Relations : 5

N Geetha
AM & CS
PSG Tech

Equivalence Relations

- **Equivalence relations** are used to relate objects that are similar in some way.

- **Definition:** A relation on a set A is called an equivalence relation if it is reflexive, symmetric, and transitive.

- Two elements a and b are related by an equivalence relation are called **equivalent**, denoted by $a \sim b$, a and b are equivalent elements w.r.t. a particular equivalence relation

Equivalence Relations

- Since R is **symmetric**, a is equivalent to b whenever b is equivalent to a.
- Since R is **reflexive**, every element is equivalent to itself.
- Since R is **transitive**, if a and b are equivalent and b and c are equivalent, then a and c are equivalent.
- Eg1. : Let R be the relation on the set of integers s.t. aRb iff a=b or a=-b
- Eg2: Let R be the relation on the set of real numbers s.t. aRb iff a-b is an integer. Is R an equivalence relation?

Equivalence Relations

- **Example:** Suppose that R is the relation on the set of strings that consist of English letters such that aRb if and only if l(a) = l(b), where l(x) is the length of the string x. Is R an equivalence relation?
- **Solution:**
  - R is reflexive, because l(a) = l(a) and therefore aRa for any string a.
  - R is symmetric, because if l(a) = l(b) then l(b) = l(a), so if aRb then bRa.
  - R is transitive, because if l(a) = l(b) and l(b) = l(c), then l(a) = l(c), so aRb and bRc implies aRc.
- R is an equivalence relation.

Equivalence Classes

- **Definition:** Let R be an equivalence relation on a set A. The set of all elements that are related to an element a of A is called the **equivalence class** of a.
- The equivalence class of a with respect to R is denoted by **$[a]_R$**.
- $[a]_R = \{s \mid (a, s) \in R\}$.
- When only one relation is under consideration, we will delete the subscript R and write **[a]** for this equivalence class.
- If $b \in [a]_R$, b is called a **representative** of this equivalence class.

Equivalence Classes

- **Example:** In the previous example (strings of identical length), what is the equivalence class of the word mouse, denoted by [mouse] ?

- **Solution:** [mouse] is the set of all English words containing five letters.

- For example, ‘horse’ would be a representative of this equivalence class.

-

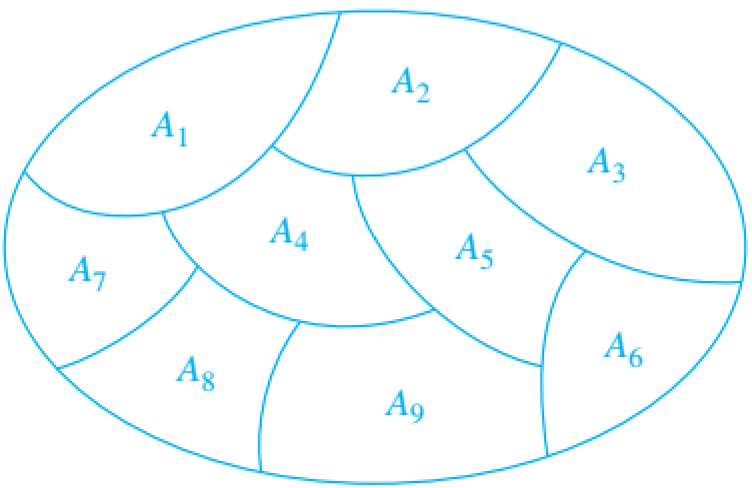
Equivalence Classes

•**Theorem:** Let R be an equivalence relation on a set A. The following statements are equivalent:

- aRb
- [a] = [b]
- $[a] \cap [b] \neq \varnothing$

Definition: A **partition** of a set S is a collection of disjoint nonempty subsets of S that have S as their union. In other words, the collection of subsets A_i, $i \in I$, forms a partition of S if and only if

- $A_i \neq \varnothing$ for $i \in I$
- $A_i \cap A_j = \varnothing$, if $i \neq j$
- $\cup_{i \in I} A_i = S$



Equivalence Classes

- **Examples:** Let S be the set {u, m, b, r, o, c, k, s}.

Do the following collections of sets partition S ?

{{m, o, c, k}, {r, u, b, s}}	yes.
{{c, o, m, b}, {u, s}, {r}}	no (k is missing).
{{b, r, o, c, k}, {m, u, s, t}}	no (t is not in S).
{{u, m, b, r, o, c, k, s}}	yes.
{{b, o, o, k}, {r, u, m}, {c, s}}	yes ({b,o,o,k} = {b,o,k}).
{{u, m, b}, {r, o, c, k, s}, ∅}	no (∅ not allowed).

Equivalence Classes

- **Theorem:** Let R be an equivalence relation on a set S. Then the **equivalence classes** of R form a **partition** of S. Conversely, given a partition $\{A_i \mid i \in I\}$ of the set S, there is an equivalence relation R that has the sets A_i, $i \in I$, as its equivalence classes.

Equivalence Classes

- **Example:** Let us assume that Frank, Suzanne and George live in Boston, Stephanie and Max live in Lübeck, and Jennifer lives in Sydney.

- Let R be the **equivalence relation** {(a, b) | a and b live in the same city} on the set P = {Frank, Suzanne, George, Stephanie, Max, Jennifer}.

- Then R = {(Frank, Frank), (Frank, Suzanne), (Frank, George), (Suzanne, Frank), (Suzanne, Suzanne), (Suzanne, George), (George, Frank), (George, Suzanne), (George, George), (Stephanie, Stephanie), (Stephanie, Max), (Max, Stephanie), (Max, Max), (Jennifer, Jennifer)}.

Equivalence Classes

- Then the **equivalence classes** of R are:
- {{Frank, Suzanne, George}, {Stephanie, Max}, {Jennifer}}.
- This is a **partition** of P.

- The equivalence classes of any equivalence relation R defined on a set S constitute a partition of S, because every element in S is assigned to **exactly one** of the equivalence classes.

