Discrete Mathematics, KOM1062
Lecture #6

Instructor: Dr. Yavuz Eren

Lecture Book: **"Discrete Mathematics, Seventh Edt., Kenneth H. Rosen, 2007, McGraw Books Discrete Mathematics and Applications, Susanna S. Epp, Brooks, 4th Edt., 2011".**

Spring 2024

1

Equivalence Relations

Equivalence relations are important throughout mathematics and computer science. One reason for this is that in an equivalence relation, when two elements are related it makes sense to say they are equivalent.

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

Definition: Two elements *a and b that are related by an equivalence relation are called equivalent. The* notation *$a \sim b$ is often used to denote that a and b are equivalent elements with respect to a* particular equivalence relation.

For the notion of equivalent elements to make sense, every element should be equivalent to itself, as the reflexive property guarantees for an equivalence relation. It makes sense to say that *a and b are related (not just that a is related to b) by an equivalence relation, because* when *a is related to b, by the symmetric property, b is related to a. Furthermore, because an* equivalence relation is transitive, if *a and b are equivalent and b and c are equivalent, it follows* that *a and c are equivalent.*

Ex. 1: Let *R be the relation on the set of integers such that aRb if and only if $a = b$ or $a = -b$.* Then we know that from the previous lecture note, *R* is reflexive, symmetric, and transitive. It follows that *R* is an equivalence relation

2

Equivalence Relations

Ex. 2: Let R be the relation on the set of real numbers such that aRb if and only if (a − b) is an integer. Is R an equivalence relation?

Solution: Because a−a = 0 is an integer for all real numbers a, aRa for all real numbers a. Hence, R is reflexive.

Now suppose that aRb. Then a−b is an integer, so b−a is also an integer. Hence, bRa. It follows that R is symmetric.

If aRb and bRc, then a−b and b−c are integers. Therefore, a−c = (a−b) + (b−c) is also an integer. Hence, aRc. Thus, R is transitive.

Consequently, R is an equivalence relation.

Equivalence Relations

Ex. 3: Congruence Modulo *m: Let m be an integer with m > 1. Show that the relation*

$R = \{(a, b) \mid a \equiv b \pmod{m}\}$

is an equivalence relation on the set of integers.

Solution: We know the definition of congruence modulo as follows

$a \equiv b \pmod{m}$ if and only if m divides $a - b$.

Reflexivity: $a - a = 0$ is divisible by m, because $0 = 0*m$. Hence, $a \equiv a \pmod{m}$, so congruence modulo m is reflexive.

Symmetry: $a \equiv b \pmod{m}$. Then $a - b$ is divisible by m, so $a - b = k*m$, where k is an integer. It follows that $b - a = (-k)*m$, so $b \equiv a \pmod{m}$. Hence, congruence modulo m is symmetric.

Transitivity: Suppose that $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$. Then m divides both $a - b$ and $b - c$. Therefore, there are integers k and l with $a - b = k*m$ and $b - c = l*m$. Adding these two equations shows that $a - c = (a - b) + (b - c) = k*m + l*m = (k + l)*m$. Thus, $a \equiv c \pmod{m}$. Therefore, congruence modulo m is transitive.

Then we conclude that congruence modulo m is an equivalence relation.

Ex. 4: Suppose that *R is the relation on the set of strings 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: Because *l(a) = l(a)*, it follows that *aRa* whenever *a* is a string, so that *R* is reflexive.

Next, suppose that *aRb*, so that *l(a) = l(b)*. Then *bRa*, because *l(b) = l(a)*. Hence, *R* is symmetric.

Finally, suppose that *aRb* and *bRc*. Then *l(a) = l(b)* and *l(b) = l(c)*. Hence, *l(a) = l(c)*, so *aRc*. Consequently, *R* is transitive.

Because *R* is reflexive, symmetric, and transitive, it is an equivalence relation.

Ex. 4: Show that the "divides" relation is the set of positive integers in not an equivalence relation.

