

Discrete Structures 2

Lecture 7:

Relations

Part 2

Dr Abdullah Alharbi

Second Semester
(1447 AH – 2026 AD)

Lecture's content is taken from "Discrete Mathematics and Its Applications" 8th edition by Kenneth H. Rosen. McGraw-Hill, 2019.

Lectures Contents:

- **Relations:**
  - **Relations and Their Properties**
  - **n-ary Relations and Their Applications**
  - **Representing Relations**
  - Closures of Relations
  - **Equivalence Relations**
  - **Partial Orderings**

Lecture Contents

- **Relations and Their Properties:**
  - Combining Relations
  - Composite of relations
- ***n*-ary Relations and Their Applications:**
  - Databases & Relations
- **Representing Relations:**
  - Representation using Matrices
  - Representation using Directed Graphs
- **Equivalence Relations:**
  - Equivalence Classes
- **Partial Orderings**

Combining Relations

- **Relations** are **sets**, and therefore, we can apply the usual **set operations** to them.
- Set operations of: **union**, **intersection**, **difference** and **symmetric difference**.
- If we have two relations R_1 and R_2, and both of them are from a set **A** to a set **B**, then we can combine them to:
  - $R_1 \cup R_2$
  - $R_1 \cap R_2$
  - $R_1 - R_2$
  - $R_2 - R_1$
- In each case, the **result** will be **another relation from A to B**

Combining Relations

- **Example:**
  - Let **A** = { 1, 2, 3 } and **B** = { u, v } and
  - $\mathbf{R}_1$ = { (1, u), (2, u), (2, v), (3, u) }
  - $\mathbf{R}_2$ = { (1, v), (3, u), (3, v) }
  - What is $\mathbf{R}_1 \cup \mathbf{R}_2$, $\mathbf{R}_1 \cap \mathbf{R}_2$, $\mathbf{R}_1 - \mathbf{R}_2$, $\mathbf{R}_2 - \mathbf{R}_1$?
- **Solution:**
  - $\mathbf{R}_1 \cup \mathbf{R}_2$ = { (1, u), (1, v), (2, u), (2, v), (3, u), (3, v) }
  - $\mathbf{R}_1 \cap \mathbf{R}_2$ = { (3, u) }
  - $\mathbf{R}_1 - \mathbf{R}_2$ = { (1, u), (2, u), (2, v) }
  - $\mathbf{R}_2 - \mathbf{R}_1$ = { (1, v), (3, v) }

Combining Relations

- **Representation of operations on relations:**
  - **Question:**
    - Can the **relation** be formed by taking the **union** or **intersection** or **composition** of two relations R_1 and R_2 be represented in terms of matrix operations?
  - **Answer:**
    - **Yes**

Combining Relations: Join

- **Definition:** The **join**, denoted by ∨, of two ***m***-by-***n*** matrices ($\mathbf{a}_{ij}$) and ($\mathbf{b}_{ij}$) of **0**s and **1**s is an ***m***-by-***n*** matrix ($\mathbf{m}_{ij}$) where $\mathbf{m}_{ij} = \mathbf{a}_{ij} \vee \mathbf{b}_{ij}$ for all *i*,*j*
  - **Pairwise OR** (**Disjunction**).
- **Example:**
  - Let **A** = {1, 2, 3} and **B** = {u, v} and
  - $\mathbf{R}_1$ = { (1, u), (2, u), (2, v), (3, u) }
  - $\mathbf{R}_2$ = { (1, v), (3, u), (3, v) }

$\mathbf{MR}_1$

1	0
1	1
1	0

$\mathbf{MR}_2$

0	1
0	0
1	1

$\mathbf{M}(\mathbf{R}_1 \vee \mathbf{R}_2)$

1	1
1	1
1	1

Combining Relations: Meet

- **Definition:** The **meet**, denoted by $\wedge$, of two ***m***-by-***n*** matrices $(\mathbf{a}_{ij})$ and $(\mathbf{b}_{ij})$ of **0**s and **1**s is an ***m***-by-***n*** matrix $(\mathbf{m}_{ij})$ where $\mathbf{m}_{ij} = \mathbf{a}_{ij} \wedge \mathbf{b}_{ij}$ for all *i,j*
  - **Pairwise AND** (**Conjunction**).
- **Example:**
  - Let **A** = {1, 2, 3} and **B** = {u, v} and
  - $\mathbf{R}_1$ = { (1, u), (2, u), (2, v), (3, u) }
  - $\mathbf{R}_2$ = { (1, v), (3, u), (3, v) }

