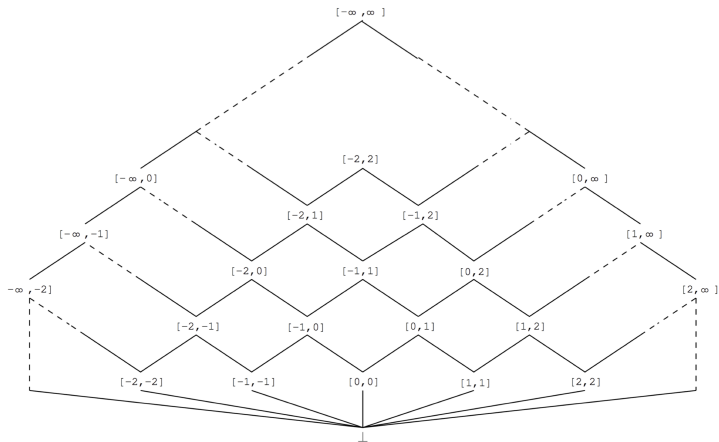


COSE312: Compilers

Lecture 14 — Semantic Analysis (2)

Hakjoo Oh
2025 Spring

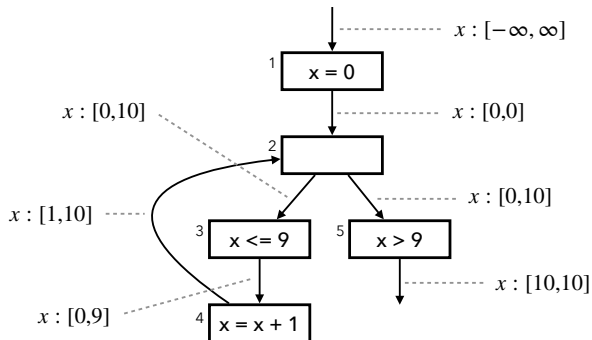
Interval Domain



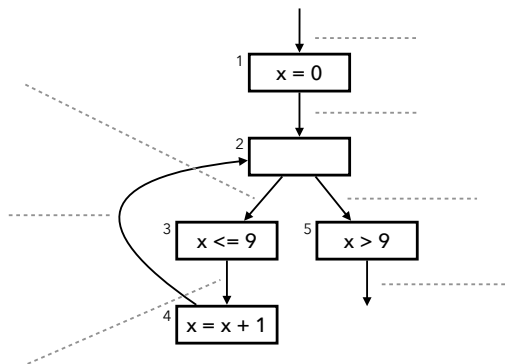
Example Program

```
x = 0;
```

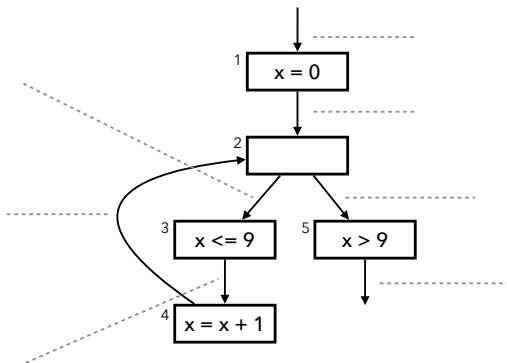
```
while (x <= 9)  
  x = x + 1;
```



Fixed Point Computation



Fixed Point Computation w/ Widening and Narrowing



Interval Domain

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

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

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

- Abstraction and concretization functions:

$$\begin{aligned} \alpha_{\hat{\mathbb{Z}}}(\emptyset) &= \perp_{\hat{\mathbb{Z}}} & \gamma_{\hat{\mathbb{Z}}}(\perp_{\hat{\mathbb{Z}}}) &= \emptyset \\ \alpha_{\hat{\mathbb{Z}}}(S) &= [\min(S), \max(S)] & \gamma_{\hat{\mathbb{Z}}}([l, u]) &= \{z \in \mathbb{Z} \mid l \leq z \leq u\} \end{aligned}$$

- Join and meet:

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

Interval Domain

- Widening:

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

- Narrowing:

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

Abstract Booleans

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

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

$$\hat{b}_1 \sqsubseteq_{\hat{\mathbf{T}}} \hat{b}_2 \iff \hat{b}_1 = \hat{b}_2 \vee \hat{b}_1 = \perp_{\hat{\mathbf{T}}} \vee \hat{b}_2 = \top_{\hat{\mathbf{T}}}$$