Solution: We know that the "divides" relation is reflexive and transitive from the previous chapter. However, this relation is not symmetric (for instance, 2 divides 4 but 4 not divides 2). We conclude that the "divides" relation on the set of positive integers is not an equivalence relation.

Ex. 5: Let R be the relation on the set of real numbers such that xRy if and only if x and y are real numbers that differ by less than 1, that is $|x - y| < 1$. Show that R is not an equivalence relation.

Solution:

R is reflexive because $|x - x| = 0 < 1$ whenever $x \in R$.

R is symmetric, for if xRy, where x and y are real numbers, then $|x - y| < 1$, which tells us that $|y - x| = |x - y| < 1$, so that yRx.

However, R is not an equivalence relation because it is not transitive. Take $x = 2.8$, $y = 1.9$, and $z = 1.1$, so that $|x - y| = |2.8 - 1.9| = 0.9 < 1$, $|y - z| = |1.9 - 1.1| = 0.8 < 1$, but $|x - z| = |2.8 - 1.1| = 1.7 > 1$. That is, $2.8 R 1.9$, $1.9 R 1.1$, but $2.8 \not R 1.1$.

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* $[a]_R$. *When only one relation is under consideration, we can delete the* subscript *R and write [a] for this equivalence class.*

If $b \in [a]_R$, *then b is called a representative of this equivalence class. Any element of a class* can be used as a representative of this class.

Ex. 6: What are the equivalence classes of 0 and 1 for congruence modulo 4?

Solution: The equivalence class of 0 contains all integers *a* such that $a \equiv 0 \pmod 4$. The integers in this class are those divisible by 4. Hence, the equivalence class of 0 for this relation is

$$[0] = \{\ldots, -8, -4, 0, 4, 8, \ldots\}.$$

The equivalence class of 1 contains all the integers *a* such that $a \equiv 1 \pmod 4$. The integers in this class are those that have a remainder of 1 when divided by 4. Hence, the equivalence class of 1 for this relation is

$$[1] = \{\ldots, -7, -3, 1, 5, 9, \ldots\}.$$

Note that : This example can easily be generalized, replacing 4 with any positive integer *m*. The equivalence classes of the relation congruence modulo *m* are called the congruence classes modulo *m*. The congruence class of an integer *a* modulo *m* is denoted by $[a]_m$, so

$$[a]_m = \{\ldots, a - 2m, a - m, a, a + m, a + 2m, \ldots\}.$$

For this example, it follows that $[0]_4 = \{\ldots, -8, -4, 0, 4, 8, \ldots\}$ and $[1]_4 = \{\ldots, -7, -3, 1, 5, 9, \ldots\}$.

Equivalence Classes and Partitions

Let *A be the set of students at your school who are majoring in exactly one subject, and let R* be the relation on *A consisting of pairs (x, y), where x and y are students with the same major.* Then *R is an equivalence relation.*

We can see that R splits all students in *A into a collection of disjoint subsets, where each subset contains students with* a specified major.

For instance, one subset contains all students majoring (just) in computer science, and a second subset contains all students majoring in history.

Furthermore, these subsets are equivalence classes of *R. This example illustrates how the equivalence classes of an* equivalence relation partition a set into disjoint, nonempty subsets.

We will make these notions more precise in the following discussion.

Theorem: Let *R be an equivalence relation on a set A. These statements for elements a and b of A are* equivalent:

(i) aRb
(ii) $[a] = [b]$
(iii) $[a] \cap [b] = \emptyset$

We are now in a position to show how an equivalence relation *partitions a set. Let R be an* equivalence relation on a set *A. The union of the equivalence classes of R is all of A, because* an element *a of A is in its own equivalence class, namely,* $[a]_R$*. Namely,*

$$\bigcup_{a \in A} [a]_R = A.$$

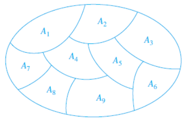
By the theorem, it follows that these equivalence classes are either equal or disjoint $[a]_R \cap [b]_R = \emptyset$ when $[a]_R \neq [b]_R$.

