D &= \Box^\sharp([\![e_1]\!]^\sharp\ D, \ldots, [\![e_n]\!]^\sharp\ D) \end{aligned}$$

Fact: $\rho \ \Delta \ D$ and $[\![e]\!]\ \rho$ is defined $\Longrightarrow [\![e]\!]\ \rho \ \Delta \ [\![e]\!]^\sharp\ D$.

Monotonic abstract evaluation of expressions

For $D \neq \bot$,
$$\begin{aligned}
[\![x]\!]^\sharp\ D &= D(x)\\
[\![n]\!]^\sharp\ D &= [n, n]\\
[\![\Box(e_1, \ldots, e_n)]\!]^\sharp\ D &= \Box^\sharp([\![e_1]\!]^\sharp\ D, \ldots, [\![e_n]\!]^\sharp\ D)
\end{aligned}$$

Fact: $\rho\ \ \Delta\ \ D$ and $[\![e]\!]\ \rho$ is defined $\Longrightarrow [\![e]\!]\ \rho\ \ \Delta\ \ [\![e]\!]^\sharp\ D$.

Case e is x: since $\rho\ \ \Delta\ \ D$ hence $[\![x]\!]\ \rho = \rho(x)\ \ \Delta\ \ D(x) = [\![x]\!]^\sharp\ D$

Monotonic abstract evaluation of expressions

For $D \neq \bot$,

$$\begin{aligned}
[\![x]\!]^\sharp\ D &= D(x) \\
[\![n]\!]^\sharp\ D &= [n, n] \\
[\![\Box(e_1, \ldots, e_n)]\!]^\sharp\ D &= \Box^\sharp([\![e_1]\!]^\sharp\ D, \ldots, [\![e_n]\!]^\sharp\ D)
\end{aligned}$$

Fact: $\rho\ \ \Delta\ \ D$ and $[\![e]\!]\ \rho$ is defined $\Longrightarrow [\![e]\!]\ \rho\ \ \Delta\ \ [\![e]\!]^\sharp\ D$.

Case e is x: since $\rho\ \ \Delta\ \ D$ hence $[\![x]\!]\ \rho = \rho(x)\ \ \Delta\ \ D(x) = [\![x]\!]^\sharp\ D$

Case e is n: $[\![n]\!]\ \rho = n\ \ \Delta\ \ [n, n] = [\![n]\!]^\sharp\ D$

Monotonic abstract evaluation of expressions

For $D \neq \bot$,

$$\begin{aligned} [\![x]\!]^\sharp\ D &= D(x) \\ [\![n]\!]^\sharp\ D &= [n, n] \\ [\![\Box(e_1, \ldots, e_n)]\!]^\sharp\ D &= \Box^\sharp([\![e_1]\!]^\sharp\ D, \ldots, [\![e_n]\!]^\sharp\ D) \end{aligned}$$

Fact: $\rho \ \Delta \ D$ and $[\![e]\!]\ \rho$ is defined $\Longrightarrow [\![e]\!]\ \rho \ \Delta \ [\![e]\!]^\sharp\ D$.

Case e is x: since $\rho \ \Delta \ D$ hence $[\![x]\!]\ \rho = \rho(x) \ \Delta \ D(x) = [\![x]\!]^\sharp\ D$

Case e is n: $[\![n]\!]\ \rho = n \ \Delta \ [n, n] = [\![n]\!]^\sharp\ D$

Case e is $\Box(e_1, \ldots, e_n)$: since each $[\![e_i]\!]\ \rho \ \Delta \ [\![e_i]\!]^\sharp\ D$ hence

$$\begin{aligned} [\![\Box(e_1, \ldots, e_n)]\!]\ \rho &= \Box([\![e_1]\!]\ \rho, \ldots, [\![e_n]\!]\ \rho) \\ &\Delta \\ \Box^\sharp([\![e_1]\!]^\sharp\ D, \ldots, [\![e_n]\!]^\sharp\ D) &= [\![\Box^\sharp(e_1, \ldots, e_n)]\!]^\sharp\ D \end{aligned}$$

Finally, the monotonic abstract transformations induced by edges

$$
\begin{array}{lrl}
 & [\![l]\!]^\sharp\ \bot & = \bot \\
\text{For } D \neq \bot, & [\![;]\!]^\sharp\ D & = D \\
 & [\![x = e;]\!]^\sharp\ D & = D \oplus \{x \mapsto [\![e]\!]^\sharp\ D\} \\
 & [\![e]\!]^\sharp\ D & = \begin{cases} \bot & \text{if } [\![e]\!]^\sharp\ D = [0, 0] \\ D & \text{otherwise} \end{cases}
\end{array}
$$

Finally, the monotonic abstract transformations induced by edges

$$
\begin{aligned}
[\![l]\!]^\sharp \perp &= \perp \\
\text{For } D \neq \perp, \qquad [\![;]\!]^\sharp D &= D \\
[\![x = e;]\!]^\sharp D &= D \oplus \{x \mapsto [\![e]\!]^\sharp D\} \\
[\![e]\!]^\sharp D &= \begin{cases} \perp & \text{if } [\![e]\!]^\sharp D = [0, 0] \\ D & \text{otherwise} \end{cases}
\end{aligned}
$$

