CS477 Formal Software Development Methods

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

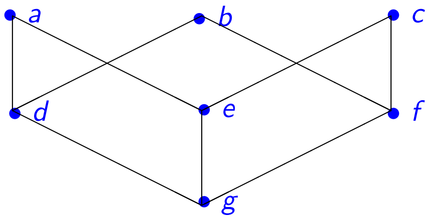
Slides based in part on previous lectures by Mahesh Vishwanathan, and by Gul Agha

April 26, 2013

Partial orders and Complete Lattices

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

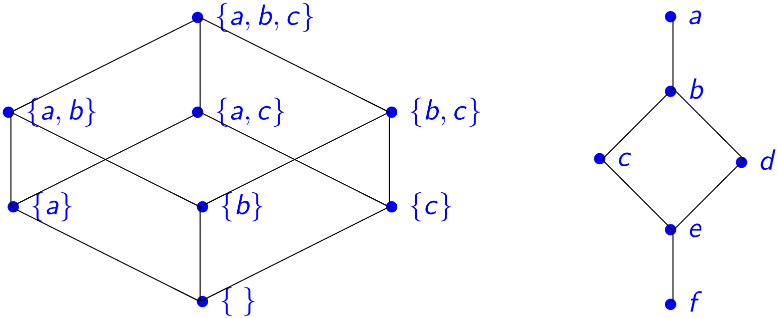
- **[Refl]** $s \leq s$ for all $s \in S$
- **[Antisym]** $s \leq t$ and $t \leq s$ impilies $s = t$, for all $s,\ t \in S$
- **[Trans]** $s \leq t$ and $t \leq u$ impilies $s \leq u$, for all $s,\ t, \in S$



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 $\mathrm{lub}(X)$ 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



Control-Flow Graphs

A Control-Flow Graph is a tuple (N, E, l, k) where

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

Environments Revisited

Previously: environments are *partial* functions from identifiers (Ident) to values.

Add to usual values two new constants: $\text{Values} = \{v | v \text{ a value}\} \cup \{\top, \bot\}$.

$\bot$ means undefined

$\top$ means error

Order Values by $v_1 \leq v_2$ implies $v_1 = v_2$, and $\bot \leq v$ and $v \leq \top$ for all values v, v_1, v_2

Redefine $\text{Env} = \text{Ident} \rightarrow \text{Values}$, all *total* functions from identifiers to extended values

Can view every old environment as an element of Env

Fix control flow graph $G = (N, E, l, k)$
$\text{States} = E \times \text{Env}$
Note: Entry is never the label on node on the end of the edge in a state.
Can define next state of state $s \in \text{States}$: only need the end state of the edge and the environment.
next state is a transition relation.
From next state can define a run.