Basic Discrete Structures

Sets, Functions, Sequences, Matrices, and Relations
(Lecture – 11)

Dr. Nirnay Ghosh

Upper & Lower Bounds

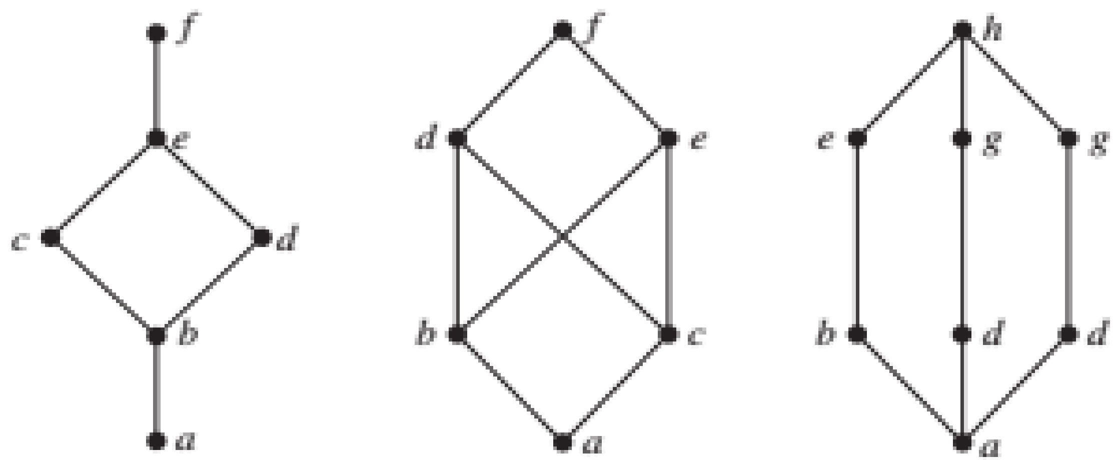
- Upper bound
  - Sometimes it is possible to find an element that is greater than or equal to all the elements in a subset A of a poset $(S, \preccurlyeq)$.
  - If u is an element of S such that $a \preccurlyeq u$ for all elements $a \in A$, then u is called an **upper bound** of A.
- Least upper bound/Supremum/Join
  - An element x is called the **least upper bound** of the subset A if x is an upper bound that is less than every other upper bound of A.
  - Because there is only one such element, if it exists, it makes sense to call this element *the* least upper bound
  - x is the least upper bound of A if $a \preccurlyeq x$ whenever $a \in A$, and $x \preccurlyeq z$ whenever z is an upper bound of A.
  - The least upper bound of set A is denoted as *lub* (A).

Upper & Lower Bounds

- Lower bound
  - Sometimes it is possible to find an element that is less than or equal to all the elements in a subset A of a poset $(S, \preccurlyeq)$.
  - If l is an element of S such that $l \preccurlyeq a$ for all elements $a \in A$, then l is called a **lower bound** of A.
- Greatest lower bound/Infimum/Meet
  - The element y is called the **greatest lower bound** of the subset A if y is an lower bound that is greater than every other lower bound of A.
  - Because there is only one such element, if it exists, it makes sense to call this element *the* greatest lower bound
  - y is the greatest lower bound of A if $z \preccurlyeq y$ whenever z is an lower bound of A.
  - The greatest lower bound of set A is denoted as $glb\ (A)$.

Lattices

- A partially ordered set in which every pair of elements has both a *least upper bound* and a *greatest lower bound* is called a **lattice**.
- Lattices are used in many different applications such as models of information flow and play an important role in Boolean algebra.
- Ex: *Determine whether the posets represented by each of the Hasse diagrams in the following figure are lattices:*



(a) (b) (c)

The Lattice Model of Information Flow

- In many settings the flow of information from one person or computer program to another is restricted via security clearances.
- We can use a lattice model to represent different information flow policies.
- Example: one common information flow policy is the *multilevel security policy* used in government and military systems.
- Each piece of information is assigned to a security class, and each security class is represented by a pair (A, C) where A is an *authority level* and C is a *category*.
- People and computer programs are then allowed access to information from a specific restricted set of security classes.
- The typical authority levels used in the U.S. government are *unclassified (0), confidential (1), secret (2), and top secret (3).*

The Lattice Model of Information Flow

- Categories used in security classes are the subsets of a set of all *compartments* relevant to a particular area of interest.
- Each compartment represents a particular subject area.
- For example, if the set of compartments is {*spies, moles, double agents*}, then there are eight different categories, one for each of the eight subsets of the set of compartments, such as {*spies, moles*}.
- We can order security classes by specifying that $(A1, C1) \preccurlyeq (A2, C2)$ if and only if $A1 \leq A2$ and $C1 \subseteq C2$.
- Information is permitted to flow from security class $(A1, C1)$ into security class $(A2, C2)$ if and only if $(A1, C1) \preccurlyeq (A2, C2)$.
- Example:
  - Information is permitted to flow from the security class *(secret, {spies, moles})* into the security class *(top secret, {spies, moles, double agents})*
  - Information is not allowed to flow from the security class *(top secret, {spies, moles})* into either of the security classes *(secret, {spies, moles, double agents})* or *(top secret, {spies})*

Topological Sorting

- Is it possible to input the sets of $P(\{a, b, c\})$ into a computer in a way that is *compatible* with the subset relation $\subseteq$ in the sense that if set U is a subset of set V, then U is input before V?
  - $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$, $\{a, b, c\}$
  - $\emptyset$, $\{a\}$, $\{b\}$, $\{a, b\}$, $\{c\}$, $\{b, c\}$, $\{a, c\}$, $\{a, b, c\}$

• Definition

Given partial order relations $\preceq$ and $\preceq'$ on a set A, $\preceq'$ is **compatible** with $\preceq$ if, and only if, for all a and b in A, if $a \preceq b$ then $a \preceq' b$.

• Definition

Given partial order relations $\preceq$ and $\preceq'$ on a set A, $\preceq'$ is a **topological sorting** for $\preceq$ if, and only if, $\preceq'$ is a total order that is compatible with $\preceq$.

- Constructing a topological sorting for a general finite partially ordered set is based on the principle that any partially ordered set that is finite and nonempty has a minimal element.

Topological Sorting (Contd…)

- Lemma:

Every finite nonempty poset $(S, \preccurlyeq)$ has at least one minimal element.

ALGORITHM 1 Topological Sorting.

procedure *topological sort* $((S, \preccurlyeq)$: finite poset)
$k := 1$
while $S \neq \emptyset$
    $a_k := $ a minimal element of S {such an element exists by Lemma 1}
    $S := S - \{a_k\}$
    $k := k + 1$
return $a_1, a_2, \ldots, a_n$ $\{a_1, a_2, \ldots, a_n$ is a compatible total ordering of $S\}$