AAA616: Program Analysis

Lecture 3 — Introduction to Program Analysis

Hakjoo Oh
2016 Fall

Static Program Analysis

A general method for
automatic and sound approximation of
sw run-time behaviors
before the execution

- "before": statically, without running sw
- "automatic": sw analyzes sw
- "sound": all possibilities into account
- "approximation": cannot be exact
- "general": for any source language and property
  - C, C++, C#, F#, Java, JavaScript, ML, Scala, Python, JVM, Dalvik, x86, Excel, etc
  - "buffer-overrun?", "memory leak?", "type errors?", "x = y at line 2?", "memory use $\leq 2K$?", etc

Program Analysis is Undecidable

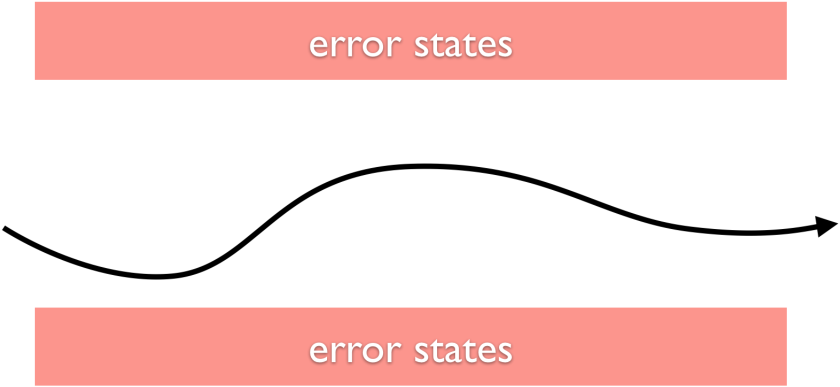
Reasoning about program behavior involves the Halting Problem: e.g.,

$$\texttt{if } \cdots \texttt{ then } x := 1 \texttt{ else } (S; x := 2); y := x$$

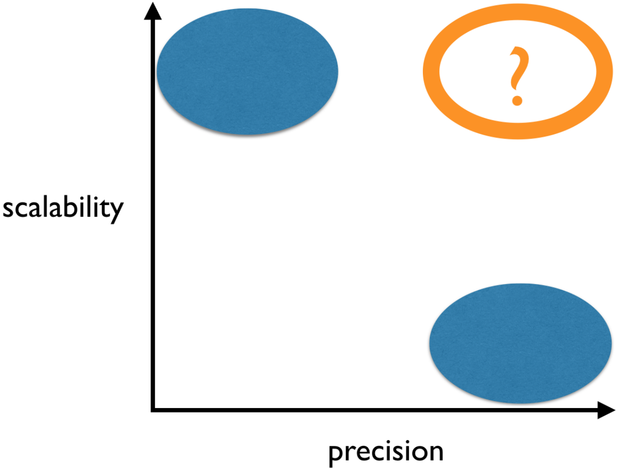
What are the possible values of x at the last statement?

Alan Turing (1912–1954)

Side-Stepping Undecidability



Key Challenge in Static Analysis



The While Language

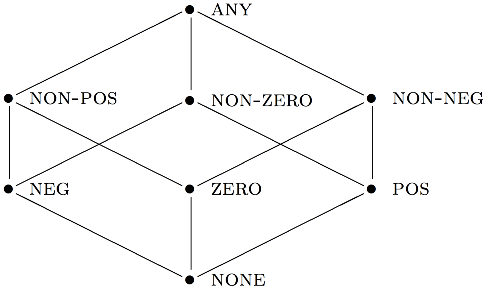
$$
\begin{array}{rcl}
a & \rightarrow & n \mid x \mid a_1 + a_2 \mid a_1 \star a_2 \mid a_1 - a_2 \\
b & \rightarrow & \texttt{true} \mid \texttt{false} \mid a_1 = a_2 \mid a_1 \leq a_2 \mid \neg b \mid b_1 \wedge b_2 \\
c & \rightarrow & x := a \mid \texttt{skip} \mid c_1 ; c_2 \mid \texttt{if}\ b\ c_1\ c_2 \mid \texttt{while}\ b\ c
\end{array}
$$

Example 1: Sign Analysis

$$\texttt{if} \ \cdots \ \texttt{then} \ x := 1 \ \texttt{else} \ (S; x := 2); y := x$$

Sign Domain

The complete lattice $(\mathbf{Sign}, \sqsubseteq)$:



The lattice is an abstraction of integers:

$$\alpha_{\mathbf{Z}} : \wp(\mathbf{Z}) \to \mathbf{Sign}, \qquad \gamma_{\mathbf{Z}} : \mathbf{Sign} \to \wp(\mathbf{Z})$$

