08 Relations

CS201 Discrete Mathematics

Instructor: Shan Chen

Relations

- Relationships between elements of sets occur in many contexts.
- Every day we deal with relationships such as:
  - a business and its telephone number
  - an employee and his or her salary
  - a person and a relative
  - …

Relations and Their Properties

Binary Relations

- **Definition:** Let A, B be two sets. A binary relation R from A to B is a subset of the Cartesian product $A \times B$.
  - By definition, a binary relation $R \subseteq A \times B$ is a set of ordered pairs of the form (a, b) with $a \in A$ and $b \in B$.
  - We use $a\ R\ b$ to denote $(a, b) \in R$, and $a\ \not R\ b$ to denote $(a, b) \notin R$.

- Example: Let $A = \{a, b, c\}$ and $B = \{1, 2, 3\}$
  - Is $R = \{(a, 1), (b, 2), (c, 2)\}$ a relation from A to B?

    Yes
  - Is $Q = \{(1, a), (2, b)\}$ a relation from A to B?

    No, it's a relation from B to A
  - Is $P = \{(a, a), (b, c), (b, a)\}$ a relation from A to A?

    Yes

Binary Relations vs Functions

- Functions can also be visualized as graphs, but they map each element in the domain to exactly one element in the codomain.
- Binary relations are able to represent one-to-many relationships between elements in A and B.
- Binary relations can be viewed as generalization of functions.

Relations between Finite Sets

- **Theorem:** There are 2^{nm} binary relations from an n-element set A to an m-element set B.

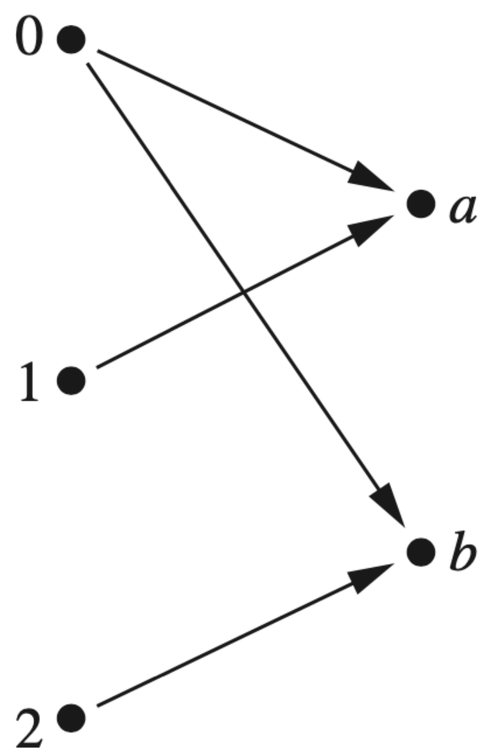
- Proof:
  - The cardinality of the Cartesian product $|A \times B| = nm$.
  - R is a binary relation from A to B if and only if $R \subseteq A \times B$.
  - The number of subsets of a set with nm elements is 2^{nm}.

- **Matrix representation:** A relation R between finite sets can be represented using a zero–one matrix M_R.

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

Representing Binary Relations

- We can visually represent a binary relation R:
  - as a directed graph: if $a\ R\ b$, then draw an arrow from a to b: $a \rightarrow b$
  - as a table (matrix): if $a\ R\ b$, then mark the table cell at (a, b)
- Example: $A = \{0, 1, 2\}$, $B = \{a, b\}$, $R = \{(0, a), (0, b), (1, a), (2, b)\}$

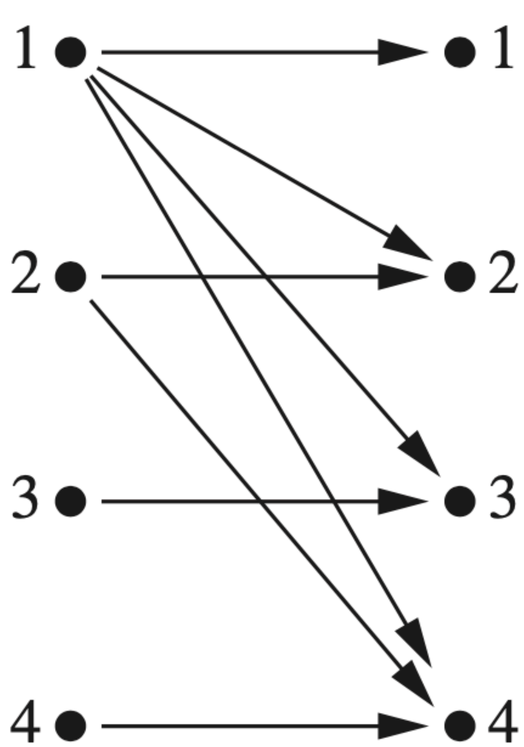


R	a	b
0	×	×
1	×	
2		×

Relations on a Set

- **Definition:** A relation on a set A is a relation from A to A.
- Example: Let $A = \{1, 2, 3, 4\}$ and $R_{div} = \{(a, b) : a \mid b\}$
  - What does R_{div} consist of?

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



R	1	2	3	4
1	×	×	×	×
2		×		×
3			×	
4				×

Properties of Relations

- **Reflexive Relation:** A relation R on a set A is called reflexive if $(a, a) \in R$ for every element $a \in A$.

- Example: Consider relations on $A = \{1, 2, 3, 4\}$
  - Is $R_{div} = \{(a, b) : a \mid b\}$ reflexive?

    Yes, because $(1, 1), (2, 2), (3, 3), (4, 4) \in R_{div}$
  - Is $R = \{(1, 2), (2, 2), (3, 3)\}$ reflexive?

    No, because $(1, 1), (4, 4) \notin R$

- A relation R is reflexive if and only if M_R has 1 in every position on its main diagonal.

$$MR_{div} = \begin{matrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Properties of Relations

- **Irreflexive Relation:** A relation R on a set A is called irreflexive if $(a, a) \notin R$ for every element $a \in A$.

- Example: Consider relations on $A = \{1, 2, 3, 4\}$
  - Is $R_{\neq} = \{(a, b) : a \neq b\}$ irreflexive?

    Yes, because $(1, 1), (2, 2), (3, 3), (4, 4) \notin R_{\neq}$
  - Is $R = \{(1, 2), (2, 2), (3, 3)\}$ irreflexive?

    No, because $(2, 2), (3, 3) \in R$ * *actually R is not reflexive either*

- A relation R is irreflexive if and only if M_R has 0 in every position on its main diagonal.

$$MR = \begin{matrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{matrix}$$

Properties of Relations

- **Symmetric Relation:** A relation R on a set A is called symmetric if $(b, a) \in R$ whenever $(a, b) \in R$ for all $a, b \in A$.
- Example: Consider relations on $A = \{1, 2, 3, 4\}$
  - Is $R_{div} = \{(a, b) : a \mid b\}$ symmetric?

    No, because $(1, 2) \in R_{div}$ but $(2, 1) \notin R_{div}$
  - Is $R_{\neq} = \{(a, b) : a \neq b\}$ symmetric?

    Yes, because if $(a, b) \in R_{\neq}$ then $(b, a) \in R_{\neq}$
- A relation R is symmetric if and only if M_R is symmetric.

$$MR = \begin{matrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{matrix}$$

Properties of Relations

- **Antisymmetric Relation:** A relation R on a set A is called antisymmetric if $(b, a) \in R$, $(a, b) \in R$ implies $a = b$ for all $a, b \in A$.

- Example: Consider relations on $A = \{1, 2, 3, 4\}$
  - Is $R = \{(1, 2), (2, 2), (2, 1), (3, 3)\}$ antisymmetric?

    No, because both $(1, 2) \in R$ and $(2, 1) \in R$ but $1 \neq 2$
  - Is $R = \{(2, 2), (3, 3)\}$ antisymmetric?

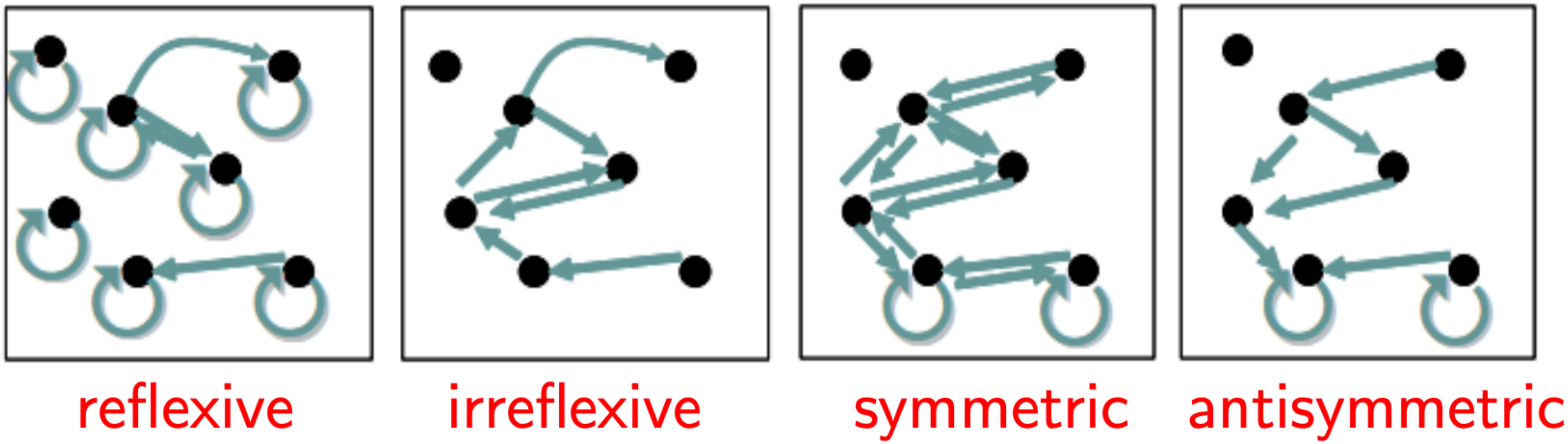
    Yes ** actually R is also symmetric*

- A relation R is antisymmetric if and only if $m_{ij} = 1$ implies $m_{ji} = 0$ for $i \neq j$, where m_{ij} is the (i, j)-*th* element of M_R.

$$MR_{div} = \begin{matrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Relation Properties in Digraphs

- Recall that a relation can be represented as a directed graph:



Exercise *(5 mins)*

- Consider binary relations on a finite set A with $|A| = n$:
  Hint: think of a binary relation as a zero-one matrix
  - How many reflexive relations?
  - How many irreflexive relations?
  - How many symmetric relations?
  - How many antisymmetric relations?

