

CS477 Formal Software Dev Methods

Elsa L Gunter
2112 SC, UIUC
egunter@illinois.edu
<http://courses.engr.illinois.edu/cs477>

Slides based in part on previous lectures
by Mahesh Vishwanathan, and by Gul Agha

April 25, 2018

Domain for Standard Interpretation

- Given (N, I, K, E) a control flow graph with labels using variables from Var
- Let $Val = values \cup \{\top, \perp\}$, the extended set of values, ordered as before
 - Val is a complete lattice.
- Let $Env = \{\rho \mid \rho : Var \rightarrow Val\}$
 - Env is a complete lattice
 - An env used to be a partial function; now map undefined to \perp
 - $val : (Exp \times Env) \rightarrow Val$
 - Will assume $\{true, false\} \subseteq values$
 - $bval : (BExp \times Env) \rightarrow \{true, false\} \cup \{\top, \perp\} \subseteq Val$
- Let $States = (E \cup \{\top, \perp\}) \times Env$
 - $States$ is a complete lattice assuming the order $((e, \rho) \leq (e', \rho')) \equiv ((e \leq e') \wedge (\rho \leq \rho'))$.

Transitions in Control Flow Graphs

- $next_state : States \rightarrow States$
- $next_state(\top, \rho) = (\top, \rho)$; $next_state(\perp, \rho) = (\perp, \rho)$
- $next_state((m, k, n), \rho)$ defined by cases on $I(n)$:

Transitions in Control Flow Graphs

- $next_state : States \rightarrow States$
- $next_state(\top, \rho) = (\top, \rho)$; $next_state(\perp, \rho) = (\perp, \rho)$
- $next_state((m, k, n), \rho)$ defined by cases on $I(n)$:
 - $I(n) \neq \text{Enter}$

Transitions in Control Flow Graphs

- $next_state : States \rightarrow States$
- $next_state(\top, \rho) = (\top, \rho)$; $next_state(\perp, \rho) = (\perp, \rho)$
- $next_state((m, k, n), \rho)$ defined by cases on $I(n)$:
 - $I(n) \neq \text{Enter}$
 - $I(n) = \text{Exit} \Rightarrow next_state((m, k, n), \rho) = ((m, k, n), \rho)$

Transitions in Control Flow Graphs

- $next_state : States \rightarrow States$
- $next_state(\top, \rho) = (\top, \rho)$; $next_state(\perp, \rho) = (\perp, \rho)$
- $next_state((m, k, n), \rho)$ defined by cases on $I(n)$:
 - $I(n) \neq \text{Enter}$
 - $I(n) = \text{Exit} \Rightarrow next_state((m, k, n), \rho) = ((m, k, n), \rho)$
 - $I(n) = (i := e)$, then n has unique successor node p , $(n, suc, p) \in E$.

