Discrete Mathematics for Computer Science

Lecture 17: Relation

Dr. Ming Tang

Department of Computer Science and Engineering
Southern University of Science and Technology (SUSTech)
Email: tangm3@sustech.edu.cn

Binary Relation

Definition: Let A and B be two sets. A binary relation from A to B is a subset of a Cartesian product $A \times B$.

- Reflexive Relation: A relation R on a set A is called reflexive if $(a, a) \in R$ for every element $a \in A$.
- Irreflexive Relation: A relation R on a set A is called irreflexive if $(a, a) \notin R$ for every element $a \in A$.
- Symmetric Relation: A relation R on a set A is called symmetric if $(b, a) \in R$ whenever $(a, b) \in R$ for all $a, b \in A$.
- Antisymmetric Relation: A relation R on a set A is called antisymmetric if $(b, a) \in R$ and $(a, b) \in R$ implies $a = b$ for all $a, b \in A$.
- Transitive Relation: A relation R on a set A is called transitive if $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$ for all $a, b, c \in A$

n-ary Relations

Definition: An n-ary relation R on sets $A_1, ..., A_n$, written as $R : A_1, ..., A_n$, is a subset $R \subseteq A_1 \times \cdots \times A_n$.

A relational database is essentially an n-ary relation R.

- Primary key, composite key
- Selection operator: $C : A \to \{T, F\}$

$$s_C(R) = \{a \in R | C(a) = T\}.$$

- Projection operator: $P_{i_k} : A \to A_{i_1} \times \cdots \times A_{i_m}$, where $i_1 < i_2 < \cdots < i_m$.
- Join operator: Puts two relations together to form a sort of combined relation.

Representing Relations

Some ways to represent n-ary relations:

- with an explicit list or table of its tuples
- with a function from the domain to $\{T, F\}$

Some special ways to represent binary relations:

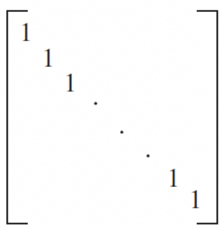
- with a zero-one matrix
- with a directed graph

Zero-One Matrix

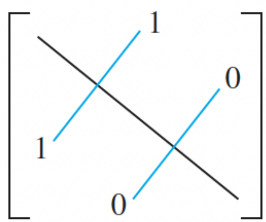
Example: Suppose that $A = \{1, 2, 3\}$ and $B = \{1, 2\}$. Let R be the relation from A to B containing (a, b) if $a \in A$, $b \in B$, and $a > b$.

Solution: $R = \{(2, 1), (3, 1), (3, 2)\}$

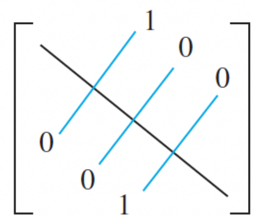
$$\mathbf{M}_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$



Reflexive



Symmetric



Antisymmetric

Zero-One Matrix: Composite of Relations

Let $A = [a_{ij}]$ be an $m \times k$ zero–one matrix and $B = [b_{ij}]$ be a $k \times n$ zero–one matrix. Then, the Boolean product of A and B, denoted by $A \odot B$, is the $m \times n$ matrix with (i, j)-th entry c_{ij} where

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \cdots \vee (a_{ik} \wedge b_{kj}).$$

Example:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

$$\begin{aligned}
\mathbf{A} \odot \mathbf{B} &= \begin{bmatrix} (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \\ (0 \wedge 1) \vee (1 \wedge 0) & (0 \wedge 1) \vee (1 \wedge 1) & (0 \wedge 0) \vee (1 \wedge 1) \\ (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \end{bmatrix} \\
&= \begin{bmatrix} 1 \vee 0 & 1 \vee 0 & 0 \vee 0 \\ 0 \vee 0 & 0 \vee 1 & 0 \vee 1 \\ 1 \vee 0 & 1 \vee 0 & 0 \vee 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.
\end{aligned}$$

Zero-One Matrix: Composite of Relations

Suppose that R is a relation from A to B and S is a relation from B to C:

$$M_{S\circ R} = M_R \odot M_S.$$

The ordered pair (a_i, c_j) belongs to $S \circ R$ if and only if there is an element b_k such that (a_i, b_k) belongs to R and (b_k, c_j) belongs to S.

Example:

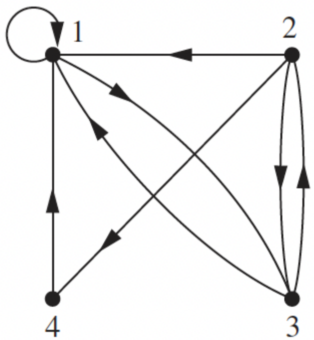
$$\mathbf{M}_R = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{M}_S = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

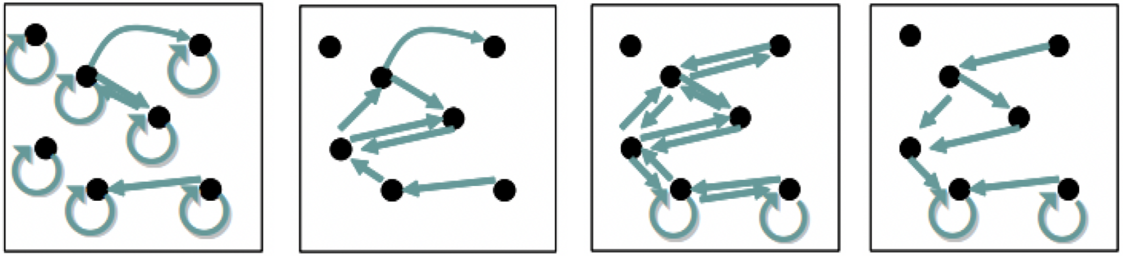
$$\mathbf{M}_{S\circ R} = \mathbf{M}_R \odot \mathbf{M}_S = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Directed Graph