- **Theorem:** There are 2^{nm} binary relations from an n-element set A to an m-element set B.
- Proof:
  - The cardinality of the Cartesian product $|A \times B| = nm$.
  - R is a binary relation from A to B if and only if $R \subseteq A \times B$.
  - The number of subsets of a set with nm elements is 2^{nm}.

Exercise (5 mins)

- Consider binary relations on a finite set A with $|A| = n$:
  Hint: think of a binary relation as a zero-one matrix
  - How many reflexive relations?
    $2^{n(n-1)}$
  - How many irreflexive relations?
    $2^{n(n-1)}$
  - How many symmetric relations?
    $2^{n(n+1)/2}$
  - How many antisymmetric relations?
    $2^n 3^{n(n-1)/2}$
    ** First, values on the main diagonal m_{ii} can be chosen arbitrarily. Then, for each pair of matrix elements (m_{ij}, m_{ji}) with $i \neq j$ (there are $n(n-1)/2$ such pairs), it has 3 possible choices: (0, 0), (0, 1), (1, 0).*

Transitive Relation

- **Transitive Relation:** A relation R on a set A is called transitive if $(a, b) \in R, (b, c) \in R$ implies $(a, c) \in R$ for all $a, b, c \in A$.

- Example: Consider relations on $A = \{1, 2, 3, 4\}$
  - Is $R_{div} = \{(a, b) : a \mid b\}$ transitive?

    Yes, because if $a \mid b$ and $b \mid c$ then $a \mid c$
  - Is $R_{\neq} = \{(a, b) : a \neq b\}$ transitive?

    No, because $(1, 2), (2, 1) \in R_{\neq}$ but $(1, 1) \notin R_{\neq}$
  - Is $R = \{(1, 2), (2, 2), (3, 3)\}$ transitive?

    Yes

SUSTech

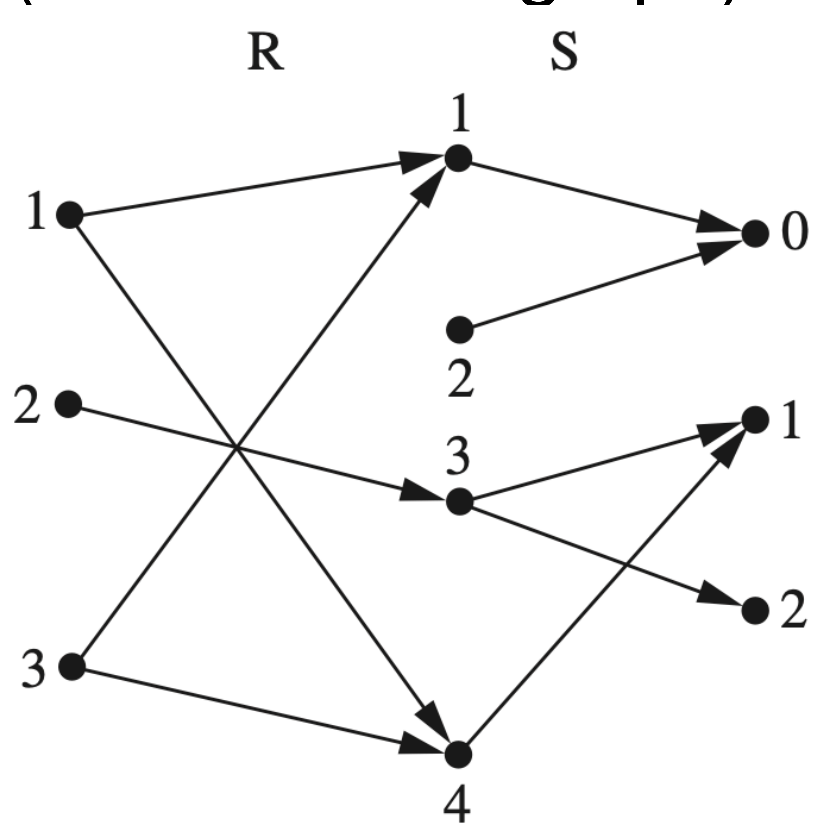
Combining Relations

- Since relations are sets, we can combine relations via set operations: union, intersection, complement, difference, etc.
  - typically both relations are defined on the same sets
- Example: consider relations from $A = \{1, 2, 3\}$ to $B = \{u, v\}$
  - $R_1 = \{(1, u), (2, u), (2, v), (3, u)\}$, $R_2 = \{(1, v), (3, u), (3, v)\}$
  - What are $R_1 \cup R_2$, $R_1 \cap R_2$, $R_1 - R_2$, $R_2 - R_1$?
- We can also combine relations by matrix operations.
  - E.g., can get $R_1 \cap R_2$ from **element-wise "and"**: $M_{R1} \wedge M_{R2}$
    ** what about other set operations?*

SUSTech

Composite of Relations

- **Definition:** Let *R* be a relation from a set *A* to a set *B* and *S* be a relation from *B* to *C*. The **composite** of *R* and *S* is the relation consisting of the ordered pairs *(a, c)* where $a \in A$ and $c \in C$ and for which there exists a $b \in B$ such that $(a, b) \in R$ and $(b, c) \in S$.
  - We denote the composite of *R* and *S* by $S \circ R$.
- Example 1: (as a directed graph)



$1 \to 1 \to 0$	(1, 0)
$1 \to 4 \to 1$	(1, 1)
$2 \to 3 \to 1$	(2, 1)
$2 \to 3 \to 2$	(2, 2)
$3 \to 1 \to 0$	(3, 0)
$3 \to 4 \to 1$	(3, 1)

Composite of Relations

- **Definition:** Let R be a relation from a set A to a set B and S be a relation from B to C. The **composite** of R and S is the relation consisting of the ordered pairs (a, c) where $a \in A$ and $c \in C$ and for which there exists a $b \in B$ such that $(a, b) \in R$ and $(b, c) \in S$.
  - We denote the composite of R and S by $S \circ R$.

- Example 2: (as a matrix)
  - $A = \{1, 2\}, B = \{1, 2, 3\}, C = \{a, b\}$
  - $R = \{(1, 2), (1, 3), (2, 1)\} \subseteq A \times B,\ \ S = \{(1, a), (3, a), (3, b)\} \subseteq B \times C$
  - $S \circ R = \{(1, a), (1, b), (2, a)\}$

$$M_R = \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{matrix} \qquad M_S = \begin{matrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{matrix} \qquad \boxed{M_R \odot M_S = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}}$$

Boolean product $\odot$ of matrices: replace $+$ with $\vee$ and $\times$ with $\wedge$

Composing a Relation with Itself

- **Definition:** Let R be a relation on a set A. The powers R^n for $n = 1, 2, 3, \ldots$ is defined inductively by $R^1 = R$ and $R^{n+1} = R^n \circ R$.
- Example: Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 2), (2, 3), (2, 4), (3, 3)\}$
  - $R^1 = R$
  - $R^2 = R \circ R = \{(1, 3), (1, 4), (2, 3), (3, 3)\}$
  - $R^3 = R^2 \circ R = \{(1, 3), (2, 3), (3, 3)\}$
  - $R^4 = R^3 \circ R = \{(1, 3), (2, 3), (3, 3)\}$
  - $R^k = ?\ (k > 4)$

Transitive Relation and R^n

- **Theorem:** The relation R on a set A is transitive if and only if

$$R^n \subseteq R \text{ for } n = 1, 2, 3, \ldots$$

- Proof:
  - **"if" part:** For $n = 2$, we have $R^2 \subseteq R$. If $(a, b) \in R$ and $(b, c) \in R$, then by the definition of composition, we have $(a, c) \in R^2 \subseteq R$.
  - **"only if" part:** Proof by induction. * *the proof is left as an exercise*

- Note that R^n can be computed by Boolean product of matrices:

$$M_{R^n} = M_R \odot M_R \odot \cdots \odot M_R$$

SUSTech

n-ary Relations and Databases

n-ary Relations

- **Definition:** An n-ary relation R on sets $A_1, A_2, \ldots, A_n$, written as $R : A_1, \ldots, A_n$, is a subset of $A_1 \times \cdots \times A_n$.
  - The sets A_i s are called the domains of R.
  - The degree of R is n.
  - R is functional in domain A_i if for any $a_i \in A_i$ the relation R contains at most one n-tuple of the form $(\cdots, a_i, \cdots)$. * *R can be indexed by a_i*
- Some ways to represent n-ary relations:
  - as an explicit list or table of its tuples
  - as a function from the domains to $\{T, F\}$