Abstract States

The complete lattice of abstract states $(\widehat{\textbf{State}}, \sqsubseteq)$:

$$\widehat{\textbf{State}} = \textbf{Var} \to \textbf{Sign}$$

with the pointwise ordering:

$$\hat{s}_1 \sqsubseteq \hat{s}_2 \iff \forall x \in \textbf{Var}.\ \hat{s}_1(x) \sqsubseteq \hat{s}_2(x).$$

The least upper bound of $Y \subseteq \widehat{\textbf{State}}$,

$$\bigsqcup Y = \lambda x. \bigsqcup_{\hat{s} \in Y} \hat{s}(x).$$

Lemma

Let S be a non-empty set and $(D, \sqsubseteq)$ be a poset. Then, the poset $(S \to D, \sqsubseteq)$ with the ordering

$$f_1 \sqsubseteq f_2 \iff \forall s \in S.\ f_1(s) \sqsubseteq f_2(s)$$

is a complete lattice (resp., CPO) if D is a complete lattice (resp., CPO).

Abstract States

The complete lattice of abstract states $(\widehat{\textbf{State}}, \sqsubseteq)$:

$$\widehat{\textbf{State}} = \textbf{Var} \rightarrow \textbf{Sign}$$

with the pointwise ordering:

$$\hat{s}_1 \sqsubseteq \hat{s}_2 \iff \forall x \in \textbf{Var}.\ \hat{s}_1(x) \sqsubseteq \hat{s}_2(x).$$

The least upper bound of $Y \subseteq \widehat{\textbf{State}}$,

$$\bigsqcup Y = \lambda x. \bigsqcup_{\hat{s} \in Y} \hat{s}(x).$$

$$\alpha : \wp(\textbf{State}) \rightarrow \widehat{\textbf{State}}$$

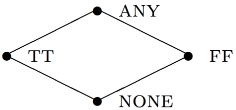
$$\alpha(S) = \lambda x. \bigsqcup_{s \in S} \alpha_{\mathbf{Z}}(\{s(x)\})$$

$$\gamma : \widehat{\textbf{State}} \rightarrow \wp(\textbf{State})$$

$$\gamma(\hat{s}) = \{s \in \textbf{State} \mid \forall x \in \textbf{Var}.\ s(x) \in \gamma_{\mathbf{Z}}(\hat{s}(x))\}$$

Abstract Booleans

The truth values $\mathbf{T} = \{true, false\}$ are abstracted by the complete lattice $(\widehat{\mathbf{T}}, \sqsubseteq)$:



The abstraction and concretization functions for the lattice:

$$\alpha_{\mathbf{T}} : \wp(\mathbf{T}) \to \widehat{\mathbf{T}}, \qquad \gamma_{\mathbf{T}} : \widehat{\mathbf{T}} \to \wp(\mathbf{T})$$

Abstract Semantics

$$\begin{array}{rcl}
\widehat{\mathcal{A}}[\![a]\!] & : & \widehat{\textsf{State}} \rightarrow \textsf{Sign} \\
\widehat{\mathcal{A}}[\![n]\!](\hat{s}) & = & \alpha_{\mathsf{Z}}(\{n\}) \\
\widehat{\mathcal{A}}[\![x]\!](\hat{s}) & = & \hat{s}(x) \\
\widehat{\mathcal{A}}[\![a_1 + a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) +_S \widehat{\mathcal{A}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{A}}[\![a_1 \star a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) \star_S \widehat{\mathcal{A}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{A}}[\![a_1 - a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) -_S \widehat{\mathcal{A}}[\![a_2]\!](\hat{s})
\end{array}$$

Abstract Semantics

$+_S$	NONE	NEG	ZERO	POS	NON-POS	NON-ZERO	NON-NEG	ANY
NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
NEG	NONE	NEG	NEG	ANY	NEG	ANY	ANY	ANY
ZERO	NONE	POS	ZERO	POS	NON-POS	NON-ZERO	NON-NEG	ANY
POS	NONE	ANY	POS	POS	ANY	ANY	POS	ANY
NON-POS	NONE	NEG	NON-POS	ANY	NON-POS	ANY	ANY	ANY
NON-ZERO	NONE	ANY	NON-ZERO	ANY	ANY	ANY	ANY	ANY
NON-NEG	NONE	ANY	NON-NEG	POS	ANY	ANY	NON-NEG	ANY
ANY	NONE	ANY	ANY	ANY	ANY	ANY	ANY	ANY

