IS893: Advanced Software Security

7. Data-flow Analysis

Kihong Heo

KAIST

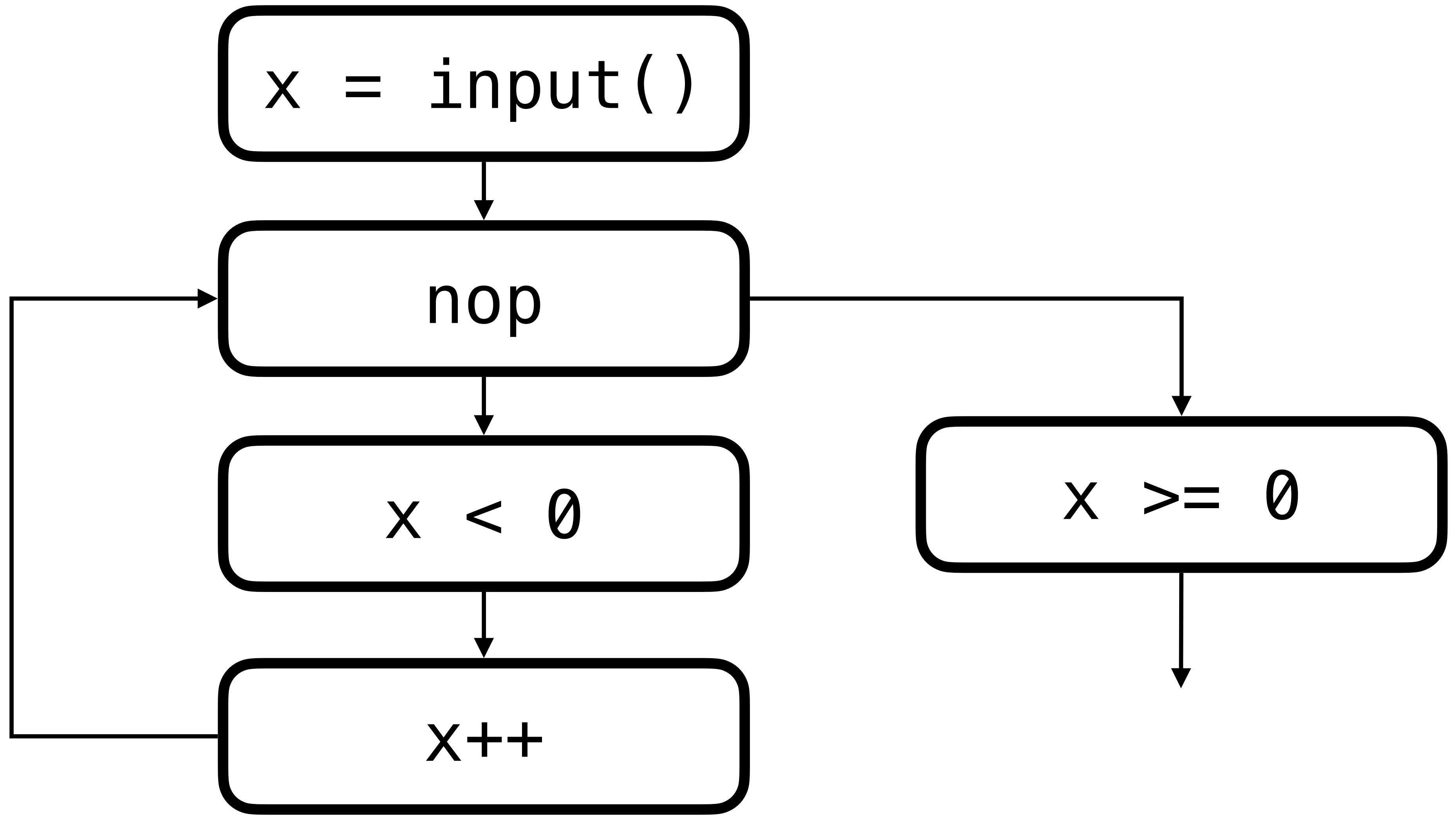
Data-flow Analysis

- A specialized framework: static analysis by equations
- Static analysis = equations setup + equations resolution
- Assumption 1: the control-flow is fixed beforehand (e.g, CFG)
  - That is why this approach is called “data-flow” analysis
  - Not true for modern languages (e.g., higher order functions, exceptions, etc)
- Assumption 2: the abstract domain has finite height

