CS477 Formal Software Development Methods

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

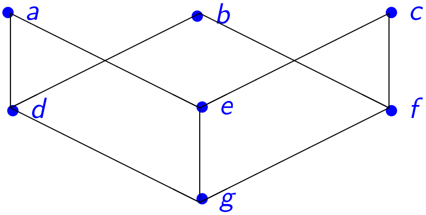
Slides based in part on previous lectures by Mahesh Vishwanathan, and by Gul Agha

May 14, 2013

Partial orders and Complete Lattices

A partial order on a set S is a binary relation $\leq$ on S such that

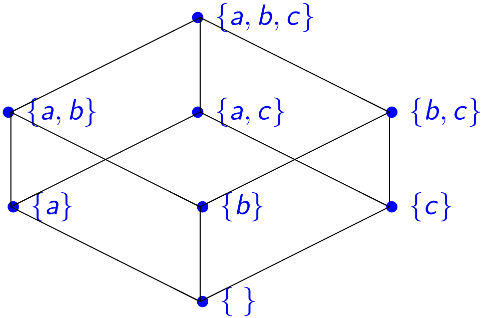
- **[Refl]** $s \leq s$ for all $s \in S$
- **[Antisym]** $s \leq t$ and $t \leq s$ impilies $s = t$, for all $s,\ t \in S$
- **[Trans]** $s \leq t$ and $t \leq u$ impilies $s \leq u$, for all $s,\ t,\ \in S$

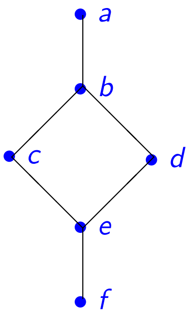


Upper Bounds and Complete Latices

- In a partial order $(S, \leq)$, given $X \subseteq S$, y is an upper bound for X if for all $x \in X$ we have $x \leq y$.
- y is a least upper bound of X, y is an upper bound of X and whenever z is an upper bound of X, $y \leq z$.
- **Note:** Least upper bounds are unique.
- A complete lattice is a partial order $(L, \leq)$ such that for all $X \subseteq S$ there exists a (unique) least upper bound.
- Write $\text{lub}(X)$ or $\bigvee X$ for the least upper bound of X.
- Write $x \vee y$ for $\bigvee\{x, y\}$
- **Note:** $x \vee y = x \iff y \leq x$
- **Note:** Given a set S, $(\mathcal{P}(S), \subseteq)$ is a complete lattice.
- Write $\bot = \bigvee\{\ \}$ and $\top = bigveeS$

Example Complete Lattices





Partial Orders, Functions, and Complete Lattices

- Let X be an arbitrary set and A and B be partial orders.
- A function $f : A \to B$ is order-preserving if, for all $x, y \in A$ with $x \leq y$ we have $f(x) \leq f(y)$
- Function $f, g : X \to A$ may be ordered by pointwise comparrison:
  - Write $f \leq_{fun} g$ to mean that for all $x \in X$ we have $f(x) \leq g(x)$
  - Will leave off the subcript in general
- **Fact:** $(\{f \,.\, f : X \to B\}, \leq_{fun})$ is a partial order.
- **Fact:** $(\{f \,.\, f : X \to B\}, \leq_{fun})$ is a complete lattice if B is.
- **Fact:** $(\{f \,.\, f : A \to B,\ f \text{ order-preserving}\} \leq_{fun})$ is a complete lattice if B is.