$\star_S$	NEG	ZERO	POS
NEG	POS	ZERO	NEG
ZERO	ZERO	ZERO	ZERO
POS	NEG	ZERO	POS

$-_S$	NEG	ZERO	POS
NEG	ANY	NEG	NEG
ZERO	POS	ZERO	NEG
POS	POS	POS	ANY

Abstract Semantics

$$
\begin{array}{rcl}
\widehat{\mathcal{B}}[\![b]\!] & : & \widehat{\textsf{State}} \to \widehat{\textsf{T}} \\
\widehat{\mathcal{B}}[\![\texttt{true}]\!](\hat{s}) & = & \mathrm{TT} \\
\widehat{\mathcal{B}}[\![\texttt{false}]\!](\hat{s}) & = & \mathrm{FF} \\
\widehat{\mathcal{B}}[\![a_1 = a_2]\!](\hat{s}) & = & \widehat{\mathcal{B}}[\![a_1]\!](\hat{s}) =_S \widehat{\mathcal{B}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{B}}[\![a_1 \leq a_2]\!](\hat{s}) & = & \widehat{\mathcal{B}}[\![a_1]\!](\hat{s}) \leq_S \widehat{\mathcal{B}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{B}}[\![\neg b]\!](\hat{s}) & = & \neg_S \widehat{\mathcal{B}}[\![b]\!](\hat{s}) \\
\widehat{\mathcal{B}}[\![b_1 \wedge b_2]\!](\hat{s}) & = & \widehat{\mathcal{B}}[\![b_1]\!](\hat{s}) \wedge_S \widehat{\mathcal{B}}[\![b_2]\!](\hat{s})
\end{array}
$$

Abstract Semantics

$=_S$	NEG	ZERO	POS
NEG	ANY	FF	FF
ZERO	FF	TT	FF
POS	FF	FF	ANY

$\leq_S$	NEG	ZERO	POS
NEG	ANY	TT	TT
ZERO	FF	TT	TT
POS	FF	FF	ANY

$\neg_T$	
NONE	NONE
TT	FF
FF	TT
ANY	ANY

$\wedge_T$	NONE	TT	FF	ANY
NONE	NONE	NONE	NONE	NONE
TT	NONE	TT	FF	ANY
FF	NONE	FF	FF	FF
ANY	NONE	ANY	FF	ANY

Abstract Semantics

$$\begin{aligned}
\widehat{\mathcal{C}}[\![c]\!] &: \widehat{\textbf{State}} \rightarrow \widehat{\textbf{State}} \\
\widehat{\mathcal{C}}[\![x := a]\!] &= \lambda\hat{s}.\hat{s}[x \mapsto \widehat{\mathcal{A}}[\![a]\!](\hat{s})] \\
\widehat{\mathcal{C}}[\![\texttt{skip}]\!] &= \textbf{id} \\
\widehat{\mathcal{C}}[\![c_1; c_2]\!] &= \widehat{\mathcal{C}}[\![c_2]\!] \circ \widehat{\mathcal{C}}[\![c_1]\!] \\
\widehat{\mathcal{C}}[\![\texttt{if}\ b\ c_1\ c_2]\!] &= \widehat{\textbf{cond}}(\widehat{\mathcal{B}}[\![b]\!], \widehat{\mathcal{C}}[\![c_1]\!], \widehat{\mathcal{C}}[\![c_2]\!]) \\
\widehat{\mathcal{C}}[\![\texttt{while}\ b\ c]\!] &= \mathit{fix}\,\widehat{F} \\
& \quad \text{where } \widehat{F}(g) = \widehat{\textbf{cond}}(\widehat{\mathcal{B}}[\![b]\!], g \circ \widehat{\mathcal{C}}[\![c]\!], \textbf{id})
\end{aligned}$$

$$\widehat{\textbf{cond}}(f, g, h)(\hat{s}) = \begin{cases} \bot & \cdots f(\hat{s}) = \text{NONE} \\ f(\hat{s}) & \cdots f(\hat{s}) = \text{TT} \\ g(\hat{s}) & \cdots f(\hat{s}) = \text{FF} \\ f(\hat{s}) \sqcup g(\hat{s}) & \cdots f(\hat{s}) = \text{ANY} \end{cases}$$

Example

$$y := 1; \texttt{while}\ x \neq 0\ (y := y \star x; x := x - 1)$$

Example 2: Taint Analysis (Information Flow Analysis)

Can the information from the untrustworthy source be transferred to the sink?

```
x:=source(); ...; sink(y)
```

Applications to sw security:

- privacy leak
- SQL injection
- buffer overflow
- integer overflow
- XSS
- ...

Abstract Domain

- The complete lattice of the abstract values $(\widehat{\mathbf{T}}, \sqsubseteq)$:

$$\widehat{\mathbf{T}} = \{\mathrm{LOW}, \mathrm{HIGH}\}$$

  with the ordering $\mathrm{LOW} \sqsubseteq \mathrm{HIGH}$, $\mathrm{LOW} \sqsubseteq \mathrm{LOW}$, and $\mathrm{HIGH} \sqsubseteq \mathrm{HIGH}$.