Program as a Graph

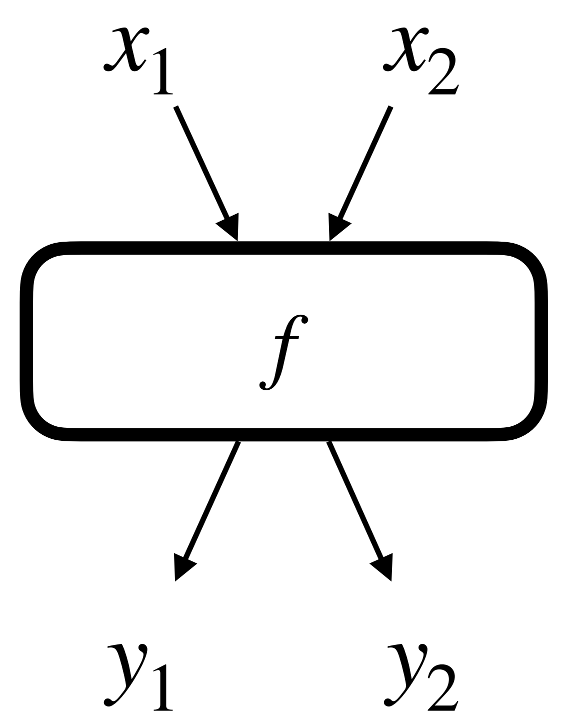
- Control-flow graph: a directed graph $G = (Node, \rightarrow)$
  - Nodes: atomic statements or conditions
  - Edges: execution order between the statements

```
x = input();
while (x < 0) {
  x++;
}
```



Equations

- Describe the abstract states that flow at each edge of the CFG

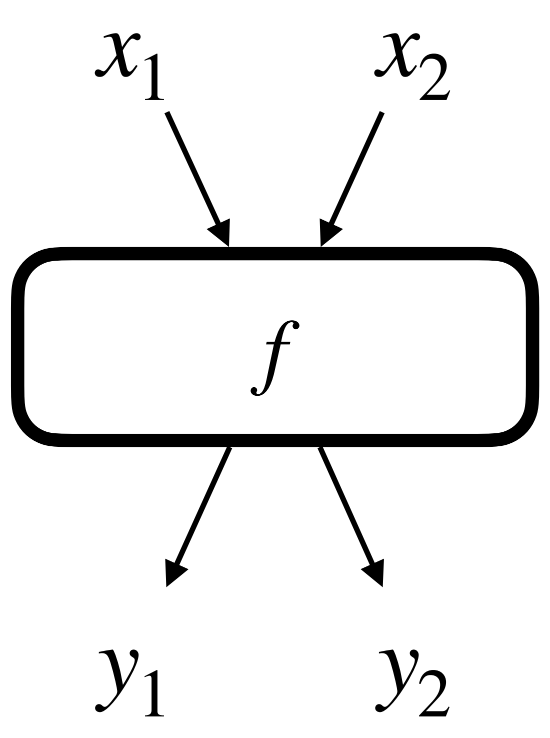


$$y_1 = f(x_1 \sqcup x_2)$$
$$y_2 = f(x_1 \sqcup x_2)$$

- f : a state transfer function for the corresponding statement (i.e., abstract semantics)
- x_i : incoming pre-states to the node (elements of an abstract domain)
- y_i : post-states flowing out along the edges (elements of an abstract domain)