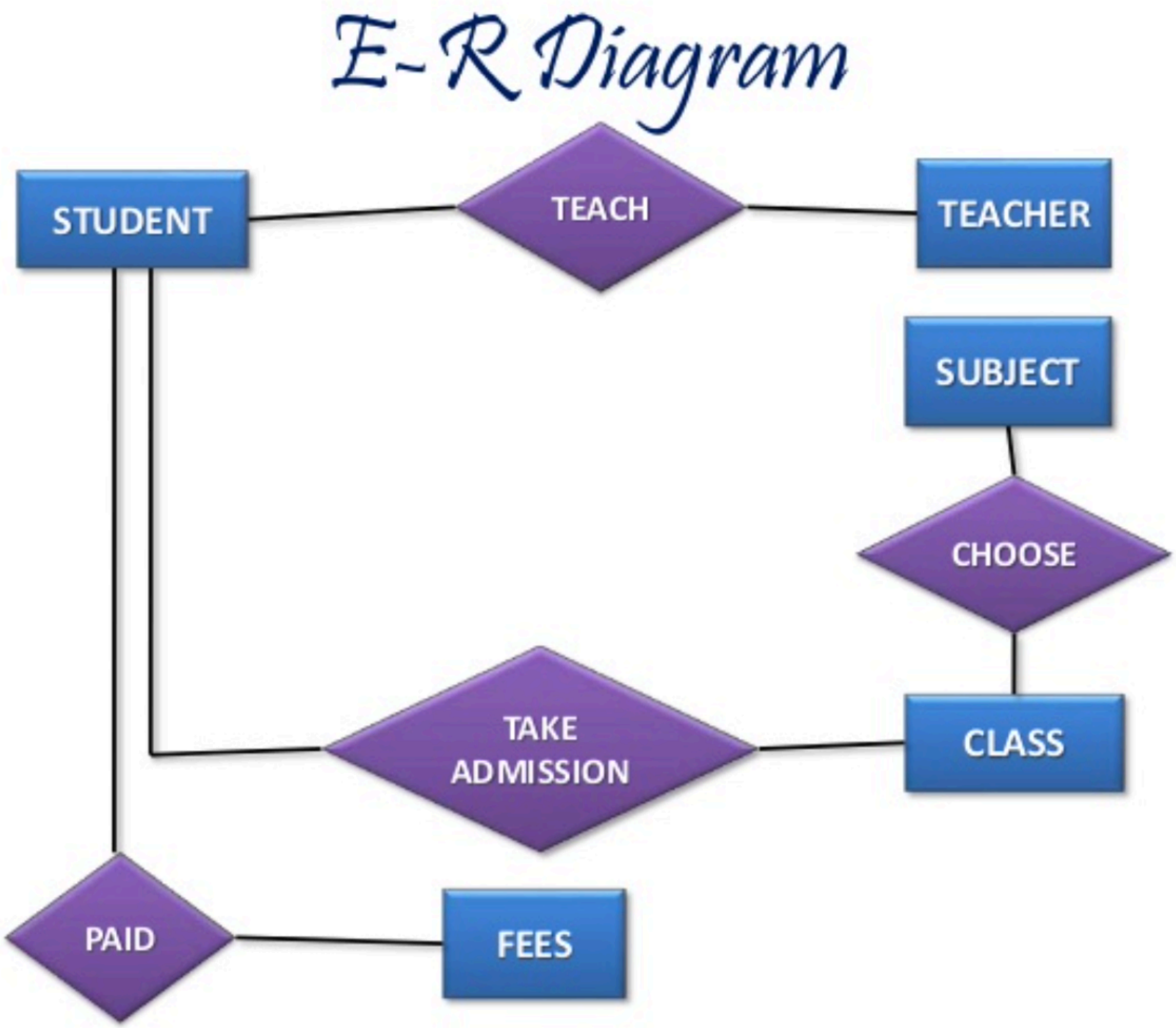
Relational Databases

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

- A domain A_i is a primary key for the database if the relation R is functional in A_i. * *because R can be indexed by a_i*

- A composite key for the relational database is a set of domains $\{\cdots, A_i, \cdots, A_j, \cdots\}$ such that R contains at most one n-tuple of the form $(\cdots, a_i, \cdots, a_j, \cdots)$ for each composite value $(a_i, a_j) \in A_i \times A_j$.
  * *R can be indexed by (a_i, a_j)*

Entity-Relationship (ER) Diagrams



Selection Operators

- Let $\boldsymbol{A}$ be an n-ary domain $\boldsymbol{A} = A_1 \times \cdots \times A_n$, and let $C : \boldsymbol{A} \to \{T, F\}$ be any condition (predicate) on elements (n-tuples) of $\boldsymbol{A}$.

- The **selection operator** s_C is the operator that maps any n-ary relation R on $\boldsymbol{A}$ to the n-ary relation consisting of all n-tuples from R that satisfy C. * *note that $s_C(\boldsymbol{A})$ is also an n-ary relation*

$$\forall R \subseteq \boldsymbol{A},\ s_C(R) = R \cap \{a \in \boldsymbol{A} \mid C(a) = T\} = \{a \in R \mid C(a) = T\}$$

- Example: Consider $\boldsymbol{A} = StudentName \times Standing \times SIDs$
  - Condition *UpperLevel(name, standing, sid)* is defined as

    $(standing = junior) \lor (standing = senior)$

  - Then, $s_{UpperLevel}$ is the selection operator that takes any relation R on $\boldsymbol{A}$ (database of all students) and produces a relation $R' \subseteq R$ consisting of just the upper-level students (juniors or seniors).

Projection Operators

- Let $\boldsymbol{A}$ be an n-ary domain $\boldsymbol{A} = A_1 \times \cdots \times A_n$, and let $\{i_1, \ldots, i_m\}$ be a sequence of indices such that $1 \leq i_1 < \cdots < i_m \leq n$ and $m < n$.
- The **projection operator** $P_{\{i_1, \ldots, i_m\}} : \boldsymbol{A} \to A_{i_1} \times \cdots \times A_{i_m}$ is the operator that maps any n-ary relation R on $\boldsymbol{A}$ to the m-ary relation consisting of m-tuples specified by indices $\{i_1, \ldots, i_m\}$:

$$P_{\{i_1, \ldots, i_m\}}(a_1, \ldots, a_n) = (a_{i_1}, \ldots, a_{i_m})$$

- Example: Consider $Cars = Model \times Year \times Color$
  - Index sequence is $\{1, 3\}$.
  - The projection operator $P_{\{1, 3\}}$ simply maps each 3-tuple, e.g., $(a_1, a_2, a_3) = (Tesla, 2020, black)$ to $(a_1, a_3) = (Tesla, black)$.
  - This operator can be applied to any relation $R \subseteq Cars$ to obtain a list of model-color combinations available.

Join Operators

- The **join operator** puts two relations together to form a sort of combined relation.
  - If the tuple (a, b) appears in $R_1 \subseteq A \times B$, and the tuple (b, c) appears in $R_2 \subseteq B \times C$, then the tuple (a, b, c) appears in their **join** $J(R_1, R_2)$.
  - In general, A, B, C can each be the Cartesian product of multiple domains rather than a single domain.
- Example:
  - Let relation R_1 be a teaching assignment table, relating Professors to Courses.
  - Let relation R_2 be a classroom assignment table, relating Courses to Rooms and Times.
  - Then, the relation $J(R_1, R_2)$ is like your class schedule, listing tuples of the form (professor, course, room, time).

Closures of Relations

Closures of Relations

- Properties of Relations:
  - reflexive
  - irreflexive
  - symmetric
  - antisymmetric
  - transitive
- Closures of Relations: (as a superset of the considered relation)
  - reflexive closures
  - symmetric closures
  - transitive closures

Example: Reflexive Closures

- Consider $R = \{(1, 1), (1, 2), (2, 1), (3, 2)\}$ defined on $A = \{1, 2, 3\}$.
- **Q:** Is relation R reflexive?
  - ***No***, $(2, 2)$ and $(3, 3)$ are not in R
- What is the minimal relation $S \supseteq R$ that is reflexive?
  - How to make R reflexive by adding minimum number of pairs?
  - Add $(2, 2)$ and $(3, 3)$: $S = \{(1, 1), (1, 2), (2, 1), (3, 2), (2, 2), (3, 3)\} \supseteq R$ is reflexive.
- The minimal set $S \supseteq R$ is called the **reflexive closure** of R.
  ** what about **irreflexive closure**? does it make sense?*

Definition of Closures

- **Definition:** Let R be a relation on a set A. A relation S on A with property P is called the closure of R with respect to P if S is the minimal set containing R satisfying the property P.
  - Minimal S: for every relation $Q \supseteq R$ that satisfies P, we have $S \subseteq Q$.

- Examples:
  - **reflexive closure** ** just showed in the previous slide*
  - **symmetric closure:** relation $R = \{(1, 2), (1, 3), (2, 2)\}$ on $A = \{1, 2, 3\}$

    ** how to make it symmetric?*

    $S = \{(1, 2), (1, 3), (2, 2)\} \cup \{(2, 1), (3, 1)\}$
  - **transitive closure:** relation $R = \{(1, 2), (2, 2), (2, 3)\}$ on $A = \{1, 2, 3\}$

    ** how to make it transitive?*

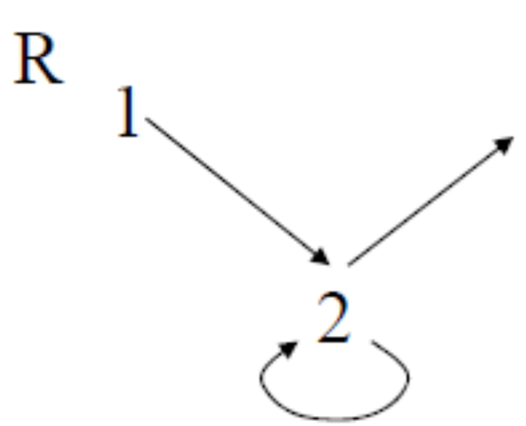
    $S = \{(1, 2), (2, 2), (2, 3)\} \cup \{(1, 3)\}$

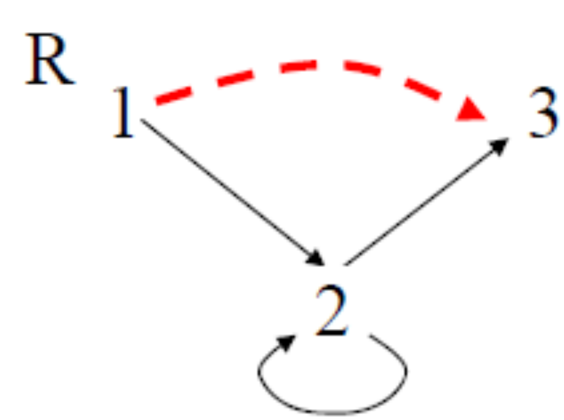
Paths and Transitive Closures

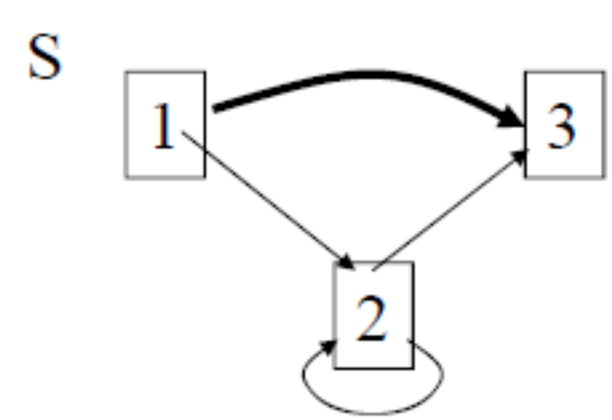
- **Definition:** A (directed) **path** from a to b in a directed graph G is a sequence of edges $(x_0, x_1), (x_1, x_2), \dots, (x_{n-1}, x_n)$ in graph G, where $n \geq 0$, $x_0 = a$ and $x_n = b$.