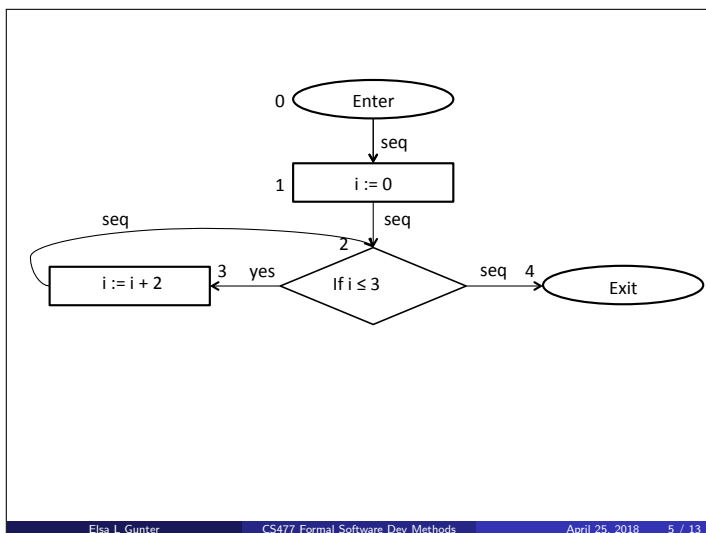
Transitions in Control Flow Graphs

- $\text{next_state} : \text{States} \rightarrow \text{States}$
- $\text{next_state}(\top, \rho) = (\top, \rho)$; $\text{next_state}(\perp, \rho) = (\perp, \rho)$
- $\text{next_state}((m, k, n), \rho)$ defined by cases on $l(n)$:
 - $l(n) \neq \text{Enter}$
 - $l(n) = \text{Exit} \Rightarrow \text{next_state}((m, k, n), \rho) = ((m, k, n), \rho)$
 - $l(n) = (i := e)$, then n has unique successor node p , $(n, \text{succ}, p) \in E$.
 - $\text{next_state}((m, k, n), \rho) = ((n, \text{succ}, p), \rho[i \mapsto \text{val}(e, \rho)])$
 - $l(n) = (\text{if } b)$, then n has two out arcs: (n, yes, p) and (n, seq, q)
 - if $\text{bval}(b, \rho) = \perp$ then $\text{next_state}((m, k, n), \rho) = (\perp, \rho)$
 - if $\text{bval}(b, \rho) = \top$ then $\text{next_state}((m, k, n), \rho) = (\top, \rho)$
 - $\text{bval}(b, \rho) = \text{true}$ then $\text{next_state}((m, k, n), \rho) = ((n, \text{yes}, p), \rho)$
 - $\text{bval}(b, \rho) = \text{false}$ then $\text{next_state}((m, k, n), \rho) = ((n, \text{seq}, q), \rho)$
- next_state is transition semantics for control flow graphs

Example

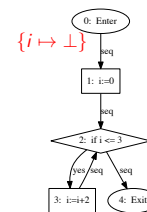
Consider the following control flow graph (N, l, K, E) where:

- $\text{Var} = \{i\}$, $\text{values} = \mathbb{Z}$
- $N = \{0, 1, 2, 3, 4, 5, 6\}$
- $l(0) = \text{Enter}$, $l(1) = i := 0$, $l(2) = \text{if } i \leq 3$, $l(3) = i := i + 2$, $l(4) = \text{Exit}$
- $K = \{\text{yes}, \text{seq}\}$
- $E = \left\{ \begin{array}{l} (0, \text{seq}, 1), (1, \text{seq}, 2), \\ (2, \text{yes}, 3), (2, \text{seq}, 4), \\ (3, \text{seq}, 2) \end{array} \right\}$



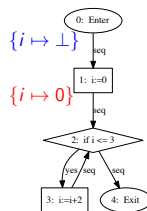
Example: next_state

- $\text{next_state}((0, \text{seq}, 1), \{i \mapsto \perp\}) =$



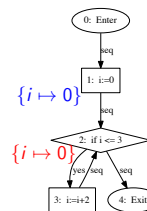
Example: next_state

- $\text{next_state}((0, \text{seq}, 1), \{i \mapsto \perp\}) = ((1, \text{seq}, 2), \{i \mapsto 0\})$



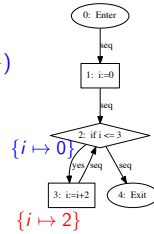
Example: next_state

- $\text{next_state}((0, \text{seq}, 1), \{i \mapsto \perp\}) = ((1, \text{seq}, 2), \{i \mapsto 0\})$
- $\text{next_state}((1, \text{seq}, 2), \{i \mapsto 0\}) = ((2, \text{yes}, 3), \{i \mapsto 0\})$



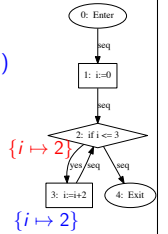
Example: next_state

- $\text{next_state}((0, \text{seq}, 1), \{i \mapsto \perp\}) = ((1, \text{seq}, 2), \{i \mapsto 0\})$
- $\text{next_state}((1, \text{seq}, 2), \{i \mapsto 0\}) = ((2, \text{yes}, 3), \{i \mapsto 0\})$
- $\text{next_state}((2, \text{yes}, 3), \{i \mapsto 0\}) = ((3, \text{seq}, 2), \{i \mapsto 0\}[i \mapsto 0 + 2]) = ((3, \text{seq}, 2), \{i \mapsto 2\})$