Next we must check the condition:

$$\rho \;\Delta\; D \quad \wedge \quad [\![l]\!] \rho = \rho_1 \quad \wedge \quad [\![l]\!]^\sharp D = D_1 \quad \Longrightarrow \quad \rho_1 \;\Delta\; D_1.$$

Finally, the monotonic abstract transformations induced by edges

$$
\begin{array}{rl}
 & [\![l]\!]^\sharp \perp \ = \perp \\
\text{For } D \neq \perp, & [\![;]\!]^\sharp \ D \ = D \\
 & [\![x = e;]\!]^\sharp \ D \ = D \oplus \{x \mapsto [\![e]\!]^\sharp \ D\} \\
 & [\![e]\!]^\sharp \ D \ = \begin{cases} \perp & \text{if } [\![e]\!]^\sharp \ D = [0, 0] \\ D & \text{otherwise} \end{cases}
\end{array}
$$

Next we must check the condition:

$$\rho \ \ \Delta \ \ D \quad \wedge \quad [\![l]\!] \ \rho = \rho_1 \quad \wedge \quad [\![l]\!]^\sharp \ D = D_1 \quad \Longrightarrow \quad \rho_1 \ \ \Delta \ \ D_1.$$

Clearly $D \neq \perp$ here.

To check: $\rho \;\Delta\; D \;\wedge\; [\![l]\!]\,\rho = \rho_1 \;\wedge\; [\![l]\!]^\sharp\,D = D_1 \implies \rho_1 \;\Delta\; D_1.$

Case l is ;

$$\rho_1 = \rho \;\Delta\; D = D_1.$$

To check: $\rho \;\Delta\; D \;\wedge\; [\![l]\!]\, \rho = \rho_1 \;\wedge\; [\![l]\!]^\sharp\, D = D_1 \implies \rho_1 \;\Delta\; D_1.$

Case l is ;

$$\rho_1 = \rho \;\Delta\; D = D_1.$$

Case l is $x = e$;

$$\rho_1 = \rho \oplus \{x \mapsto [\![e]\!]\, \rho\} \quad \text{and} \quad D_1 = D \oplus \{x \mapsto [\![e]\!]^\sharp\, D\}$$

As $[\![e]\!]\, \rho \;\Delta\; [\![e]\!]^\sharp\, D$ hence $\rho_1 \;\Delta\; D_1$.

To check: $\rho \;\Delta\; D \;\wedge\; [\![l]\!]\, \rho = \rho_1 \;\wedge\; [\![l]\!]^\sharp\, D = D_1 \implies \rho_1 \;\Delta\; D_1$.

Case l is ;

$\rho_1 = \rho \;\Delta\; D = D_1$.

Case l is $x = e$;

$\rho_1 = \rho \oplus \{x \mapsto [\![e]\!]\, \rho\}$ and $D_1 = D \oplus \{x \mapsto [\![e]\!]^\sharp\, D\}$

As $[\![e]\!]\, \rho \;\Delta\; [\![e]\!]^\sharp\, D$ hence $\rho_1 \;\Delta\; D_1$.

Case e is some condition e

Since the tranformation $[\![e]\!]\, \rho$ is defined,

hence the expression evaluation $[\![e]\!]\, \rho \neq 0$, and $\rho_1 = \rho$.

Since $\rho \;\Delta\; D$,

hence the abstract expression evaluation $[\![e]\!]^\sharp\, D \neq [0, 0]$, and $D_1 = D$.

Recall, for a path $\pi = k_1 \dots k_n$,

$$[\![\pi]\!]\ \rho = ([\![k_n]\!]\ \circ \dots \circ [\![k_1]\!]\ )\rho$$

$$[\![\pi]\!]^\sharp\ D = ([\![k_n]\!]^\sharp\ \circ \dots \circ [\![k_1]\!]^\sharp\ )D$$

We conclude from above:

if $\rho\ \ \Delta\ \ D$ and $[\![\pi]\!]\ \rho$ is defined then $[\![\pi]\!]\ \rho\ \ \Delta\ \ [\![\pi]\!]^\sharp\ D$.

$$\begin{tikzcd}
D \arrow[r, "{[\![k_1]\!]^\sharp}"] & \cdot \arrow[r, "{[\![k_2]\!]^\sharp}"] & \cdot \arrow[r, "{[\![k_3]\!]^\sharp}"] & \cdot \arrow[r, phantom, "\cdots"] & \cdot \arrow[r, "{[\![k_n]\!]^\sharp}"] & \cdot \\
\rho \arrow[u, "\Delta"] \arrow[r, "{[\![k_1]\!]}"'] & \cdot \arrow[u, dashed, "\Delta"] \arrow[r, "{[\![k_2]\!]}"'] & \cdot \arrow[u, dashed, "\Delta"] \arrow[r, "{[\![k_3]\!]}"'] & \cdot \arrow[u, dashed, "\Delta"] & \cdot \arrow[u, dashed, "\Delta"] \arrow[r, "{[\![k_n]\!]}"'] & \cdot \arrow[u, dashed, "\Delta"]
\end{tikzcd}$$