These two observations show that the equivalence classes form a partition of *A, because* they split *A into disjoint subsets. More precisely, 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 ∈ I (where I is an index set) forms a partition of S if and only if*

$$A_i \neq \emptyset \text{ for } i \in I,$$

$$A_i \cap A_j = \emptyset \text{ when } i \neq j,$$

$$\bigcup_{i \in I} A_i = S.$$



Note that, the notation with union operator represents the union of the sets A_i *for all i ∈ I as illustrated in the figure(a partition of set) above.*

Ex. : Suppose that *S = {1, 2, 3, 4, 5, 6}. The collection of sets A1 = {1, 2, 3}, A2 = {4, 5}, and A3 = {6} forms a partition of S, because these sets are disjoint and their union is S.*

We have seen that the equivalence classes of an equivalence relation on a set form a partition of the set. The subsets in this partition are the equivalence classes. To see this, assume that {A_i *| i ∈ I } is a partition on S. Let R be the relation on S consisting* of the pairs *(x, y), where x and y belong to the same subset* A_i *in the partition. To show that R* is an equivalence relation we must show that *R is reflexive, symmetric, and transitive*.

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 *| i ∈ I } of the set S, there is an equivalence relation R that has the sets* A_i *, i ∈ I , as its equivalence classes.*

Ex. List the ordered pairs in the equivalence relation *R produced by the partition A1 = {1, 2, 3}, A2 = {4, 5}, and A3 = {6} of S = {1, 2, 3, 4, 5, 6}.*
Solution: The subsets in the partition are the equivalence classes of *R*. The pair *(a, b) ∈ R* if and only if *a* and *b* are in the same subset of the partition.

The pairs *(*1, 1*)*, *(*1, 2*)*, *(*1, 3*)*, *(*2, 1*)*, *(*2, 2*)*, *(*2, 3*)*, *(*3, 1*)*, *(*3, 2*)*, and *(*3, 3*)* belong to *R* because *A*1 = {1, 2, 3} is an equivalence class;

the pairs *(*4, 4*)*, *(*4, 5*)*, *(*5, 4*)*, and *(*5, 5*)* belong to *R* because *A*2 = {4, 5} is an equivalence class;

and finally the pair *(*6, 6*)* belongs to *R* because {6} is an equivalence class. No pair other than those listed belongs to *R*.

Note that, the congruence classes modulo m provide a useful illustration of the above theorem. There are m different congruence classes modulo m, corresponding to the m different remainders possible when an integer is divided by m. These m congruence classes are denoted by $[0]_m$, $[1]_m$, . . . , $[m − 1]_m$. They form a partition of the set of integers.

Ex. What are the sets in the partition of the integers arising from congruence modulo 4?

Solution: There are four congruence classes, corresponding to $[0]_4$, $[1]_4$, $[2]_4$, and $[3]_4$. They are the sets

$$[0]_4 = \{\dots, -8, -4, 0, 4, 8, \dots\},$$
$$[1]_4 = \{\dots, -7, -3, 1, 5, 9, \dots\},$$
$$[2]_4 = \{\dots, -6, -2, 2, 6, 10, \dots\},$$
$$[3]_4 = \{\dots, -5, -1, 3, 7, 11, \dots\}.$$

Note also that, these congruence classes are disjoint, and every integer is in exactly one of them.

In other words, as the theorem says, these congruence classes form a partition

Ex. What are the sets in the partition of the set of all bit strings arising from the relation R_3 *on the set of all bit strings? (Recall that* sR_3t *, where s and t are bit* strings, if $s = t$ ***or*** *s and t are bit strings with at least three bits that agree in their first three bits.)*

Solution: Note that every bit string of length less than three is equivalent only to itself.

Hence $[\lambda]R_3= \{\lambda\}$, $[0]R_3= \{0\}$, $[1]R_3= \{1\}$, $[00]R_3= \{00\}$, $[01]R_3= \{01\}$, $[10]R_3= \{10\}$, and $[11]R_3= \{11\}$.

