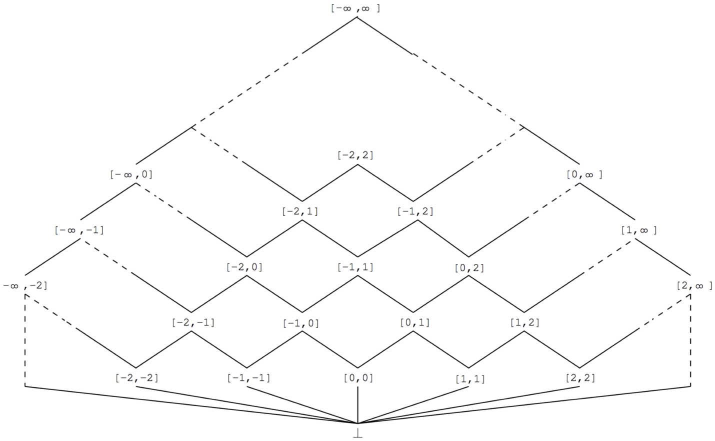
COSE312: Compilers

Lecture 14 — Semantic Analysis (2)

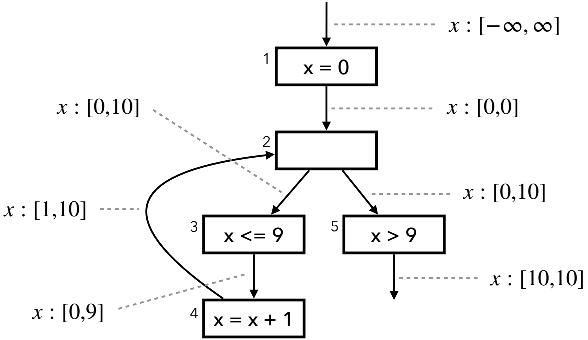
Hakjoo Oh
2025 Spring

Interval Domain

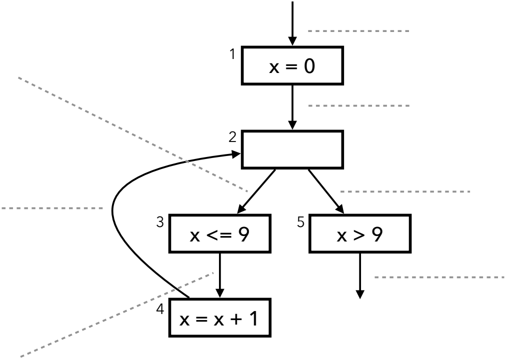


Example Program

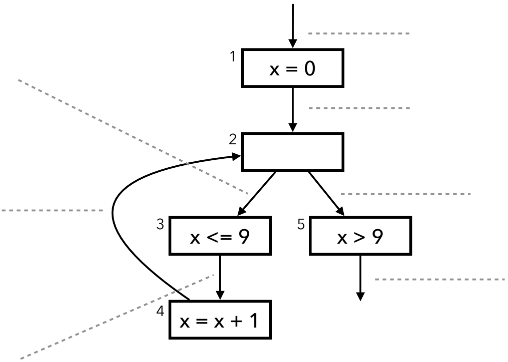
```
x = 0;

while (x <= 9)
  x = x + 1;
```



Fixed Point Computation



Fixed Point Computation w/ Widening and Narrowing



Interval Domain

- Concrete integers ($\mathbb{Z}$) are abstracted by the complete lattice $(\widehat{\mathbb{Z}}, \sqsubseteq_{\widehat{\mathbb{Z}}})$:

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

$$\bot \sqsubseteq_{\widehat{\mathbb{Z}}} \hat{z} \ (\forall \hat{z} \in \widehat{\mathbb{Z}}) \quad [l_1, u_1] \sqsubseteq_{\widehat{\mathbb{Z}}} [l_2, u_2] \iff l_2 \leq l_1 \wedge u_1 \leq u_2$$