- The lattice of states:

$$\widehat{\mathbf{State}} = \mathbf{Var} \rightarrow \widehat{\mathbf{T}}$$

Abstract Semantics

$$
\begin{array}{rcl}
\widehat{\mathcal{A}}[\![a]\!] & : & \widehat{\textsf{State}} \to \widehat{\textsf{T}} \\
\widehat{\mathcal{A}}[\![n]\!](\hat{s}) & = & \begin{cases} \text{LOW} & \cdots n \text{ is public} \\ \text{HIGH} & \cdots n \text{ is private} \end{cases} \\
\widehat{\mathcal{A}}[\![x]\!](\hat{s}) & = & \hat{s}(x) \\
\widehat{\mathcal{A}}[\![a_1 + a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) \sqcup \widehat{\mathcal{A}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{A}}[\![a_1 \star a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) \sqcup \widehat{\mathcal{A}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{A}}[\![a_1 - a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) \sqcup \widehat{\mathcal{A}}[\![a_2]\!](\hat{s})
\end{array}
$$

Abstract Semantics

$$\begin{array}{rcl}
\widehat{\mathcal{B}}[\![b]\!] & : & \widehat{\textsf{State}} \to \widehat{\textsf{T}} \\
\widehat{\mathcal{B}}[\![\texttt{true}]\!](\hat{s}) & = & \text{LOW} \\
\widehat{\mathcal{B}}[\![\texttt{false}]\!](\hat{s}) & = & \text{LOW} \\
\widehat{\mathcal{B}}[\![a_1 = a_2]\!](\hat{s}) & = & \widehat{\mathcal{B}}[\![a_1]\!](\hat{s}) \sqcup \widehat{\mathcal{B}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{B}}[\![a_1 \leq a_2]\!](\hat{s}) & = & \widehat{\mathcal{B}}[\![a_1]\!](\hat{s}) \sqcup \widehat{\mathcal{B}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{B}}[\![\neg b]\!](\hat{s}) & = & \widehat{\mathcal{B}}[\![b]\!](\hat{s}) \\
\widehat{\mathcal{B}}[\![b_1 \wedge b_2]\!](\hat{s}) & = & \widehat{\mathcal{B}}[\![b_1]\!](\hat{s}) \sqcup \widehat{\mathcal{B}}[\![b_2]\!](\hat{s})
\end{array}$$

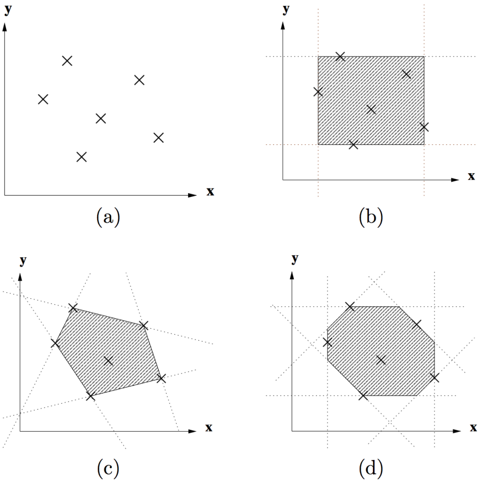
Abstract Semantics

$$
\begin{array}{rcl}
\widehat{\mathcal{C}}[\![c]\!] & : & \widehat{\textsf{State}} \to \widehat{\textsf{State}} \\
\widehat{\mathcal{C}}[\![x := a]\!] & = & \lambda\hat{s}.\hat{s}[x \mapsto \widehat{\mathcal{A}}[\![a]\!](\hat{s})] \\
\widehat{\mathcal{C}}[\![\texttt{skip}]\!] & = & \textsf{id} \\
\widehat{\mathcal{C}}[\![c_1; c_2]\!] & = & \widehat{\mathcal{C}}[\![c_2]\!] \circ \widehat{\mathcal{C}}[\![c_1]\!] \\
\widehat{\mathcal{C}}[\![\texttt{if}\ b\ c_1\ c_2]\!] & = & \lambda\hat{s}.\widehat{\mathcal{C}}[\![c_1]\!](\hat{s}) \sqcup \widehat{\mathcal{C}}[\![c_2]\!](\hat{s}) \\
\widehat{\mathcal{C}}[\![\texttt{while}\ b\ c]\!] & = & \mathit{fix}\widehat{F} \\
 & & \text{where } \widehat{F}(g) = \lambda\hat{s}.\hat{s} \sqcup (g \circ \widehat{\mathcal{C}}[\![c]\!])(\hat{s})
\end{array}
$$