Note that every bit string of length three or more is equivalent to one of the eight bit strings 000, 001, 010, 011, 100, 101, 110, and 111. We have

$$[000]_{R_3} = \{000, 0000, 0001, 00000, 00001, 00010, 00011, \dots\},$$
$$[001]_{R_3} = \{001, 0010, 0011, 00100, 00101, 00110, 00111, \dots\},$$
$$[010]_{R_3} = \{010, 0100, 0101, 01000, 01001, 01010, 01011, \dots\},$$
$$[011]_{R_3} = \{011, 0110, 0111, 01100, 01101, 01110, 01111, \dots\},$$
$$[100]_{R_3} = \{100, 1000, 1001, 10000, 10001, 10010, 10011, \dots\},$$
$$[101]_{R_3} = \{101, 1010, 1011, 10100, 10101, 10110, 10111, \dots\},$$
$$[110]_{R_3} = \{110, 1100, 1101, 11000, 11001, 11010, 11011, \dots\},$$
$$[111]_{R_3} = \{111, 1110, 1111, 11100, 11101, 11110, 11111, \dots\}.$$

These 14 equivalence classes are disjoint and every bit string is in exactly one of them. As the theorem 2 tells us, these equivalence classes partition the set of all bit strings.

Partial Orderings

We often use relations to order some or all of the elements of sets. For instance, we order words using the relation containing pairs of words *(x, y), where x comes before y in the dictionary.*

We schedule projects using the relation consisting of pairs *(x, y), where x and y are tasks in* a project such that *x must be completed before y begins.*

We order the set of integers using the relation containing the pairs *(x, y), where x is less than y.*

When we add all of the pairs of the form *(x, x) to these relations, we obtain a relation that is reflexive, anti-symmetric, and* transitive. These are properties that characterize relations used to order the elements of sets.

Definition: A relation *R on a set S is called a partial ordering or partial order if it is reflexive, anti-symmetric,* and transitive. A set *S together with a partial ordering R is called a partially ordered set, or poset, and is denoted by (S,R). Members of S are called elements of the poset.*

Partial Orderings

Ex. Show that the "greater than or equal" relation (≥) is a partial ordering on the set of integers.
Solution:

Because $a \geq a$ for every integer a, ≥ is reflexive.

If $a \geq b$ and $b \geq a$, then $a = b$. Hence, ≥ is antisymmetric.

Finally, ≥ is transitive because $a \geq b$ and $b \geq c$ imply that $a \geq c$.

It follows that ≥ is a partial ordering on the set of integers and ($\mathbf{Z}$, ≥) is a poset.

Ex. Show that the inclusion relation ⊆ is a partial ordering on the power set of a set S.
Solution:

Because A ⊆ A whenever A is a subset of S, ⊆ is reflexive.

It is antisymmetric because A ⊆ B and B ⊆ A imply that A = B.

Finally, ⊆ is transitive, because A ⊆ B and B ⊆ C imply that A ⊆ C.

Hence, ⊆ is a partial ordering on P(S), and (P (S), ⊆) is a poset.

Partial Orderings

Ex. Let R be the relation on the set of people such that xRy if x and y are people and x is older than y. Show that R is not a partial ordering.

Solution:
Note that R is antisymmetric because if a person x is older than a person y, then y is not older than x. That is, if xRy, then $y Rx$.

The relation R is transitive because if person x is older than person y and y is older than person z, then x is older than z. That is, if xRy and yRz, then xRz.

However, R is not reflexive, because no person is older than himself or herself. That is, xRx for all people x. It follows that R is not a partial ordering.

Partial Orderings

$\preccurlyeq$

In different posets different symbols such as ≤, ⊆, and |, are used for a partial ordering. However, we need a symbol that we can use when we discuss the ordering relation in an arbitrary poset. Customarily, the notation $a \preccurlyeq b$ *is used to denote that* $(a, b) \in R$ *in an arbitrary* poset (S,R).