Control-Flow Graphs

A Control-Flow Graph is a tuple (N, I, K, E) where

- N is a finite set of nodes
- $I : N \to \{\text{Entry}, \text{Exit,i:=e}, \text{ifb}, \}$
- $K = \{\text{yes}, \text{no}, \text{seq}\}$
- $E \subseteq N \times K \times N$ such that
  - for all $m \in N$ we have $|\{n\,.\,\exists k \in K.\,(m, k, n) \in E\}| \leq 2$
  - if $m \in N$ and $I(m) = \text{Exit}$ then $|\{n\,.\,\exists k \in K.\,(m, k, n) \in E\}| = 0$
  - if $m \in N$ and $I(m) = \text{Entry}$ or $I(m) = i := e$ for some identifier i and expression e, then $|\{k, n\,.\,(m, k, n) \in E\}| = 1$
  - if $m \in N$ and $I(m) = \text{if } b$ for some boolean expression b, then $|\{n\,.\,\exists k \in K.\,(m, k, n) \in E\}| = 2$
- $k : E \to \{\text{seq}, \text{yes}, \text{no}\}$ such that
  - if $(m, k, n) \in E$ and $I(m) = \text{Entry}$ or $I(m) = i := e$, then $k = \text{seq}$
  - if $m, \in N$ and $I(m) = \text{if } b$, then $\{k\,.\,(m, k, n) \in E\} = \{\text{yes}, \text{no}\}$

Abstract Interpretation

- Let (N, I, K, E) be a control flow graph.
- An abstract interpretation of control flow graphs is a pair $(A, \mathcal{I})$ where
  - A is a complete latice and
  - $\mathcal{I} : ((E \rightarrow A) \times E) \rightarrow A$ (think *next state information vector*)
  - for all $a, b \in A$, for all $e \in E$, if $a \leq b$ then $\mathcal{I}(e, a) \leq \mathcal{I}(e, b)$

Abstract Semantics

- Can define $\overline{\mathcal{I}} : (E \to A) \to (E \to A)$ by $\overline{\mathcal{I}}(f)(e) = \mathcal{I}(f, e)$
- **Fact:** $\overline{\mathcal{I}}$ is order-preserving
- **Tarski's Fixed-Point Theorem:** If A is a complete lattice and $f : A \to A$ is order-preserving, then f has both a least and a greatest fixed-point (may or may not be the same).
- **Fact:** There exist $c : E \to A$ such that $oI(c) = c$, and that c is the least such.
- Write $\mu\overline{\mathcal{I}}$ for the least fixed point of $\overline{\mathcal{I}}$
- $\mu\overline{\mathcal{I}}$ is the abstract semantics of (N, I, K, E) with respect to $(A, \mathcal{I})$.

Standard Interpretation and Semantics

- Let (N, I, K, E) be a control flow graph with labels using variables from Var
- Let $Val = values \cup \{\top, \bot\}$, the extended set of values, ordered as before; $val : Exp \to Val$
- Let $Env = \mathcal{P}(\{\rho \,.\, \rho : Var \to Val\})$. Env is a complete lattice.
- Let $States = E \times Env$
- $\mathrm{next\_state} : States \to States$; $\mathrm{next\_state}((m, k, n), \rho)$ defined by cases on $I(n)$:
    - $I(n) \neq \mathrm{Enter}$
    - $I(n) = \mathrm{Exit} \Rightarrow \mathrm{next\_state}((m, k, n), \rho) = ((m, k, n), \rho)$
    - $I(n) = (i := e)$, then n has unique successor node p, $(n, \mathrm{suc}, p) \in E$.

$$\mathrm{next\_state}((m, k, n), \rho) = ((n, \mathrm{suc}, p), \rho[i \mapsto val(e, \rho)])$$

- Let $\mathit{Interp}(\theta, (m, k, n))$ is the lifting of next_state to sets of environments
  - $l(m) = \text{Enter} \Rightarrow \mathit{Interp}(\theta, (m, k, n)) = \{\bot_{Env}\}$
  - $l(m) \neq \text{Exit} \Rightarrow$
    $$\begin{array}{l} \mathit{Interp}(\theta, (m, k, n)) = \\ \quad \{\rho \mid \exists m',\ k',\ \rho' \,.\, (m', k', m) \in E \wedge \\ \qquad\qquad\qquad\qquad \rho' \in \theta((m', k', m)) \wedge \\ \qquad\qquad\qquad\qquad \text{next\_state}((m', k', m), \rho') = ((m, k, n), \rho)\} \end{array}$$
- If θ tells all the environments we might come into our edge with, $\mathit{Interp}(\theta, (m, k, n))$ tells us the set of environemts we may leave with
- **Fact:** $(\mathit{Env}, \mathit{Interp})$ is an abtract interpretation
- $\mu\, \mathit{overline Interp}$ tells us the best knowledge we can know statically about our program