Transfer Function

- A monotonic function that describes the behavior of each statement
  - Defined on abstract domain: $f \in D \to D$
  - Monotonic: $\forall x, y \in D \,.\, x \sqsubseteq y \implies f(x) \sqsubseteq f(y)$
- Larger (less precise) inputs will result in larger outputs
  - Larger = more information merged = less precise



f: "increase by 1"

x_1: Pos
x_2: Pos
y_1, y_2: Pos

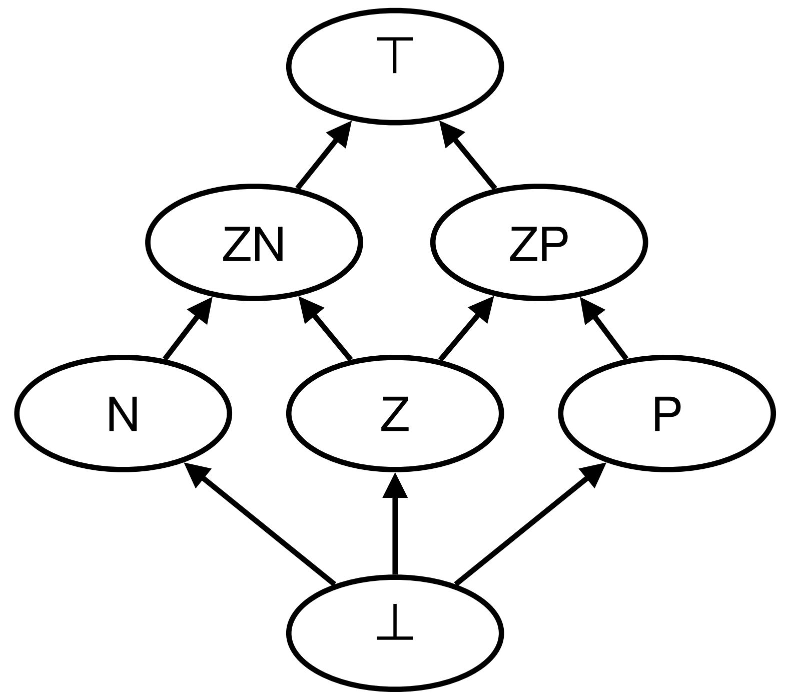
x_1: Pos
x_2: Neg
y_1, y_2: Unknown

Example: Transfer Functions

- Consider a simple grammar with only one variable x
- Goal: estimate the value of x with the sign abstract domain

$$
\begin{array}{rcl}
C & \rightarrow & \texttt{nop} \\
 & | & x := E \\
 & | & x \lessdot E \\
E & \rightarrow & n \\
 & | & x \\
 & | & E \oplus E \\
 & | & \texttt{input}()
\end{array}
$$

Grammar



Abstract Domain

$$
f_n(x) = \begin{cases} N & \text{if } n < 0 \\ Z & \text{if } n = 0 \\ P & \text{if } n > 0 \end{cases}
$$

$$
f_x(x) = x
$$

$$
f_{E_1+E_2}(x) = f_{E_1}(x) \oplus f_{E_2}(x)
$$

$$
f_{\texttt{input}()}(x) = \top
$$

Transfer functions for E

$$
f_{\texttt{nop}}(x) = x
$$

$$
f_{x:=E}(x) = f_E(x)
$$

$$
f_{x<E}(x) = \begin{cases} x \sqcap N & \text{if } f_E(x) = N, Z \text{ or } ZN \\ x & \text{otherwise} \end{cases}
$$

