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 27, 2018

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 abtract interpretation
- Recall: $Interp : ((E \to Contexts) \times E) \to Contexts$ so $\overline{Interp} : (E \to Contexts) \to (E \to Contexts)$
- $\overline{Interp}(\theta)(e) = Interp(\theta, e) = \overline{Interp}^1(\theta)(e)$
- $\overline{Interp}^{n+1}(\theta)(e) = \overline{Interp}(\overline{Interp}^n(\theta))(e)$
- $\mu\,\overline{Interp}(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n\{e' \mapsto \{\,\} \mid e' \in E\}(e)$
- $\mu\,\overline{Interp}$ tells us the best knowledge we can know about our program
- Problem: May take an unbounded amount of computation; as informative as transition semantics

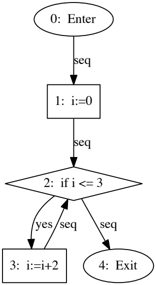
Example: *Interp*

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 $Var = \{i\}$,
  $Interp(\theta, (0, \mathrm{seq}, 1)) = \{\mathrm{next\_state}((0, \mathrm{seq}, 1), \{i \mapsto \bot\})\} = \{\{i \mapsto 0\}\}$
- Since $l(2) =$ `if i <= 3` we have $Interp(\theta, (1, \mathrm{seq}, 2)) = \theta((0, \mathrm{seq}, 1))$
- $Interp(\theta, (2, \mathrm{yes}, 3)) =$
  $\{\rho[i \mapsto \rho(i) + 2] \mid \rho \in (\theta(1, \mathrm{seq}, 2) \cup \theta(3, \mathrm{seq}, 2)) \wedge \rho(i) \leq 3\}$
- $Interp(\theta, (3, \mathrm{seq}, 2)) = \theta(2, \mathrm{yes}, 3)$
- $Interp(\theta, (2, \mathrm{no},)) = \{\rho \mid \rho \in (\theta(1, \mathrm{seq}, 2) \cup \theta(3, \mathrm{seq}, 2)) \wedge \rho(i) > 3\}$

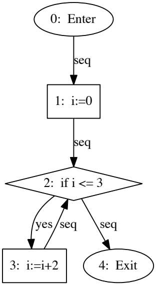
Example: $\mu\,\overline{Interp}$

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



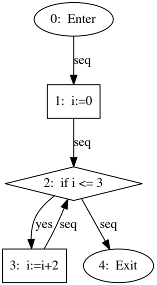
Example: $\mu\,\overline{Interp}$

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



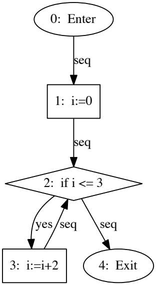
Example: $\mu\,\overline{Interp}$

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



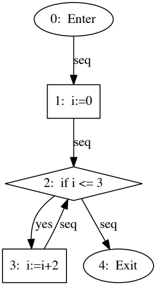
Example: $\mu\,\overline{Interp}$

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



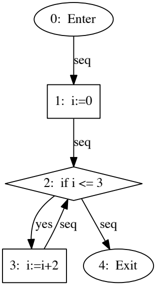
Example: $\mu\,\overline{Interp}$

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



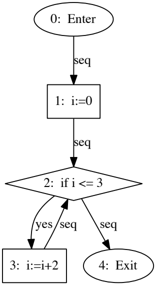
Example: $\mu\, \overline{Interp}$

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



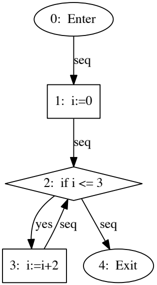
Example: $\mu\,\overline{Interp}$

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



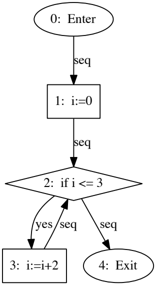
Example: $\mu\,\overline{Interp}$

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



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

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



Soundness of Abstract Semantics

Fact: An abstract interpretation $(A, \mathcal{I})$ is sound (or consistent) with respect to $(\mathit{Contexts}, \mathit{Interp})$ if and only if there exist $\alpha,\ \beta$ such that

- $\alpha : \mathit{Contexts} \to A$, $\beta : A \to \mathit{Contexts}$
- $\alpha,\ \beta$ order preserving
- For all $a \in A$ have $\alpha(\beta(a)) = a$
- For all $S \in \mathit{Contexts}$, have $S \subseteq \beta(\alpha(S))$
  - The abtract interpretation gives us more possibilities, is less precise
- For all $e \in E$, $\alpha(\mu\,\overline{\mathit{Interp}}(e)) = \mu\overline{\mathcal{I}}(e)$
- The pair (α, β) is called a *Galois Insertion*.

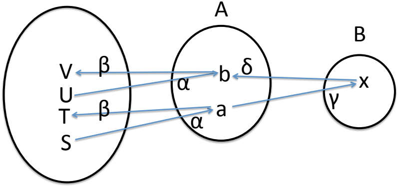
- Game: find useful A where we can compute $\mu\overline{\mathcal{I}}$, usually in time proportional to size of program

Composing Abstract Interpretations

- Observe: Can abstract an abstraction:
  - Given $\alpha : \mathit{Contexts} \to A$, $\beta : A \to \mathit{Contexts}$ such that for all $a \in A$ have $\alpha(\beta(a)) = a$ and for all $S \in \mathit{Contex}$, have $S \subseteq \beta(\alpha(S))$
  - Given $\gamma : A \to B$, $\delta : B \to A$ 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 $(B, \gamma \circ \mu\overline{\mathcal{I}})$ is another abstract interpretation.
  - If $A = \{a \mid \exists S \in \mathit{Contexts}.\ a = \alpha(S)\}$, then

$$\beta(a) = \bigcup\{S \in \mathit{Contexts} \mid \alpha(S) = a\}$$

  - If our abstract domain is the image of *Contexts*, then we only need α, because we can compute β.