Soundness of Abstract Semantics

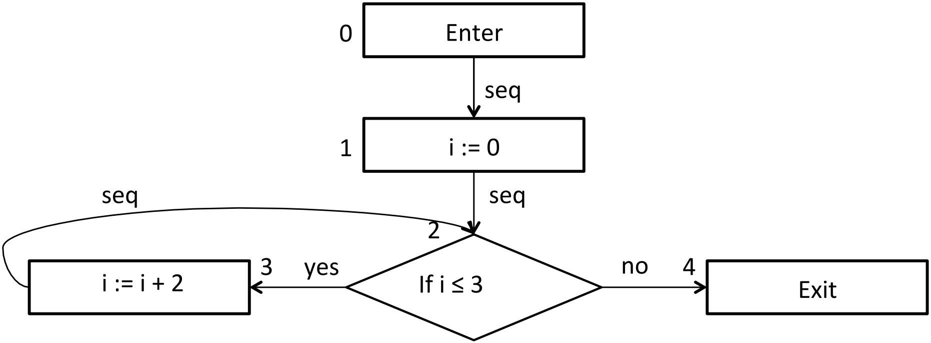
Fact: An abstract interpretation $(A, \mathcal{I})$ is sound (or consistent) with respect to $(Env, Interp)$ if and only if there exist $\alpha,\ \beta$ such that

- $\alpha : Env \rightarrow A$, $\beta : A \rightarrow Env$
- $\alpha,\ \beta$ order preserving
- For all $a \in A$ have $\alpha(\beta(a)) = a$
- For all $S \in Env$, have $S \leq \beta(\alpha(S))$ – We have more possibilities

Example

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

- $Var = \{i\}, \quad val = \mathbb{Z}$
- $N = \{0, 1, 2, 3, 4, 5, 6\}$
- $I(0) = \text{Enter}$, $I(1) = \texttt{i:=0}$, $I(2) = \texttt{if 1} \leq \texttt{3}$,
  $I(3) = \texttt{i:=i+2}$, $I(4) = \text{Exit}$
- $K = \{\text{yes}, \text{no}, \text{seq}\}$
- $E = \left\{ \begin{array}{ll} (0, \text{seq}, 1), & (1, \text{seq}, 2), \\ (2, \text{yes}, 3), & (2, \text{no}, 4), \\ (3, \text{seq}, 2) & \end{array} \right\}$


0
Enter
seq
1
i := 0
seq
seq
2
If i ≤ 3
3
yes
i := i + 2
no
4
Exit

Example: next_state

- $\mathrm{next\_state}((0, \mathrm{seq}, 1), \rho) = ((1, \mathrm{seq}, 2)\{i \mapsto \bot\})$
- $\mathrm{next\_state}((1, \mathrm{seq}, 2), \rho) = ((2, \mathrm{yes}, 3), \{i \mapsto 0\})$
- $\mathrm{next\_state}((2, \mathrm{yes}, 3), \rho) = ((3, \mathrm{seq}, 2), \rho)$
- if $\rho(i) \leq 1$ then $\mathrm{next\_state}((3, \mathrm{seq}, 2), \rho) = ((2, \mathrm{yes}, 3), \{i \mapsto \rho(i) + 2\})$
- if $\rho > 1$ then $\mathrm{next\_state}((3, \mathrm{seq}, 2), \rho) = ((2, \mathrm{no}, 4), \{i \mapsto \rho(i) + 2\})$

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:

- $Interp(\Theta, (0, \mathrm{seq}, 1)) = \{\{i \mapsto \bot\}\}$
- $$\begin{aligned} Interp(\Theta, (1, \mathrm{seq}, 2)) &= \{\rho \mid \exists \rho' \in \Theta(0, \mathrm{seq}, 1) \,.\, \rho = \rho'[i \mapsto 0]\} \\ &= \{\{i \mapsto 0\}\} \text{ if } \Theta(0, \mathrm{seq}, 1) \neq \{\,\} \text{ since } Var = \{i\} \end{aligned}$$
- $Interp(\Theta, (2, \mathrm{yes}, 3)) = \Theta(1, \mathrm{seq}, 2) \cup \{\rho \in \Theta(3, \mathrm{seq}, 2) \mid \rho(i) \leq 3\}$
- $Interp(\Theta, (3, \mathrm{seq}, 2)) = \{\rho \mid \exists \rho' \in \Theta(2, \mathrm{yes}, 3) \,.\, \rho = \rho‘[i \mapsto \rho'(i) + 2]\}$
- $Interp(\Theta, (2, \mathrm{no}, 4)) = \{\rho \in \Theta(3, \mathrm{seq}, 2) \mid \rho(i) > 3\}$