$\mathbf{MR}_1$

1	0
1	1
1	0

$\mathbf{MR}_2$

0	1
0	0
1	1

0	0
0	0
1	0

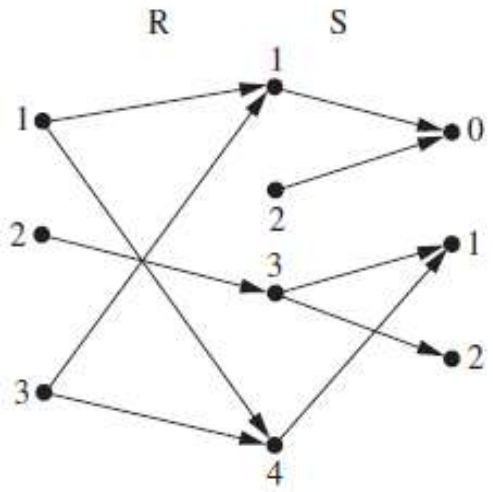
$\mathbf{M}\ (\mathbf{R}_1 \wedge \mathbf{R}_2)$

Composite of Relations

- … and there is **another important way to combine relations**.
- **Definition:** Let **R** be a relation from a set **A** to a set **B** and **S** a relation from **B** to a set **C**. The **composite** of **R** and **S** is the relation consisting of ordered pairs (**a**, **c**), where **a**$\in$**A**, **c**$\in$**C**, and for which there exists an element **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**
- In other words, if relation **R** contains a pair (**a**, **b**) and relation **S** contains a pair (**b**, **c**), then **S** $\circ$ **R** contains a pair (**a**, **c**)

Composite of Relations

- **Examples:**
  - **R** = { (1, 1), (1, 4), (2, 3), (3, 1), (3, 4) } from { 1, 2, 3 } to { 1, 2, 3, 4 }
  - **S** = { (1, 0), (2, 0), (3, 1), (3, 2), (4, 1) } from { 1, 2, 3, 4 } to { 0, 1, 2 }
  - What is **S** ◦ **R**?
- **Solution:**
  - **S** ◦ **R** = { (1, 0), (1, 1), (2, 1), (2, 2), (3, 0), (3, 1) }



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

Composite of Relations

- **Examples:**
  - Let **A** = {1, 2, 3}, **B** = {0, 1, 2} and **C** = {a, b}
  - **R** = { (1, 0), (1, 2), (3, 1), (3, 2) }
  - **S** = { (0, b), (1, a), (2, b) }
  - What is **S** ∘ **R**?
- **Solution:**
  - **S** ∘ **R** = { (1, b), (3, a), (3, b) }

Composite of Relations

- **Example:** Let **D** and **S** be relations on **A** = {1, 2, 3, 4}
- **D** = { (a, b) | b = 5 - a } "b equals (5 – a)"
- **S** = { (a, b) | a < b } "a is smaller than b"
- **D** = { (1, 4), (2, 3), (3, 2), (4, 1) }
- **S** = { (1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4) }
- **S** ∘ **D** = { (2, 4), (3, 3), (3, 4), (4, 2), (4, 3), (4, 4) }
- **D** maps an element a to the element (5 – a), and afterwards **S** maps (5 – a) to all elements larger than (5 – a), resulting in **S** ∘ **D** = **{ (a, b) | b > 5 – a }** or **S** ∘ **D** = **{ (a, b) | a + b > 5 }**

Composite of Relations

- We already know that **functions** are just **special cases** of **relations** (namely those that *map each element in the **domain** onto exactly one element in the **codomain***).
- If we formally convert two functions into relations, that is, write them down as sets of ordered pairs, the composite of these **relations** will be exactly the same as the composite of the **functions** (as defined earlier).