Congruence Modulo Definition

- Two numbers are said to be congruent modulo m if their difference is divisible by m. Each integer belongs to one of m congruence (or residue) classes modulo m.
- for some integer k (positive or negative):
- $a=b+km$ or $a-b=km$
- For a mod b **Method 1: add/subtract b**
  Step 1) If a >b keep subtracting b from a until a is between 0 and b-1
  Step 2)If a <0 keep adding b to a until a is between 0 and b-1
  Step 3) return a
- **Method 3: long division allowing negative remainders**
  Step 1) perform long division to divide a by b, without worrying about whether the remainder r is negative or positive. (you will obtain a remainder r between -(b-1) and +(b-1)
  Step 2) if the remainder r is <0 add b
  Step 3) return r

Congruence Modulo Example

- Let m be a positive integer with m>1. Show that the relation R={(a,b)|a ≡b (mod m)} is an equivalence relation on Z
- a≡b (mod m) iff m divides a-b. a-a=0 is divided by m. Hence a≡a (mod m). It is reflexive
- Suppose a≡b(mod m), then a-b is divisible by m, so a-b=km, where k is an integer. It follows b-a=(-k)m, so b≡a(mod m). Hence, It is symmetric
- Suppose a≡b(mod m) and b≡c(mod m). Then m divides both a-b and b-c. Thus, there are integers k and l with a-b=km and b-c=lm
- Together a-c=(a-b)+(b-c) = km+lm = (k+l)m. Thus a≡c(mod m). So, congruence modulo m is transitive
- Therefore congruence modulo m is an equivalence relation

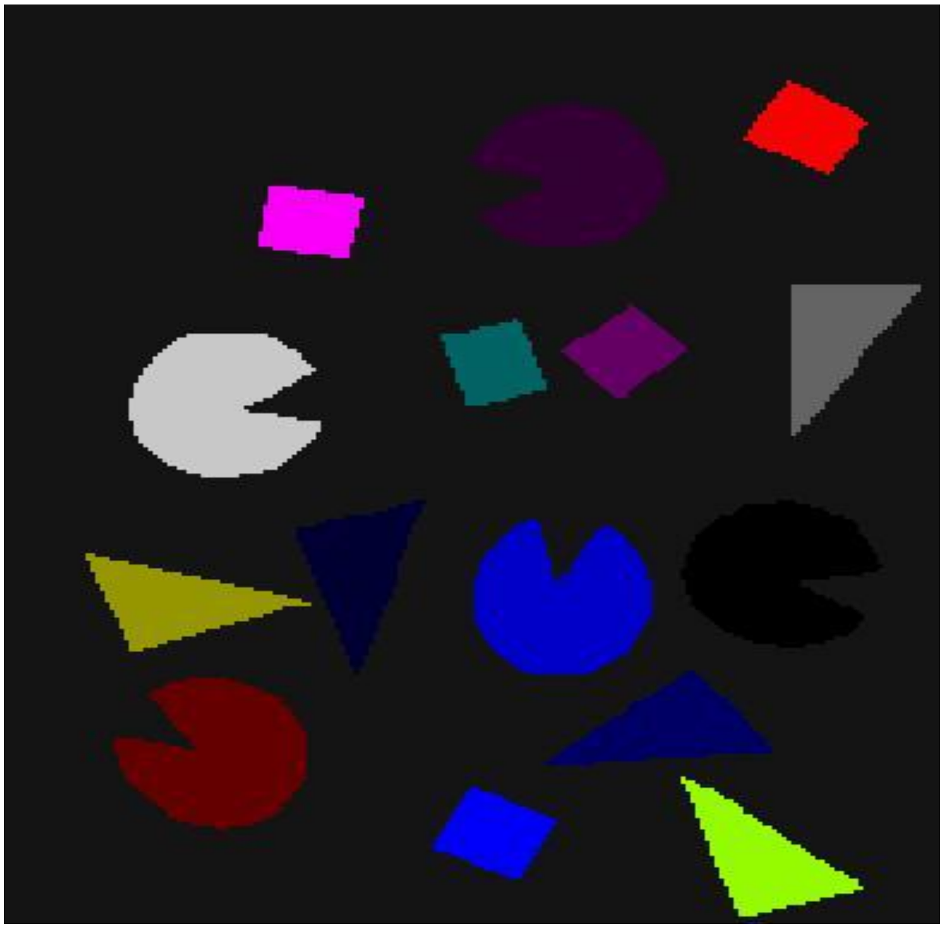
Equivalence Classes

- **Example:** Let R be the relation $\{(a, b) \mid a \equiv b \pmod{3}\}$ on the set of integers.
- Is R an equivalence relation?
- Yes, R is reflexive, symmetric, and transitive.
- What are the equivalence classes of R ?
- $\{\ \{\ldots, -6, -3, 0, 3, 6, \ldots\},\ \ \{\ldots, -5, -2, 1, 4, 7, \ldots\},$
  $\{\ldots, -4, -1, 2, 5, 8, \ldots\}\ \}$

Example: the integers mod m

- a R b = { (a, b) | a ≡ b (mod m) }
- If m=5 then we have 5 classes induced by R
- $[0]_R$ = { …., -15,-10,-5, 0, 5, 10, 15, …}
- $[1]_R$ = { …., -14, -9, -4, 1, 6, 11, 16, …}
- $[2]_R$ = { …., -13, -8, -3, 2, 7, 12, 17, …}
- $[3]_R$ = { …..,-12, -7, -2, 3, 8, 13, 18, …}
- $[4]_R$ = { ….., -11, -6,-1, 4, 9, 14, 19, …}
- Note that every integer belongs to one and only one of these classes – ***R partitions Z***.

Example: pixel adjacency partitions a binary image into “blobs” or “objects”



12 subsets of pixels. Within each subset, pixels are connected by a path through neighbors. Pixels in each subset are not connected to pixels in any other subset. (Connecting paths are not allowed to go through background pixels.)