- $\overline{Interp}(\Theta)(e) = Interp(\Theta, e)$
- $\overline{Interp}^{0}(\Theta)(e) = \{\,\}$ $\quad \overline{Interp}^{n+1}(\Theta)(e) = \overline{Interp}(\overline{Interp}^{n}(\Theta))(e)$

Example: $\mu Interp$

- $\mu Inter : E \rightarrow \mathcal{P}(Env)$
- Start with minimal Θ_0 assigning no evnironments to any edge: $\Theta_0(e) = \{\ \}$
- $\mu Interp(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^{n}(e)$
- $\mu Interp(0, \text{seq}, 1) = \{ \qquad\qquad \}$
- $\mu Interp(1, \text{seq}, 2) = \{ \qquad\qquad \}$
- $\mu Interp(2, \text{yes}, 3) = \{ \qquad\qquad\qquad\qquad \}$
- $\mu Interp(3, \text{seq}, 2) = \{ \qquad\qquad\qquad\qquad \}$
- $Interp(\Theta, (2, \text{no}, 4)) = \{ \qquad\qquad \}$

Example: $\mu Interp$

- $\mu Inter : E \to \mathcal{P}(Env)$
- Start with minimal Θ_0 assigning no evnironments to any edge: $\Theta_0(e) = \{\,\}$
- $\mu Interp(e) = \bigcup_{n\in\mathbb{N}} \overline{Interp}^n(e)$
- $\mu Interp(0, \mathrm{seq}, 1) = \{\{i \mapsto \bot\}\}$
- $\mu Interp(1, \mathrm{seq}, 2) = \{\qquad\qquad\}$
- $\mu Interp(2, \mathrm{yes}, 3) = \{\qquad\qquad\qquad\qquad\}$
- $\mu Interp(3, \mathrm{seq}, 2) = \{\qquad\qquad\qquad\qquad\}$
- $Interp(\Theta, (2, \mathrm{no}, 4)) = \{\qquad\qquad\}$

Example: $\mu Interp$

- $\mu Inter : E \to \mathcal{P}(Env)$
- Start with minimal Θ_0 assigning no evnironments to any edge: $\Theta_0(e) = \{\ \}$
- $\mu Interp(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n(e)$
- $\mu Interp(0, \mathrm{seq}, 1) = \{\{i \mapsto \bot\}\}$
- $\mu Interp(1, \mathrm{seq}, 2) = \{\{i \mapsto 0\}\}$
- $\mu Interp(2, \mathrm{yes}, 3) = \{ \qquad\qquad\qquad \}$
- $\mu Interp(3, \mathrm{seq}, 2) = \{ \qquad\qquad\qquad \}$
- $Interp(\Theta, (2, \mathrm{no}, 4)) = \{ \qquad\qquad \}$

Example: $\mu Interp$

- $\mu Inter : E \to \mathcal{P}(Env)$
- Start with minimal Θ_0 assigning no evnironments to any edge: $\Theta_0(e) = \{\,\}$
- $\mu Interp(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n(e)$
- $\mu Interp(0, \text{seq}, 1) = \{\{i \mapsto \bot\}\}$
- $\mu Interp(1, \text{seq}, 2) = \{\{i \mapsto 0\}\}$
- $\mu Interp(2, \text{yes}, 3) = \{\{i \mapsto 0\}, \qquad \}$
- $\mu Interp(3, \text{seq}, 2) = \{ \qquad \}$
- $Interp(\Theta, (2, \text{no}, 4)) = \{ \qquad \}$