Transfer functions for C

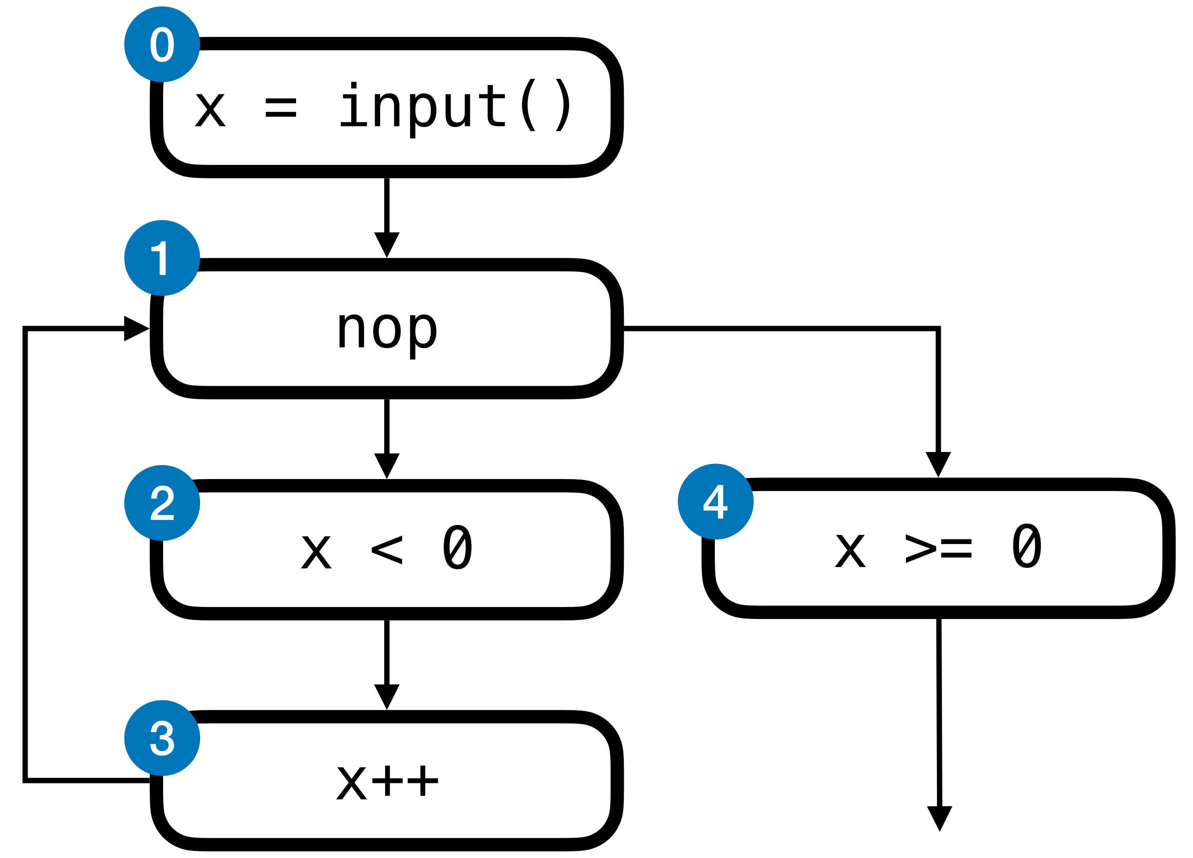
Static Analysis by Equation

- Static analysis = equation setup + equations resolution
- Semantics = solution of the equation = fixed point of the transfer function