Relation R is defined on $A = \{1, 2, 3, 4\}$:

$$R = \{(1,1), (1,3), (2,1), (2,3), (2,4), (3,1), (3,2), (4,1)\}$$

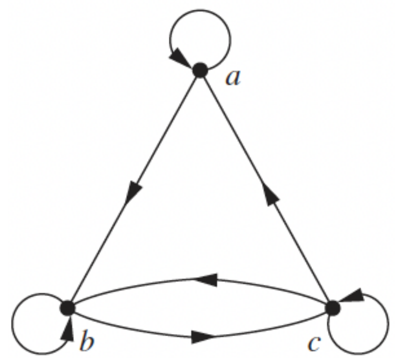




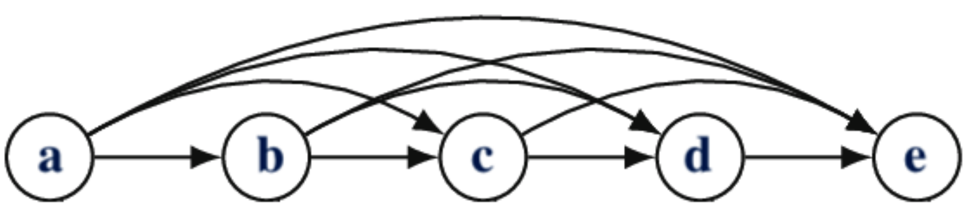
reflexive irreflexive symmetric antisymmetric

Transitive Relation: Example

Determine whether the relations for the directed graphs are reflexive, symmetric, antisymmetric, and/or transitive.



Reflexive, not symmetric, not antisymmetric, not transitive (because there is an edge from a to b and an edge from b to c, but no edge from a to c).



Closures of Relations

The minimal set $S \supseteq R$ is called the reflexive closure of R.

- contains R
- is reflexive
- is minimal (is contained in every reflexive relation Q that contains R ($R \subseteq Q$), i.e., $S \subseteq Q$)

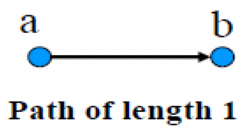
Symmetric closures, transitive closures

Path Length

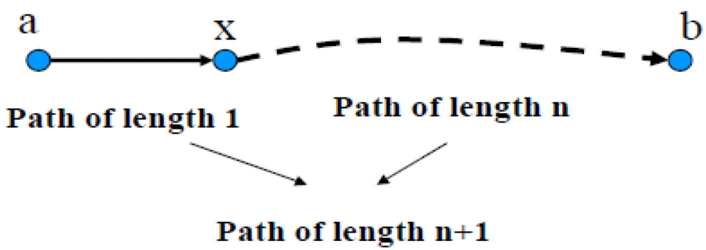
Theorem: Let R be relation on a set A. There is a path of length n from a to b if and only if $(a, b) \in R^n$.

Proof (by mathematical induction):

- Basic Step:

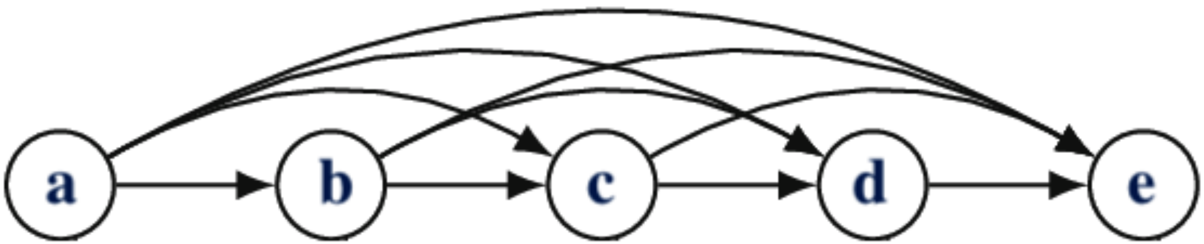


- Inductive Step: Suppose the theorem is true for the positive integer n. Thus, there is a path of length $n+1$ from a to b if and only if there is an element x with $(a, x) \in R$ and $(x, b) \in R_n$. There is such an element x if and only if $(a, b) \in R^{n+1}$.



Transitive Relation and R^n

Theorem: The relation R on a set A is transitive if and only if $R^n \subseteq R$ for $n = 1, 2, 3, \ldots$



Connectivity Relation

Definition: Let R be a relation on a set A. The connectivity relation R^* consists of all pairs (a, b) such that there is a path (of any length) between a and b in R:

$$R^* = \bigcup_{k=1}^{\infty} R^k$$

Lemma: Let A be a set with n elements, and R be a relation on A. If there is a path from a to b with $a \neq b$, then there exists a path of length $\leq n - 1$.

Lemma: If there is a path of length at least one in R from a to b, then there is such a path with length not exceeding n.

Connectivity

Theorem: The transitive closure of a relation R equals the connectivity relation $R^* = \bigcup_{k=1}^{\infty} R^k$.

Recall: Finding a transitive closure corresponds to finding all pairs of elements that are connected with a directed path.

Recall: The connectivity relation R^* consists of all pairs (a, b) such that there is a path (of any length) between a and b in R:

Connectivity

Theorem: The transitive closure of a relation R equals the connectivity relation $R^* = \bigcup_{k=1}^{\infty} R^k$.

- R^* is transitive

  If $(a, b) \in R^*$ and $(b, c) \in R^*$, then there are paths from a to b and from b to c in R. Thus, there is a path from a to c in R^*. This means that $(a, c) \in R^*$.

- $R^* \subseteq S$ whenever S is a transitive relation containing R
  - Suppose that S is a transitive relation containing R.
  - $S^n \subseteq S$ for integer $n \geq 1$. (Recall S is transitive iff $S^n \subseteq S$).
  - We have $S^* \subseteq S$.
  - If $R \subseteq S$, then $R^* \subseteq S^*$, because any path in R is also a path in S.
  - Thus, $R^* \subseteq S^* \subseteq S$.

Find Transitive Closure

Recall that if there is a path of length at least one in R from a to b, then there is such a path with length not exceeding n. Thus,

$$R^* = R \cup R^2 \cup R^3 \cup \cdots \cup R^n.$$

Theorem: Let M_R be the zero–one matrix of the relation R on a set with n elements. Then the zero–one matrix of the transitive closure R^* is

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]} \vee \cdots \vee M_R^{[n]},$$

where $M_R^{[n]} = \underbrace{M_R \odot M_R \odot \cdots \odot M_R}_{n\ M_R's}$

Find Transitive Closure: Example

Find the zero–one matrix of the transitive closure of the relation R where

$$\mathbf{M}_R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

Solution:

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]}$$

$$\mathbf{M}_{R^*} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Simple Transitive Closure Algorithm

ALGORITHM 1 A Procedure for Computing the Transitive Closure.

procedure *transitive closure* ($\mathbf{M}_R$: zero–one $n \times n$ matrix)
$\mathbf{A} := \mathbf{M}_R$
$\mathbf{B} := \mathbf{A}$
for $i := 2$ **to** n
    $\mathbf{A} := \mathbf{A} \odot \mathbf{M}_R$
    $\mathbf{B} := \mathbf{B} \vee \mathbf{A}$
return $\mathbf{B}$ {$\mathbf{B}$ is the zero–one matrix for R^*}

- $n - 1$ Boolean products
- Each of these Boolean products use $n^2(2n - 1)$ bit operations.
- $O(n^4)$ bit operations.