Example 3: Interval Analysis

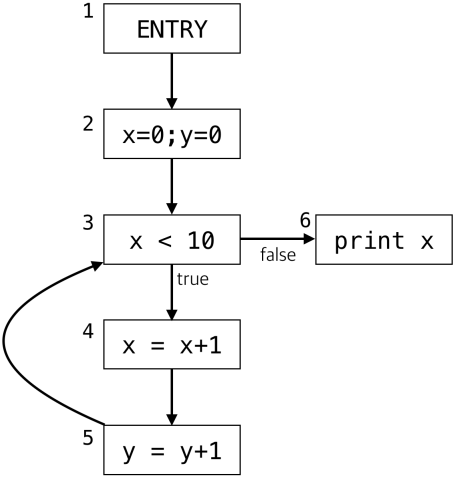
- ```
  x = 0;
  while (x < 10) {
    assert (x < 10);
    x++;
  }
  assert (x == 10);
  ```
- ```
  x = 0;
  y = 0;
  while (x < 10) {
    assert (y < 10);
    x++; y++;
  }
  assert (y == 10);
  ```

cf) Numerical Abstractions



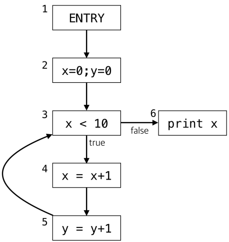
(image from The Octagon Abstract Domain by Antonine Mine)

Example 3: Interval Analysis



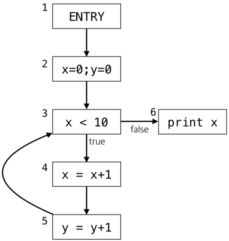
Node	Result
1	$x \mapsto \bot$ $y \mapsto \bot$
2	$x \mapsto [0, 0]$ $y \mapsto [0, 0]$
3	$x \mapsto [0, 9]$ $y \mapsto [0, +\infty]$
4	$x \mapsto [1, 10]$ $y \mapsto [0, +\infty]$
5	$x \mapsto [1, 10]$ $y \mapsto [1, +\infty]$
6	$x \mapsto [10, 10]$ $y \mapsto [0, +\infty]$

Fixed Point Computation Does Not Terminate



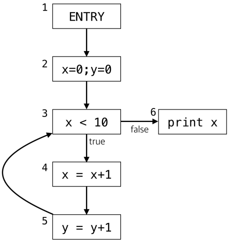
Node	initial	1	2	3	10	11	k	∞
1	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$
2	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,0]$ $y \mapsto [0,0]$
3	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,1]$ $y \mapsto [0,1]$	$x \mapsto [0,2]$ $y \mapsto [0,2]$	$x \mapsto [0,9]$ $y \mapsto [0,9]$	$x \mapsto [0,9]$ $y \mapsto [0,10]$	$x \mapsto [0,9]$ $y \mapsto [0,k-1]$	$x \mapsto [0,9]$ $y \mapsto [0,+\infty]$
4	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto [1,1]$ $y \mapsto [0,0]$	$x \mapsto [1,2]$ $y \mapsto [0,1]$	$x \mapsto [1,3]$ $y \mapsto [0,2]$	$x \mapsto [1,10]$ $y \mapsto [0,9]$	$x \mapsto [1,10]$ $y \mapsto [0,10]$	$x \mapsto [1,10]$ $y \mapsto [0,k-1]$	$x \mapsto [1,10]$ $y \mapsto [0,+\infty]$
5	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto [1,1]$ $y \mapsto [1,1]$	$x \mapsto [1,2]$ $y \mapsto [1,2]$	$x \mapsto [1,3]$ $y \mapsto [1,3]$	$x \mapsto [1,10]$ $y \mapsto [1,10]$	$x \mapsto [1,10]$ $y \mapsto [1,11]$	$x \mapsto [1,10]$ $y \mapsto [1,k]$	$x \mapsto [1,10]$ $y \mapsto [1,+\infty]$
6	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto [0,0]$	$x \mapsto \bot$ $y \mapsto [0,1]$	$x \mapsto \bot$ $y \mapsto [0,2]$	$x \mapsto [10,10]$ $y \mapsto [0,9]$	$x \mapsto [10,10]$ $y \mapsto [0,10]$	$x \mapsto [10,10]$ $y \mapsto [0,k-1]$	$x \mapsto [10,10]$ $y \mapsto [0,+\infty]$

Fixed Point Computation with Widening and Narrowing



Node	initial	**1**	**2**	**3**
1	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$
2	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,0]$ $y \mapsto [0,0]$
3	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,9]$ $y \mapsto [0,+\infty]$	$x \mapsto [0,9]$ $y \mapsto [0,+\infty]$
4	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto [1,1]$ $y \mapsto [0,0]$	$x \mapsto [1,10]$ $y \mapsto [0,+\infty]$	$x \mapsto [1,10]$ $y \mapsto [0,+\infty]$
5	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto [1,1]$ $y \mapsto [1,1]$	$x \mapsto [1,10]$ $y \mapsto [1,+\infty]$	$x \mapsto [1,10]$ $y \mapsto [1,+\infty]$
6	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto [0,0]$	$x \mapsto [10,+\infty]$ $y \mapsto [0,+\infty]$	$x \mapsto [10,+\infty]$ $y \mapsto [0,+\infty]$