- An abstract boolean denotes a set of concrete booleans:

$\alpha_{\hat{\mathbf{T}}} : \mathcal{P}(\mathbf{T}) \rightarrow \hat{\mathbf{T}}$	$\gamma_{\hat{\mathbf{T}}} : \hat{\mathbf{T}} \rightarrow \mathcal{P}(\mathbf{T})$
$\alpha_{\hat{\mathbf{T}}}(\emptyset) = \perp_{\hat{\mathbf{T}}}$	$\gamma_{\hat{\mathbf{T}}}(\perp_{\hat{\mathbf{T}}}) = \emptyset$
$\alpha_{\hat{\mathbf{T}}}(\{true\}) = \widehat{true}$	$\gamma_{\hat{\mathbf{T}}}(\widehat{true}) = \{true\}$
$\alpha_{\hat{\mathbf{T}}}(\{false\}) = \widehat{false}$	$\gamma_{\hat{\mathbf{T}}}(\widehat{false}) = \{false\}$
$\alpha_{\hat{\mathbf{T}}}(\mathbf{T}) = \top_{\hat{\mathbf{T}}}$	$\gamma_{\hat{\mathbf{T}}}(\top_{\hat{\mathbf{T}}}) = \mathbf{T}$

- Join and meet:

$\hat{a} \sqcup_{\hat{\mathbf{T}}} \hat{b} = \hat{a} \ (\hat{b} \sqsubseteq_{\hat{\mathbf{T}}} \hat{a})$	$\hat{a} \sqcap_{\hat{\mathbf{T}}} \hat{b} = \hat{b} \ (\hat{b} \sqsubseteq_{\hat{\mathbf{T}}} \hat{a})$
$\hat{a} \sqcup_{\hat{\mathbf{T}}} \hat{b} = \hat{b} \ (\hat{a} \sqsubseteq_{\hat{\mathbf{T}}} \hat{b})$	$\hat{a} \sqcap_{\hat{\mathbf{T}}} \hat{b} = \hat{a} \ (\hat{a} \sqsubseteq_{\hat{\mathbf{T}}} \hat{b})$
$\hat{a} \sqcup_{\hat{\mathbf{T}}} \hat{b} = \top_{\hat{\mathbf{T}}}$	$\hat{a} \sqcap_{\hat{\mathbf{T}}} \hat{b} = \perp_{\hat{\mathbf{T}}}$

Abstract States

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

$$\widehat{\text{State}} = \text{Var} \rightarrow \widehat{\mathbb{Z}}$$

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

- An abstract state denotes a set of concrete states:

$$\begin{aligned} \alpha_{\widehat{\text{State}}} &: \mathcal{P}(\text{State}) \rightarrow \widehat{\text{State}} \\ \alpha_{\widehat{\text{State}}}(S) &= \lambda x. \bigsqcup_{s \in S} \alpha_{\widehat{\mathbb{Z}}}(\{s(x)\}) \end{aligned}$$

$$\begin{aligned} \gamma_{\widehat{\text{State}}} &= \widehat{\text{State}} \rightarrow \mathcal{P}(\text{State}) \\ \gamma_{\widehat{\text{State}}}(\hat{s}) &= \{s \in \text{State} \mid \forall x \in \text{Var}. s(x) \in \gamma_{\widehat{\mathbb{Z}}}(\hat{s}(x))\} \end{aligned}$$

- Join and meet:

$$\begin{aligned} \hat{s}_1 \sqcup_{\widehat{\text{State}}} \hat{s}_2 &= \lambda x. \hat{s}_1(x) \sqcup_{\widehat{\mathbb{Z}}} \hat{s}_2(x) \\ \hat{s}_1 \sqcap_{\widehat{\text{State}}} \hat{s}_2 &= \lambda x. \hat{s}_1(x) \sqcap_{\widehat{\mathbb{Z}}} \hat{s}_2(x) \end{aligned}$$

- Widening and narrowing:

$$\begin{aligned} \hat{s}_1 \nabla_{\widehat{\text{State}}} \hat{s}_2 &= \lambda x. \hat{s}_1(x) \nabla_{\widehat{\mathbb{Z}}} \hat{s}_2(x) \\ \hat{s}_1 \triangle_{\widehat{\text{State}}} \hat{s}_2 &= \lambda x. \hat{s}_1(x) \triangle_{\widehat{\mathbb{Z}}} \hat{s}_2(x) \end{aligned}$$