- Abstraction and concretization functions:

$$\begin{array}{rclrcl} \alpha_{\widehat{\mathbb{Z}}}(\emptyset) & = & \bot_{\widehat{\mathbb{Z}}} & \gamma_{\widehat{\mathbb{Z}}}(\bot_{\widehat{\mathbb{Z}}}) & = & \emptyset \\ \alpha_{\widehat{\mathbb{Z}}}(S) & = & [\min(S), \max(S)] & \gamma_{\widehat{\mathbb{Z}}}([l, u]) & = & \{z \in \mathbb{Z} \mid l \leq z \leq u\} \end{array}$$

- Join and meet:

$$\begin{array}{rcl} \bot \sqcup_{\widehat{\mathbb{Z}}} \hat{z} & = & \hat{z} \\ \hat{z} \sqcup_{\widehat{\mathbb{Z}}} \bot & = & \hat{z} \\ [l_1, u_1] \sqcup_{\widehat{\mathbb{Z}}} [l_2, u_2] & = & [\min(l_1, l_2), \max(u_1, u_2)] \\ \\ \bot \sqcap_{\widehat{\mathbb{Z}}} \hat{z} & = & \bot \\ \hat{z} \sqcap_{\widehat{\mathbb{Z}}} \bot & = & \bot \\ [l_1, u_1] \sqcap_{\widehat{\mathbb{Z}}} [l_2, u_2] & = & [l_2, u_1] \ (l_1 \leq l_2 \wedge l_2 \leq u_1) \\ [l_1, u_1] \sqcap_{\widehat{\mathbb{Z}}} [l_2, u_2] & = & [l_1, u_2] \ (l_2 \leq l_1 \wedge l_1 \leq u_2) \\ [l_1, u_1] \sqcap_{\widehat{\mathbb{Z}}} [l_2, u_2] & = & \bot \ \text{(otherwise)} \end{array}$$

Interval Domain

- Widening:

$$\begin{array}{rcl} \perp \bigtriangledown_{\widehat{\mathbb{Z}}} \hat{z} & = & \hat{z} \\ \hat{z} \bigtriangledown_{\widehat{\mathbb{Z}}} \perp & = & \hat{z} \\ [l_1, u_1] \bigtriangledown_{\widehat{\mathbb{Z}}} [l_2, u_2] & = & [l_1 > l_2 ?\ -\infty : l_1, u_1 < u_2 ?\ \infty : u_1] \end{array}$$

- Narrowing:

$$\begin{array}{rcl} \perp \bigtriangleup_{\widehat{\mathbb{Z}}} \hat{z} & = & \perp \\ \hat{z} \bigtriangleup_{\widehat{\mathbb{Z}}} \perp & = & \perp \\ [l_1, u_1] \bigtriangleup_{\widehat{\mathbb{Z}}} [l_2, u_2] & = & [l_1 = -\infty ?\ l_2 : l_1, u_1 = \infty ?\ u_2 : u_1] \end{array}$$

Abstract Booleans

- The truth values $\mathbf{T} = \{true, false\}$ are abstracted by $(\widehat{\mathbf{T}}, \sqsubseteq_{\widehat{\mathbf{T}}})$:

$$\widehat{\mathbf{T}} = \{\top_{\widehat{\mathbf{T}}}, \bot_{\widehat{\mathbf{T}}}, \widehat{true}, \widehat{false}\}$$

$$\hat{b}_1 \sqsubseteq_{\widehat{\mathbf{T}}} \hat{b}_2 \iff \hat{b}_1 = \hat{b}_2 \lor \hat{b}_1 = \bot_{\widehat{\mathbf{T}}} \lor \hat{b}_2 = \top_{\widehat{\mathbf{T}}}$$