- Let R be a relation on the set of bit strings. What is R_3. What are the sets of the partition of the set of all bit strings ?

$$[000]_{R_3} = \{000, 0000, 0001, 00000, 00001, 00010, 00011, \ldots\},$$

$$[001]_{R_3} = \{001, 0010, 0011, 00100, 00101, 00110, 00111, \ldots\},$$

$$[010]_{R_3} = \{010, 0100, 0101, 01000, 01001, 01010, 01011, \ldots\},$$

$$[011]_{R_3} = \{011, 0110, 0111, 01100, 01101, 01110, 01111, \ldots\},$$

$$[100]_{R_3} = \{100, 1000, 1001, 10000, 10001, 10010, 10011, \ldots\},$$

$$[101]_{R_3} = \{101, 1010, 1011, 10100, 10101, 10110, 10111, \ldots\},$$

$$[110]_{R_3} = \{110, 1100, 1101, 11000, 11001, 11010, 11011, \ldots\},$$

$$[111]_{R_3} = \{111, 1110, 1111, 11100, 11101, 11110, 11111, \ldots\}.$$

Partial ordering

- We use relations to order some or all of the elements of sets
- Example: order words, schedule projects
- A relation R on a set S is called **partial ordering** or **partial order** if it is *reflexive*, *antisymmetric*, and *transitive*
- A set S together with a partial ordering R is called **partially ordered set**, or **poset**, and is denoted by (S,R)
- Members of S are called **elements** of the poset

Example

- Show that the “greater than or equal” relation ($\geq$) is a partial ordering on the set of integers
- $\geq$ is reflexive as $a \geq a$
- $\geq$ is antisymmetric as if $a \geq b$ and $b \geq a$ then $a=b$
- $\geq$ is transitive as if if $a\geq b$ and $b\geq c$ then $a \geq c$
- $(Z, \geq)$ is a poset

Example

- The divisbility relation | is a partial ordering on the set of positive integers, as it is reflexive, antisymmetric, and transitive
- We see that $(Z^+, |)$ is a poset

Example

- Show that inclusion ⊆ is a partial ordering (the relation of one set being a subset of another is called inclusion) on the power set of a set S
- Example: power set of {0,1,2} is P({0,1,2})= {Ø, {0}, {1}, {2}, {0,1}, {0,2}, {1,2}, {0,1,2}}
- A ⊆ A whenever A is a subset of S, so ⊆ is reflexive
- It is antiysmmetric as A ⊆ B and B ⊆ A imply that A=B
- It is transitive as A ⊆ B and B ⊆ C imply that A ⊆ C. Hence ⊆ is a partial ordering on P(S), and (P(S), ⊆) is a poset

Example

- Let R be the relation on the set of people s.t. xRy if x and y are people and x is older than y. Show that R is not a partial ordering
- R is antisymmetric if a person x is order than a person y, then y is not order than x
- R is transitive
- R is not reflexive as no person is older than himself/herself

Relation in any poset

- In different posets different symbols such as $\leq$, $\subseteq$ and $|$ are used for a partial ordering
- Need a symbol we can use when we discuss the ordering in an **arbitrary** poset
- The notation $a \preccurlyeq b$ is used to denote $(a,b) \in R$ in an arbitrary poset (S,R)
- $\preccurlyeq$ is used as the "less than or equal to" relation on the set of real numbers is the most familiar example of a partial ordering, and similar to $\leq$ symbol
- $\preccurlyeq$ is used for *any poset*, not just "less than or equals" relation , i.e., $(S, \preccurlyeq)$

Lexicographic order

Definition: Given two posets $(A_1,\preccurlyeq_1)$ and $(A_2,\preccurlyeq_2)$, the *lexicographic ordering* on $A_1 \times A_2$ is defined by specifying that (a_1, a_2) is less than (b_1,b_2), that is,

$$(a_1, a_2) \prec (b_1,b_2),$$

either if $a_1 \prec_1 b_1$ or if $a_1 = b_1$ and $a_2 \prec_2 b_2$.

- This definition can be easily extended to a lexicographic ordering on strings (*see text*).

Example: Consider strings of lowercase English letters. A lexicographic ordering can be defined using the ordering of the letters in the alphabet. This is the same ordering as that used in dictionaries.

- *discreet* $\prec$ *discrete*, because these strings differ in the seventh position and $e \prec t$.
- *discreet* $\prec$ *discreetness*, because the first eight letters agree, but the second string is longer.

Comparable

- The elements a and b of a poset $(S, \preccurlyeq)$ are called **comparable** if either $a \preccurlyeq b$ or $b \preccurlyeq a$. When a and b are elements of S s.t. neither $a \preccurlyeq b$ nor $b \preccurlyeq a$, a and b are **incomparable**
- In the poset $(Z+, |)$, are the integers 3 and 9 comparable? Are 5 and 7 comparable?
- The integers 3 and 9 are comparable as $3|9$. the integers 5 and 7 are incomparable as 5 does not divide 7 and 7 does not divide 5

Total ordering

- Pairs of elements may be incomparable and thus we have "partial" ordering
- When every 2 elements in the set are comparable, the relation is called **total ordering**
- If $(S, \preccurlyeq)$ is a poset and every two elements of S are comparable, S is called a **totally ordered** or **linearly ordered set**, and $\preccurlyeq$ is called a **total order** or a **linear order**
- A totally ordered set is also called a **chain**

Example

- The poset $(Z, \leq)$ is totally ordered as $a \leq b$ or $b \leq a$ whenever a and b are integers
- The poset $(Z^+, |)$ is not totally ordered as it contains elements that are incomparable, such as 5 and 7