Program

```
x = input();
while (x < 0) {
  x++;
}
```

CFG



Equation

x_i : abstract value of x after program point i

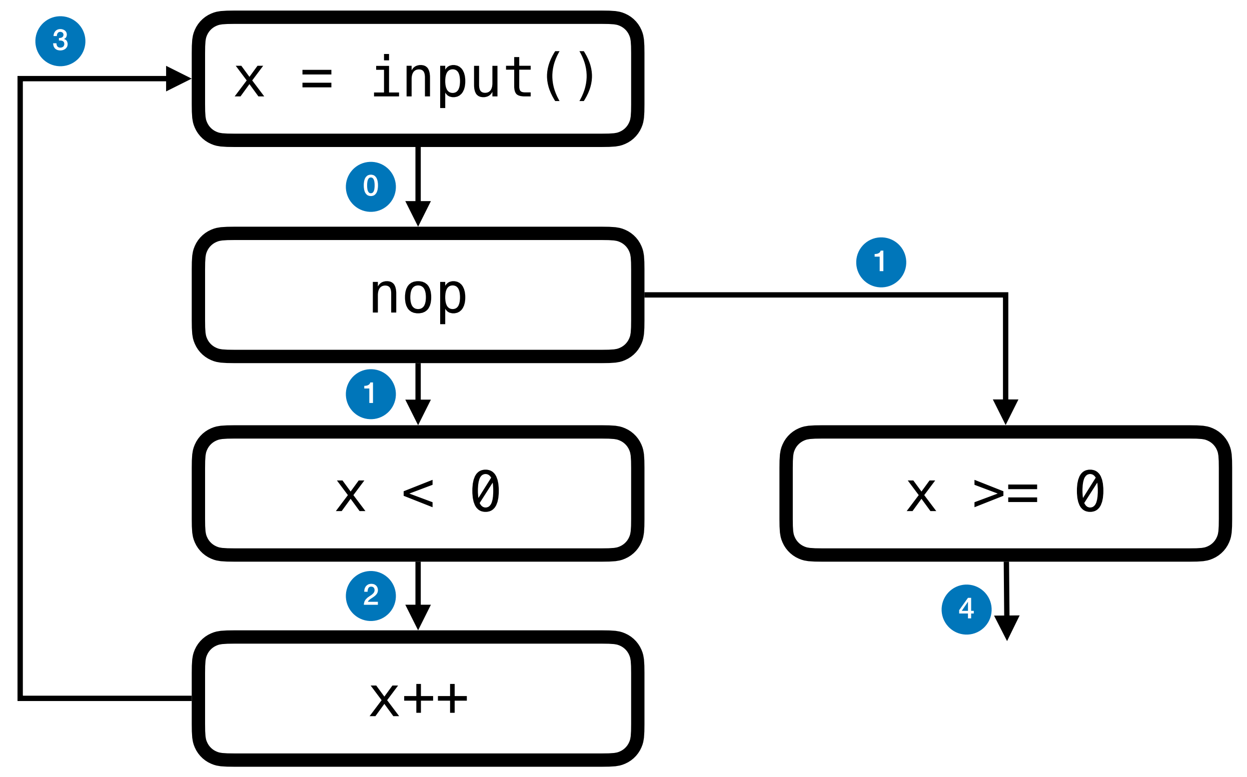
$$
\begin{aligned}
x_0 &= \top \\
x_1 &= x_0 \sqcup x_3 \\
x_2 &= x_1 \sqcap N \\
x_3 &= x_2 +^\sharp P \\
x_4 &= x_1 \sqcap ZP
\end{aligned}
$$

Solution (Fixed point)

$$
\begin{aligned}
(x_0, x_1, x_2, x_3, x_4) &= F(x_0, x_1, x_2, x_3, x_4) \\
&= \text{fix}F
\end{aligned}
$$

$$
\begin{aligned}
x_0 &= \top \\
x_1 &= \top \\
x_2 &= N \\
x_3 &= \top \\
x_4 &= ZP
\end{aligned}
$$