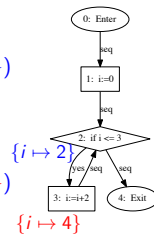
Example: next_state

- $\text{next_state}((0, \text{seq}, 1), \{i \mapsto \perp\}) = ((1, \text{seq}, 2), \{i \mapsto 0\})$
- $\text{next_state}((1, \text{seq}, 2), \{i \mapsto 0\}) = ((2, \text{yes}, 3), \{i \mapsto 0\})$
- $\text{next_state}((2, \text{yes}, 3), \{i \mapsto 0\}) = ((3, \text{seq}, 2), \{i \mapsto 0\}[i \mapsto 0 + 2]) = ((3, \text{seq}, 2), \{i \mapsto 2\})$
- Since $\{i \mapsto 2\}(i) = 2 \leq 3$
 $\text{next_state}((3, \text{seq}, 2), \{i \mapsto 2\}) = ((2, \text{yes}, 3), \{i \mapsto 2\})$



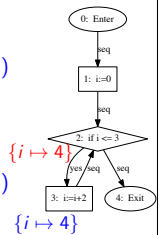
Example: next_state

- $\text{next_state}((0, \text{seq}, 1), \{i \mapsto \perp\}) = ((1, \text{seq}, 2), \{i \mapsto 0\})$
- $\text{next_state}((1, \text{seq}, 2), \{i \mapsto 0\}) = ((2, \text{yes}, 3), \{i \mapsto 0\})$
- $\text{next_state}((2, \text{yes}, 3), \{i \mapsto 0\}) = ((3, \text{seq}, 2), \{i \mapsto 0\}[i \mapsto 0 + 2]) = ((3, \text{seq}, 2), \{i \mapsto 2\})$
- Since $\{i \mapsto 2\}(i) = 2 \leq 3$
 $\text{next_state}((3, \text{seq}, 2), \{i \mapsto 2\}) = ((2, \text{yes}, 3), \{i \mapsto 2\})$
- $\text{next_state}((2, \text{yes}, 3), \{i \mapsto 2\}) = ((3, \text{seq}, 2), \{i \mapsto 2\}[i \mapsto 2 + 2]) = ((3, \text{seq}, 2), \{i \mapsto 4\})$



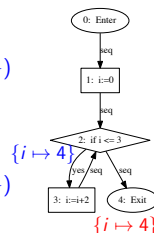
Example: next_state

- $\text{next_state}((0, \text{seq}, 1), \{i \mapsto \perp\}) = ((1, \text{seq}, 2), \{i \mapsto 0\})$
- $\text{next_state}((1, \text{seq}, 2), \{i \mapsto 0\}) = ((2, \text{yes}, 3), \{i \mapsto 0\})$
- $\text{next_state}((2, \text{yes}, 3), \{i \mapsto 0\}) = ((3, \text{seq}, 2), \{i \mapsto 0\}[i \mapsto 0 + 2]) = ((3, \text{seq}, 2), \{i \mapsto 2\})$
- Since $\{i \mapsto 2\}(i) = 2 \leq 3$
 $\text{next_state}((3, \text{seq}, 2), \{i \mapsto 2\}) = ((2, \text{yes}, 3), \{i \mapsto 2\})$
- $\text{next_state}((2, \text{yes}, 3), \{i \mapsto 2\}) = ((3, \text{seq}, 2), \{i \mapsto 2\}[i \mapsto 2 + 2]) = ((3, \text{seq}, 2), \{i \mapsto 4\})$
- Since $\{i \mapsto 4\}(i) = 4 \leq 3$
 $\text{next_state}((3, \text{seq}, 2), \{i \mapsto 4\}) = ((2, \text{seq}, 4), \{i \mapsto 4\})$