Well-ordered set

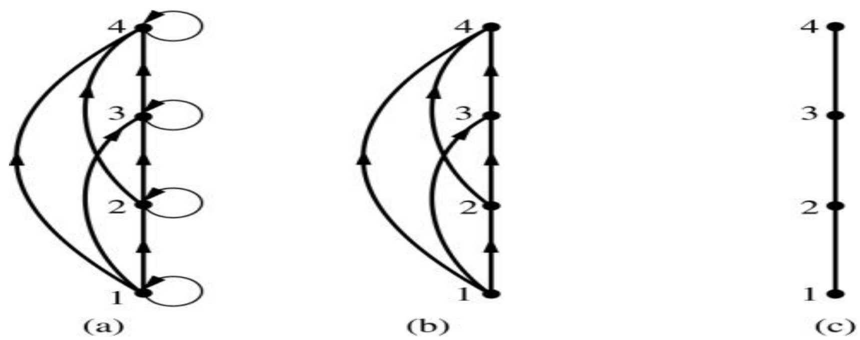
- $(S, \preccurlyeq)$ is a **well-ordered set** if it is a poset s.t. $\preccurlyeq$ is a total ordering and every nonempty subset of S has a **least** element
- The set of ordered pairs of positive integers, $Z^+ \times Z^+$ with (a_1, a_2) (b_1, b_2) if $a_1 < b_1$ or if $a_1 = b_1$ and $a_2 \leq b_2$ (the lexicographic ordering), is a well-ordered set
- The set Z, with the usual ordering, is not well-ordered as the set of negative integers has no least element

Hasse Diagrams

- A *Hasse diagram* is a drawing that represents a partial order relation.
- Named after Helmut Hasse who made extensive use of them
- Draw a diagram in which
  - $a \preccurlyeq b$ is represented by $a \longrightarrow b$; a is drawn below b.
  - Draw the digraph for the partial order
  - Remove the loops as reflexive property is implied
  - If there exists c such that $a \preccurlyeq c$ and $c \preccurlyeq b$, then we represent only $a \leq c$ and $c \preccurlyeq b$; $a \preccurlyeq b$ is implied by transitivity. Remove the transitive edges.
  - Remove the direction arrows. It is from bottom to top.
  - The resulting diagram is the Hasse Diagram

Hasse diagrams

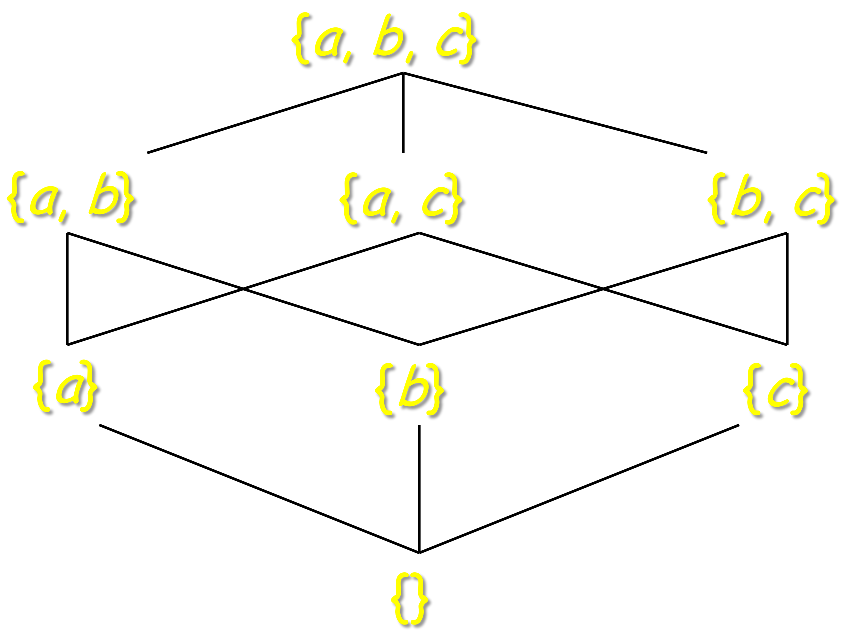
Definition: A *Hasse diagram* is a visual representation of a partial ordering that leaves out edges that must be present because of the reflexive and transitive properties.



A partial ordering is shown in (a) of the figure above. The loops due to the reflexive property are deleted in (b). The edges that must be present due to the transitive property are deleted in (c). The Hasse diagram for the partial ordering (a), is depicted in (c).

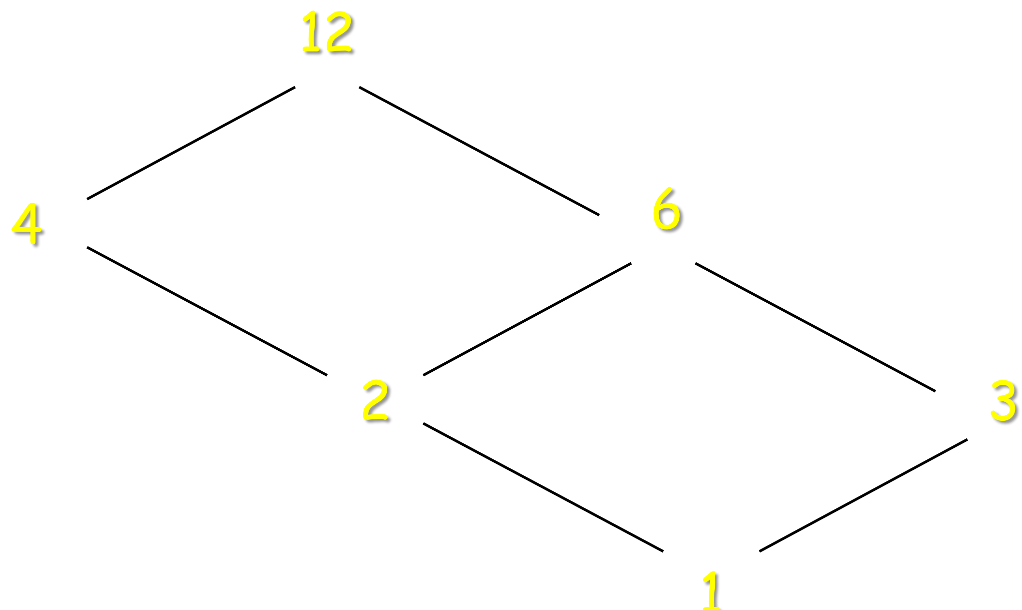
Example: Hasse Diagram

- Let the relation be $\subseteq$ on $\wp(\{a, b, c\})$.