Roy-Warshall Algorithm

The transitive closure can be found by Warshall's algorithm using only $O(n^3)$ bit operations.

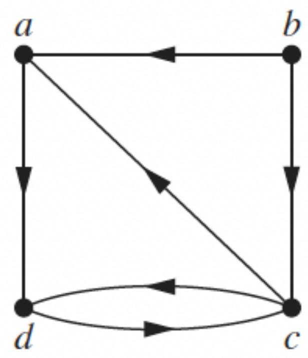
If $a, x_1, x_2, \ldots, x_{m-1}, b$ is a path, its interior vertices are $x_1, x_2, \ldots, x_{m-1}$.

Consider a list of vertices $v_1, v_2, \ldots, v_k, \ldots, v_n$. Define a zero-one matrix

$$\mathbf{W}_k = [w_{ij}^{(k)}],$$

where $w_{ij}^{(k)} = 1$ if there is a path from v_i to v_j such that all the interior vertices of this path are in the set $\{v_1, v_2, \ldots, v_k\}$ and is 0 otherwise.

Example of $\mathbf{W}_k$

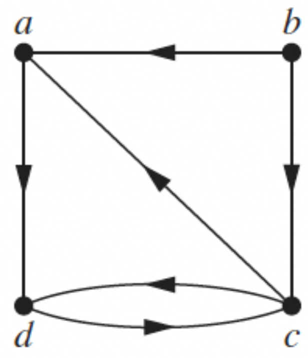


Let $v_1 = a$, $v_2 = b$, $v_3 = c$, and $v_4 = d$.

W_0 is the matrix of the relation.

$$\mathbf{W}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Example of $\mathbf{W}_k$



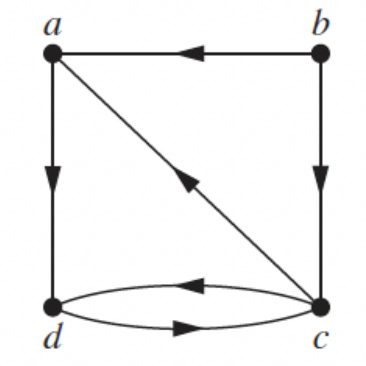
Let $v_1 = a$, $v_2 = b$, $v_3 = c$, and $v_4 = d$.

W_1 has 1 as its (i, j)th entry if

- W_0 has 1 as its (i, j)th entry (i.e., no interior) or
- there is a path from v_i to v_j that has only $v_1 = a$ as an interior vertex

$$\mathbf{W}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Example of $\mathbf{W}_k$



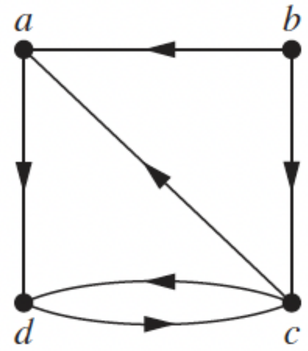
Let $v_1 = a$, $v_2 = b$, $v_3 = c$, and $v_4 = d$.

W_2 has 1 as its (i, j)th entry if

- W_1 has 1 as its (i, j)th entry (i.e., no interior, $v_1 = a$ as interior) or
- there is a path from v_i to v_j that has $v_2 = b$ as an interior vertex

$$W_2 = W_1$$

Example of $\mathbf{W}_k$



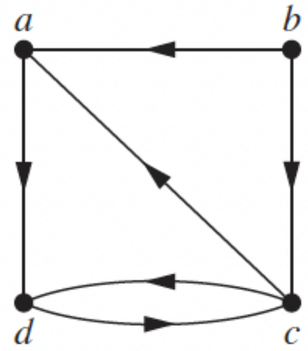
Let $v_1 = a$, $v_2 = b$, $v_3 = c$, and $v_4 = d$.

W_3 has 1 as its (i, j)th entry if

- W_2 has 1 as its (i, j)th entry (i.e., no interior, $v_1 = a$ and/or $v_2 = b$ as interior) or
- there is a path from v_i to v_j that has $v_3 = c$ as an interior vertex

$$\mathbf{W}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

Example of $\mathbf{W}_k$



Let $v_1 = a$, $v_2 = b$, $v_3 = c$, and $v_4 = d$.

W_4 has 1 as its (i, j)th entry if

- W_3 has 1 as its (i, j)th entry (i.e., no interior, $v_1 = a$ and/or $v_2 = b$ and/or $v_3 = c$ as interior) or
- there is a path from v_i to v_j that has $v_4 = d$ as an interior vertex

$$\mathbf{W}_4 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

Roy-Warshall Algorithm

Consider a list of vertices $v_1, v_2, \ldots, v_k, \ldots, v_n$. Define a zero-one matrix

$$\mathbf{W}_k = [w_{ij}^{(k)}],$$

where $w_{ij}^{(k)} = 1$ if there is a path from v_i to v_j such that all the interior vertices of this path are in the set $\{v_1, v_2, \ldots, v_k\}$ and is 0 otherwise.

Note that $\mathbf{W}_n = M_{R^*}$, because the (i, j)th entry of M_{R^*} is 1 if and only if there is a path from v_i to v_j with all interior vertices in the set $\{v_1, v_2, \ldots, v_n\}$.

Roy-Warshall Algorithm

Warshall's algorithm computes M_{R^*} by efficiently computing $\mathbf{W}_0 = M_R, W_1, W_2, ..., \mathbf{W}_n = M_{R^*}$.