- An abstract boolean denotes a set of concrete booleans:

$$\begin{array}{rcl} \alpha_{\widehat{\mathbf{T}}} & : & \mathcal{P}(\mathbf{T}) \to \widehat{\mathbf{T}} \\ \alpha_{\widehat{\mathbf{T}}}(\emptyset) & = & \bot_{\widehat{\mathbf{T}}} \\ \alpha_{\widehat{\mathbf{T}}}(\{true\}) & = & \widehat{true} \\ \alpha_{\widehat{\mathbf{T}}}(\{false\}) & = & \widehat{false} \\ \alpha_{\widehat{\mathbf{T}}}(\mathbf{T}) & = & \top_{\widehat{\mathbf{T}}} \end{array} \qquad \begin{array}{rcl} \gamma_{\widehat{\mathbf{T}}} & : & \widehat{\mathbf{T}} \to \mathcal{P}(\mathbf{T}) \\ \gamma_{\widehat{\mathbf{T}}}(\bot_{\widehat{\mathbf{T}}}) & = & \emptyset \\ \gamma_{\widehat{\mathbf{T}}}(\widehat{true}) & = & \{true\} \\ \gamma_{\widehat{\mathbf{T}}}(\widehat{false}) & = & \{false\} \\ \gamma_{\widehat{\mathbf{T}}}(\top_{\widehat{\mathbf{T}}}) & = & \mathbf{T} \end{array}$$

- Join and meet:

$$\begin{array}{l} \hat{a} \sqcup_{\widehat{\mathbf{T}}} \hat{b} = \hat{a} \ (\hat{b} \sqsubseteq_{\widehat{\mathbf{T}}} \hat{a}) \\ \hat{a} \sqcup_{\widehat{\mathbf{T}}} \hat{b} = \hat{b} \ (\hat{a} \sqsubseteq_{\widehat{\mathbf{T}}} \hat{b}) \\ \hat{a} \sqcup_{\widehat{\mathbf{T}}} \hat{b} = \top_{\widehat{\mathbf{T}}} \end{array} \qquad \begin{array}{l} \hat{a} \sqcap_{\widehat{\mathbf{T}}} \hat{b} = \hat{b} \ (\hat{b} \sqsubseteq_{\widehat{\mathbf{T}}} \hat{a}) \\ \hat{a} \sqcap_{\widehat{\mathbf{T}}} \hat{b} = \hat{a} \ (\hat{a} \sqsubseteq_{\widehat{\mathbf{T}}} \hat{b}) \\ \hat{a} \sqcap_{\widehat{\mathbf{T}}} \hat{b} = \bot_{\widehat{\mathbf{T}}} \end{array}$$

Abstract States

- Concrete states ($\mathbf{State}$) are abstracted by $(\widehat{\mathbf{State}}, \sqsubseteq_{\widehat{\mathbf{State}}})$:

$$\widehat{\mathbf{State}} = \mathbf{Var} \to \widehat{\mathbb{Z}}$$

$$\hat{s}_1 \sqsubseteq_{\widehat{\mathbf{State}}} \hat{s}_2 \iff \forall x \in \mathbf{Var}.\ \hat{s}_1(x) \sqsubseteq_{\widehat{\mathbb{Z}}} \hat{s}_2(x).$$

- An abstract state denotes a set of concrete states:

$$\begin{array}{rcl} \alpha_{\widehat{\mathbf{State}}} & : & \mathcal{P}(\mathbf{State}) \to \widehat{\mathbf{State}} \\ \alpha_{\widehat{\mathbf{State}}}(S) & = & \lambda x.\ \bigsqcup_{s \in S} \alpha_{\widehat{\mathbb{Z}}}(\{s(x)\}) \\ & & \\ \gamma_{\widehat{\mathbf{State}}} & = & \widehat{\mathbf{State}} \to \mathcal{P}(\mathbf{State}) \\ \gamma_{\widehat{\mathbf{State}}}(\hat{s}) & = & \{s \in \mathbf{State} \mid \forall x \in \mathbf{Var}.\ s(x) \in \gamma_{\widehat{\mathbb{Z}}}(\hat{s}(x))\} \end{array}$$