Example: next_state

- $\text{next_state}((0, \text{seq}, 1), \{i \mapsto \perp\}) = ((1, \text{seq}, 2), \{i \mapsto 0\})$
- $\text{next_state}((1, \text{seq}, 2), \{i \mapsto 0\}) = ((2, \text{yes}, 3), \{i \mapsto 0\})$
- $\text{next_state}((2, \text{yes}, 3), \{i \mapsto 0\}) = ((3, \text{seq}, 2), \{i \mapsto 0\}[i \mapsto 0 + 2]) = ((3, \text{seq}, 2), \{i \mapsto 2\})$
- Since $\{i \mapsto 2\}(i) = 2 \leq 3$
 $\text{next_state}((3, \text{seq}, 2), \{i \mapsto 2\}) = ((2, \text{yes}, 3), \{i \mapsto 2\})$
- $\text{next_state}((2, \text{yes}, 3), \{i \mapsto 2\}) = ((3, \text{seq}, 2), \{i \mapsto 2\}[i \mapsto 2 + 2]) = ((3, \text{seq}, 2), \{i \mapsto 4\})$
- Since $\{i \mapsto 4\}(i) = 4 \leq 3$
 $\text{next_state}((3, \text{seq}, 2), \{i \mapsto 4\}) = ((2, \text{seq}, 4), \{i \mapsto 4\})$
- Since $l(4) = \text{Exit}$ we have
 $\text{next_state}((2, \text{seq}, 4), \{i \mapsto 4\}) = ((2, \text{seq}, 4), \{i \mapsto 4\})$



Standard Interpretation and Semantics

- Let $\text{Interp}(\theta, (m, k, n))$ be the lifting of next_state to sets of environments (contexts)
- If θ tells all the environments we might come into our edge with, $\text{Interp}(\theta, (m, k, n))$ tells us the set of environments we may leave with
 - Note: $l(m) \neq \text{Exit}$
 - If $l(m) = \text{Enter}$ there are no inbound edges to m , so $\text{Interp}(\theta, (m, k, n)) = \{\text{next_state}((m, k, n), \{v \mapsto \perp \mid v \in \text{Var}\})\}$
 - $l(m) \neq \text{Enter} \Rightarrow$
 $\text{Interp}(\theta, (m, k, n)) = \{\rho \mid \exists m', k', \rho' \mid (m', k', m) \in E \wedge \rho' \in \theta((m', k', m)) \wedge \text{next_state}((m', k', m), \rho') = ((m, k, n), \rho)\}$

Standard Interpretation and Semantics

- Let $Contexts = \mathcal{P}(Env)$
 - $Contexts$ is a complete lattice
 - A context corresponds to a formula in predicate logic over the program variables (recall how we encoded Hoare Logic)
- If for all $e \in E$ we have $\theta(e) \subseteq \phi(e)$, then for all $e' \in E$ we have $Interp(\theta, e') \subseteq Interp(\phi, e')$
- Result:** $(Contexts, Interp)$ is an abstract interpretation
- Recall: $Interp : ((E \rightarrow Contexts) \times E) \rightarrow Contexts$ so $\overline{Interp} : (E \rightarrow Contexts) \rightarrow (E \rightarrow Contexts)$
- $\overline{Interp}(\theta)(e) = Interp(\theta, e) = \overline{Interp}^1(\theta)(e)$
- $\overline{Interp}^{n+1}(\theta)(e) = \overline{Interp}(\overline{Interp}^n(\theta))(e)$
- $\mu \overline{Interp}(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n \{e' \mapsto \{\} \mid e' \in E\}(e)$
- $\mu \overline{Interp}$ tells us the best knowledge we can know about our program
- Problem:** May take an unbounded amount of computation; as informative as transition semantics

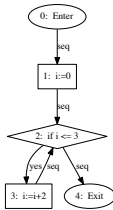
Example: $Interp$

Let θ map edges to sets of environments. $Interp$ will tell us the set of environments $next_state$ will associate with each edge assuming θ gives a set of (possibly) possible environments for each predecessor edge:

- Since $Var = \{i\}$,
 $Interp(\theta, (0, seq, 1)) = \{next_state((0, seq, 1), \{i \mapsto \perp\})\} = \{\{i \mapsto 0\}\}$
- Since $l(2) = \text{if } i \leq 3$ we have $Interp(\theta, (1, seq, 2)) = \theta((0, seq, 1))$
- $Interp(\theta, (2, yes, 3)) = \{\rho \mid \rho(i) + 2 \mid \rho \in (\theta(1, seq, 2) \cup \theta(3, seq, 2)) \wedge \rho(i) \leq 3\}$
- $Interp(\theta, (3, seq, 2)) = \theta(2, yes, 3)$
- $Interp(\theta, (2, no,)) = \{\rho \mid \rho \in (\theta(1, seq, 2) \cup \theta(3, seq, 2)) \wedge \rho(i) > 3\}$

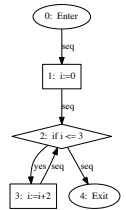
Example: $\mu \overline{Interp}$

- $\mu \overline{Interp} : E \rightarrow Contexts = \mathcal{P}(Env)$
- Start with minimal θ_0 assigning no environments to any edge:
 $\theta_0(e) = \{\}$
- $\mu \overline{Interp}(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n(e)$
- $\mu \overline{Interp}(0, seq, 1) = \{\}$
- $\mu \overline{Interp}(1, seq, 2) = \{\}$
- $\mu \overline{Interp}(2, yes, 3) = \{\}$
- $\mu \overline{Interp}(3, seq, 2) = \{\}$
- $\mu \overline{Interp}((2, no, 4)) = \{\}$



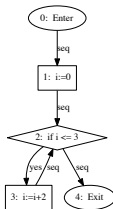
Example: $\mu \overline{Interp}$

- $\mu \overline{Interp} : E \rightarrow Contexts = \mathcal{P}(Env)$
- Start with minimal θ_0 assigning no environments to any edge:
 $\theta_0(e) = \{\}$
- $\mu \overline{Interp}(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n(e)$
- $\mu \overline{Interp}(0, seq, 1) = \{\{i \mapsto 0\}\}$
- $\mu \overline{Interp}(1, seq, 2) = \{\}$
- $\mu \overline{Interp}(2, yes, 3) = \{\}$
- $\mu \overline{Interp}(3, seq, 2) = \{\}$
- $\mu \overline{Interp}((2, no, 4)) = \{\}$



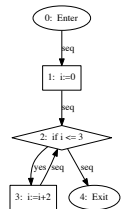
Example: $\mu \overline{Interp}$

- $\mu \overline{Interp} : E \rightarrow Contexts = \mathcal{P}(Env)$
- Start with minimal θ_0 assigning no environments to any edge:
 $\theta_0(e) = \{\}$
- $\mu \overline{Interp}(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n(e)$
- $\mu \overline{Interp}(0, seq, 1) = \{\{i \mapsto 0\}\}$
- $\mu \overline{Interp}(1, seq, 2) = \{\{i \mapsto 0\}\}$
- $\mu \overline{Interp}(2, yes, 3) = \{\}$
- $\mu \overline{Interp}(3, seq, 2) = \{\}$
- $\mu \overline{Interp}((2, no, 4)) = \{\}$



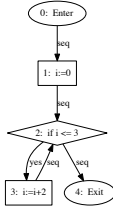
Example: $\mu \overline{Interp}$

- $\mu \overline{Interp} : E \rightarrow Contexts = \mathcal{P}(Env)$
- Start with minimal θ_0 assigning no environments to any edge:
 $\theta_0(e) = \{\}$
- $\mu \overline{Interp}(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n(e)$
- $\mu \overline{Interp}(0, seq, 1) = \{\{i \mapsto 0\}\}$
- $\mu \overline{Interp}(1, seq, 2) = \{\{i \mapsto 0\}\}$
- $\mu \overline{Interp}(2, yes, 3) = \{\{i \mapsto 2\}\}$
- $\mu \overline{Interp}(3, seq, 2) = \{\}$
- $\mu \overline{Interp}((2, no, 4)) = \{\}$