- Recall that we can represent a relation using a directed graph. Then, finding a **transitive closure** corresponds to finding all pairs of elements that are connected with a directed path.

- Example: Relation $R = \{(1, 2), (2, 2), (2, 3)\}$ on $A = \{1, 2, 3\}$
  - The **transitive closure** of R is $S = \{(1, 2), (2, 2), (2, 3), (1, 3)\}$





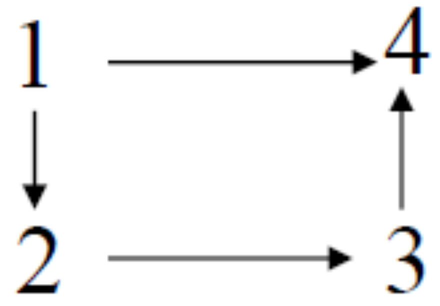


The Connectivity Relation R^*

- **Definition:** R is 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) from a to b in R.

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

- Example: Relation R on $A = \{1, 2, 3, 4\}$ shown in the figure below
  - $R = \{(1, 2), (1, 4), (2, 3), (3, 4)\}$
  - $R^2 = \{(1, 3), (2, 4)\}$
  - $R^3 = \{(1, 4)\}$
  - $R^4 = \emptyset$
  - $R^* = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$



R^* = Transitive Closure of R

- **Theorem:** The transitive closure of a relation R equals the connectivity relation R^*.

- Proof:
  - R^* is transitive. * *view $(a, b) \in R^*$ as pairs connected by a path in R*
  - $R^* \subseteq S$ whenever S is a transitive relation containing R.

    Since S is a transitive relation, we have $S^n \subseteq S$. * *proved before*

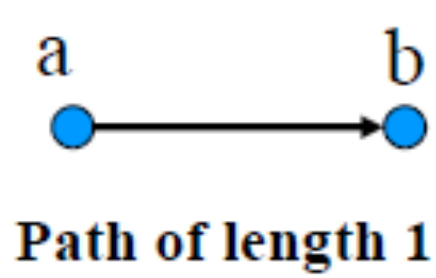
    Therefore, $S^* \subseteq S$. * *think about the infinite union definition of S^**

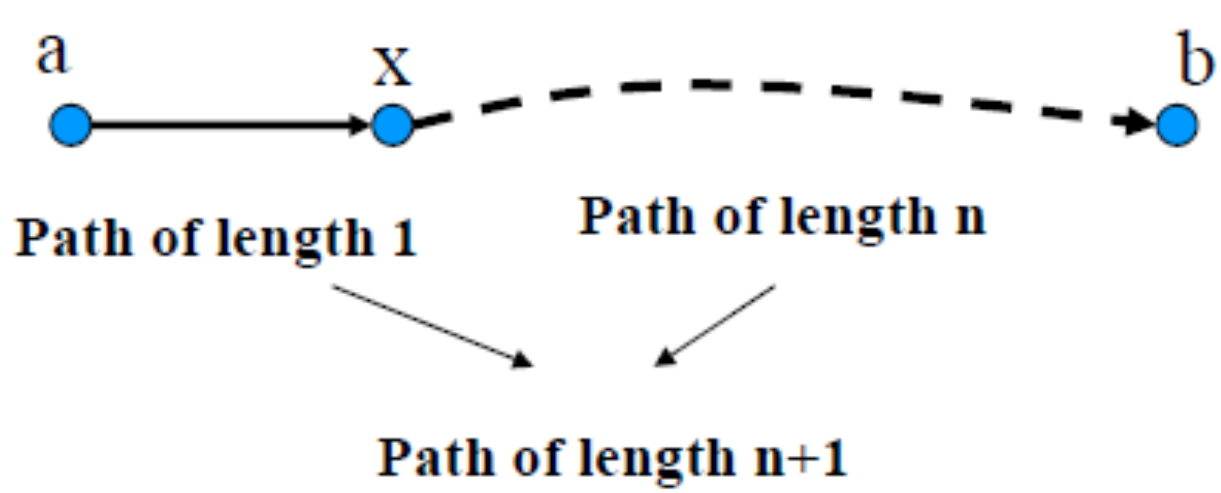
    Since $R \subseteq S$, we have $R^* \subseteq S^*$. * *any path in R is also a path in S*

    Together, we have $R^* \subseteq S^* \subseteq S$.

Paths and Powers of a Relation

- **Theorem:** Let R be relation on a set A. There is a path of length n from a to b in R (as a directed graph) if and only if $(a, b) \in R^n$.
- Proof by induction: (recall that R^{n+1} is defined as $R^n \circ R$)





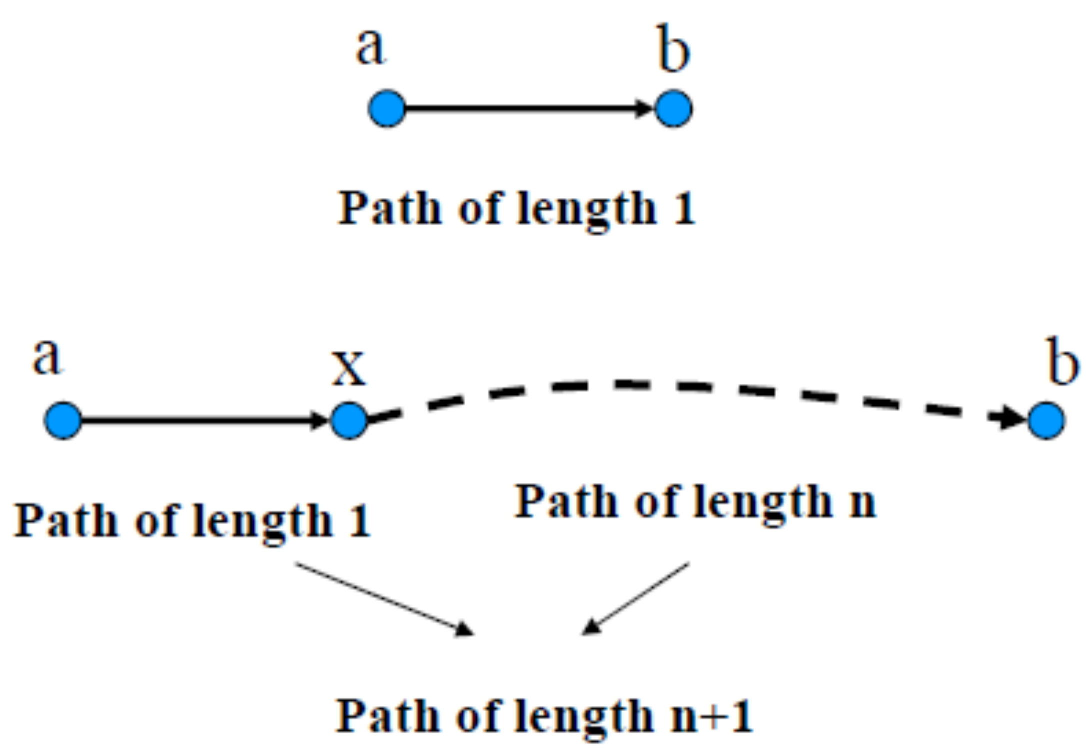
SUSTech

Exercise *(5 mins)*

- Prove that "If R is transitive, then R^n is also transitive."

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

- Proof by induction: (recall that R^{n+1} is defined as $R^n \circ R$)



Exercise (5 mins)

- Prove that "If R is transitive, then R^n is also transitive."
- Proof by induction: (Let $P(n)$ be the theorem statement.)
  - **Basis step:** $n = 1$: The statement is trivially true.
  - **Inductive step:** From the inductive hypothesis, i.e., $P(k)$ is true for an arbitrary integer $k \geq 1$, we need to show that $P(k + 1)$ is true. By the previous **Theorem**, $P(k + 1)$ means: for any path $p(a, b)$ of length $k + 1$ from a to b and any path $p(b, c)$ of length $k + 1$ from b to c, our goal is to find a path $p(a, c)$ from a to c of length $k + 1$.

    Let $p(a, b) = (a, x) + (x, y) + p(y, b)$, then $p(y, b)$ is of length $k - 1 \geq 0$.

    Since R is transitive, $(a, x), (x, y) \in R$ implies that $(a, y) \in R$.

    So, $(a, y) + p(y, b)$ is a path from a to b of length $1 + k - 1 = k$.

    Let $p(b, c) = p(b, z) + (z, c)$, then $p(b, z)$ is also of length k.

    By the inductive hypothesis, we can find a path $p(a, z)$ from a to z of length k. So, $p(a, z) + (z, c)$ is a path from a to c of length $k + 1$.

Finding Transitive Closures

- Recall that finding a transitive closure corresponds to finding the connectivity relation, which consists of all pairs of elements that are connected with a directed path.