Example: Hasse Diagram

- Let the relation be $|$ on {1, 2, 3, 4, 6, 12}

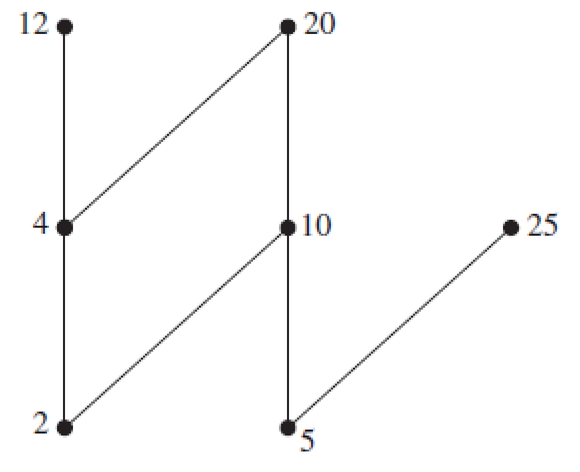


Maximal and Greatest Elements

- Let S be a partially ordered set under $\preccurlyeq$.
- An element $a \in S$ is a *maximal element* if, for all $b \in S$, either $b \preccurlyeq a$ or a and b are not comparable.
- An element $a \in S$ is a *greatest element* if, for all $b \in S$, $b \preccurlyeq a$.

Example

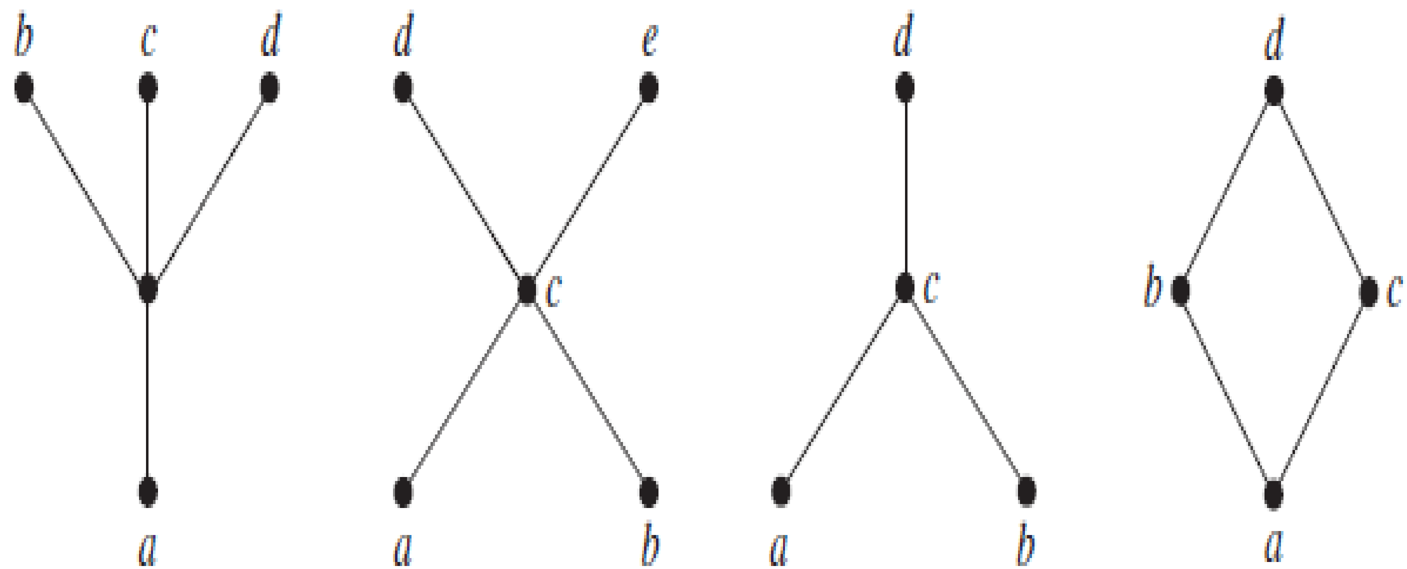
- Let $S = \{2, 4, 5, 10, 12, 20, 25\}$ and let the relation be $a \mid b$.
- Are there any maximal elements?
- Are there any greatest elements?



Minimal and Least Elements

- Minimal and least elements are similar to maximal and greatest elements, respectively.
- An element $a \in S$ is a *minimal element* if, for all $b \in S$, either $a \preccurlyeq b$ or a and b are not comparable.
- An element $a \in S$ is a *least element* if, for all $b \in S$, $a \preccurlyeq b$.

Examples

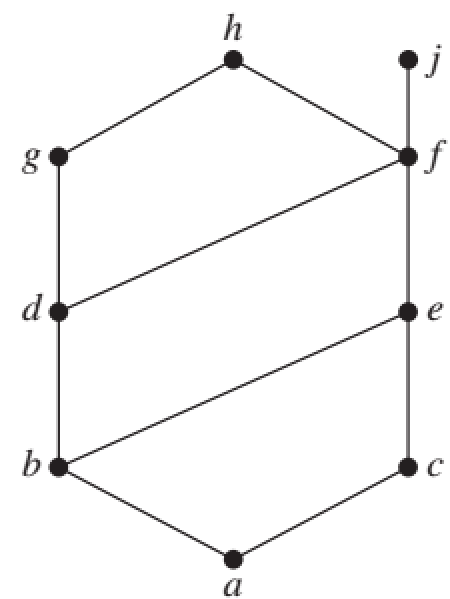


(a) (b) (c) (d)

Upper bound and lub; Lower bound and glb

- A is a subset of a poset $(S, \preccurlyeq)$.
- If u is an element of S such that $a \preccurlyeq u$ for all $a \in A$, then u is called an upper bound of A.
- If l is an element of S such that $l \preccurlyeq a$ for all $a \in A$, then l is called a lower bound of A.