Equations Resolution



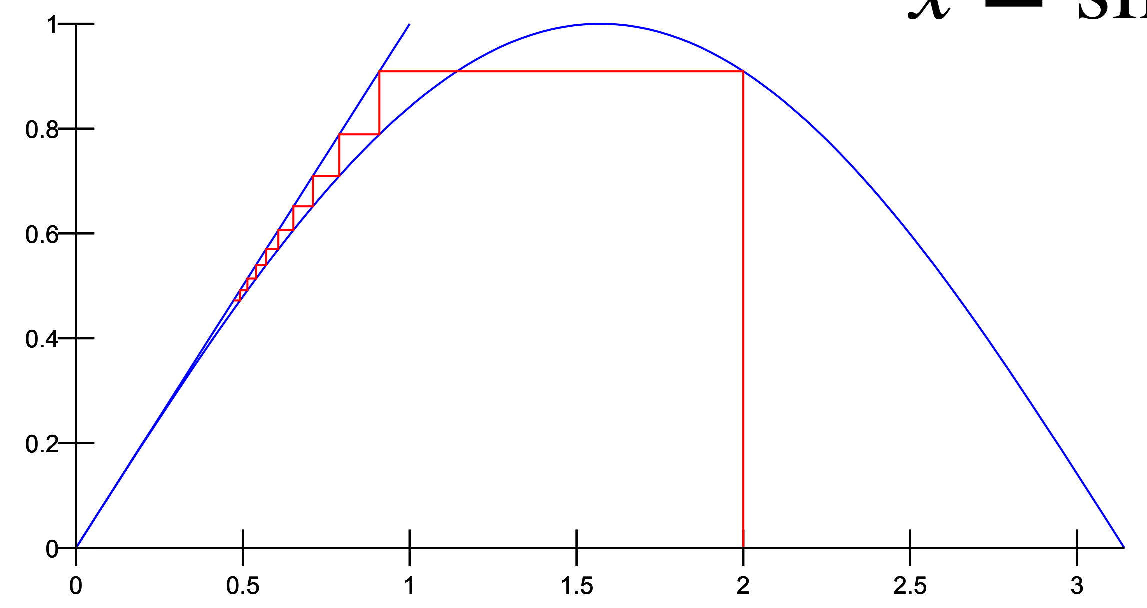
$$
\begin{aligned}
x_0 &= \top \\
x_1 &= x_0 \sqcup x_3 \\
x_2 &= x_1 \sqcap N \\
x_3 &= x_2 +^\sharp P \\
x_4 &= x_1 \sqcap ZP
\end{aligned}
$$

$$F : (Node \to Z^\sharp) \to (Node \to Z^\sharp)$$

$$F(X) = \lambda c \in Node. f_c\Big(\bigsqcup_{c' \to c} (X(c'))\Big)$$

- Static analysis result = a solution of the equations = a fixed point of F
- Q: How to solve the equations?
- A: A generic iterative algorithm!

$x = \sin(x)$



Fixed-point Iteration

1. Start from the bottom element
2. Apply the transfer function
3. If the result is stable, return, otherwise goto 2 with the result

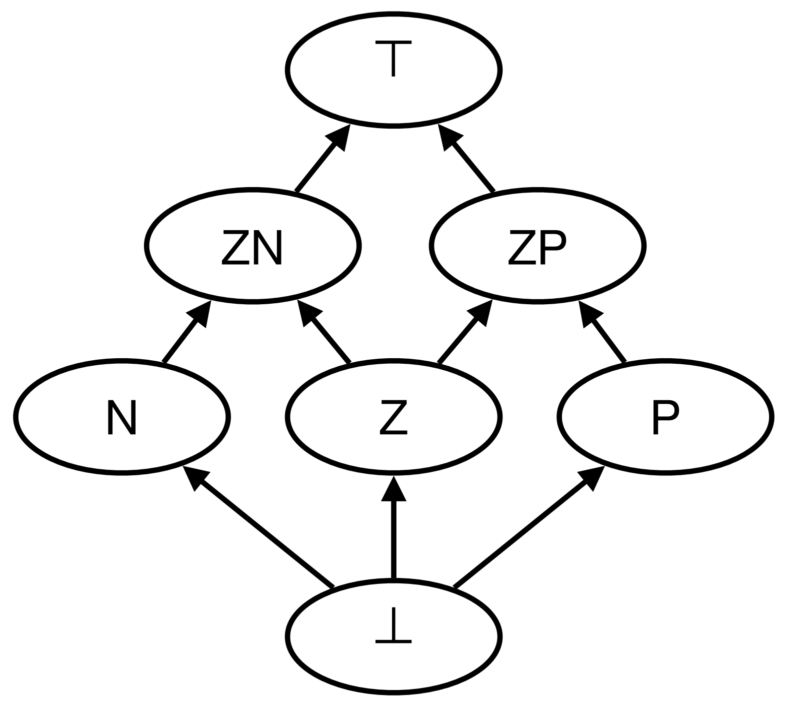
$$(x_0, x_1, x_2, x_3, x_4) = F(x_0, x_1, x_2, x_3, x_4)$$

$$\begin{aligned} x_0 &= \top \\ x_1 &= x_0 \sqcup x_3 \\ x_2 &= x_1 \sqcap N \\ x_3 &= x_2 +^\sharp P \\ x_4 &= x_1 \sqcap ZP \end{aligned}$$

x	Iter 0	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
0	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$
1	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$
2	$\bot$	$\bot$	$\bot$	N	N	N
3	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$
4	$\bot$	$\bot$	$\bot$	$\bot$	ZP	ZP