- The following lemma shows that it is sufficient to examine paths containing no more than n edges, where n is the number of elements in the set.

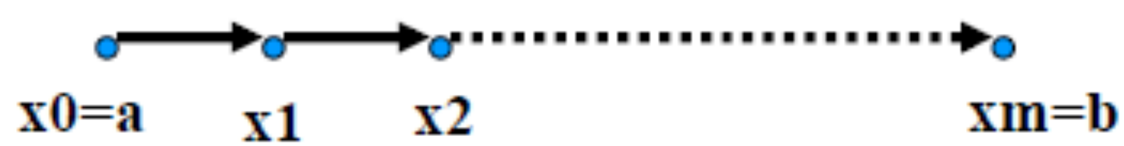
- **Lemma:** Let A be a set with n elements and R be a relation on A. If there is a path in R from a to b, then there is such a path with length $\leq n$. Moreover, when $a \neq b$, if there is a path from a to b, then there is such a path with length $\leq n - 1$. That is,

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

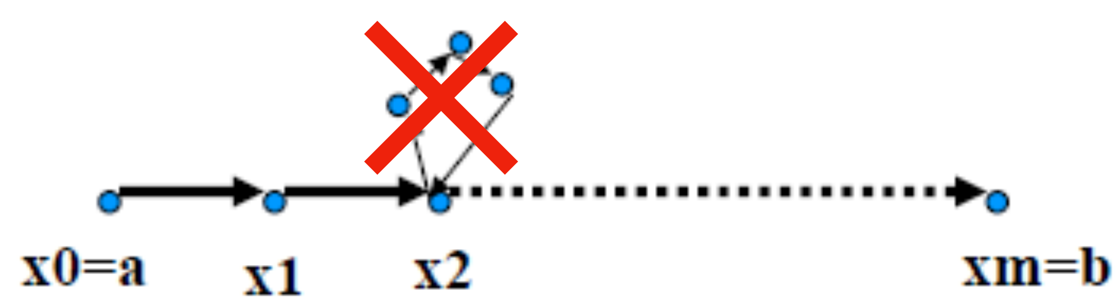
Finding Transitive Closures

- **Lemma:** Let A be a set with n elements and R be a relation on A. If there is a path in R from a to b, then there is such a path with length $\leq n$. Moreover, when $a \neq b$, if there is a path from a to b, then there is such a path with length $\leq n - 1$.

- Proof intuition:
  - The longest path is of length $n - 1$ if it does not have cycles.



  - Cycles may increase the path length but the same node will be visited more than once, so we can remove all cycles.



SUSTech

Finding Transitive Closures

- Recall that from the previous **Lemma** we have

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

- **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

$$\mathbf{M}_{R^*} = \mathbf{M}_R \vee \mathbf{M}_R^{[2]} \vee \mathbf{M}_R^{[3]} \vee \cdots \mathbf{M}_R^{[n]}$$

  - the matrix superscripts denote the powers of Boolean product of matrices, i.e., $M_R^{[n]} = M_R \odot M_R \odot \cdots \odot M_R = M_{R^n}$
  - *the proof is easy by applying the previous **Lemma***

SUSTech

Finding Transitive Closures

- **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

$$\mathbf{M}_{R^*} = \mathbf{M}_R \vee \mathbf{M}_R^{[2]} \vee \mathbf{M}_R^{[3]} \vee \cdots \mathbf{M}_R^{[n]}$$

- Example: what is the transitive closure for M_R?

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

$$\mathbf{M}_{R^*} = \mathbf{M}_R \vee \mathbf{M}_R^{[2]} \vee \mathbf{M}_R^{[3]}$$

Finding Transitive Closures

- **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

$$\mathbf{M}_{R^*} = \mathbf{M}_R \vee \mathbf{M}_R^{[2]} \vee \mathbf{M}_R^{[3]} \vee \cdots \mathbf{M}_R^{[n]}$$

- Naive algorithm that successively computes $M_R, M_R^{[2]}, \ldots, M_R^{[n]}$:

procedure transClosure ($\mathbf{M}_R$: zero-one $n \times n$ matrix)
// computes R^* with zero-one matrices
$A := B := \mathbf{M}_R$;
for $i := 2$ to n
  $A := A \odot \mathbf{M}_R$
  $B := B \vee A$
return B
// B is the zero-one matrix for R^*

This algorithm takes $\Theta(n^4)$ time.

Finding Transitive Closures

- The **Floyd-Warshall** algorithm**:**
  - By definition, R^* consists of all pairs (a, b) such that there exists a path from a to b in the graph representation of R.
  - This algorithm computes M_{R^*} by iterating on k: in the k-th iteration, the intermediate nodes of all paths are from the first k nodes only.

procedure Warshall ($\mathbf{M}_R$: zero-one $n \times n$ matrix)
// computes R^* with zero-one matrices
$W := \mathbf{M}_R$;
for $k := 1$ to n
  for $i := 1$ to n
    for $j := 1$ to n
      $\boxed{w_{ij} := w_{ij} \vee (w_{ik} \wedge w_{kj})}$
return W
// W is the zero-one matrix for R^*

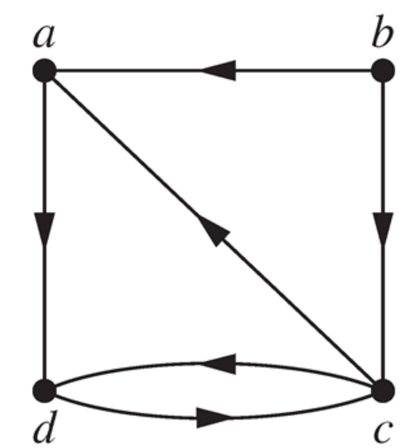
$w_{ij} = 1$ means there is a path from i to j going only through nodes $\leq k$.

$$W_{ij}^{[k]} = W_{ij}^{[k-1]} \vee \left(W_{ik}^{[k-1]} \wedge W_{kj}^{[k-1]}\right)$$

This algorithm takes $\Theta(n^3)$ time.

Exercise (5 mins)

- For the relation R shown in the figure, find the **Floyd-Warshall** matrices W_1, W_2, W_3, W_4. (Here W_k is the matrix after the k-th iteration and W_4 is the transitive closure of R.)



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

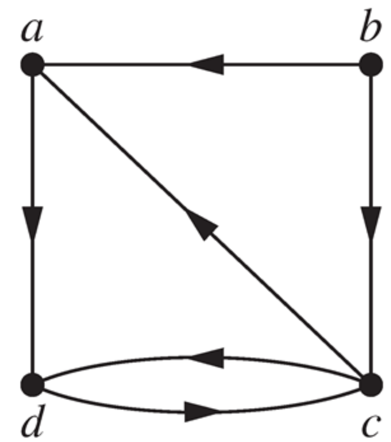
$$W_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

procedure Warshall ($\mathbf{M}_R$: zero-one $n \times n$ matrix)
// computes R^* with zero-one matrices
$W := \mathbf{M}_R$;
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 W
// W is the zero-one matrix for R^*

SUSTech

Exercise (5 mins)

- For the relation R shown in the figure, find the **Floyd-Warshall** matrices W_1, W_2, W_3, W_4. (Here W_k is the matrix after the k-th iteration and W_4 is the transitive closure of R.)



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

$$W_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad W_2 = W_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad W_4 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

Equivalence Relations

Equivalence Relations

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

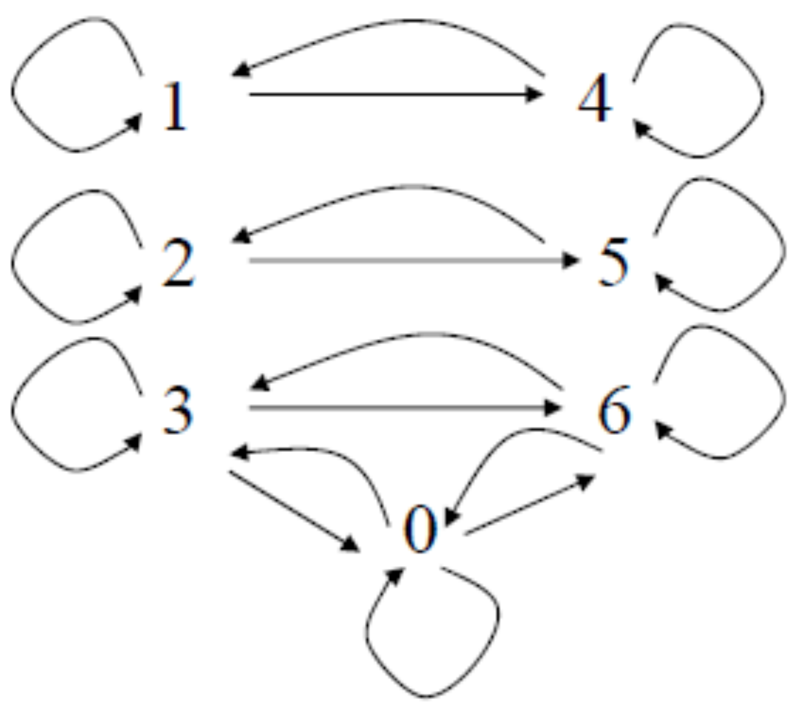
- Example: $R = \{(a, b) : a \equiv b \bmod 3\}$ on $S = \{0, 1, 2, 3, 4, 5, 6\}$
  - R has the following pairs:

    $(0, 0), (0, 3), (3, 0), (0, 6), (6, 0), (3, 3), (3, 6), (6, 3), (6, 6), (1, 1), (1, 4), (4, 1), (4, 4), (2, 2), (2, 5), (5, 2), (5, 5)$
  - Is R reflexive?

    Yes
  - Is R symmetric?

    Yes
  - Is R transitive?

    Yes
  - Therefore, R is an equivalence relation.



Equivalence Relations

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

- Are the following relations equivalence relations?
  - "Integers a and b have the same absolute value."

    Yes
  - "The relation $\geq$ between real numbers."