- Join and meet:

$$\begin{array}{c} \hat{s}_1 \sqcup_{\widehat{\mathbf{State}}} \hat{s}_2 = \lambda x.\ \hat{s}_1(x) \sqcup_{\widehat{\mathbb{Z}}} \hat{s}_2(x) \\ \hat{s}_1 \sqcap_{\widehat{\mathbf{State}}} \hat{s}_2 = \lambda x.\ \hat{s}_1(x) \sqcap_{\widehat{\mathbb{Z}}} \hat{s}_2(x) \end{array}$$

- Widening and narrowing:

$$\begin{array}{c} \hat{s}_1 \bigtriangledown_{\widehat{\mathbf{State}}} \hat{s}_2 = \lambda x.\ \hat{s}_1(x) \bigtriangledown_{\widehat{\mathbb{Z}}} \hat{s}_2(x) \\ \hat{s}_1 \bigtriangleup_{\widehat{\mathbf{State}}} \hat{s}_2 = \lambda x.\ \hat{s}_1(x) \bigtriangleup_{\widehat{\mathbb{Z}}} \hat{s}_2(x) \end{array}$$

Abstract Semantics

$$
\begin{array}{rcl}
\widehat{\mathcal{A}}[\![a]\!] & : & \widehat{\textbf{State}} \to \widehat{\mathbb{Z}} \\
\widehat{\mathcal{A}}[\![n]\!](\hat{s}) & = & \alpha_{\widehat{\mathbb{Z}}}(\{n\}) \\
\widehat{\mathcal{A}}[\![x]\!](\hat{s}) & = & \hat{s}(x) \\
\widehat{\mathcal{A}}[\![a_1 + a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) +_{\widehat{\mathbb{Z}}} \widehat{\mathcal{A}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{A}}[\![a_1 \star a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) \star_{\widehat{\mathbb{Z}}} \widehat{\mathcal{A}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{A}}[\![a_1 - a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) -_{\widehat{\mathbb{Z}}} \widehat{\mathcal{A}}[\![a_2]\!](\hat{s})
\end{array}
$$

$$
\begin{array}{rcl}
\widehat{\mathcal{B}}[\![b]\!] & : & \widehat{\textbf{State}} \to \widehat{\mathbf{T}} \\
\widehat{\mathcal{B}}[\![\texttt{true}]\!](\hat{s}) & = & \widehat{true} \\
\widehat{\mathcal{B}}[\![\texttt{false}]\!](\hat{s}) & = & \widehat{false} \\
\widehat{\mathcal{B}}[\![a_1 = a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) =_{\widehat{\mathbb{Z}}} \widehat{\mathcal{A}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{B}}[\![a_1 \leq a_2]\!](\hat{s}) & = & \widehat{\mathcal{A}}[\![a_1]\!](\hat{s}) \leq_{\widehat{\mathbb{Z}}} \widehat{\mathcal{A}}[\![a_2]\!](\hat{s}) \\
\widehat{\mathcal{B}}[\![\neg b]\!](\hat{s}) & = & \neg_{\widehat{\mathbf{T}}} \widehat{\mathcal{B}}[\![b]\!](\hat{s}) \\
\widehat{\mathcal{B}}[\![b_1 \wedge b_2]\!](\hat{s}) & = & \widehat{\mathcal{B}}[\![b_1]\!](\hat{s}) \wedge_{\widehat{\mathbf{T}}} \widehat{\mathcal{B}}[\![b_2]\!](\hat{s})
\end{array}
$$