Let $\mathbf{W}_k = [w_{ij}^{[k]}]$ be the zero–one matrix that has a 1 in its (i, j)th position if and only if there is a path from v_i to v_j with interior vertices from the set $\{v_1, v_2, \ldots, v_k\}$. Then

$$w_{ij}^{[k]} = w_{ij}^{[k-1]} \vee (w_{ik}^{[k-1]} \wedge w_{kj}^{[k-1]}),$$

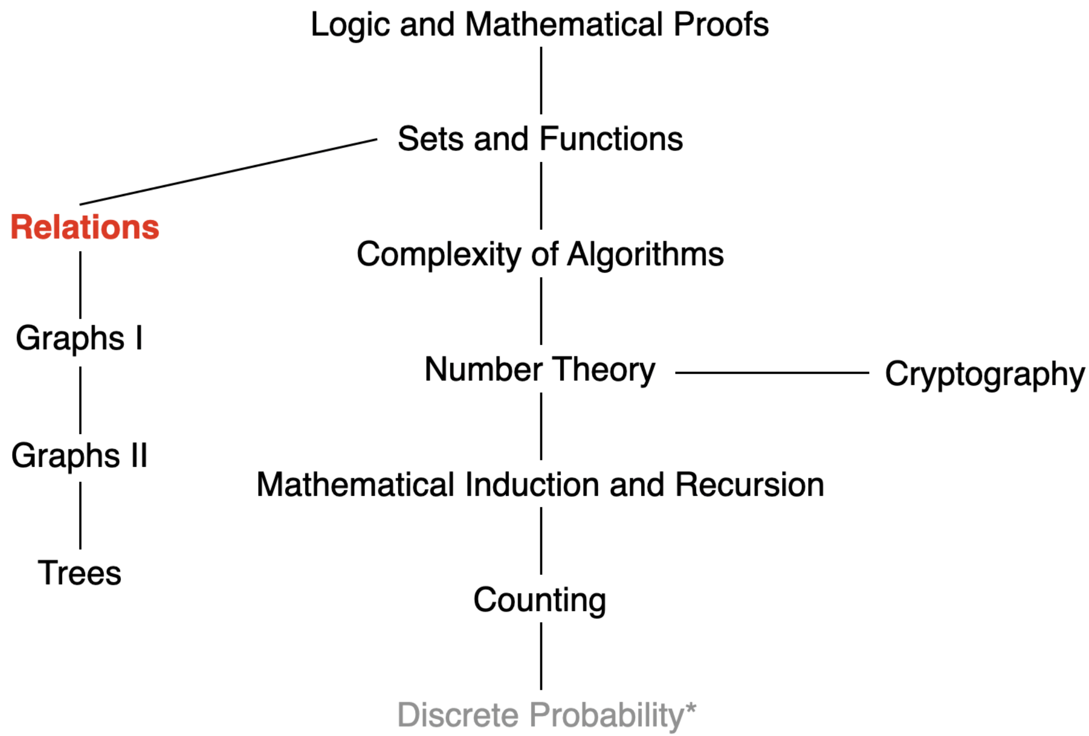
whenever i, j, and k are positive integers not exceeding n.

ALGORITHM 2 Warshall Algorithm.

procedure *Warshall* ($\mathbf{M}_R$: $n \times n$ zero–one matrix)
$\mathbf{W} := \mathbf{M}_R$
for $k := 1$ **to** n
  for $i := 1$ **to** n
    for $j := 1$ **to** n
      $w_{ij} := w_{ij} \vee (w_{ik} \wedge w_{kj})$
return $\mathbf{W}\{\mathbf{W} = [w_{ij}]$ is $\mathbf{M}_{R^*}\}$

The transitive closure can be found by Warshall's algorithm using only $O(n^3)$ bit operations.

This Lecture



Relation, *n*-ary Relations, Representing Relations, Closures of Relations, Relation Equivalence, ...

Equivalence Relation

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

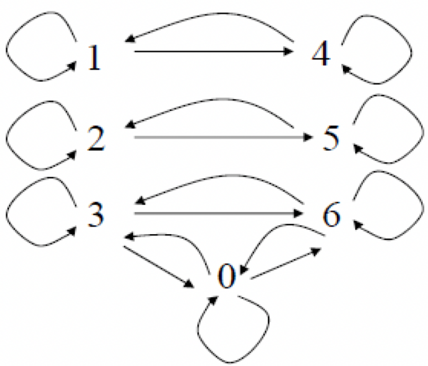
Example:
$A = \{0, 1, 2, 3, 4, 5, 6\}$ $R = \{(a, b) : a \equiv b \mod 3\}$
R has the following pairs:

- (0, 0), (0, 3), (3, 0), (0, 6), (6, 0), (3, 3), (3, 6), (6, 3), (6, 6)
- (1, 1), (1, 4), (4, 1), (4, 4)
- (2, 2), (2, 5), (5, 2), (5, 5)

Equivalence Relation

Relation R on $A = \{0, 1, 2, 3, 4, 5, 6\}$ has the pairs:

- (0, 0), (0, 3), (3, 0), (0, 6), (6, 0), (3, 3), (3, 6), (6, 3), (6, 6)
- (1, 1), (1, 4), (4, 1), (4, 4)
- (2, 2), (2, 5), (5, 2), (5, 5)



Is R reflexive? Yes

Is R symmetric? Yes

Is R transitive? Yes

R is an equivalence relation.

Examples of Equivalence Relations

- "Strings a and b have the same length."
- "Integers a and b have the same absolute value."
- "Real numbers a and b have the same fractional part (i.e., $a - b \in \mathrm{Z}$)."

Equivalence Class

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, denoted by $[a]_R$. When only one relation is considered, we use the notation $[a]$.

$$[a]_R = \{b : (a, b) \in R\}$$

Example: $A = \{0, 1, 2, 3, 4, 5, 6\}$
$R = \{(a, b) : a \equiv b \mod 3\}$

$[0] = [3] = [6] = \{0, 3, 6\}$
$[1] = [4] = \{1, 4\}$
$[2] = [5] = \{2, 5\}$

Examples of Equivalence Classes

"Strings a and b have the same length."
$[a] =$ the set of all strings of the same length as a.