    No, not symmetric
  - "Real numbers a and b have the same fractional part: $a - b \in \boldsymbol{Z}$."

    Yes
  - "Natural numbers have a common factor > 1."

    No, not reflexive, e.g., $(1, 1) \notin R$

SUSTech

Equivalence Classes

- **Definition:** Let R be an equivalence relation on a set S. The set of all elements that are related to an element a of S is called the **equivalence class** of a, denoted by $[a]_R$.

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

  - When only one relation is considered, $[a]_R$ can be simplified as $[a]$.

- Example: $R = \{(a, b) : a \equiv b \bmod 3\}$ on $S = \{0, 1, 2, 3, 4, 5, 6\}$
  - $[0] = [3] = [6] = \{0, 3, 6\}$
  - $[1] = [4] = \{1, 4\}$
  - $[2] = [5] = \{2, 5\}$

Equivalence Classes

- **Definition:** Let R be an equivalence relation on a set S. The set of all elements that are related to an element a of S is called the **equivalence class** of a, denoted by $[a]_R$.

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

  - When only one relation is considered, $[a]_R$ can be simplified as $[a]$.

- Find $[a]$ for the following relations:
  - "Strings a and b have the same length."

    $[a]$ = the set of all strings of the same length as string a
  - "Integers a and b have the same absolute value."

    $[a] = \{a, -a\}$
  - "Real numbers a and b have the same fractional part: $a - b \in \mathbf{Z}$."

    $[a] = \{\ldots, a - 2, a - 1, a, a + 1, a + 2, \ldots\}$

Theorem on Equivalence Classes

- **Theorem:** Let R be an equivalence relation on a set S. The following statements are equivalent:
  - (i) $(a, b) \in R$
  - (ii) $[a] = [b]$
  - (iii) $[a] \cap [b] \neq \varnothing$
- Proof:
  - (i) → (ii): prove $[a] \subseteq [b]$ and $[b] \subseteq [a]$
  - (ii) → (iii): $[a]$ is not empty (R is reflexive and hence $a \in [a]$)
  - (iii)→(i): there exists $c \in S$ such that $c \in [a]$ and $c \in [b]$

Theorem on Equivalence Classes

- **Theorem:** Let R be an equivalence relation on a set S. The following statements are equivalent:
  - (i) $(a, b) \in R$
  - (ii) $[a] = [b]$
  - (iii) $[a] \cap [b] \neq \varnothing$

- **Corollary:** Let R be an equivalence relation on a set S. Then the union of all the equivalence classes of R is S:

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

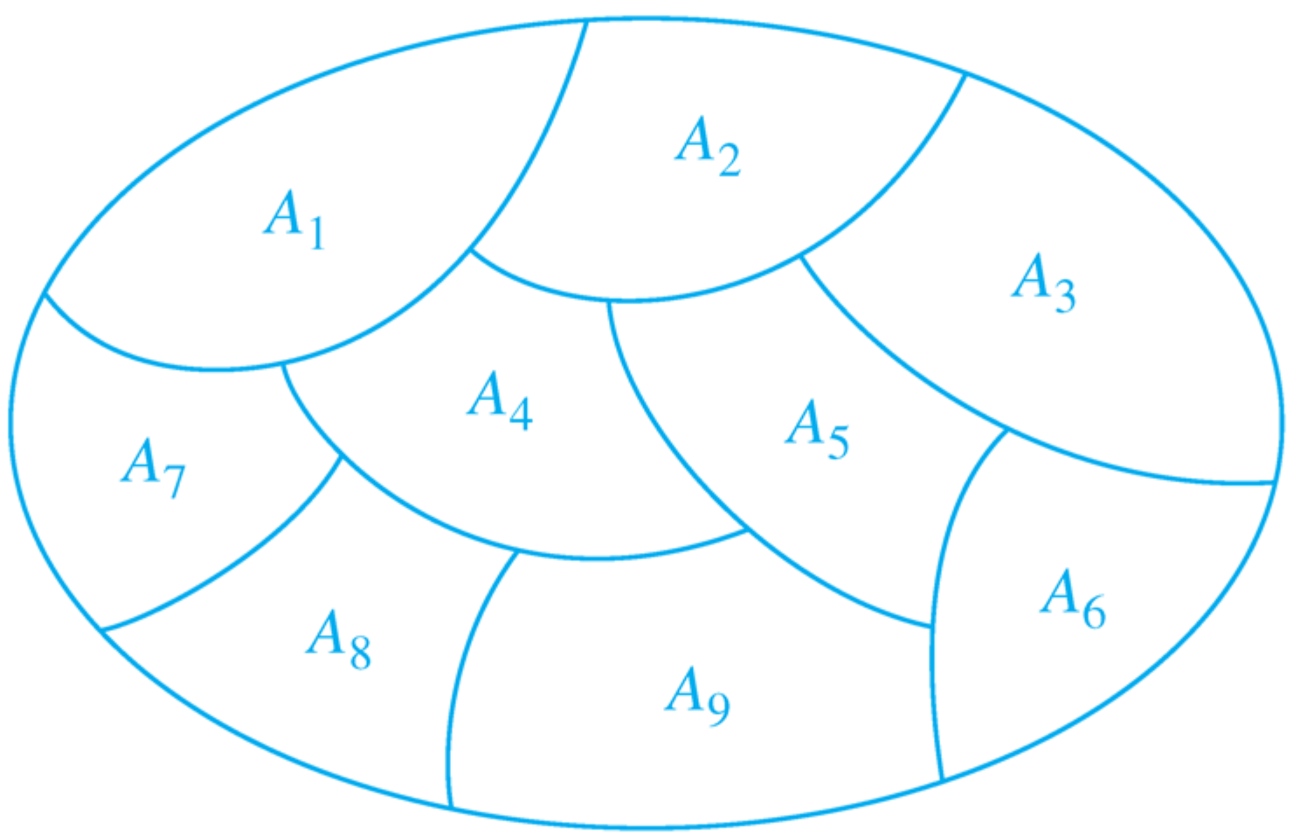
  - *the proof is left as an exercise*

SUSTech

Partition of a Set S

- **Definition:** Let S be a set. A collection of nonempty subsets of S $A_1, A_2, \ldots, A_k$ is called a **partition** of S if:

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



SUSTech

Equivalence Classes and Partitions

- **Theorem:** The equivalence classes form a partition of S.
  - *the proof is left as an exercise*

- **Theorem:** Let $\{A_1, A_2, \ldots, A_i, \ldots\}$ be a partition of S. Then there is an equivalence relation R on S, which has the sets A_i as its equivalence classes.
  - *the proof is left as an exercise*

Partial Orderings

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) or simply $(S, \preccurlyeq)$ in general. Members of S are called elements of the poset.

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

      Yes
    - Is R antisymmetric?

      Yes
    - Is R transitive?

      Yes
    - Therefore, R is a 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) or simply $(S, \preccurlyeq)$ in general. Members of S are called elements of the poset.

- Example 2: $S = \{1, 2, 3, 4, 5\}$, R denotes the " $|$ " relation
  - Is R a partial ordering?

    Yes, *R is reflexive, antisymmetric, and transitive*

- **Notation:** The notation $a \prec b$ denotes that $a \preccurlyeq b$ but $a \neq b$. Also, we say "a is less than b" or "b is greater than a" if $a \prec b$.

Comparability

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

- Example: $S = \{1, 2, 3, 4, 5\}$, R denotes the " $|$ " relation
  - Is $2, 4$ comparable?

    Yes
  - Is $5, 5$ comparable?

    Yes
  - Is $3, 5$ comparable?

    No, because neither of $3 \mid 5$ and $5 \mid 3$ holds

SUSTech

Total Ordering

- **Definition:** If $(S, \preccurlyeq)$ is a poset and every two elements of S are comparable, then $\preccurlyeq$ is called a **total ordering** or **total order**.
  - $(S, \preccurlyeq)$ is called a **totally ordered set** or **linearly ordered set**. A totally ordered set is also called a **chain.**

- Example: Consider $S = \{1, 2, 3, 4, 5\}$
  - Is $(S, \geq)$ a totally (linearly) ordered set?
    Yes
  - Is $(S, |)$ a totally (linearly) ordered set?
    No, because $3, 5$ are not comparable

Example: Lexicographic Ordering

- **Definition:** Given two posets $(A_1, \preccurlyeq_1)$ and $(A_2, \preccurlyeq_2)$, the lexicographic ordering $\preccurlyeq$ on $A_1 \times A_2$ is defined by specifying that
  (a_1, a_2) is less than (b_1, b_2), i.e., $(a_1, a_2) \prec (b_1, b_2)$,
  either if $a_1 \prec_1 b_1$, or if both $a_1 = b_1$ and $a_2 \prec_2 b_2$.
  Then, we obtain a partial ordering $\preccurlyeq$ by adding equality "=" to the above ordering $\prec$ defined on $A_1 \times A_2$.
  - It is easy to see that, if $(A_1, \preccurlyeq_1)$ and $(A_2, \preccurlyeq_2)$ are totally ordered sets, then the lexicographic ordering is also a total order.

- Example: Consider strings of lowercase English letters.
  - The lexicographic ordering can be defined from the total ordering of letters in the alphabet plus the empty letter (less than all letters).
    ** this is the same ordering used in dictionaries*
  - E.g., "discreet" $\prec$ "discrete", "discreet   " $\prec$ "discreetness", etc.

Well-Ordered Induction

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

- **The principle of well-ordered induction:** Suppose that S is a well-ordered set (e.g., $\mathbf{Z}^+$). To prove that $P(x)$ is true for all $x \in S$, we complete two steps:
  - **Basis step:** prove $P(x_0)$ is true for the least element x_0 of S
  - **Inductive step:** prove, for every $x_0 \neq y \in S$, if $P(x)$ is true for all $x \in S$ with $x \prec y$, then $P(y)$ is true.