Termination

- Q: Does the fix-point iteration algorithm always terminate?
  - Static analysis must terminate even though the target program does not
- A: Yes! because,
  - Abstract domains have finite height + transfer functions are monotone



$$\forall x, y \in D\,.\, x \sqsubseteq y \implies f(x) \sqsubseteq f(y)$$

Extension to Handle Memory

- How to analyze memory state?
  - Abstract memory: a mapping from variables to abstract values
  - E.g., $Mem^\sharp = Var \rightarrow Z^\sharp$
- How to make a CPO for abstract memory?

Constructions on CPOs

- If S is a set, and D_1 and D_2 are CPOs, then the followings are CPOs
  - Lifted set : $D = S_\perp$
  - Cartesian product : $D = D_1 \times D_2$
  - Separated sum : $D = D_1 + D_2$
  - Function : $D = D_1 \to D_2$

Abstract Semantics with Memory

- Abstract domain: $Mem^\sharp = Var_\bot \rightarrow Z^\sharp$
  - Order: $m_1^\sharp \sqsubseteq m_2^\sharp \iff \forall x \in Var_\bot \,.\, m_1^\sharp(x) \sqsubseteq m_2^\sharp(x)$
  - Join: $m_1^\sharp \sqcup m_2^\sharp = \lambda x \,.\, m_1^\sharp(x) \sqcup m_2^\sharp(x)$
- Abstract semantics