Definition: 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 such that neither* $a \preccurlyeq b$ *nor* $b \preccurlyeq a$, *a and b are called incomparable.*

Ex. In the poset $(\mathbf{Z}^+, |)$, *are the integers 3 and 9 comparable? Are 5 and 7 comparable?*

Solution: The integers 3 and 9 are comparable, because 3 | 9. The integers 5 and 7 are incomparable, because 5 not divide 7 and 7 not divide 5.

Definition: 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.*

Ex. The poset $(Z, \leq)$ *is totally ordered, because* $a \leq b$ *or* $b \leq a$ *whenever a and b are integers.*

Definition: $(S, \preccurlyeq)$ *is a well-ordered set if it is a poset such that is a total ordering and every nonempty* subset of *S has a least element.*

Ex. 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.*

Hasse Diagrams

Many edges in the directed graph for a finite poset do not have to be shown because they must be present.

For instance, consider the directed graph for the partial ordering *{(a, b) | a ≤ b} on the* set {1*, 2, 3, 4}, shown in Figure a.*

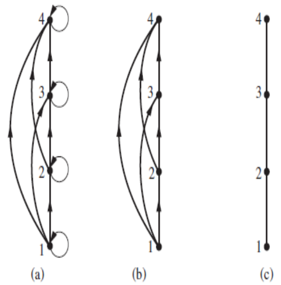
Because this relation is a partial ordering, it is reflexive, and its directed graph has loops at all vertices.

Consequently, we do not have to show these loops because they must be present; in Figure b loops are not shown.

Because a partial ordering is transitive, we do not have to show those edges that must be present because of transitivity.

For example, in Figure c the edges *(1, 3), (1, 4), and (2, 4) are not shown because they must be* present.

If we assume that all edges are pointed "upward" (as they are drawn in the figure), we do not have to show the directions of the edges; Figure 2(c) does not show directions.

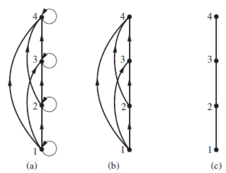


Hasse Diagrams

In general, we can represent a finite poset *(S, ≼) using this procedure:*

- *Start with the* directed graph for this relation. Because a partial ordering is reflexive, a loop *(a, a) is present at* every vertex *a. Remove these loops.*
- *Next, remove all edges that must be in the partial ordering* because of the presence of other edges and transitivity. That is, remove all edges *(x, y) for* which there is an element *z ∈S such that x ≺z and z ≺x.*
- *Finally, arrange each edge so that* its initial vertex is below its terminal vertex. Remove all the arrows on the directed edges, because all edges point "upward" toward their terminal vertex.

When all the steps have been taken, the resulting diagram contains sufficient information to find the partial ordering, as we will explain later. The resulting diagram is called the Hasse diagram of *(S, ≼)*



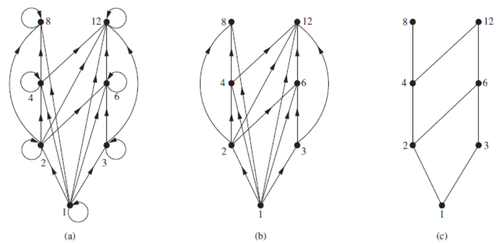
Ex. Draw the Hasse diagram representing the partial ordering {*(a, b) |a divides b} on* {*1, 2, 3, 4, 6, 8, 12}*.

Solution: Begin with the digraph for this partial order, as shown in Figure (a).

Remove all loops, as shown in Figure (b).

Then delete all the edges implied by the transitive property. These are *(1, 4)*, *(1, 6)*, *(1, 8)*, *(1, 12)*, *(2, 8)*, *(2, 12)*, and *(3, 12)*.

Arrange all edges to point upward, and delete all arrows to obtain the Hasse diagram. The resulting Hasse diagram is shown in Figure (c).