Well-Ordered Induction

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

- **The principle of well-ordered induction:** Suppose that S is a well-ordered set (e.g., $\mathbf{Z}^+$). To prove that $P(x)$ is true for all $x \in S$, we complete two steps:
  - ~~**Basis step:** prove $P(x_0)$ is true for the least element x_0 of S~~
  - **Inductive step:** prove, for every $\cancel{x_0 \neq}\ y \in S$, if $P(x)$ is true for all $x \in S$ with $x \prec y$, then $P(y)$ is true.
  - *Technically, Basis step is not necessary as it can be derived from Inductive step when y equals the least element x_0.*

- Proof by **contradiction**: consider the set $\{x \in S : P(x) \text{ is false}\}$.
  ** the rest of the proof is left as an exercise*

Well-Ordered Induction Example

- Let $a_{m,n}$ be defined recursively for *(m, n)* $\in$ ***N*** $\times$ ***N*** by $a_{0,0} = 0$ and

$$a_{m,n} = \begin{cases} a_{m-1,n} + 1, & \text{if } n = 0 \text{ and } m > 0 \\ a_{m,n-1} + n, & \text{if } n > 0 \end{cases}$$

  Show that $a_{m,n} = m + n(n+1)/2$ for all *(m, n)* $\in$ ***N*** $\times$ ***N***.

- Proof by **well-ordered induction**:
  - Let ***N*** $\times$ ***N*** with the lexicographic ordering $\preccurlyeq$ be the well-ordered set.
  - **Basis step:** The equality holds for *(0, 0)*, i.e., $a_{0,0} = 0 = 0 + 0 \cdot 1/2$.
  - **Inductive step:** By inductive hypothesis, the equality holds for all *(m', n')* $\prec$ *(m, n)* $\neq$ *(0, 0)*:

    It holds for *(m – 1, n)* if *n = 0* and *(m, n – 1)* if *n > 0*.

    To prove the equality holds for *(m, n)*, just plug the equality for the above two pairs into the two cases that recursively define $a_{m,n}$.

Hasse Diagram

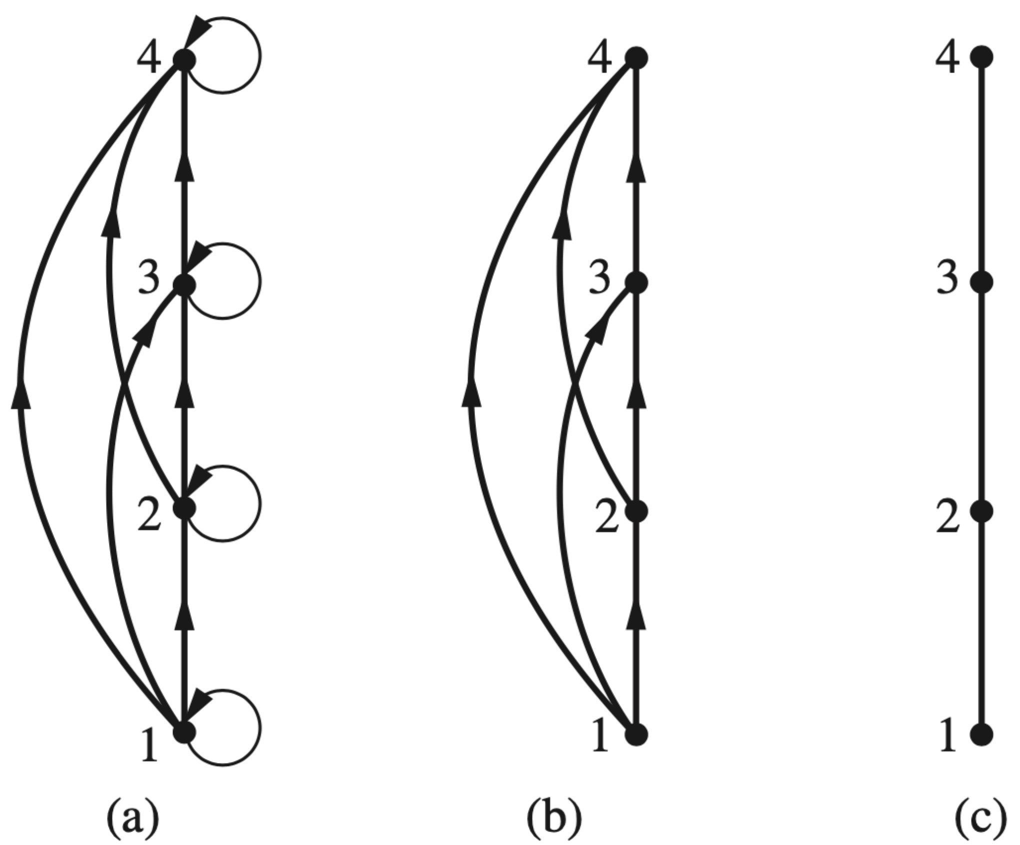
- **Definition:** A **Hasse diagram** is a visual representation of a partial ordering that leaves out edges that must be present because of the reflexive and transitive properties.

- Example: Construct the Hasse diagram of $(\{1, 2, 3, 4\}, \leq)$.

  (a) Draw the directed graph for the partial ordering.

  (b) Remove the loops due to the reflexive property.

  (c) Remove the edges due to the transitive property; and remove all arrows and ensure that all edges point upwards toward their terminal vertex.



Exercise (3 mins)

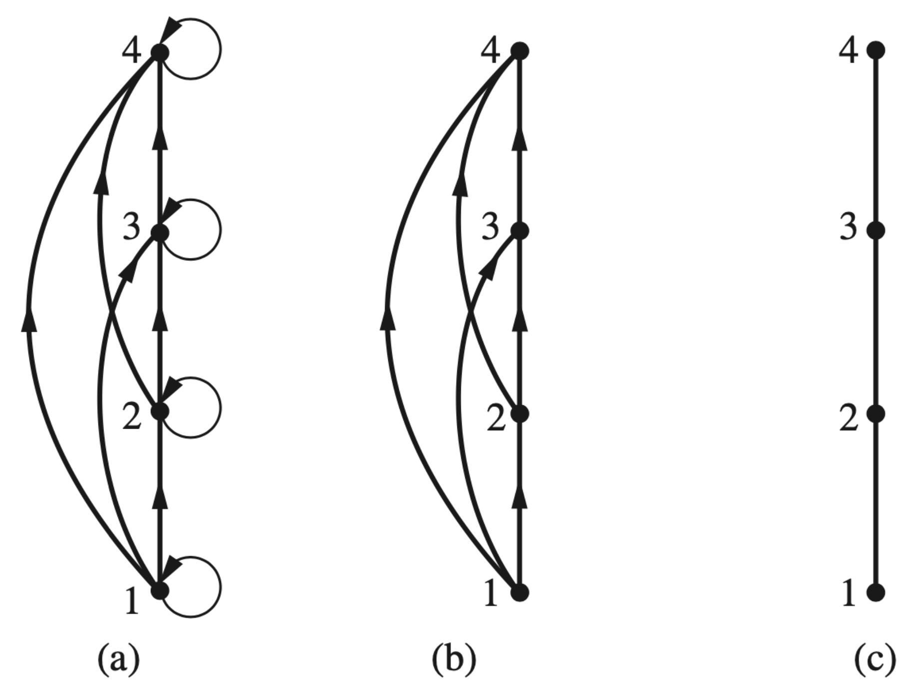
- Construct the Hasse diagram of $(\{1, 2, 3, 4, 6, 8, 12\}, \mid)$.

- Example: Construct the Hasse diagram of $(\{1, 2, 3, 4\}, \leq)$.

  (a) Draw the directed graph for the partial ordering.

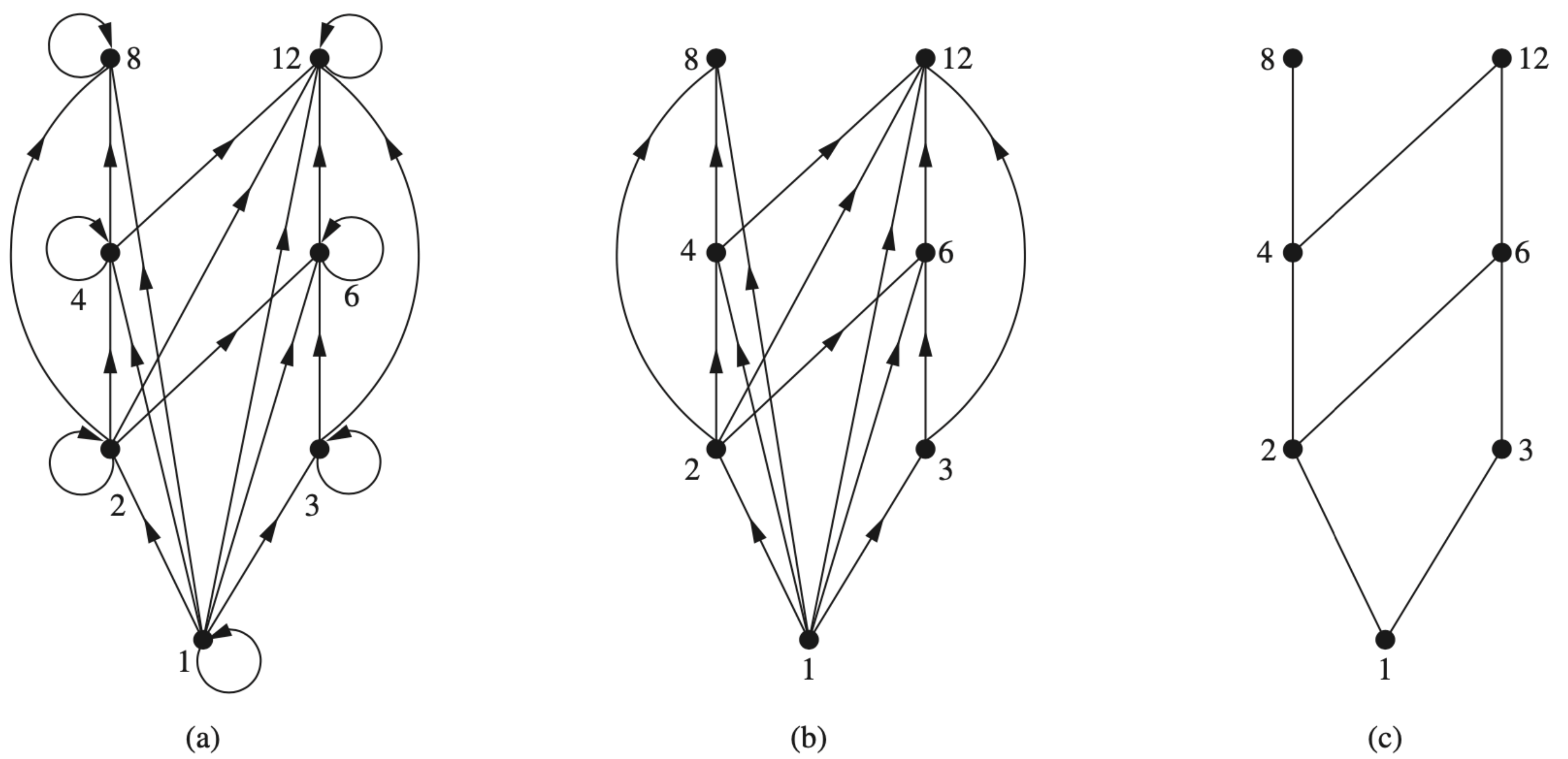
  (b) Remove the loops due to the reflexive property.