Composite of Relations

- **Definition:** Let **R** be a relation on the set **A**. The **powers** $\mathbf{R}^n$, n = 1, 2, 3, …, are defined inductively by
  - $\mathbf{R}^1 = \mathbf{R}$
  - $\mathbf{R}^{n+1} = \mathbf{R}^n \circ \mathbf{R}$

- In other words:
- $\mathbf{R}^n = \mathbf{R} \circ \mathbf{R} \circ \ldots \circ \mathbf{R}$ (n times the letter **R**)

Composite of Relations

- **Examples**
  - **R** = { (1, 2), (2, 3), (2, 4), (3, 3) } is a relation on **A** = {1, 2, 3, 4}
  - $\mathbf{R}^1$ = **R** = { (1, 2), (2, 3), (2, 4), (3, 3) }
  - $\mathbf{R}^2$ = { (1, 3), (1, 4), (2, 3), (3, 3) }
  - $\mathbf{R}^3$ = { (1, 3), (2, 3), (3, 3) }
  - $\mathbf{R}^4$ = { (1, 3), (2, 3), (3, 3) }

Composite of Relations

- **Theorem:** The relation **R** on a set **A** is **transitive** if and only if $R^n \subseteq R$ for all positive integers ***n***
- Remember the definition of **transitivity**:
- **Definition:** A relation **R** on a set **A** is called **transitive** if whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$ for $a, b, c \in A$
- The composite of **R** with itself contains exactly these pairs **(a, c)**
- Therefore, for a **transitive** relation **R**, $R \circ R$ **does not** contain any pairs that are **not** in **R**, so $R \circ R \subseteq R$
- Since $R \circ R$ does not introduce any pairs that are not already in **R**, it must also be **true** that $(R \circ R) \circ R \subseteq R$, and so on, so that $R^n \subseteq R$

n-ary Relations

- In order to study an interesting application of relations, namely **databases**, we first need to generalize the concept of **binary relations** to ***n*-ary relations**.
- **Definition:** Let A_1, A_2, …, A_n be sets. An ***n*-ary relation** on these sets is a *subset* of $A_1 \times A_2 \times \ldots \times A_n$
- The sets A_1, A_2, …, A_n are called the **domains** of the relation, and ***n*** is called its **degree**.

n-ary Relations

- **Example:**
  - Let **R** = { (a, b, c) | a = 2b $\wedge$ b = 2c with a, b, c $\in$ **Z** }
  - What is the degree of **R**?
  - The degree of R is 3, so its elements are triples.
  - What are its domains?
  - Its domains are all equal to the set of integers.
  - Is (2, 4, 8) in **R**?
  - No
  - Is (4, 2, 1) in **R**?
  - Yes

Databases & Relations

- Let us take a look at a type of database representation that is based on relations, namely the **relational data model.**
- A database consists of *n*-tuples called **records**, which are made up of **fields**.
- These **fields** are the **entries** of the *n-tuples*.
- The **relational data model** represents a database as an *n*-ary relation, that is, a set of records.

Databases & Relations

- **Example:** Consider a database of students, whose records are represented as *4*-tuples with the fields **Student Name**, **ID Number**, **Major**, and **GPA**:

- **R** = { (Ackermann, 231455, CS, 3.88),
  (Adams, 888323, Physics, 3.45),
  (Chou, 102147, CS, 3.79),
  (Goodfriend, 453876, Math, 3.45),
  (Rao, 678543, Math, 3.90),
  (Stevens, 786576, Psych, 2.99) }

- Relations that represent databases are also called **tables**, since they are often displayed as tables.

Databases & Relations

- A **domain** of an *n*-ary relation is called a **primary key** if the ***n-tuples*** are uniquely determined by their **values** from this domain.
- This means that no two records have the same value from the same primary key.
- In our example, which of the fields **Student Name**, **ID Number**, **Major**, and **GPA** are primary keys?
- **Student Name** and **ID Number** are primary keys, because no two students have identical values in these fields.
- In a **real** student database, only **ID Number** would be a primary key.

Databases & Relations

- In a database, a **primary key** should remain **one** even if **new records** are added.
- Therefore, we should use a **primary key** of the **intension** of the database, containing all the *n-tuples* that can ever be included in our database.
- **Combinations of domains** can also uniquely identify *n-tuples* in an *n*-ary relation.
- When the **values** of a **set of domains** determine an *n-tuple* in a relation, the **Cartesian product** of these domains is called a **composite key**.