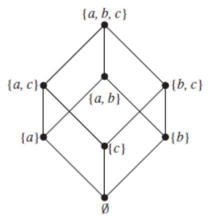
Ex. Draw the Hasse diagram for the partial ordering {*(A,B) | A ⊆ B} on the power set P(S) where S = {a, b, c}.*

Solution: The Hasse diagram for this partial ordering is obtained from the associated digraph by deleting all the loops and all the edges that occur from transitivity, namely,

(∅, {a, b}), (∅, {a, c}), (∅, {b, c}), (∅, {a, b, c}), ({a}, {a, b, c}), ({b}, {a, b, c}), and *({c}, {a, b, c}).*

Finally all edges point upward, and arrows are deleted.

The resulting Hasse diagram of(P({a,b,c}), ⊆) is illustrated in Figure 4.



Maximal and Minimal Elements

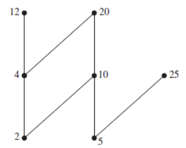
Elements of posets that have certain extremal properties are important for many applications. An element of a poset is called maximal if it is not less than any element of the poset.

That is, a is maximal in the poset $(S, \preccurlyeq)$ *if there is no* $b \in S$ *such that* $a \prec b$.

Similarly, an element of a poset is called minimal if it is not greater than any element of the poset. That is, *a is minimal* if there is no element $b \in S$ *such that* $b \prec a$.

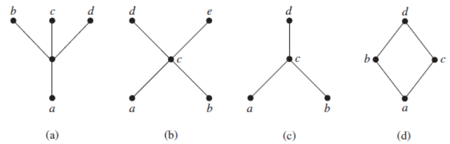
Maximal and minimal elements are easy to spot using a Hasse diagram. They are the "top" and "bottom" elements in the diagram.

Ex. Which elements of the poset *({2, 4, 5, 10, 12, 20, 25}, |) are maximal, and which are minimal?*
Solution:
The Hasse diagram in figure below for this poset shows that the maximal elements are 12, 20, and 25, and the minimal elements are 2 and 5. As this example shows, a poset can have more than one maximal element and more than one minimal element.



Sometimes there is an element in a poset that is greater than every other element. Such an element is called the greatest element. That is, *a is the* greatest element of the poset $(S, \preccurlyeq)$. if $b \preccurlyeq a$ *for all* $b \in S$. Likewise, an element is called the least element if it is less than all the other elements in the poset. That is, *a is the least element of* $(S, \preccurlyeq)$ *if* $a \preccurlyeq b$ *for all* $b \in S$.

Ex. Determine whether the posets represented by each of the Hasse diagrams in Figure below have a greatest element and a least element.
Solution:



The least element of the poset with Hasse diagram (a) is *a*. This poset has no greatest element.

The poset with Hasse diagram (b) has neither a least nor a greatest element.

The poset with Hasse diagram (c) has no least but it has greatest element(d).

The poset with Hasse diagram (d) has least element *a* and greatest element *d*.

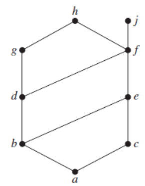
Ex. Let *S be a set. Determine whether there is a greatest element and a least element in the poset (P (S),⊆).*

Solution:
The least element is the empty set, because $\emptyset \subseteq T$ *for any subset T of S.*
The set S is the greatest element in this poset, because $T \subseteq S$ *whenever T is a subset of S.*

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, ≼). If u is an element of S such that a ≼ u for all elements a ∈ A,* then *u is called an upper bound of A. Likewise, there may be an element less than or equal to* all the elements in *A. If l is an element of S such that l ≼ a for all elements a ∈ A, then l is* called a lower bound of *A.*

Ex. Find the lower and upper bounds of the subsets {*a, b, c*}, *{j, h}, and {a, c, d, f } in the poset* with the Hasse diagram shown in Figure below.

Solution:



The upper bounds of {*a, b, c*} are *e, f, j*, and *h*, and its only lower bound is *a*.

There are no upper bounds of {*j, h*}, and its lower bounds are *a, b, c, d, e*, and *f* .

The upper bounds of {*a, c, d, f* } are *f* , *h*, and *j* , and its lower bound is *a*.