  (c) Remove the edges due to the transitive property; and remove all arrows and ensure that all edges point upwards toward their terminal vertex.



Exercise *(3 mins)*

- Construct the Hasse diagram of *({1, 2, 3, 4, 6, 8, 12}, |)*.
- Solution: Figure (c) is the Hasse diagram.



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: Consider the poset $(\{2, 4, 5, 10, 12, 20, 25\}, \mid)$
  - What are the maximal elements?

    $12, 20, 25$
  - What are the minimal elements?

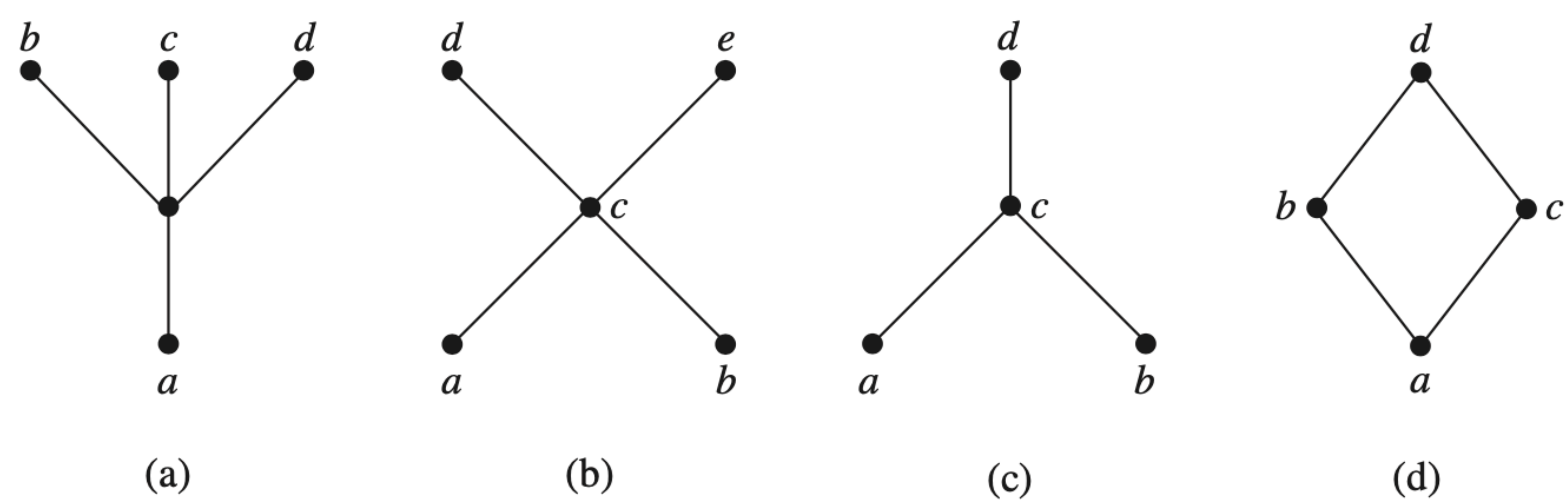
    $2, 5$

Greatest and Least Elements

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

- Example: Find the greatest and least elements, if any.

  (a) least: a (b) none (c) greatest: d (d) least: a greatest: d



SUSTech

Upper and Lower Bounds

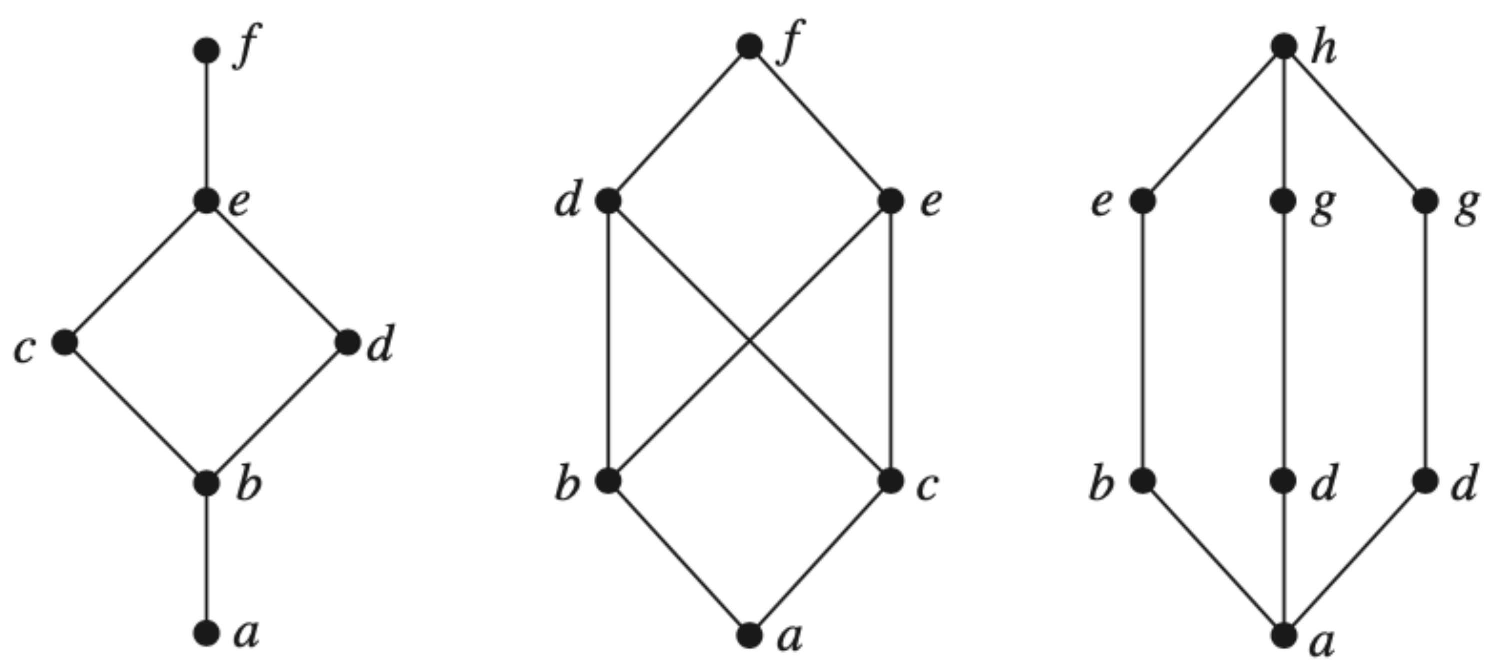
- **Definition:** Let A be a subset of a poset $(S, \preccurlyeq)$ and let $s \in S$.
  - s is called an upper bound (resp. lower bound) of A if for all $a \in A$ we have $a \preccurlyeq s$ (resp. $s \preccurlyeq a$). * *note that $s \in S$ may not belong to A*
  - s is called the least upper bound (resp. greatest lower bound) of A if s is an upper bound (resp. lower bound) that is less (resp. greater) than all other upper bounds (resp. lower bounds) of A.
- Example: Find the greatest lower bound and the least upper bound of set $A = \{1, 2, 4, 5, 10\}$, if they exist, in the poset $(\mathbf{Z}^+, \mid)$.
  - greatest lower bound: 1 least upper bound: 20

SUSTech

Lattices

- **Definition:** 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**.
- Example: Are the following lattices?

  (a) ***Yes*** (b) ***No***, e.g., *(d, e)* has no greatest lower bound (c) ***Yes***



(a) (b) (c)

SUSTech

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?

- Given a partial ordering R, a total ordering $\preccurlyeq$ is said to be **compatible** with R if $a \preccurlyeq b$ whenever $a\ R\ b$.
  - Intuitively, this means the total ordering maintains the structure of the underlying partial ordering.

- **Topological sorting:** construct a compatible total ordering from a partial ordering.

Topological Sorting for Finite Posets

- **Theorem:** Every finite nonempty poset $(S, \preccurlyeq)$ has at least one minimal element.
  - *the proof is left as an exercise*
- Topological sorting algorithm for finite posets:
  - For any minimal element a_k of S, we have $(S - \{a_k\}, \preccurlyeq)$, if not empty, is still a poset. * *the proof is left as an exercise*

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 $S\}$

09 Graphs and Trees

To be continued…

Assignment 5 and Final Exam

- Deadline for Assignment 5: Dec 13
- Final exam will take place on Dec 30 and it captures all materials, including slides, assignments, and relevant sections in textbook.
  - Final exam is closed-book.
  - Please write your answers in English.

SUSTech