"Integers a and b have the same absolute value."
$[a] =$ the set $\{a, -a\}$

"Real numbers a and b have the same fractional part (i.e., $a - b \in Z$)."
$[a] =$ the set $\{..., a-2, a-1, a, a+1, a+2, ...\}$

Equivalence Class

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

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

Proof:

- (i) $\rightarrow$ (ii): prove $[a] \subseteq [b]$ and $[b] \subseteq [a]$
  Suppose $c \in [a]$. Then, aRc.
  Because aRb and R is symmetric, we know that bRa.
  Since R is transitive and bRa and aRc, it follows that bRc.
  Hence, $c \in [b]$. This shows that $[a] \subseteq [b]$.

Equivalence Class

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

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

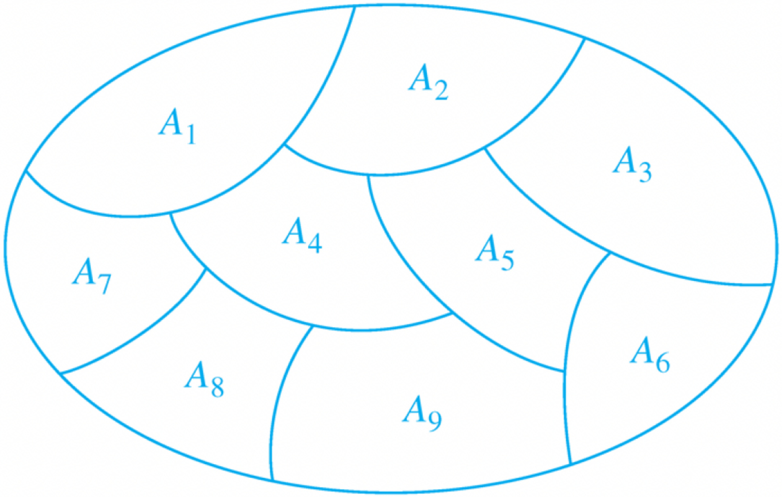
Proof:

- (i) $\rightarrow$ (ii): prove $[a] \subseteq [b]$ and $[b] \subseteq [a]$
- (ii) $\rightarrow$ (iii): Assume that $[a] = [b]$. It follows that $[a] \cap [b] \neq \emptyset$ because $[a]$ is nonempty (because $a \in [a]$ as R is reflexive).
- (iii) $\rightarrow$ (i): Suppose that$[a] \cap [b] \neq \emptyset$. There exists a c such that $c \in [a]$ and $c \in [b]$, i.e., aRc and bRc.
  By the symmetric property, cRb.
  Then by transitivity, because aRc and cRb, we have aRb.

Partition of a Set S

Definition: Let S be a set. A collection of nonempty subsets of S, i.e A_1, A_2, . . . , A_k, is called a partition of S if:

$$A_i \cap A_j = \emptyset, i \neq j \text{ and } S = \bigcup_{i=1}^{k} A_i$$



Partition of a Set S

Definition: Let S be a set. A collection of nonempty subsets of S, i.e A_1, A_2, . . . , A_k, is called a partition of S if:

$$A_i \cap A_j = \emptyset, i \neq j \text{ and } S = \bigcup_{i=1}^{k} A_i$$

Example: $A = \{0, 1, 2, 3, 4, 5, 6\}$
$A_1 = \{0, 3, 6\}$, $A_2 = \{1, 4\}$, $A_3 = \{2, 5\}$
Is A_1, A_2, A_3 a partition of S?

Equivalence Classes and Partitions

Theorem: Let R be an equivalence relation on a set A. Then, union of all the equivalence classes of R is A:

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

Theorem: The equivalence classes form a partition of A.

Theorem: Let $\{A_1, A_2, ..., A_i, ...\}$ be a partition of S. Then, there is an equivalence relation R on S, that has the sets A_i as its equivalence classes.

Equivalence Classes and Partitions: Example

List the ordered pairs in the equivalence relation R produced by the partition $A_1 = \{1, 2, 3\}$, $A_2 = \{4, 5\}$, and $A_3 = \{6\}$ of $S = \{1, 2, 3, 4, 5, 6\}$.

Solution: The subsets in the partition are the equivalence classes of R. The pair $(a, b) \in R$ if and only if a and b are in the same subset of the partition:

- (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;
- (4, 4), (4, 5), (5, 4), and (5, 5) belong to R because $A_2 = \{4, 5\}$ is an equivalence class;
- (6, 6) belongs to R because 6 is an equivalence class.

Equivalence Classes and Partitions: Example

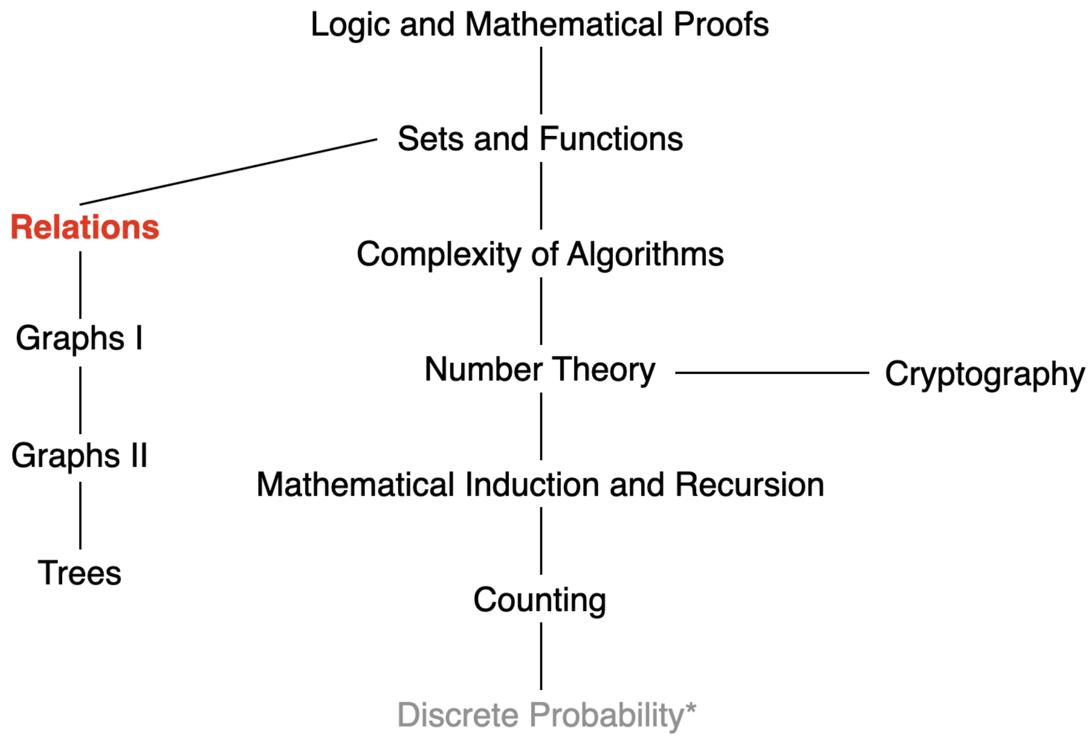
What are the sets in the partition of the integers arising from congruence modulo 4?