Databases & Relations

- We can apply a variety of **operations** on *n*-ary relations to form new relations.
- **Definition:** The **projection** $P_{i_1, i_2, \ldots, i_m}$ ***maps*** the ***n-tuple*** $(a_1, a_2, \ldots, a_n)$ to the ***m-tuple*** $(a_{i_1}, a_{i_2}, \ldots, a_{i_m})$, where $m \leq n$
- In other words, a **projection** $P_{i_1, i_2, \ldots, i_m}$ keeps the ***m*** components $a_{i_1}, a_{i_2}, \ldots, a_{i_m}$ of an ***n-tuple*** and deletes its $(n - m)$ other components.
- **Example:** What is the result when we apply the **projection** $P_{2,4}$ to the student record (Stevens, 786576, Psych, 2.99) ?
- **Solution:** It is the pair (786576, 2.99)

Databases & Relations

- In some cases, applying a **projection** to an entire **table** may **not** only result in **fewer columns**, but also in **fewer rows**.
- Why is that?
- Some **records** may only have differed in those fields that were **deleted**, so they become **identical**, and there is no need to list identical records more than once.

Databases & Relations

- We can use the **join operation** to combine two tables into one if they share some identical fields.

- **Definition:** Let **R** be a relation of degree ***m*** and **S** a relation of degree ***n***. The **join** $J_p(R, S)$, where $p \leq m$ and $p \leq n$, is a relation of **degree** $m + n - p$ that consists of all $(m + n - p)$-tuples $(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p})$, where the m-tuple $(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p)$ belongs to **R** and the n-tuple $(c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p})$ belongs to **S**.

Databases & Relations

- In other words, to generate $\mathbf{J_p(R, S)}$, we have to find all the elements in **R** whose **p** last components match the **p** first components of an element in **S**.

- The **new relation** contains exactly these matches, which are combined to tuples that contain each matching field only once.

Databases & Relations

- **Example:**
  - What is $\mathbf{J}_1(\mathbf{Y}, \mathbf{R})$, where **Y** contains the fields **Student Name** and **Year of Birth**,
  - **Y** = { (1978, Ackermann),
    (1972, Adams),
    (1917, Chou),
    (1984, Goodfriend),
    (1982, Rao),
    (1970, Stevens) },
  - and **R** contains the **student records** as defined before ?

Databases & Relations

- **Solution:**
  - The resulting relation is:
  - { (1978, Ackermann, 231455, CS, 3.88),
    (1972, Adams, 888323, Physics, 3.45),
    (1917, Chou, 102147, CS, 3.79),
    (1984, Goodfriend, 453876, Math, 3.45),
    (1982, Rao, 678543, Math, 3.90),
    (1970, Stevens, 786576, Psych, 2.99) }
  - Since **Y** has two fields and **R** has four, the relation $\mathbf{J_1}$(**Y**, **R**) has **2 + 4 – 1 = 5 fields**.

Representing Relations

- We already know different ways of representing relations.
- We will now take a closer look at **two** ways of representation:
  - **Zero-One Matrices**
  - **Directed Graphs (Digraphs)**
- If **R** is a **relation** from **A** = { $a_1, a_2, \ldots, a_m$ } to **B** = { $b_1, b_2, \ldots, b_n$ }, then **R** can be represented by the zero-one matrix $\mathbf{M}_R = [m_{ij}]$ with
  - $m_{ij} = 1$, if $(a_i, b_j) \in R$
  - $m_{ij} = 0$, if $(a_i, b_j) \notin R$
- Note that for creating this matrix we first need to list the elements in **A** and **B** in a **particular, but arbitrary order**.

Representing Relations using Matrices

- Let us now assume that the zero-one matrices $\mathbf{M}_A = [a_{ij}]$, $\mathbf{M}_B = [b_{ij}]$ and $\mathbf{M}_C = [c_{ij}]$ represent relations **A**, **B**, and **C**, respectively.
- **Remember:** For $\mathbf{M}_C = \mathbf{M}_A \circ \mathbf{M}_B$ we have:
  - $c_{ij} = 1$ if and only if at least one of the terms $(a_{in} \wedge b_{nj}) = 1$ for some n; ***otherwise*** $c_{ij} = 0$
- In terms of the **relations**, this means that **C** contains a **pair** (x_i, z_j) if and only if there is an **element** y_n such that (x_i, y_n) is in relation **A** and (y_n, z_j) is in relation **B**.
  - Therefore, **C** = **B** ◦ **A** (**composite** of **A** and **B**)