Example: $\mu Interp$

- $\mu Inter : E \to \mathcal{P}(Env)$
- Start with minimal Θ_0 assigning no evnironments to any edge: $\Theta_0(e) = \{\,\}$
- $\mu Interp(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n(e)$
- $\mu Interp(0, \text{seq}, 1) = \{\{i \mapsto \bot\}\}$
- $\mu Interp(1, \text{seq}, 2) = \{\{i \mapsto 0\}\}$
- $\mu Interp(2, \text{yes}, 3) = \{\{i \mapsto 0\}, \qquad\qquad \}$
- $\mu Interp(3, \text{seq}, 2) = \{\{i \mapsto 2\}, \qquad\qquad \}$
- $Interp(\Theta, (2, \text{no}, 4)) = \{ \qquad\qquad \}$

Example: $\mu Interp$

- $\mu Inter : E \to \mathcal{P}(Env)$
- Start with minimal Θ_0 assigning no evnironments to any edge: $\Theta_0(e) = \{\,\}$
- $\mu Interp(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^{n}(e)$
- $\mu Interp(0, \text{seq}, 1) = \{\{i \mapsto \bot\}\}$
- $\mu Interp(1, \text{seq}, 2) = \{\{i \mapsto 0\}\}$
- $\mu Interp(2, \text{yes}, 3) = \{\{i \mapsto 0\}, \{i \mapsto 2\}\}$
- $\mu Interp(3, \text{seq}, 2) = \{\{i \mapsto 2\}, \qquad\qquad \}$
- $Interp(\Theta, (2, \text{no}, 4)) = \{ \qquad\qquad \}$

Example: $\mu Interp$

- $\mu Inter : E \to \mathcal{P}(Env)$
- Start with minimal Θ_0 assigning no evnironments to any edge: $\Theta_0(e) = \{\,\}$
- $\mu Interp(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n(e)$
- $\mu Interp(0, \mathrm{seq}, 1) = \{\{i \mapsto \bot\}\}$
- $\mu Interp(1, \mathrm{seq}, 2) = \{\{i \mapsto 0\}\}$
- $\mu Interp(2, \mathrm{yes}, 3) = \{\{i \mapsto 0\}, \{i \mapsto 2\}\}$
- $\mu Interp(3, \mathrm{seq}, 2) = \{\{i \mapsto 2\}, \{i \mapsto 4\}\}$
- $Interp(\Theta, (2, \mathrm{no}, 4)) = \{ \qquad\qquad \}$

Example: $\mu Interp$

- $\mu Inter : E \to \mathcal{P}(Env)$
- Start with minimal Θ_0 assigning no evnironments to any edge: $\Theta_0(e) = \{\,\}$
- $\mu Interp(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^{n}(e)$
- $\mu Interp(0, \text{seq}, 1) = \{\{i \mapsto \bot\}\}$
- $\mu Interp(1, \text{seq}, 2) = \{\{i \mapsto 0\}\}$
- $\mu Interp(2, \text{yes}, 3) = \{\{i \mapsto 0\}, \{i \mapsto 2\}\}$
- $\mu Interp(3, \text{seq}, 2) = \{\{i \mapsto 2\}, \{i \mapsto 4\}\}$
- $Interp(\Theta, (2, \text{no}, 4)) = \{\{i \mapsto 4\}\}$

Example: $\mu Interp$

- $\mu Inter : E \to \mathcal{P}(Env)$
- Start with minimal Θ_0 assigning no evnironments to any edge: $\Theta_0(e) = \{\ \}$
- $\mu Interp(e) = \bigcup_{n \in \mathbb{N}} \overline{Interp}^n(e)$
- $\mu Interp(0, \mathrm{seq}, 1) = \{\{i \mapsto \bot\}\}$
- $\mu Interp(1, \mathrm{seq}, 2) = \{\{i \mapsto 0\}\}$
- $\mu Interp(2, \mathrm{yes}, 3) = \{\{i \mapsto 0\}, \{i \mapsto 2\}\}$
- $\mu Interp(3, \mathrm{seq}, 2) = \{\{i \mapsto 2\}, \{i \mapsto 4\}\}$
- $Interp(\Theta, (2, \mathrm{no}, 4)) = \{\{i \mapsto 4\}\}$