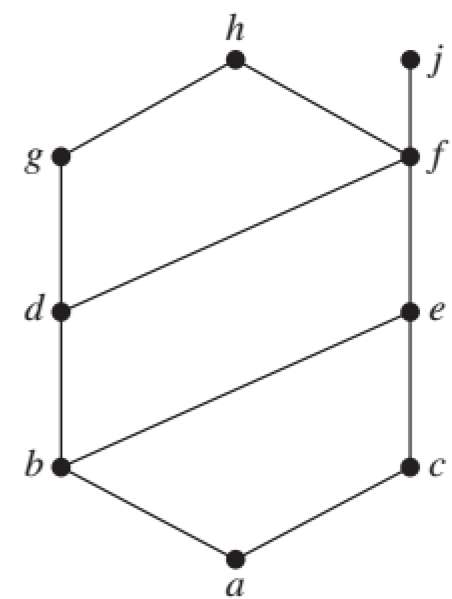
- Upper bounds of {a,b,c} = {e,f,j,h};
- Lower bounds of {a,b,c} = {a};
- Upper bounds of {j,h} = { };
- Lower bounds of {j,h} = {a,b,c,d,e,f};



Upper bound and lub; Lower bound and glb

- *A* is a subset of a poset (S, ≼).
- The 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*. lub is unique if it exists.
- The element *y* is called the greatest lower bound of the subset *A* if *y* is a lower bound that is greater than every other lower bound of *A*. glb is unique if it exists.

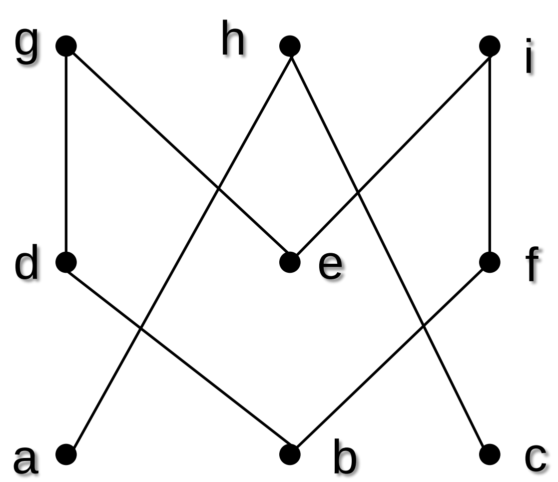
- Upper bounds of {b,d,g} = {g,h};
- lub of {b,d,g} = {g};
- Lower bounds of {b,d,g} = {a,b };
- glb of {b,d,g} = {b};



Extremal Elements: Example 2

Give lower/upper bounds & glb/lub of the sets:

{d,e,f}, {a,c} and {b,d}



{d,e,f}

- Lower bounds: ∅, thus no glb
- Upper bounds: ∅, thus no lub

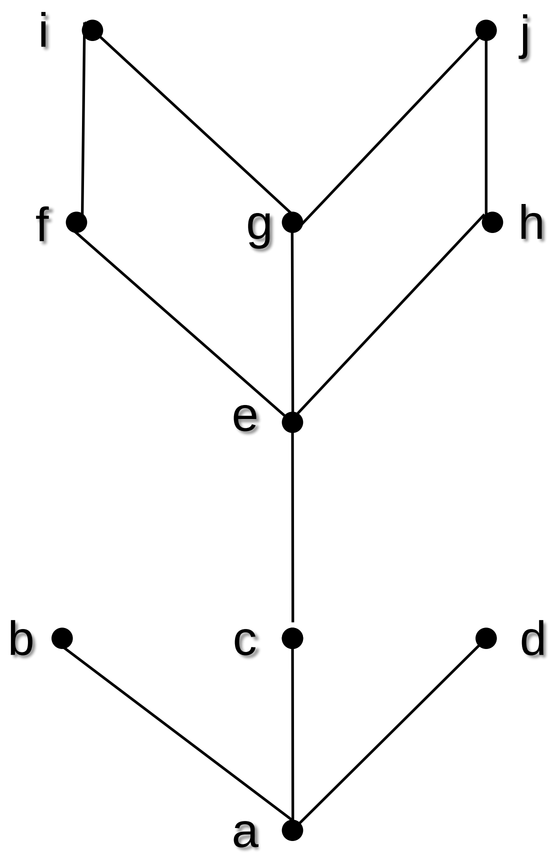
{a,c}

- Lower bounds: ∅, thus no glb
- Upper bounds: {h}, lub: h

{b,d}

- Lower bounds: {b}, glb: b
- Upper bounds: {d,g}, lub: d because d≺g

Extremal Elements: Example 3



- Minimal/Maximal elements?
  - Minimal & Least element: a
  - Maximal elements: b, d, i, j
- Bounds, glb, lub of {c,e}?
  - Lower bounds: {a,c}, thus glb is c
  - Upper bounds: {e,f,g,h,i,j}, thus lub is e
- Bounds, glb, lub of {b,c}?
  - Lower bounds: {a}, thus glb is a
  - Upper bounds: ∅, thus lub Does Not Exist

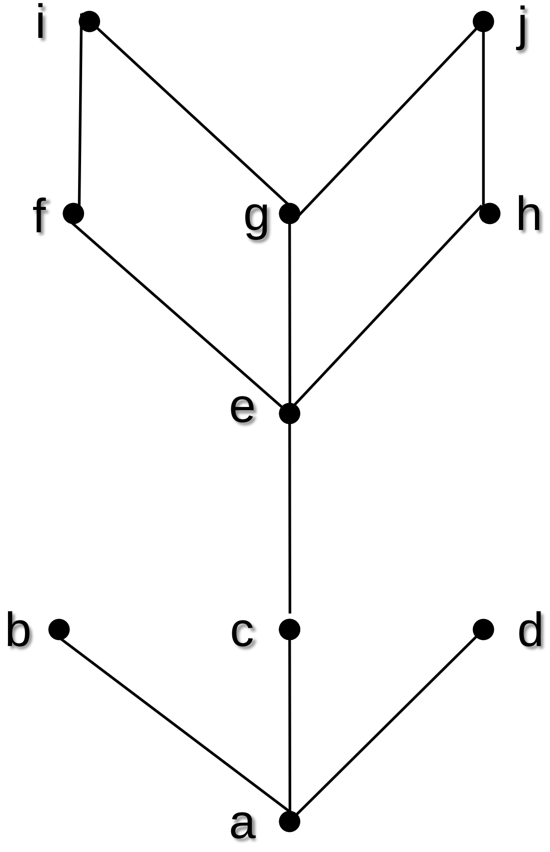
Lattices

- A special structure arises when every pair of elements in a poset has an lub and a glb
- **Definition**: A lattice is a partially ordered set in which every pair of elements has both
  - a least upper bound and
  - a greatest lower bound

Lattices: Example 1

- Is the example from before a lattice?