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

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

These congruence classes are disjoint, and every integer is in exactly one of them.

This Lecture



Relation, *n*-ary Relations, Representing Relations, Closures of Relations, Relation Equivalence, Partial Ordering, ...

Partial Ordering

Definition: A relation R on a set S is called a partial ordering, or partial order, if it is reflexive, antisymmetric, and transitive.

A set S together with a partial ordering R is called a partially ordered set, or poset, denoted by (S, R). Members of S are called elements of the poset.

Example: $S = \{1, 2, 3, 4, 5\}$, R denotes the "$\geq$" relation:

- Is R reflexive? Yes
- Is R antisymmetric? Yes
- Is R transitive? Yes

R is a partial ordering

Partial Ordering: Example

$S = \{1, 2, 3, 4, 5, 6\}$, R denotes the "$|$" relation

- Is R reflexive? Yes
- Is R antisymmetric? Yes
- Is R transitive? Yes

R is a partial ordering

Comparability

The notation $a \preccurlyeq b$ is used to denote that $(a, b) \in R$ in an arbitrary poset (S, R).

The notation $a \prec b$ denotes that $a \preccurlyeq b$, but $a \neq b$.

Definition: The elements a and b of a poset $(S, \preccurlyeq)$ are comparable if either $a \preccurlyeq b$ or $b \preccurlyeq a$. Otherwise, a and b are called incomparable.

Example: $S = \{1, 2, 3, 4, 5, 6\}$, R denotes the "$|$" relation.
$2, 4$ are comparable, $3, 5$ are incomparable.

Total Ordering

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.

Example: $S = \{1, 2, 3, 4, 5, 6\}$, R denotes the "$\geq$" relation S is a chain.

Well-Ordered Set

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

Example: The set of ordered pairs of positive integers, $\mathrm{Z}^+ \times \mathrm{Z}^+$, 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 $\leq$ ordering, is not well-ordered because the set of negative integers, which is a subset of Z, has no least element.

The Principle of Well-Ordered Induction

The Principle of Well-Ordered Induction: Suppose that $(S, \preccurlyeq)$ is a well-ordered set. Then $P(x)$ is true for all $x \in S$, if

Inductive Step: For every $y \in S$, if $P(x)$ is true for all $x \in S$ with $x \prec y$, then $P(y)$ is true.

Note: Suppose x_0 is the least element of a well ordered set, the inductive step tells us that $P(x_0)$ is true. We do not need a basis step.

The Principle of Well-Ordered Induction

The Principle of Well-Ordered Induction: Suppose that $(S, \preccurlyeq)$ is a well-ordered set. Then $P(x)$ is true for all $x \in S$, if

Inductive Step: For every $y \in S$, if $P(x)$ is true for all $x \in S$ with $x \prec y$, then $P(y)$ is true.

Proof: Suppose it is not the case that $P(x)$ is true for all $x \in S$. Then there is an element $y \in S$ such that $P(y)$ is false.

Consequently, the set $A = \{x \in S | P(x) \text{is false}\}$ is nonempty. Because S is well ordered, A has a least element a.

By the choice of a as a least element of A, we know that $P(x)$ is true for all $x \in S$ with $x \prec a$. By the inductive step, $P(a)$ is true.

This contradiction shows that $P(x)$ must be true for all $x \in S$.

Questions from Section 5 (Induction)

The Well-Ordering Property: Every nonempty set of nonnegative integers has a least element.

The principle of mathematical induction follows from the well-ordering property.

Question from students: Consider the set of negative integers. Although it does not has a least element, it has a greatest element. Can we solve it using mathematical induction?

Yes. We can solve it using the principle of well-ordered induction if we can find a relation $\preccurlyeq$ such that $(S, \preccurlyeq)$ is a well-ordered set.

Questions from Section 5 (Induction)

(i) The principle of mathematical induction, (ii) strong induction, and (iii) well-ordering property are all equivalent principles.

That is, the validity of each can be proved from either of the other two. (See Section 5.2 Exercise 41, 42, 43)

- (i) $\rightarrow$ (ii): The inductive hypothesis of a proof by mathematical induction is part of the inductive hypothesis in a proof by strong induction.
- (ii) $\rightarrow$ (iii) Use strong induction to show that the set of nonnegative integers has a least element.
- (iii) $\rightarrow$ (i) The principle of mathematical induction follows from the well-ordering property.

Lexicographic Ordering

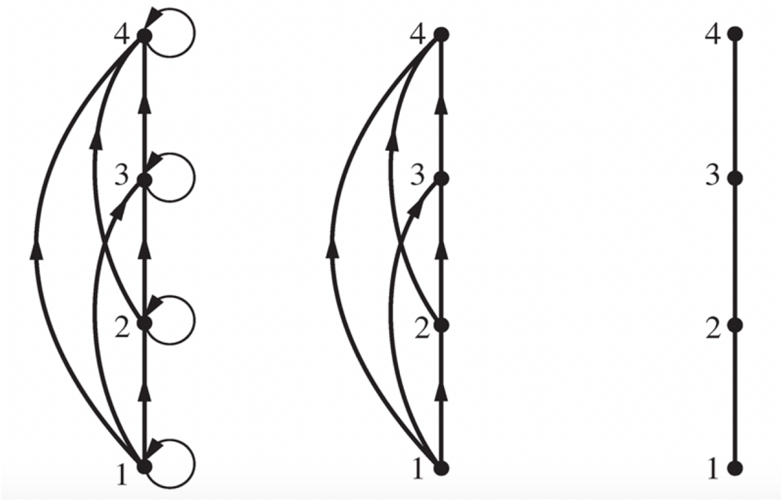
Definition: Given two posets $(A_1, \preccurlyeq_1)$ and $(A_2, \preccurlyeq_2)$, the lexicographic ordering on $A_1 \times A_2$ is defined by specifying that (a_1, a_2) is less than (b_1, b_2), i.e., $(a_1, a_2) \preccurlyeq (b_1, b_2)$, either if $a_1 \prec_1 b_1$ or if $a_1 = b_1$ then $a_2 \preccurlyeq_2 b_2$.