Abstract Semantics

- Addition / subtraction / multiplication:

$$\begin{array}{rcl} [l_1, u_1] +_{\widehat{\mathbb{Z}}} [l_2, u_2] & = & [l_1 + l_2, u_1 + u_2] \\ [l_1, u_1] -_{\widehat{\mathbb{Z}}} [l_2, u_2] & = & [l_1 - u_2, u_1 - l_2] \\ [l_1, u_1] \star_{\widehat{\mathbb{Z}}} [l_2, u_2] & = & [\min(l_1 l_2, l_1 u_2, u_1 l_2, u_1 u_2), \max(\cdots)] \end{array}$$

- Equality:

$$[l_1, u_1] =_{\widehat{\mathbb{Z}}} [l_2, u_2] = \begin{cases} \widehat{true} & \text{if } l_1 = u_1 = l_2 = u_2 \\ \widehat{false} & \text{if no overlap} \\ \top & \text{otherwise} \end{cases}$$

- Comparison:

$$[l_1, u_1] \leq_{\widehat{\mathbb{Z}}} [l_2, u_2] = \begin{cases} \widehat{true} & \text{if } u_1 \leq l_2 \\ \widehat{false} & \text{if } l_1 > u_2 \\ \top & \text{otherwise} \end{cases}$$

Abstract Semantics

- Control-flow graph $G = (N, \hookrightarrow)$ with commands, i.e., $cmd(n)$:

$$c \to x := a \mid assume(b) \mid skip$$

- Transfer function $\hat{f}_n : \widehat{\mathbf{State}} \to \widehat{\mathbf{State}}$:

$$\hat{f}_n(\hat{s}) = \begin{cases} \hat{s} & \text{if } cmd(n) = skip \\ \hat{s}[x \mapsto \hat{\mathcal{A}}[\![\, a \,]\!](\hat{s})] & \text{if } x := a \\ \textbf{Prune}_b(\hat{s}) & \text{if } assume(b) \end{cases}$$

where $\textbf{Prune}_b : \widehat{\mathbf{State}} \to \widehat{\mathbf{State}}$ computes a *pruned* abstract state such that

$$\alpha_{\widehat{\mathbf{State}}}(\{s \in \gamma_{\widehat{\mathbf{State}}}(\hat{s}) \mid \mathcal{B}[\![\, b \,]\!](s)\}) \sqsubseteq \textbf{Prune}_b(\hat{s}) \sqsubseteq \hat{s}.$$

- The analysis is to compute the least fixed point of the function:

$$\widehat{F} : (N \to \widehat{\mathbf{State}}) \to (N \to \widehat{\mathbf{State}})$$

$$\widehat{F}(X) = \lambda n.\ \hat{f}_n\Big(\bigsqcup_{n' \hookrightarrow n} X(n')\Big)$$

Fixed Point Computation

Widening phase	Narrowing phase

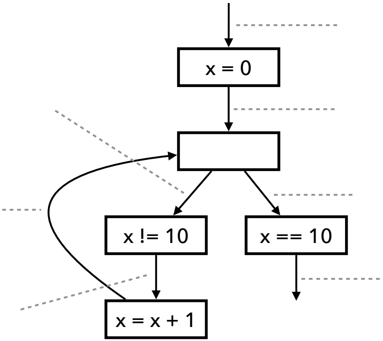
$W := N$
$X := \lambda n.\bot$
`repeat`
  $n :=$ **choose**(W)
  $W := W \setminus \{n\}$
  $s := \hat{f}_n(\bigsqcup_{n' \hookrightarrow n} X(n'))$
  `if` $s \not\sqsubseteq X(n)$
    if widening is needed
      $X(n) := X(n) \bigtriangledown s$
    `else`
      $X(n) := X(n) \sqcup s$
    $W := W \cup \{n' \mid n \hookrightarrow n'\}$
`until` $W = \emptyset$
	$W := N$
	`repeat`
	  $n :=$ **choose**(W)
	  $W := W \setminus \{n\}$
	  $s := \hat{f}_n(\bigsqcup_{n' \hookrightarrow n} X(n'))$
	  `if` $X(n) \not\sqsubseteq s$
	    $X(n) := X(n) \bigtriangleup s$
	    $W := W \cup \{n' \mid n \hookrightarrow n'\}$
	`until` $W = \emptyset$