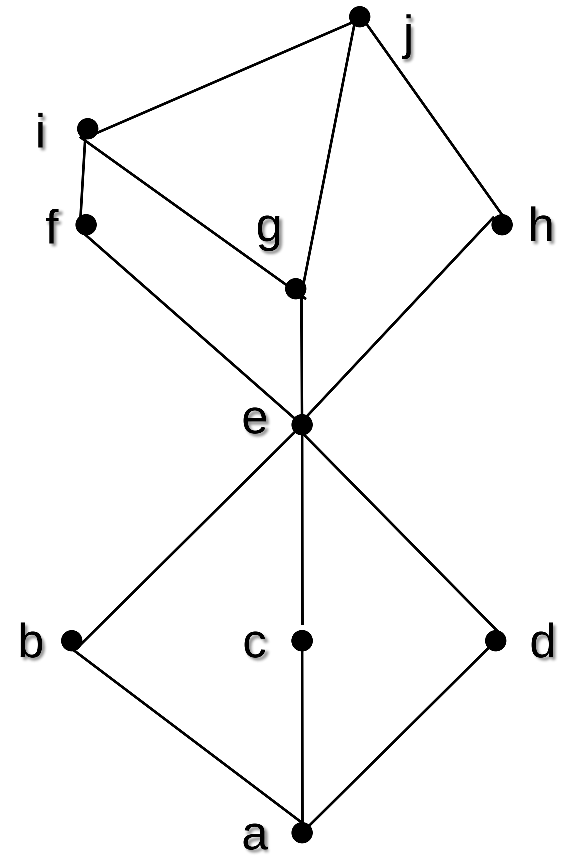
- **No, because the pair {b,c} does not have a least upper bound**



Lattices: Example 2

- What if we modified it as shown here?

- **Yes, because for any pair, there is an lub & a glb**



A Lattice Or Not a Lattice?

- To show that a partial order is not a lattice, it suffices to find a pair that does not have an lub or a glb (i.e., a counter-example)
- For a pair not to have an lub/glb, the elements of the pair must first be <u>incomparable</u>
- You can then view the upper/lower bounds on a pair as a sub-Hasse diagram: If there is no minimum element in this sub-diagram, then it is not a lattice

Compatible Order Relations

- Let $\preccurlyeq_1$ and $\preccurlyeq_2$ be partial order relations defined on a set *A*.
- We call $\preccurlyeq_2$ *compatible* with $\preccurlyeq_1$ if, whenever $a \preccurlyeq_1 b$, it is also the case that $a \preccurlyeq_2 b$.
- In other words,
- $$\forall a, b \in A, a \preccurlyeq_1 b \rightarrow a \preccurlyeq_2 b.$$
- Constructing a compatible total ordering from a partial ordering is called topological sorting

Topological Sorting

- we have got a partial order model, it would be nice to actually create a concrete schedule
- That is, given a partial order, we would like to transform it into a total order that is compatible with the partial order
- A total order is compatible if it does not violate any of the original relations in the partial order
- Essentially, we are simply imposing an order on incomparable elements in the partial order

Topological Sorting: Preliminaries (1)

- some tools to justify its correctness
- **Fact**: Every finite, nonempty poset $(S,\prec)$ has a minimal element
- We will prove the above fact by a form of *reductio ad absurdum*

Topological Sorting: Preliminaries (2)

- **Proof**:
  - Assume, to the contrary, that a nonempty finite poset $(S,\preccurlyeq)$ has no minimal element. In particular, assume that a_1 is not a minimal element.
  - Assume, w/o loss of generality, that $|S|=n$
  - If a_1 is not minimal, then there exists a_2 such that $a_2 \prec a_1$
  - But a_2 is also not minimal because of the above assumption
  - Therefore, there exists a_3 such that $a_3 \prec a_2$. This process proceeds until we have the last element a_n. Thus, $a_n \prec a_{n-1} \prec \ldots \prec a_2 \prec a_1$
  - Finally, by definition a_n is the minimal element **QED**

Topological Sorting: Intuition

- The idea of topological sorting is
  - We start with a poset $(S, \prec)$
  - We remove a minimal element, choosing arbitrarily if there is more than one. Such an element is guaranteed to exist by the previous fact
  - As we remove each minimal element, one at a time, the set S shrinks
  - Thus we are guaranteed that the algorithm will <u>terminate</u> in a finite number of steps
  - Furthermore, the order in which the elements are removed is a total order: $a_1 \prec a_2 \prec \ldots \prec a_{n-1} \prec a_n$
- Now, the algorithm itself