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
- discreet $\prec$ discreetness

Hasse Diagram

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.

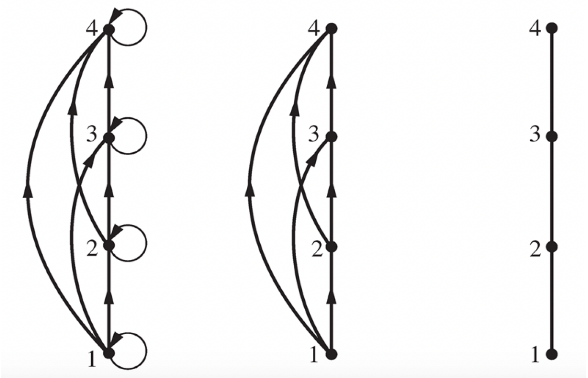


- A partial ordering. The loops are due to the reflexive property.
- The edges that must be present due to the transitive property are deleted.
- The Hasse diagram for the partial ordering (a).

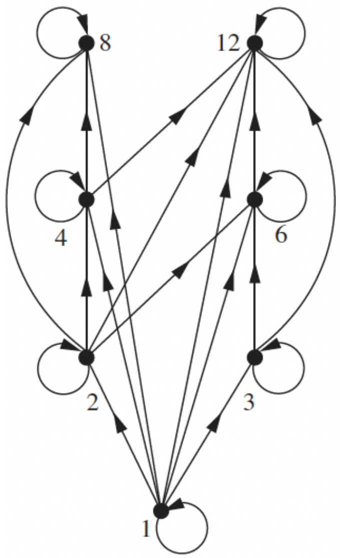
Procedure for Constructing Hasse Diagram

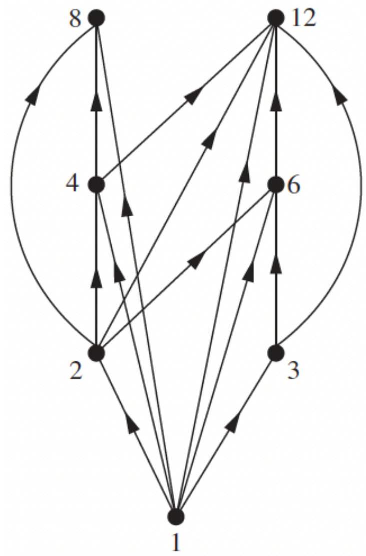
Start with the directed graph of the relation:

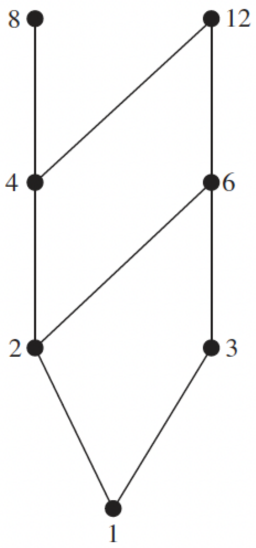
- Remove the loops (a, a) present at every vertex due to the reflexive property.
- Remove all edges (x, y) for which there is an element $z \in S$ s.t. $x \prec z$ and $z \prec y$. These are the edges that must be present due to the transitive property.
- Arrange each edge so that its initial vertex is below the terminal vertex. Remove all the arrows, because all edges point upwards toward their terminal vertex.



Hasse Diagram Example



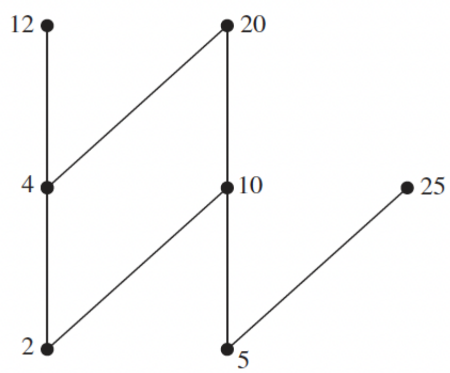




Maximal and Minimal Elements

Definition: a is a maximal (resp. minimal) element in poset $(S, \preccurlyeq)$ if there is no $b \in S$ such that $a \prec b$ (resp. $b \prec a$).

Example: Which elements of the poset $(\{2, 4, 5, 10, 12, 20, 25\}, |)$ are maximal, and which are minimal?



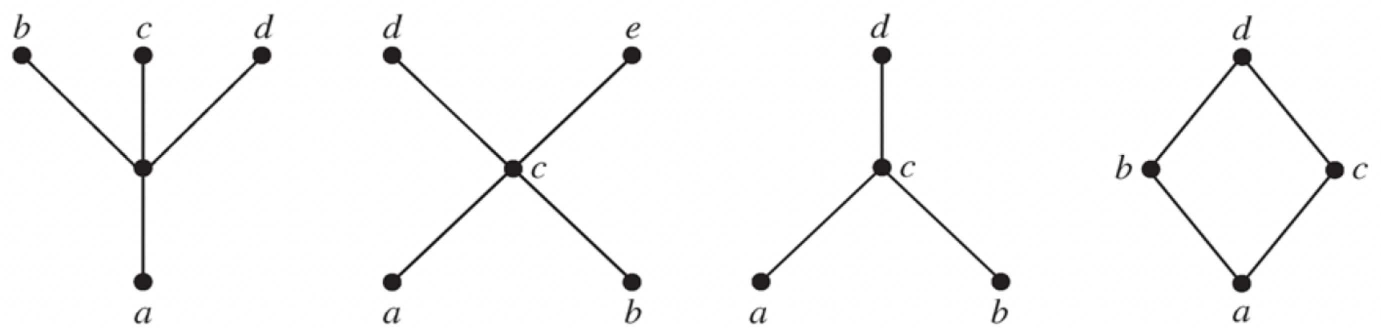
The maximal elements are 12, 20, and 25.

The minimal elements are 2 and 5.

A poset can have more than one maximal element and more than one minimal element.