Representing Relations using Matrices

- This gives us the following rule:
  - $\mathbf{M}_{B \circ A} = \mathbf{M}_A \odot \mathbf{M}_B$
- In other words, the matrix representing the composite of relations **A** and **B** is the Boolean product (⊙)of the matrices representing **A** and **B**.
- Analogously, we can find matrices representing the powers of relations:
  - $\mathbf{M}_R^n = \mathbf{M}_R^{[n]}$ (*n*-th Boolean power)

Representing Relations using Matrices

- **Example:**
  - Let **A** = { 1, 2 }, **B** = { 1, 2, 3 } **C** = { a, b }
  - **R** = { (1, 2), (1, 3), (2, 1) } is a relation from **A** to **B**
  - **S** = { (1, a), (3, b), (3, a) } is a relation from **B** to **C**
  - **S** ∘ **R** = { (1, b), (1, a), (2, a) }

0	1	1
1	0	0

$\mathbf{M}_R$

1	0
0	0
1	1

$\mathbf{M}_S$

1	1
1	0

$\mathbf{M}_{S \circ R}$

1	1
1	0

$\mathbf{M}_R \odot \mathbf{M}_S$

Representing Relations using Matrices

- **Example:** Find the matrix representing R^2, where the matrix representing R is given by

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

- **Solution:** The matrix for R^2 is given by

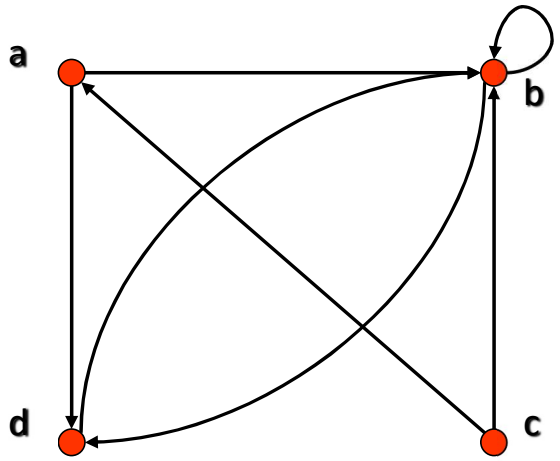
$$M_{R^2} = M_R^{[2]} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Representing Relations using Digraphs

- **Definition:** A **directed graph**, or **digraph**, consists of a set **V** of **vertices** (or **nodes**) together with a set **E** of ordered pairs of elements of **V** called **edges** (or **arcs**).
- The **vertex a** is called the **initial vertex** of the edge (**a**, **b**), and the **vertex b** is called the **terminal vertex** of this edge.
- We can use arrows to display graphs.

Representing Relations using Digraphs

- **Example:**
  - Display the **digraph** with **V** = { a, b, c, d },
  - **E** = { (a, b), (a, d), (b, b), (b, d), (c, a), (c, b), (d, b) }



An **edge** of the form (**b**, **b**) is called a **loop**

Representing Relations using Digraphs

- **Example:**
  - What are the ordered pairs in the relation $\mathbf{R}_2$ represented by the directed graph shown in Figure 1?
- **Solution:**
  - $\mathbf{R}_2$ = { (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (3, 1), (3, 3), (4, 1), (4, 3) }