Fixed Point Computation with Widening and Narrowing



Node	initial	1	2
1	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$	$x \mapsto \bot$ $y \mapsto \bot$
2	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,0]$ $y \mapsto [0,0]$	$x \mapsto [0,0]$ $y \mapsto [0,0]$
3	$x \mapsto [0,9]$ $y \mapsto [0,+\infty]$	$x \mapsto [0,9]$ $y \mapsto [0,+\infty]$	$x \mapsto [0,9]$ $y \mapsto [0,+\infty]$
4	$x \mapsto [1,10]$ $y \mapsto [0,+\infty]$	$x \mapsto [1,10]$ $y \mapsto [0,+\infty]$	$x \mapsto [1,10]$ $y \mapsto [0,+\infty]$
5	$x \mapsto [1,10]$ $y \mapsto [1,+\infty]$	$x \mapsto [1,10]$ $y \mapsto [1,+\infty]$	$x \mapsto [1,10]$ $y \mapsto [1,+\infty]$
6	$x \mapsto [10,+\infty]$ $y \mapsto [0,+\infty]$	$x \mapsto [10,10]$ $y \mapsto [0,+\infty]$	$x \mapsto [10,10]$ $y \mapsto [0,+\infty]$

Programs

A program is represented by a control-flow graph:

$$(\mathbb{C}, \rightarrow)$$

- $\mathbb{C}$: the set of program points (i.e., nodes) in the program
- $(\rightarrow) \subseteq \mathbb{C} \times \mathbb{C}$: the control-flow relation
  - $c \rightarrow c'$: c is a predecessor of c'
- Each program point c is associated with a command, denoted $\mathbf{cmd}(c)$

$$\begin{array}{rcl} cmd & \rightarrow & skip \mid x := e \mid x < n \\ e & \rightarrow & n \mid x \mid e + e \mid e - e \mid e * e \mid e/e \end{array}$$

Interval Domain

- Definition:

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

- An interval is an abstraction of a set of integers:
  - $\gamma([1, 5]) =$
  - $\gamma([3, 3]) =$
  - $\gamma([0, +\infty]) =$
  - $\gamma([-\infty, 7]) =$
  - $\gamma(\bot) =$

Concretization/Abstraction Functions

- $\gamma : \mathbb{I} \to \wp(\mathbb{Z})$ is called *concretization function*:

$$\begin{array}{rcl} \gamma(\bot) & = & \emptyset \\ \gamma([a, b]) & = & \{z \in \mathbb{Z} \mid a \leq z \leq b\} \end{array}$$

- $\alpha : \wp(\mathbb{Z}) \to \mathbb{I}$ is *abstraction function*:
  - $\alpha(\{2\}) =$
  - $\alpha(\{-1, 0, 1, 2, 3\}) =$
  - $\alpha(\{-1, 3\}) =$
  - $\alpha(\{1, 2, \ldots\}) =$
  - $\alpha(\emptyset) =$
  - $\alpha(\mathbb{Z}) =$

$$\begin{array}{rcl} \alpha(\emptyset) & = & \bot \\ \alpha(S) & = & [\min(S), \max(S)] \end{array}$$

Partial Order ($\sqsubseteq$) $\subseteq \mathbb{I} \times \mathbb{I}$

- $\bot \sqsubseteq i$ for all $i \in \mathbb{I}$
- $i \sqsubseteq [-\infty, +\infty]$ for all $i \in \mathbb{I}$.
- $[1, 3] \sqsubseteq [0, 4]$
- $[1, 3] \not\sqsubseteq [0, 2]$

Definition:

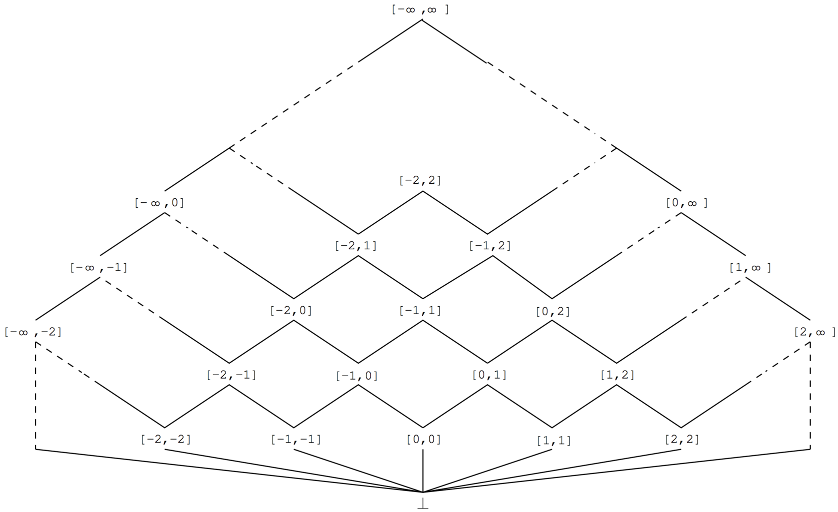
- Mathematical:

$$i_1 \sqsubseteq i_2 \text{ iff } \gamma(i_1) \subseteq \gamma(i_2)$$

- Implementable:

$$i_1 \sqsubseteq i_2 \text{ iff } \begin{cases} i_1 = \bot \lor \\ i_2 = [-\infty, +\infty] \lor \\ (i_1 = [l_1, u_1] \ \land \ i_2 = [l_2, u_2] \ \land \ l_1 \geq l_2 \ \land \ u_1 \leq u_2) \end{cases}$$

Partial Order



Join $\sqcup$ and Meet $\sqcap$ Operators

- The join operator computes the *least upper bound*:
  - $[1,3] \sqcup [2,4] = [1,4]$
  - $[1,3] \sqcup [7,9] = [1,9]$
- The conditions of $i_1 \sqcup i_2$:
  1. $i_1 \sqsubseteq i_1 \sqcup i_2 \;\wedge\; i_2 \sqsubseteq i_1 \sqcup i_2$
  2. $\forall i.\ i_1 \sqsubseteq i \;\wedge\; i_2 \sqsubseteq i \implies i_1 \sqcup i_2 \sqsubseteq i$
- Definition:

$$i_1 \sqcup i_2 = \alpha(\gamma(i_1) \cup \gamma(i_2))$$

$$\begin{array}{rcl} \bot \sqcup i & = & i \\ i \sqcup \bot & = & i \\ [l_1, u_1] \sqcup [l_2, u_2] & = & [\min(l_1, l_2), \max(l_1, l_2)] \end{array}$$

Join $\sqcup$ and Meet $\sqcap$ Operators

- The meet operator computes the *greatest lower bound*:
  - $[1,3] \sqcap [2,4] = [2,3]$
  - $[1,3] \sqcap [7,9] = \bot$
- The conditions of $i_1 \sqcap i_2$:
  1. $i_1 \sqsubseteq i_1 \sqcup i_2 \ \wedge \ i_2 \sqsubseteq i_1 \sqcup i_2$
  2. $\forall i.\ i \sqsubseteq i_1 \ \wedge \ i \sqsubseteq i_2 \implies i \sqsubseteq i_1 \sqcap i_2$
- Definition:

$$i_1 \sqcap i_2 = \alpha(\gamma(i_1) \cap \gamma(i_2))$$

$$\begin{array}{rcl} \bot \sqcap i & = & \bot \\ i \sqcap \bot & = & \bot \\ [l_1, u_1] \sqcap [l_2, u_2] & = & \begin{cases} \bot & \max(l_1, l_2) > \min(l_1, l_2) \\ [\max(l_1, l_2), \min(l_1, l_2)] & \text{o.w.} \end{cases} \end{array}$$

Widening and Narrowing

A simple widening operator for the Interval domain:

$$\begin{array}{rcll} [a,b] & \triangledown & \bot & = [a,b] \\ \bot & \triangledown & [c,d] & = [c,d] \\ [a,b] & \triangledown & [c,d] & = [(c < a ? -\infty : a), (b < d ? +\infty : b)] \end{array}$$

A simple narrowing operator:

$$\begin{array}{rcll} [a,b] & \triangle & \bot & = \bot \\ \bot & \triangle & [c,d] & = \bot \\ [a,b] & \triangle & [c,d] & = [(a = -\infty ? c : a), (b = +\infty ? d : b)] \end{array}$$

Abstract States

$$\mathbb{S} = \textbf{Var} \rightarrow \mathbb{I}$$

Partial order, join, meet, widening, and narrowing are lifted pointwise:

$$s_1 \sqsubseteq s_2 \text{ iff } \forall x \in \textbf{Var}.\ s_1(x) \sqsubseteq s_2(x)$$

$$s_1 \sqcup s_2 = \lambda x.\ s_1(x) \sqcup s_2(x)$$

$$s_1 \sqcap s_2 = \lambda x.\ s_1(x) \sqcap s_2(x)$$

$$s_1 \bigtriangledown s_2 = \lambda x.\ s_1(x) \bigtriangledown s_2(x)$$

$$s_1 \bigtriangleup s_2 = \lambda x.\ s_1(x) \bigtriangleup s_2(x)$$

The Abstract Domain

$$\mathbb{D} = \mathbb{C} \to \mathbb{S}$$

Partial order, join, meet, widening, and narrowing are lifted pointwise:

$$d_1 \sqsubseteq d_2 \text{ iff } \forall c \in \mathbb{C}.\ d_1(x) \sqsubseteq d_2(x)$$

$$d_1 \sqcup d_2 = \lambda c.\ d_1(c) \sqcup d_2(c)$$

$$d_1 \sqcap d_2 = \lambda c.\ d_1(c) \sqcap d_2(c)$$

$$d_1 \bigtriangledown d_2 = \lambda c.\ d_1(c) \bigtriangledown d_2(c)$$

$$d_1 \bigtriangleup d_2 = \lambda c.\ d_1(c) \bigtriangleup d_2(c)$$

Abstract Semantics of Expressions

$$e \;\to\; n \mid x \mid e+e \mid e-e \mid e*e \mid e/e$$

$$\begin{array}{rcl}
eval & : & e \times \mathbb{S} \to \mathbb{I} \\
eval(n, s) & = & [n, n] \\
eval(x, s) & = & s(x) \\
eval(e_1 + e_2, s) & = & eval(e_1, s) \hat{+} eval(e_2, s) \\
eval(e_1 - e_2, s) & = & eval(e_1, s) \hat{-} eval(e_2, s) \\
eval(e_1 * e_2, s) & = & eval(e_1, s) \hat{*} eval(e_2, s) \\
eval(e_1 / e_2, s) & = & eval(e_1, s) \hat{/} eval(e_2, s)
\end{array}$$

Abstract Binary Operators

$$\begin{array}{rcl} i_1 \hat{+} i_2 & = & \alpha(\{z_1 + z_2 \mid z_1 \in \gamma(i_1) \ \wedge \ z_2 \in \gamma(i_2)\}) \\ i_1 \hat{-} i_2 & = & \alpha(\{z_1 - z_2 \mid z_1 \in \gamma(i_1) \ \wedge \ z_2 \in \gamma(i_2)\}) \\ i_1 \hat{*} i_2 & = & \alpha(\{z_1 * z_2 \mid z_1 \in \gamma(i_1) \ \wedge \ z_2 \in \gamma(i_2)\}) \\ i_1 \hat{/} i_2 & = & \alpha(\{z_1 / z_2 \mid z_1 \in \gamma(i_1) \ \wedge \ z_2 \in \gamma(i_2)\}) \end{array}$$

Implementable version:

$$\begin{array}{rcl} \bot \hat{+} i & = & \\ i \hat{+} \bot & = & \\ [l_1, u_1] \hat{+} [l_2, u_2] & = & \\ [l_1, u_1] \hat{-} [l_2, u_2] & = & \\ [l_1, u_1] \hat{*} [l_2, u_2] & = & \\ [l_1, u_1] \hat{/} [l_2, u_2] & = & \end{array}$$

Abstract Execution of Commands

$$f_c : \mathbb{S} \to \mathbb{S}$$

$$f_c(s) = \begin{cases} s & \textsf{cmd}(c) = skip \\ [x \mapsto eval(e, s)]s & \textsf{cmd}(c) = x := e \\ [x \mapsto s(x) \sqcap [-\infty, n-1]]s & \textsf{cmd}(c) = x < n \end{cases}$$

Fixed Point Equation

We aim to compute

$$X : \mathbb{C} \to \mathbb{S}$$

such that

$$X = \lambda c.\ f_c(\bigsqcup_{c' \to c} X(c'))$$

In fixed point form:

$$X = F(X)$$

where

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

The solution of the equation is a fixed point of

$$F : (\mathbb{C} \to \mathbb{S}) \to (\mathbb{C} \to \mathbb{S})$$

Fixed Point Computation

The least fixed point computation may not converge:

$$\mathit{fix} F = \bigsqcup_{i \in \mathbb{N}} F^i(\bot) = F^0(\bot) \sqcup F^1(\bot) F^2(\bot) \sqcup \cdots$$

Instead, we aim to find a (not necessarily least) fixed point with widening and narrowing:

1. widening iteration:

$$\begin{array}{rcll} X_0 & = & \bot & \\ X_i & = & X_{i-1} & \text{if } F(X_{i-1}) \sqsubseteq X_{i-1} \\ & = & X_{i-1} \bigtriangledown F(X_{i-1}) & \text{otherwise} \end{array}$$

2. narrowing iteration:

$$Y_i = \begin{cases} \hat{A} & \text{if } i = 0 \\ Y_{i-1} \bigtriangleup F(Y_{i-1}) & \text{if } i > 0 \end{cases} \quad (1)$$

($\hat{A}$ is the result from the widening iteration, i.e., $\lim_i X_i$)

Need for Static Analysis Theory

Static analyses so far are based on our intuition. Questions remain:

- How to ensure that the abstract semantics is sound?
- How to ensure the soundness of widening and narrowing?
- How to ensure the termination of widening and narrowing?

Next: Abstract Interpretation Theory