Greatest and Least Elements

Definition: a is the greatest (resp. least) element of the poset $(S, \preccurlyeq)$ if $b \preccurlyeq a$ (resp. $a \preccurlyeq b$) for all $b \in S$.



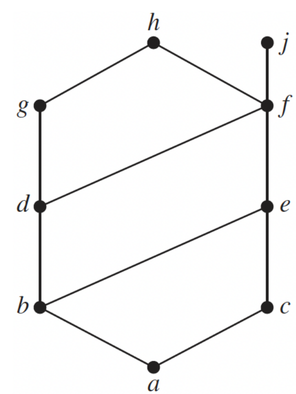
- (a): a least element a, no greatest element
- (b): neither a least nor a greatest element
- (c): no least element., a greatest element d
- (d): a least element a, a greatest element d

Upper and Lower Bound

Definition: Let A be a subset of a poset $(S, \preccurlyeq)$.

- $u \in S$ is called an upper bound (resp. lower bound) of A if $a \preccurlyeq u$ (resp. $u \preccurlyeq a$) for all $a \in A$.
- $x \in S$ is called the least upper bound (resp. greatest lower bound) of A if x is an upper bound (resp. lower bound) that is less than any other upper bounds (resp. lower bounds) of A.

Find the greatest lower bound and the least upper bound of $\{b, d, g\}$, if they exist.



g is the least upper bound, b is the greatest lower bound.

Upper and Lower Bound

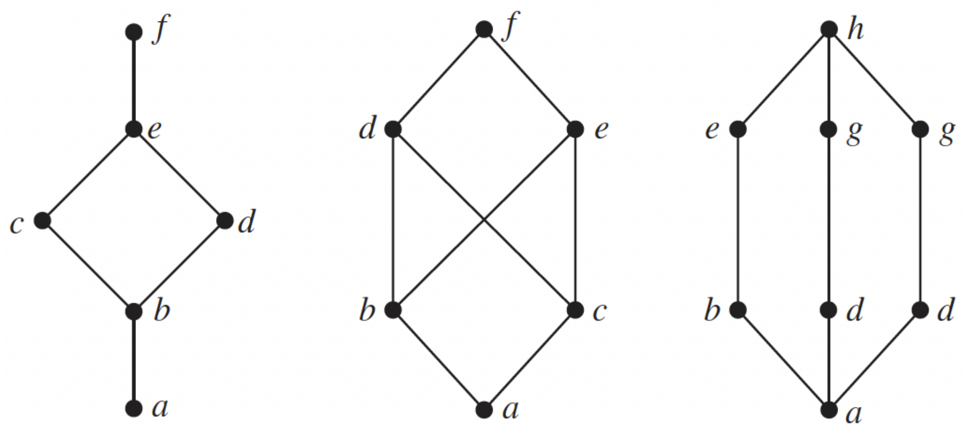
Example: Find the greatest lower bound and the least upper bound of the sets $\{3, 9, 12\}$ and $\{1, 2, 4, 5, 10\}$, if they exist, in the poset $(\mathrm{Z}^+, |)$.

- Lower bound of $\{3, 9, 12\}$: 1 and 3; the greatest lower bound: 3.
- Lower bound of $\{1, 2, 4, 5, 10\}$: 1; the greatest lower bound: 1.
- Upper bound of $\{3, 9, 12\}$: multiple of 36; the least upper bound: 36.
- Upper bound of $\{1, 2, 4, 5, 10\}$: multiple of 20; the least upper bound: 20.

SUSTech Southern University of Science and Technology

Lattices

Definition: A partial ordered set in which every pair of elements has both a least upper bound and a greatest lower bound is called a lattice.



- (a) and (c): lattices
- (b): not a lattice, because the elements b and c have no least upper bound.

Lattices: Example

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, 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
- The greatest lower bound of two elements is the smaller of the elements

Hence, this second poset is a lattice.

Topological Sorting

Motivation: 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?

Topological sorting: Given a partial ordering R, find a total ordering $\preccurlyeq$ such that $a \preccurlyeq b$ whenever aRb. $\preccurlyeq$ is said compatible with R.

Topological Sorting for Finite Posets

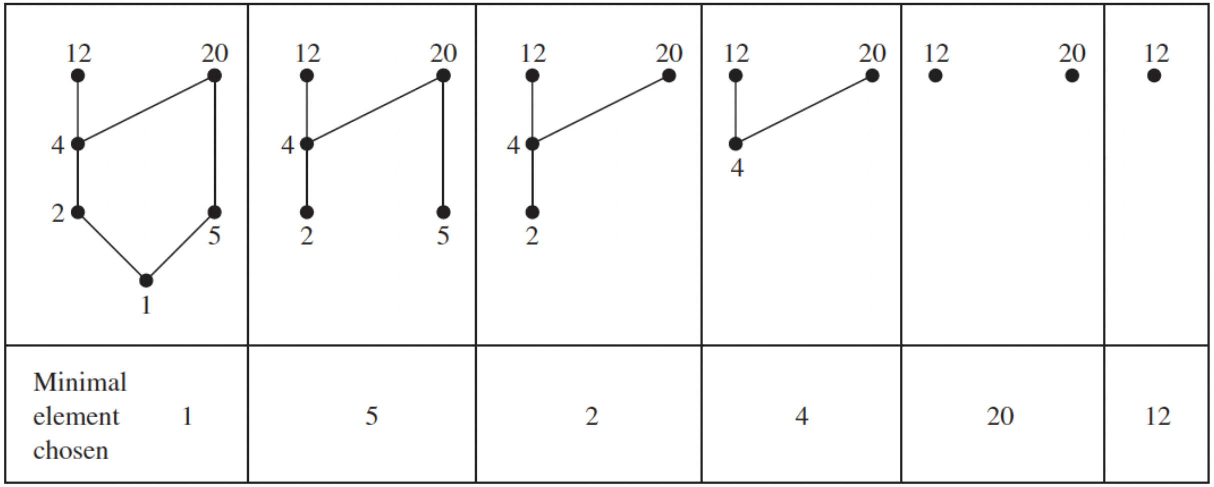
Lemma: Every finite nonempty poset $(S, \preccurlyeq)$ has at least one minimal element.

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}

Topological Sorting for Finite Posets

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



This produces the total ordering

$$1 \prec 5 \prec 2 \prec 4 \prec 20 \prec 12$$

Recall the Motivation: 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?

Next Lecture