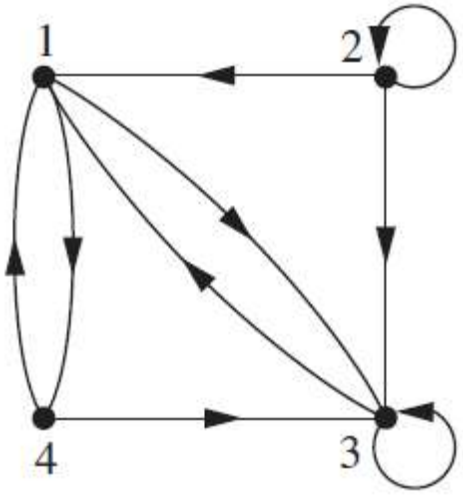


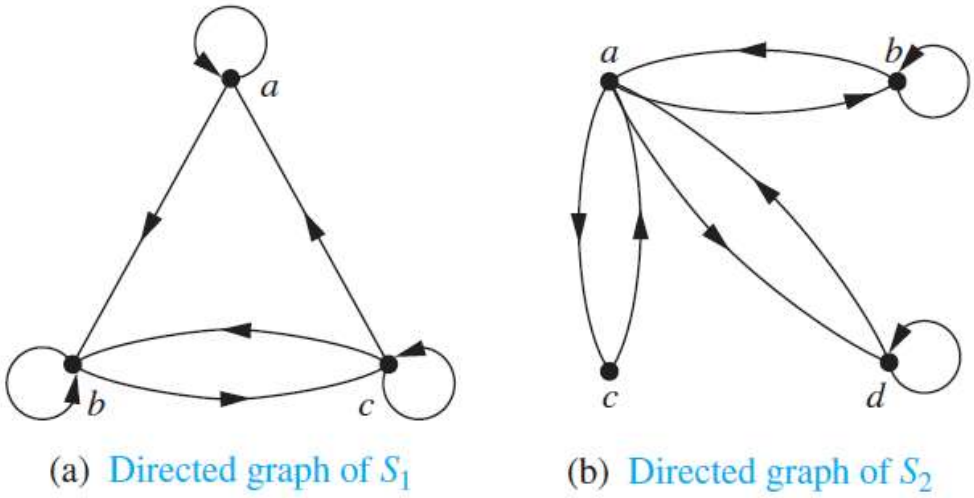
Figure 1

Representing Relations using Digraphs

- The **directed graph** representing a **relation** can be used to determine whether the **relation** has various **properties**.
  - **Reflexive**: if and only if there is a **loop** at every vertex of the directed graph.
  - **Symmetric**: if and only if for every edge between **distinct** vertices in its digraph there is an edge in the opposite direction (both (**a**, **b**) & (**b**, **a**) are on the graph).
  - **Antisymmetric**: if and only if there are **never** two edges in opposite directions between distinct vertices.
  - **Transitive**: if and only if whenever there is an edge from a vertex **a** to a vertex **b** and an edge from a vertex **b** to a vertex **c**, there is an edge from **a** to **c** (completing a triangle where each side is a directed edge with the correct direction).

Representing Relations using Digraphs

- **Example:**
  - Determine whether the relations for the directed graphs shown in **Figure 2** are **reflexive**, **symmetric**, **antisymmetric**, and/or transitive.
- **Solution:**
  - $\mathbf{S}_1$ is **reflexive**
  - $\mathbf{S}_2$ is **symmetric**



(a) Directed graph of S_1 (b) Directed graph of S_2

Figure 2

Representing Relations using Digraphs

- Obviously, we can represent any **relation R** on a **set A** by the **digraph** with **A** as its **vertices** and **all pairs** $(a, b) \in R$ as its **edges**.
- Vice versa, any **digraph** with **vertices V** and **edges E** can be represented by a **relation** on **V** containing **all the pairs** in **E**.
- This **one-to-one correspondence** between **relations** and **digraphs** means that any statement about relations also applies to digraphs, and vice versa.

Equivalence Relations

- **Equivalence relations** are used to relate objects that are similar in some way.
- **Definition:** A **relation** on a set **A** is called an **equivalence relation** if it is (1) reflexive, (2) symmetric, and (3) transitive.
- Two elements that are related by an **equivalence relation R** are called **equivalent**.

Equivalence Relations

- Since R is symmetric, a is equivalent to b whenever b is equivalent to a.
- Since R is reflexive, every element is equivalent to itself.
- Since R is transitive, if a and b are equivalent and b and c are equivalent, then a and c are equivalent.
- Obviously, these three properties are necessary for a reasonable definition of equivalence.

Equivalence Relations

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

Equivalence Classes

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

Equivalence Classes

- **Example:**
  - In the previous example (strings of identical length), what is the equivalence class of the word mouse, denoted by [mouse] ?
- **Solution:**
  - [mouse] is the set of all English words containing five letters.
  - For example, 'horse' would be a representative of this **equivalence class**.

Equivalence Classes

- **Theorem:** Let **R** be an **equivalence** relation on a set **A**. The following statements are equivalent:
  - aRb
  - $[a] = [b]$
  - $[a] \cap [b] \neq \emptyset$
- **Definition:** A **partition** of a set **S** is a collection of disjoint nonempty subsets of **S** that have **S** as their union. In other words, the collection of subsets $\mathbf{A}_i$, $i \in I$, forms a partition of **S** if and only if (i) $\mathbf{A}_i \neq \emptyset$ for $i \in I$
  - $A_i \cap A_j = \emptyset$, if $i \neq j$
  - $\cup_{i \in I} A_i = S$

Equivalence Classes

- **Example:** Let **S** be the set { u, m, b, r, o, c, k, s }. Do the following collections of sets partition **S** ?