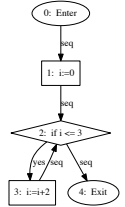
Example: $\mu \overline{\text{Interp}}$

- $\mu \overline{\text{Interp}} : E \rightarrow \text{Contexts} = \mathcal{P}(\text{Env})$
- Start with minimal θ_0 assigning no environments to any edge:
 $\theta_0(e) = \{ \}$
- $\mu \overline{\text{Interp}}(e) = \bigcup_{n \in \mathbb{N}} \overline{\text{Interp}}^n(e)$
- $\mu \overline{\text{Interp}}(0, \text{seq}, 1) = \{ \{i \mapsto 0\} \}$
- $\mu \overline{\text{Interp}}(1, \text{seq}, 2) = \{ \{i \mapsto 0\} \}$
- $\mu \overline{\text{Interp}}(2, \text{yes}, 3) = \{ \{i \mapsto 2\}, \}$
- $\mu \overline{\text{Interp}}(3, \text{seq}, 2) = \{ \{i \mapsto 2\}, \}$
- $\mu \overline{\text{Interp}}((2, \text{no}, 4)) = \{ \}$



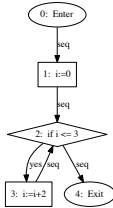
Example: $\mu \overline{\text{Interp}}$

- $\mu \overline{\text{Interp}} : E \rightarrow \text{Contexts} = \mathcal{P}(\text{Env})$
- Start with minimal θ_0 assigning no environments to any edge:
 $\theta_0(e) = \{ \}$
- $\mu \overline{\text{Interp}}(e) = \bigcup_{n \in \mathbb{N}} \overline{\text{Interp}}^n(e)$
- $\mu \overline{\text{Interp}}(0, \text{seq}, 1) = \{ \{i \mapsto 0\} \}$
- $\mu \overline{\text{Interp}}(1, \text{seq}, 2) = \{ \{i \mapsto 0\} \}$
- $\mu \overline{\text{Interp}}(2, \text{yes}, 3) = \{ \{i \mapsto 2\}, \{i \mapsto 4\} \}$
- $\mu \overline{\text{Interp}}(3, \text{seq}, 2) = \{ \{i \mapsto 2\}, \}$
- $\mu \overline{\text{Interp}}((2, \text{no}, 4)) = \{ \}$



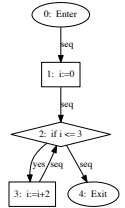
Example: $\mu \overline{\text{Interp}}$

- $\mu \overline{\text{Interp}} : E \rightarrow \text{Contexts} = \mathcal{P}(\text{Env})$
- Start with minimal θ_0 assigning no environments to any edge:
 $\theta_0(e) = \{ \}$
- $\mu \overline{\text{Interp}}(e) = \bigcup_{n \in \mathbb{N}} \overline{\text{Interp}}^n(e)$
- $\mu \overline{\text{Interp}}(0, \text{seq}, 1) = \{ \{i \mapsto 0\} \}$
- $\mu \overline{\text{Interp}}(1, \text{seq}, 2) = \{ \{i \mapsto 0\} \}$
- $\mu \overline{\text{Interp}}(2, \text{yes}, 3) = \{ \{i \mapsto 2\}, \{i \mapsto 4\} \}$
- $\mu \overline{\text{Interp}}(3, \text{seq}, 2) = \{ \{i \mapsto 2\}, \{i \mapsto 4\} \}$
- $\mu \overline{\text{Interp}}((2, \text{no}, 4)) = \{ \}$



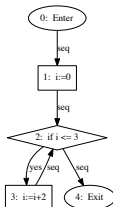
Example: $\mu \overline{\text{Interp}}$

- $\mu \overline{\text{Interp}} : E \rightarrow \text{Contexts} = \mathcal{P}(\text{Env})$
- Start with minimal θ_0 assigning no environments to any edge:
 $\theta_0(e) = \{ \}$
- $\mu \overline{\text{Interp}}(e) = \bigcup_{n \in \mathbb{N}} \overline{\text{Interp}}^n(e)$
- $\mu \overline{\text{Interp}}(0, \text{seq}, 1) = \{ \{i \mapsto 0\} \}$
- $\mu \overline{\text{Interp}}(1, \text{seq}, 2) = \{ \{i \mapsto 0\} \}$
- $\mu \overline{\text{Interp}}(2, \text{yes}, 3) = \{ \{i \mapsto 2\}, \{i \mapsto 4\} \}$
- $\mu \overline{\text{Interp}}(3, \text{seq}, 2) = \{ \{i \mapsto 2\}, \{i \mapsto 4\} \}$
- $\mu \overline{\text{Interp}}((2, \text{no}, 4)) = \{ \{i \mapsto 4\} \}$