Some Abstract Interpretations

- Replace $Contexts = \mathcal{P}(Var \to Val)$ with $A = Var \to \mathcal{P}(Val)$.
  Let $S \in Contexts$. Define

$$\alpha(S)(x) = \{v \mid \exists s \in S.s(x) = v\}$$

- Chain with $B = Var \to \{int, bool, \bot, \top\}$ and
  - $\gamma(a)(x) = int$ if $a(x)$ contains only integers (and $\bot$);
  - $\gamma(a)(x) = bool$ if $a(x)$ contains only booleans (and $\bot$);
  - $\gamma(a)(x) = \bot$ if $a(x) = \{\,\}$ or $a(x) = \{\bot\}$
  - $\gamma(a)(x) = \top$ otherwise
  - Can be used for type checking
- Chain with $B = Var \to \{\bot, \top\}$ and
  - $\gamma(a)(x) = \top$ if $a(x)$ contains something other than $\bot$;
  - $\gamma(a)(x) = \bot$ if $a(x) = \{\bot\}$ or $a(x) = \{\,\}$
  - Can be used for checking variables are initialized before they are used.

Abstraction: $A = Var \rightarrow \mathcal{P}(Val)$

- Have $\alpha(S)(x) = \{v \mid \exists s \in S.s(x) = v\}$
- Need to show
  - (onto A) $\forall x \in Var. \forall V \subseteq Val. \exists S \subseteq Env. \, V = \{v \mid \exists s \in S. \, s(x) = v\}$

Pf: Fix $x \in Var$ and $V \subseteq Val$. Let $S = \{s \in Env \mid s(x) \in V\}$. Because Env is all mappings of Var to Val, for any $v \in V$, there is a mapping s such that $s(x) = v$. Therefore, $V = \{v \mid \exists s \in S.s(x) = v\}$

  - (order preserving) $\forall S, T \subseteq Env. \, S \subseteq T \Rightarrow (\forall x \in Var. \{v \in Val \mid \exists s \in S. \, s(x) = v\} \subseteq \{v \in Val \mid \exists s \in T. \, s(x) = v\}$

Pf: (sketch) If $s \in S$ then $s \in T$.

  - This abstraction is the root of most abstraction
  - Still too informative; takes unbounded time to compute

Abstraction: $B = Var \to \{int, bool, \bot, \top\}$

- $\gamma : A \to B$ where
  - $\gamma(a)(x) = int$ if $a(x)$ contains only integers (and $\bot$);
  - $\gamma(a)(x) = bool$ if $a(x)$ contains only booleans (and $\bot$);
  - $\gamma(a)(x) = \bot$ if $a(x) = \{\}$ or $a(x) = \{\bot\}$
  - $\gamma(a)(x) = \top$ otherwise
- Still need onto and order-preserving

- Tells us for each variable if it is guaranteed to be used
- B has finite height this time
- Should be able to compute in bounded time
- But how?

New transfer (transition) functions from old

- Recall that $Interp : ((E \to \mathcal{P}(Env)) \times E) \to \mathcal{P}(Env))$ tells us how, for each edge, taking one step of computation updates the possible environments after that step of computation after that edge
- $\overline{Interp}$ tells us how to update the environments at each edge after we let each edge do its endpoint computation, assuming each starting environment
- Given $\alpha : \mathcal{P}(Env) \to A$, order preserving and onto, construct $\alpha \circ \overline{Interp} : ((E \to A) \to (E \to A))$
- Calculating $\mu\,(\alpha \circ \overline{Interp}) : E \to A$ can be done in bounded timeif calculating $\alpha \circ \overline{Interp})(\theta, e)$ can be for each $\theta \in (E \to A)$ and $e \in E$.

Worklist Algorithm

- Given transfer function $\mathcal{I} : ((E \to A) \times E) \to A$, want algorithm for finding $\mu\overline{\mathcal{I}}$
- Start with $\mathcal{I}_o : E \to A$ by $\mathcal{I}_0(e) = \bot$ for all $e \in E$
- Let $Worklist = E$
- Assume we have computed $\mathcal{I}_n$
- while $not(Worklist = \{\,\})$ do
  - Pick $(n, k, m) \in Worklist$; let $Worklist = Worklist - \{(n, k, m)\}$
  - Let $in = lub\{a \mid \exists p, j.\ (p, j, n) \in E \wedge a = \mathcal{I}_n(p, j, n)\}$
  - Let $\mathcal{I}_{n+1}(n, k, m) = \mathcal{I}(\mathcal{I}_n, (n, k, m))$ and $\mathcal{I}_{n+1}(e) = \mathcal{I}_n(e)$ if $e \neq (n, k, m)$
  - If $\mathcal{I}_{n+1}(n, k, m) = \mathcal{I}_n(n, k, m)$ then $Worklist \neq Worklist \cup \{e \mid \exists j, p.\ e = (m, j, p)\}$
  - Repeat while
- $\mu\mathcal{I}(e) = \mathcal{I}_n(e)$ where n is the last value where a change was made