Exercise 1

Describe the interval analysis on the program:

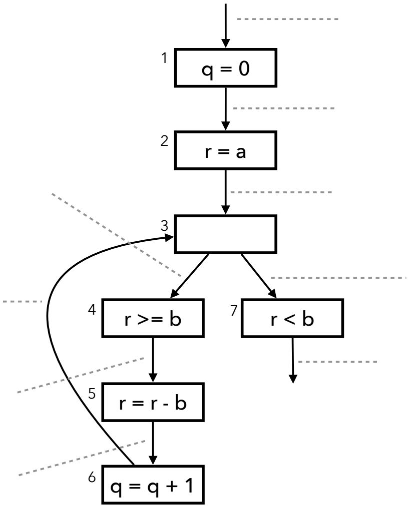
1. without widening and
2. with widening/narrowing

```
x = 0;

while (x != 10)
  x = x + 1;
```



Exercise 2

Describe the interval analysis on the program:



Goal: A Static Analyzer for S Based on the Interval Domain

$$
\begin{array}{rcll}
\mathit{program} & \rightarrow & \mathit{block} & \\
\mathit{block} & \rightarrow & \mathit{decls}\ \mathit{stmts} & \\
\mathit{decls} & \rightarrow & \mathit{decls}\ \mathit{decl} \mid \epsilon & \\
\mathit{decl} & \rightarrow & \mathit{type}\ x & \\
\mathit{type} & \rightarrow & \texttt{int} \mid \texttt{int[}n\texttt{]} & \\
\mathit{stmts} & \rightarrow & \mathit{stmts}\ \mathit{stmt} \mid \epsilon & \\
\\
\mathit{stmt} & \rightarrow & \mathit{lv}\ \texttt{=}\ e & \\
 & \mid & \texttt{if}\ e\ \mathit{stmt}\ \mathit{stmt} & \\
 & \mid & \texttt{while}\ e\ \mathit{stmt} & \\
 & \mid & \texttt{do}\ \mathit{stmt}\ \texttt{while}\ e & \\
 & \mid & \texttt{read}\ x & \\
 & \mid & \texttt{print}\ e & \\
 & \mid & \mathit{block} & \\
\\
\mathit{lv} & \rightarrow & x \mid x\texttt{[}e\texttt{]} & \\
\\
e & \rightarrow & n & \text{integer} \\
 & \mid & \mathit{lv} & \text{l-value} \\
 & \mid & e\texttt{+}e \mid e\texttt{-}e \mid e\texttt{*}e \mid e\texttt{/}e \mid \texttt{-}e & \text{airthmetic operation} \\
 & \mid & e\texttt{==}e \mid e\texttt{<}e \mid e\texttt{<=}e \mid e\texttt{>}e \mid e\texttt{>=}e & \text{conditional operation} \\
 & \mid & \texttt{!}e \mid e\texttt{||}e \mid e\texttt{\&\&}e & \text{boolean operation}
\end{array}
$$

Control-Flow Graph

- $G = (N, \hookrightarrow)$ where each node $n \in N$ contains a command:

$$c \to x = alloc(n) \mid lv = e \mid assume(e) \mid skip \mid read\ x \mid print\ e$$

- Concrete domain

$$\begin{array}{rcl} l \in Loc & = & Var + Addr \times Offset \\ v \in Value & = & \mathbb{Z} + Addr \times Size \\ Offset & = & \mathbb{N} \\ Size & = & \mathbb{N} \\ m \in Mem & = & Loc \to Value \\ a \in Addr & = & \text{Address} \end{array}$$

- Concrete semantics

$$\begin{array}{rcl} \mathcal{L}(lv) & : & Mem \to Loc \\ \mathcal{E}(e) & : & Mem \to Value \\ f_n & : & Mem \hookrightarrow Mem \end{array}$$

Abstraction of Memory Objects

Memory locations are unbounded. In typical static analysis, arrays are abstracted by their allocation sites, without distinguishing elements.