Example: $\mu \overline{\text{Interp}}$

- $\mu \overline{\text{Interp}} : E \rightarrow \text{Contexts} = \mathcal{P}(\text{Env})$
- Start with minimal θ_0 assigning no environments to any edge:
 $\theta_0(e) = \{ \}$
- $\mu \overline{\text{Interp}}(e) = \bigcup_{n \in \mathbb{N}} \overline{\text{Interp}}^n(e)$
- $\mu \overline{\text{Interp}}(0, \text{seq}, 1) = \{ \{i \mapsto 0\} \}$
- $\mu \overline{\text{Interp}}(1, \text{seq}, 2) = \{ \{i \mapsto 0\} \}$
- $\mu \overline{\text{Interp}}(2, \text{yes}, 3) = \{ \{i \mapsto 2\}, \{i \mapsto 4\} \}$
- $\mu \overline{\text{Interp}}(3, \text{seq}, 2) = \{ \{i \mapsto 2\}, \{i \mapsto 4\} \}$
- $\mu \overline{\text{Interp}}((2, \text{no}, 4)) = \{ \{i \mapsto 4\} \}$



Soundness of Abstract Semantics

Fact: An abstract interpretation (A, \mathcal{I}) is sound (or consistent) with respect to $(\text{Env}, \overline{\text{Interp}})$ if and only if there exist α, β such that

- $\alpha : \text{Contexts} \rightarrow A, \beta : A \rightarrow \text{Contexts}$
- α, β order preserving
- For all $a \in A$ have $\alpha(\beta(a)) = a$
- For all $S \in \text{Context}$, have $S \subseteq \beta(\alpha(S))$
 - The abstract interpretation gives us more possibilities, is less precise
- For all $e \in E, \alpha(\mu \overline{\text{Interp}}(e)) = \mu \overline{\mathcal{I}}(e)$
- The pair (α, β) is called a *Galois Insertion*.
- Game: find useful A where we can compute $\mu \overline{\mathcal{I}}$, usually in time proportional to size of program

Composing Abstract Interpretations

- Observe: Can abstract an abstraction:
 - Given $\alpha : Context \rightarrow A$, $\beta : A \rightarrow Contexts$ such that for all $a \in A$ have $\alpha(\beta(a)) = a$
 - and for all $S \in Context$, have $S \subseteq \beta(\alpha(S))$
 - Given $\gamma : A \rightarrow B$, $\delta : B \rightarrow A$ such that for all $b \in B$ have $\gamma(\delta(b)) = b$
 - and for all $a \in A$, have $\delta(\gamma(a)) \leq a$
 - Then $(B, \gamma \circ \mu \bar{I})$ is another abstract interpretation.

Some Abstract Interpretations

- Replace *Contexts* with $A = Var \rightarrow \mathcal{P}(Val)$. Let $S \in Contexts$. Define $\alpha(S)(x) = \{v \mid \exists s \in S. v = s(x) \wedge s(x) \neq \perp\}$
- Chain with $B = Var \rightarrow \{int, bool, \perp, \top\}$ and
 - $\gamma(a)(x) = int$ if $a(x)$ contains only integers (and \perp);
 - $\gamma(a)(x) = int$ if $a(x)$ contains only integers (and \perp);
 - $\gamma(a)(x) = \perp$ if $a(x) = \{\}$;
 - $\gamma(a)(x) = \top$ otherwise
 - Can be used for type checking
- Chain with $B = Var \rightarrow \{\perp, \top\}$ and
 - $\gamma(a)(x) = \top$ if $a(x)$ contains something other than \perp ;
 - $\gamma(a)(x) = bot$ if $a(x) = \{\perp\}$ or $a(x) = \{\}$
 - Can be used for checking variables are initialized before they are used.