Abstract Semantics

$$\widehat{\mathcal{A}}[a] : \widehat{\text{State}} \rightarrow \widehat{\mathbb{Z}}$$

$$\widehat{\mathcal{A}}[n](\hat{s}) = \alpha_{\widehat{\mathbb{Z}}}(\{n\})$$

$$\widehat{\mathcal{A}}[x](\hat{s}) = \hat{s}(x)$$

$$\widehat{\mathcal{A}}[a_1 + a_2](\hat{s}) = \widehat{\mathcal{A}}[a_1](\hat{s}) +_{\widehat{\mathbb{Z}}} \widehat{\mathcal{A}}[a_2](\hat{s})$$

$$\widehat{\mathcal{A}}[a_1 \star a_2](\hat{s}) = \widehat{\mathcal{A}}[a_1](\hat{s}) \star_{\widehat{\mathbb{Z}}} \widehat{\mathcal{A}}[a_2](\hat{s})$$

$$\widehat{\mathcal{A}}[a_1 - a_2](\hat{s}) = \widehat{\mathcal{A}}[a_1](\hat{s}) -_{\widehat{\mathbb{Z}}} \widehat{\mathcal{A}}[a_2](\hat{s})$$

$$\widehat{\mathcal{B}}[b] : \widehat{\text{State}} \rightarrow \widehat{\mathbb{T}}$$

$$\widehat{\mathcal{B}}[\text{true}](\hat{s}) = \widehat{\text{true}}$$

$$\widehat{\mathcal{B}}[\text{false}](\hat{s}) = \widehat{\text{false}}$$

$$\widehat{\mathcal{B}}[a_1 = a_2](\hat{s}) = \widehat{\mathcal{A}}[a_1](\hat{s}) =_{\widehat{\mathbb{Z}}} \widehat{\mathcal{A}}[a_2](\hat{s})$$

$$\widehat{\mathcal{B}}[a_1 \leq a_2](\hat{s}) = \widehat{\mathcal{A}}[a_1](\hat{s}) \leq_{\widehat{\mathbb{Z}}} \widehat{\mathcal{A}}[a_2](\hat{s})$$

$$\widehat{\mathcal{B}}[\neg b](\hat{s}) = \neg_{\widehat{\mathbb{T}}} \widehat{\mathcal{B}}[b](\hat{s})$$

$$\widehat{\mathcal{B}}[b_1 \wedge b_2](\hat{s}) = \widehat{\mathcal{B}}[b_1](\hat{s}) \wedge_{\widehat{\mathbb{T}}} \widehat{\mathcal{B}}[b_2](\hat{s})$$

Abstract Semantics

- Addition / subtraction / multiplication:

$$[l_1, u_1] +_{\widehat{\mathbb{Z}}} [l_2, u_2] = [l_1 + l_2, u_1 + u_2]$$

$$[l_1, u_1] -_{\widehat{\mathbb{Z}}} [l_2, u_2] = [l_1 - u_2, u_1 - l_2]$$

$$[l_1, u_1] \star_{\widehat{\mathbb{Z}}} [l_2, u_2] = [\min(l_1 l_2, l_1 u_2, u_1 l_2, u_1 u_2), \max(\dots)]$$

- 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., $cmd(n)$:

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

- Transfer function $\hat{f}_n : \widehat{\text{State}} \rightarrow \widehat{\text{State}}$:

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

where \hat{s}' is a *pruned* abstract state such that

$$\alpha_{\widehat{\text{State}}}(\{s \in \gamma_{\widehat{\text{State}}}(\hat{s}) \mid \mathcal{B}[b](s)\}) \sqsubseteq \hat{s}' \sqsubseteq \hat{s}.$$

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

$$\hat{F} : (N \rightarrow \widehat{\text{State}}) \rightarrow (N \rightarrow \widehat{\text{State}})$$