Ex. Find the greatest lower bound and the least upper bound of {*b, d, g*}*, if they exist, in the poset* shown in Figure above.

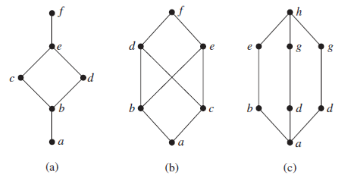
Solution:
The upper bounds of {*b, d, g*} are *g* and *h*. Because $g \prec h$, *g* is the least upper bound.
The lower bounds of {*b, d, g*} are *a* and *b*. Because $a \prec b$, *b* is the greatest lower bound.

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 have many special properties. Furthermore, 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 figure below are lattices.

Solution:



The posets represented by the Hasse diagrams in (a) and (c) are both lattices because in each poset every pair of elements has both a least upper bound and a greatest lower bound.

On the other hand, the poset with the Hasse diagram shown in (b) is not a lattice, because the elements *b* and *c* have no least upper bound. To see this, note that each of the elements *d*, *e*, and *f* is an upper bound, but none of these three elements precedes the other two with respect to the ordering of this poset.

Ex. Is the poset *($\mathbf{Z}^+$, |) a lattice?*

Solution: Let *a* and *b* be two positive integers. The least upper bound and greatest lower bound of these two integers are the least common multiple and the greatest common divisor of these integers, respectively, as the reader should verify. It follows that this poset is a lattice.

Ex. Determine whether the posets *({1, 2, 3, 4, 5}, |) and ({1, 2, 4, 8, 16}, |) are lattices.*

Solution:
Because 2 and 3 have no upper bounds in *({*1, 2, 3, 4, 5*}, |)*, they certainly do not have a least upper bound. Hence, the first poset is not a lattice.

Every two elements of the second poset have both a least upper bound and a greatest lower bound. The least upper bound of two elements in this poset is the larger of the elements and the greatest lower bound of two elements is the smaller of the elements. Hence, this second poset is a lattice.

Ex. Determine whether *(P (S),⊆) is a lattice where S is a set.*

Solution:

Let *A* and *B* be two subsets of *S*. The least upper bound and the greatest lower bound of *A* and *B* are $A \cup B$ and $A \cap B$, respectively. Hence, *(P (S),⊆)* is a lattice.

Topological Sorting

Suppose that a project is made up of 20 different tasks.

Some tasks can be completed only after others have been finished. How can an order be found for these tasks?

To model this problem we set up a partial order on the set of tasks so that *$a \prec b$ if and only if a and b are tasks where b* cannot be started until *a has been completed.*

To produce a schedule for the project, we need to produce an order for all 20 tasks that is compatible with this partial order.

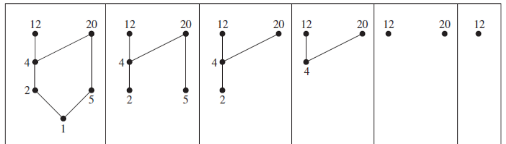
We begin with a definition. A total ordering is said to be compatible with the partial ordering *R if a b whenever aRb. Constructing a compatible total ordering from a partial* ordering is called topological sorting.

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

Topological Sorting

Ex. Find a compatible total ordering for the poset *({1, 2, 4, 5, 12, 20}, |).*

Solution:
The first step is to choose a minimal element. This must be 1, because it is the only minimal element.
Next, select a minimal element of *({2, 4, 5, 12, 20}, |).* There are two minimal elements in this poset, namely, 2 and 5. We select 5.
The remaining elements are {2, 4, 12, 20}. The only minimal element at this stage is 2.
Next, 4 is chosen because it is the only minimal element of *({4, 12, 20}, |).* Because both 12 and 20 are minimal elements of *({12, 20}, |)*, either can be chosen next. We select 20, which leaves 12 as the last element left. This produces the total ordering $1 \prec 5 \prec 2 \prec 4 \prec 20 \prec 12$.
The steps used by this sorting algorithm are displayed in Figure below.



Minimal element chosen	1	5	2	4	20	12