$$F : (Node \rightarrow Mem^\sharp) \rightarrow (Node \rightarrow Mem^\sharp)$$

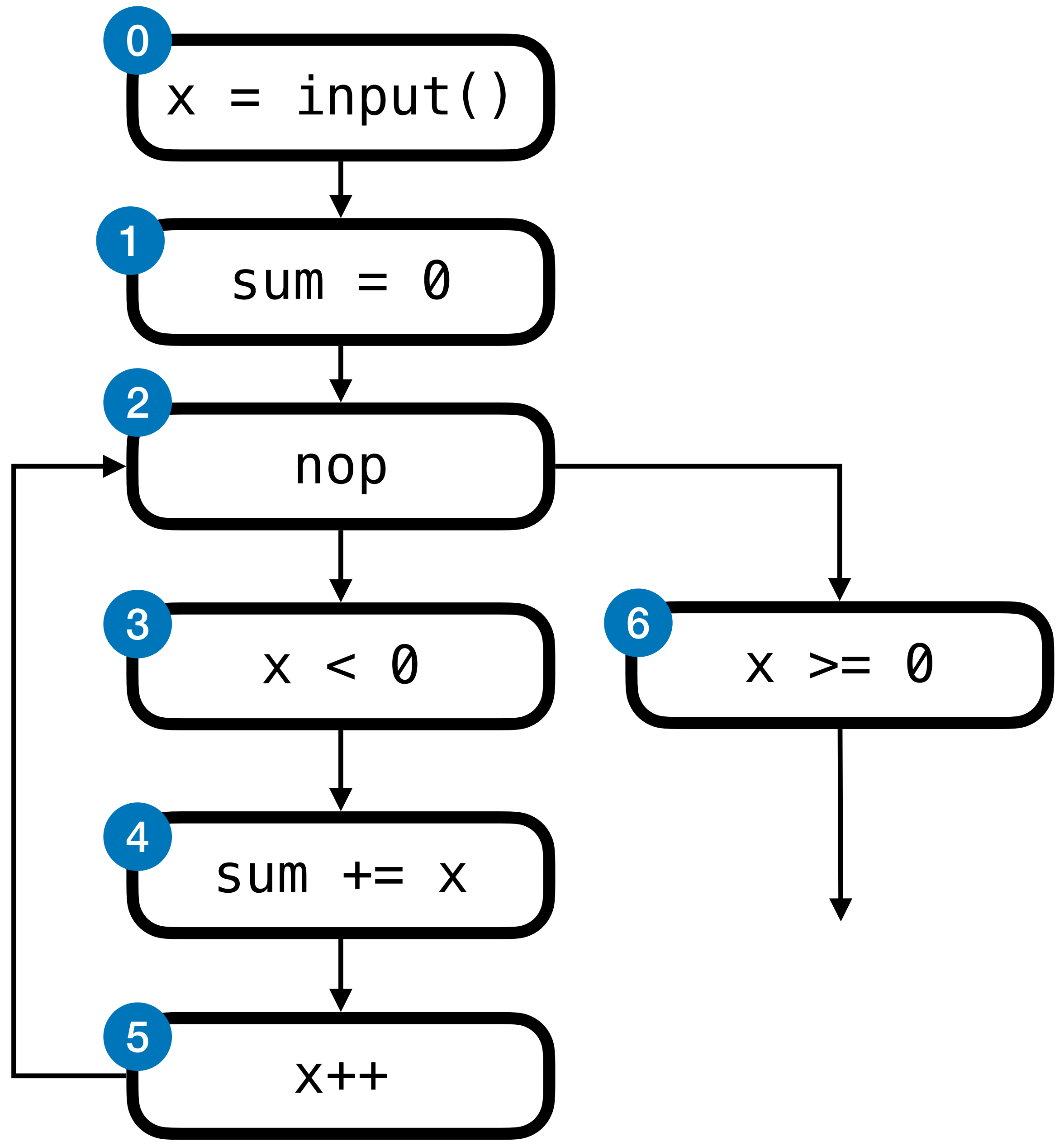
$$F(X) = \lambda c.f_n\Big(\bigsqcup_{c' \rightarrow c} (X(c'))\Big)$$

Example

Program

```
x = input();
sum = 0;
while (x < 0) {
  sum += x;
  x++;
}
```

CFG



Equation

m_i : abstract memory after program point i

$$
\begin{aligned}
m_0 &= \{x \mapsto \top\} \\
m_1 &= \{x \mapsto \top, sum \mapsto Z\} \\
m_2 &= m_1 \sqcup m_5 \\
m_3 &= m_2 \sqcap \top\{x \mapsto N\} \\
m_4 &= m_3\{sum \mapsto m_3(sum) +^\sharp m_3(x)\} \\
m_5 &= m_4\{x \mapsto m_4(x) +^\sharp P\} \\
m_6 &= m_2 \sqcap \top\{x \mapsto ZP\}
\end{aligned}
$$

Solution (Fixed point)

$$
\begin{aligned}
(m_0, m_1, m_2, m_3, m_4, m_5) &= F(m_0, m_1, m_2, m_3, m_4, m_5) \\
&= \text{fix}F
\end{aligned}
$$

$$
\begin{aligned}
m_0 &= \{x \mapsto \top\} \\
m_1 &= \{x \mapsto \top, sum \mapsto Z\} \\
m_2 &= \{x \mapsto \top, sum \mapsto ZP\} \\
m_3 &= \{x \mapsto N, sum \mapsto ZP\} \\
m_4 &= \{x \mapsto N, sum \mapsto ZP\} \\
m_5 &= \{x \mapsto \top, sum \mapsto ZP\} \\
m_6 &= \{x \mapsto P, sum \mapsto ZP\}
\end{aligned}
$$

Conclusion

- Data-flow analysis: a specialized framework of static analysis
- Static analysis by equation
  - Static analysis = equations setup + equations resolution
- Limited but powerful enough for simple properties
  - How to achieve more sophisticated static analysis?
    (e.g., infinite domains, dynamic jumps, etc)