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

Fixed Point Computation

Widening phase	Narrowing phase
$\begin{aligned} W &:= N \\ X &:= \lambda n. \perp \\ \text{repeat} \\ & n := \text{choose}(W) \\ & W := W \setminus \{n\} \\ & s := \hat{f}_n(\bigsqcup_{n' \hookrightarrow n} X(n')) \\ & \text{if } s \sqsubseteq X(n) \\ & \quad \text{if widening is needed} \\ & \quad \quad X(n) := X(n) \nabla s \\ & \quad \text{else} \\ & \quad \quad X(n) := X(n) \sqcup s \\ & \quad W := W \cup \{n' \mid n \hookrightarrow n'\} \\ \text{until } W = \emptyset \end{aligned}$	$\begin{aligned} W &:= N \\ \text{repeat} \\ & n := \text{choose}(W) \\ & W := W \setminus \{n\} \\ & s := \hat{f}_n(\bigsqcup_{n' \hookrightarrow n} X(n')) \\ & \text{if } X(n) \not\sqsubseteq s \\ & \quad X(n) := X(n) \triangle s \\ & \quad W := W \cup \{n' \mid n \hookrightarrow n'\} \\ \text{until } W = \emptyset \end{aligned}$

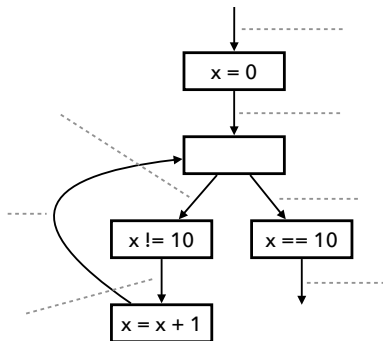
Exercise 1

Describe the interval analysis on the program:

- 1 without widening and
- 2 with widening/narrowing

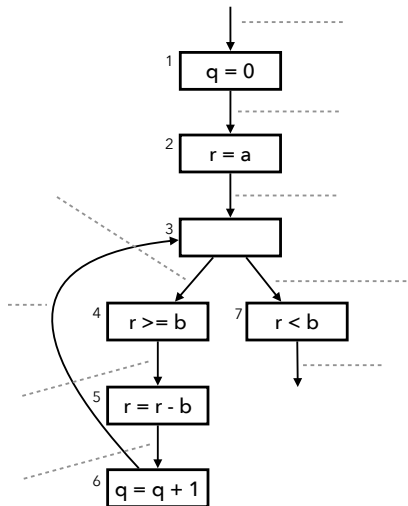
```
x = 0;
```

```
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

program → *block*
block → *decls stmts*
decls → *decls decl* | ϵ
decl → *type x*
type → *int* | *int[n]*
stmts → *stmts stmt* | ϵ

stmt → *lv = e*
| *if e stmt stmt*
| *while e stmt*
| *do stmt while e*
| *read x*
| *print e*
| *block*

lv → *x* | *x[e]*

e → *n* integer
| *lv* l-value
| *e+e* | *e-e* | *e*e* | *e/e* | *-e* arithmetic operation
| *e==e* | *e<e* | *e<=e* | *e>e* | *e>=e* conditional operation
| *!e* | *e||e* | *e&&e* boolean operation

Control-Flow Graph

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

$c \rightarrow x = \text{alloc}(n) \mid lv = e \mid \text{assume}(e) \mid \text{skip} \mid \text{read } x \mid \text{print } e$

- Concrete domain

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

- Concrete semantics

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

Abstraction of Memory Objects

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

```
❶ int i;  
   int[10] arr;  
   i = 1;  
   arr[i] = 2;
```

```
❷ 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}):

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

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

$$\hat{F} : (N \rightarrow \widehat{Mem}) \rightarrow (N \rightarrow \widehat{Mem})$$

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

Abstract Semantics

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

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

- An expression evaluates to an abstract value:

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

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

$$\left\{ \begin{array}{ll} \hat{m}[x \mapsto \widehat{\mathcal{E}}(e)(\hat{m})] & \text{if } lv := e, \mathcal{L}(lv)(\hat{m}) = \{x\} \\ \bigsqcup_{\hat{l} \in \widehat{\mathcal{L}}(lv)(\hat{m})} \hat{m}[\hat{l} \mapsto \hat{m}(\hat{l}) \sqcup \widehat{\mathcal{E}}(e)(\hat{m})] & \text{if } lv := e, \widehat{\mathcal{L}}(lv)(\hat{m}) = \dots \\ \bigsqcup \{ \hat{m}' \sqsubseteq \hat{m} \mid [1, 1] \sqsubseteq \widehat{\mathcal{E}}(e)(\hat{m}').\mathbf{1} \} & \text{if } assume(e) \end{array} \right.$$

Summary

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