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

Equivalence Classes

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

Equivalence Classes

- **Example:** Let us assume that Frank, Suzanne and George live in Boston, Stephanie and Max live in Lübeck, and Jennifer lives in Sydney.
- Let **R** be the **equivalence relation** { (a, b) | a and b live in the same city } on the set **P** = { Frank, Suzanne, George, Stephanie, Max, Jennifer }
- Then **R** = { (Frank, Frank), (Frank, Suzanne), (Frank, George), (Suzanne, Frank), (Suzanne, Suzanne), (Suzanne, George), (George, Frank), (George, Suzanne), (George, George), (Stephanie, Stephanie), (Stephanie, Max), (Max, Stephanie), (Max, Max), (Jennifer, Jennifer) }.

Equivalence Classes

- Then the **equivalence classes** of **R** are:
- { { Frank, Suzanne, George }, { Stephanie, Max }, { Jennifer } }
- This is a **partition** of **P**
- The equivalence classes of any equivalence relation **R** defined on a set **S** constitute a partition of **S**, because every element in **S** is assigned to **exactly one** of the equivalence classes.

Equivalence Classes

- **Another example:** Let **R** be the relation { (a, b) | a ≡ b (mod 3) } on the set of integers.
- Is **R** an equivalence relation?
- Yes, **R** is reflexive, symmetric, and transitive.
- What are the equivalence classes of **R** ?
- { {…, -6, -3, 0, 3, 6, …},
  {…, -5, -2, 1, 4, 7, …},
  {…, -4, -1, 2, 5, 8, …} }

Partial Orderings

- **Definition 1**: A relation *R* on a set S is called a ***partial ordering***, or *partial order*, if it is reflexive, antisymmetric, and transitive.
- A set 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***.

- **Example 1**: Show that the "greater than or equal" relation (≥) is a partial ordering on the set of integers.

Example 1 Solution

- **Example 1**: Show that the "greater than or equal" relation (≥) is a partial ordering on the set of integers.
- **Solution:**
  - *Reflexivity*: $a \geq a$ for every integer a.
  - *Antisymmetry*: If $a \geq b$ and $b \geq a$, then $a = b$.
  - *Transitivity*: If $a \geq b$ and $b \geq c$, then $a \geq c$.
  - It follows that ≥ is a partial ordering on the set of integers and (**Z**, ≥) is a poset.

- **Example 2**: Show that the divisibility relation (|) is a partial ordering on the set of positive integers.

Example 2 Solution

- **Example 2**: Show that the divisibility relation (|) is a partial ordering on the set of positive integers.
- **Solution:**
  - *Reflexivity*: $a \mid a$ for all integers a.
  - *Antisymmetry*: If a and b are positive integers with $a \mid b$ and $b \mid a$, then $a = b$.
  - *Transitivity*: Suppose that a divides b and b divides c. Then there are positive integers k and l such that $b = ak$ and $c = bl$. Hence, $c = a(kl)$, so a divides c. Therefore, the relation is transitive.
  - It follows that | is a partial ordering on the set of integers and $(\mathbf{Z}^+, |)$ is a poset.

- **Example 3**: Show that the inclusion relation ($\subseteq$) is a partial ordering on the power set of a set S.

Example 3 Solution

- **Example 3**: Show that the inclusion relation ($\subseteq$) is a partial ordering on the power set of a set S.
- **Solution:**
  - *Reflexivity*: $A \subseteq A$ whenever A is a subset of S.
  - *Antisymmetry*: If A and B are positive integers with $A \subseteq B$ and $B \subseteq A$, then $A = B$.
  - *Transitivity*: If $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.

  - It follows that $\subseteq$ is a partial ordering on the set S and (S, $\subseteq$) is a poset.

Any Questions?

- Have a look at the "**Extra Examples**" on the textbook's website.
- Keep reviewing this lecture *regularly, as well as other lectures*.
- Have a look at the **exercises** at the end of the sections we covered in this lecture in **Chapter 9** and do your best with those ones similar to what we presented in this lecture.
  - You should get more *examples* and *exercises* in the *tutorials*.
- Ask *me or ask your peers* if you have any questions.
  - Please, do not wait until the end of the semester or even right before exams.

All the best!