Topological Sorting: Algorithm

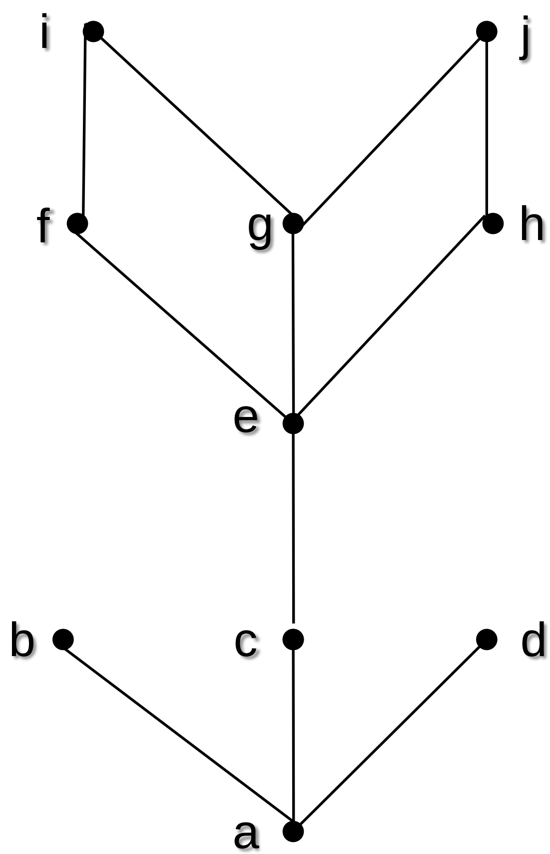
Input: $(S, \preccurlyeq)$ a poset with $|S|=n$

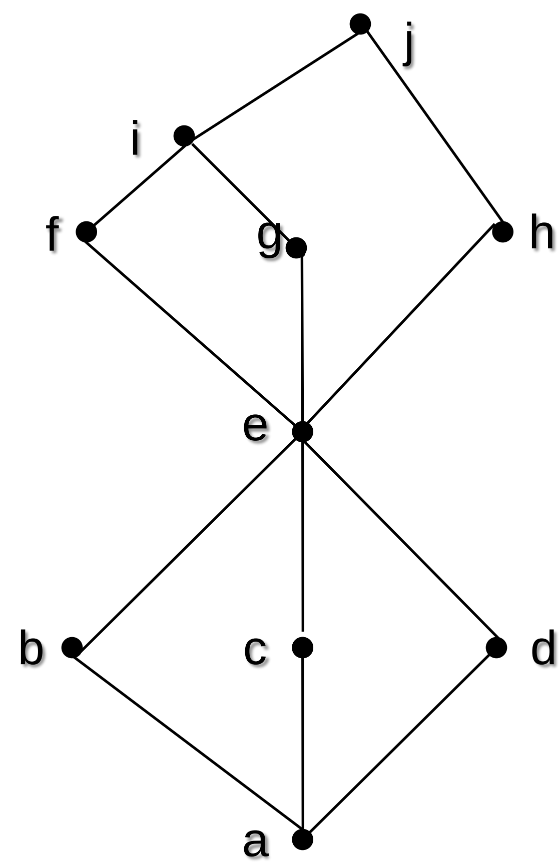
Output: A total ordering $(a_1, a_2, \ldots, a_n)$

1. $k \leftarrow 1$
2. **While** S **Do**
3. $\quad a_k \leftarrow$ a minimal element in S
4. $\quad S \leftarrow S \setminus \{a_k\}$
5. $\quad k \leftarrow k+1$
6. **End**
7. **Return** $(a_1, a_2, \ldots, a_n)$

Topological Sorting: Example

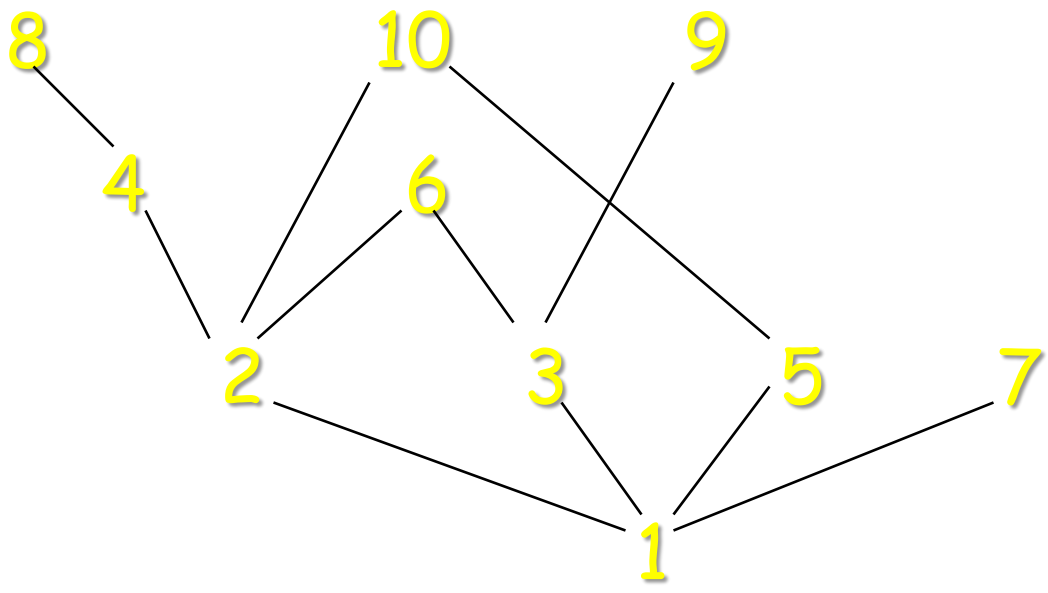
- Find a compatible ordering (topological ordering) of the poset represented by the Hasse diagrams below





Total Ordering Example

- Let $A = \{1, 2, 3, \ldots, 10\}$ under the "divides" relation.



Example

- The total ordering is

  1, 7, 5, 3, 9, 2, 10, 6, 4, 8.

- This ordering is compatible with the divide relation.
- A number of the choices were arbitrary, so there are many other topological sortings for the divide relation on this set.

Summary

- Equivalence relations, equivalence classes, partitions
- Definitions
  - Partial ordering, comparability, total ordering, well ordering
- Lexicographic orderings
  - Idea, on $A_1 \times A_2$, $A_1 \times A_2 \times \ldots \times A_n$, S^t (strings)
- Hasse Diagrams
- Extremal elements
  - Minimal/minimum, maximal/maximum, glb, lub
- Lattices
- Topological Sorting