1. ```
   int i;
   int[10] arr;
   i = 1;
   arr[i] = 2;
   ```
2. ```
   int i;
   int[10] a;
   int[2] b;
   a[0] = 1;
   a[a[0]] = 2;
   b[a[0]] = 3;
   ```

Abstract Semantics

- An abstract state maps abstract locations ($\widehat{Loc}$) to values ($\widehat{Val}$):

$$
\begin{array}{rcl}
\hat{l} \in \widehat{Loc} & = & Var + AllocSite \\
\hat{v} \in \widehat{Val} & = & \widehat{\mathbb{Z}} \times \widehat{Array} \\
\widehat{\mathbb{Z}} & = & \{\bot\} \cup \{[l, u] \mid l, u \in \mathbb{Z} \cup \{-\infty, \infty\}, l \leq u\} \\
\widehat{Array} & = & \mathcal{P}(AllocSite) \times \widehat{\mathbb{Z}} \\
\hat{m} \in \widehat{Mem} & = & \widehat{Loc} \to \widehat{Val}
\end{array}
$$

- The analysis is to compute the least fixed point of the function:

$$\widehat{F} : (N \to \widehat{Mem}) \to (N \to \widehat{Mem})$$

$$\widehat{F}(X) = \lambda n.\ \hat{f}_n\Big(\bigsqcup_{n' \hookrightarrow n} X(n')\Big)$$

Abstract Semantics

- An l-value evaluates to a set of abstract locations:

$$\begin{array}{rcl} \widehat{\mathcal{L}}(lv) & : & \widehat{Mem} \to \mathcal{P}(\widehat{Loc}) \\ \widehat{\mathcal{L}}(x)(\hat{m}) & = & \{x\} \\ \widehat{\mathcal{L}}(x[e])(\hat{m}) & = & \hat{m}(x).2.1 \end{array}$$

- An expression evaluates to an abstract value:

$$\begin{array}{rcl} \widehat{\mathcal{E}}(e) & : & \widehat{Mem} \to \widehat{Val} \\ \widehat{\mathcal{E}}(n)(\hat{m}) & = & \langle [n,n], \bot_{\widehat{Array}} \rangle \\ \widehat{\mathcal{E}}(lv)(\hat{m}) & = & \bigsqcup_{\hat{l} \in \widehat{\mathcal{L}}(lv)(\hat{m})} \hat{m}(\hat{l}) \\ \widehat{\mathcal{E}}(e_1 + e_2)(\hat{m}) & = & \widehat{\mathcal{E}}(e_1)(\hat{m}) +_{\widehat{Val}} \widehat{\mathcal{E}}(e_2)(\hat{m}) \end{array}$$

- Transfer function: $\hat{f}_n(\hat{m}) =$

$$\begin{cases} \hat{m}[x \mapsto \widehat{\mathcal{E}}(e)(\hat{m})] & \text{if } lv := e, \mathcal{L}(lv)(\hat{m}) = \{x\} \\ \bigsqcup_{\hat{l} \in \widehat{\mathcal{L}}(lv)(\hat{m})} \hat{m}[\hat{l} \mapsto \hat{m}(\hat{l}) \sqcup \widehat{\mathcal{E}}(e)(\hat{m})] & \text{if } lv := e, \widehat{\mathcal{L}}(lv)(\hat{m}) = \dots \\ \textbf{Prune}_e(\hat{m}) & \text{if } assume(e) \end{cases}$$

Summary

- Interval abstract domain
- Fixed point computation with widening and narrowing
- Interval domain-based static analysis for S