UNIVERSITY AT ALBANY
State University of New York

Lecture 32: Relations (2)

Dr. Chengjiang Long
Computer Vision Researcher at Kitware Inc.
Adjunct Professor at SUNY at Albany.
Email: **clong2@albany.edu**

Outline

- Representing Relations Using Matrices
- Representing Relations Using Digraph
- Equivalence Relations
- Equivalence Classes
- Application Example

Outline

- **Representing Relations Using Matrices**
- Representing Relations Using Digraph
- Equivalence Relations
- Equivalence Classes
- Application Example

Representing Relations Using Matrices

- We already know different ways of representing relations. We will now take a closer look at two ways of representation: **Zero-one matrices** and **directed graphs**.
- 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 $M_R = [m_{ij}]$ with
- $m_{ij} = 1$, if $(a_i, b_j) \in R$, and
- $m_{ij} = 0$, 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

- **Example:** How can we represent the relation $R = \{(2, 1), (3, 1), (3, 2)\}$ as a zero-one matrix?

- **Solution:** The matrix M_R is given by

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

Representing Relations Using Matrices

- What do we know about the matrices representing a **relation on a set** (a relation from A to A) ?
- They are **square** matrices.
- What do we know about matrices representing **reflexive** relations?
- All the elements on the **diagonal** of such matrices M_{ref} must be **1s**.

$$M_{ref} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & 1 \end{bmatrix}$$

Representing Relations Using Matrices

- What do we know about the matrices representing **symmetric relations**?
- These matrices are symmetric, that is, $M_R = (M_R)^t$.

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

symmetric matrix,
symmetric relation.

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

non-symmetric matrix,
non-symmetric relation.

Representing Relations Using Matrices

- The Boolean operations **join** and **meet** **(you remember?)** can be used to determine the matrices representing the **union** and the **intersection** of two relations, respectively.

- To obtain the **join** of two zero-one matrices, we apply the Boolean "or" function to all corresponding elements in the matrices.

- To obtain the **meet** of two zero-one matrices, we apply the Boolean "and" function to all corresponding elements in the matrices.

Representing Relations Using Matrices

- **Example:** Let the relations R and S be represented by the matrices

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

What are the matrices representing R∪S and R∩S?
Solution: These matrices are given by

$$M_{R \cup S} = M_R \vee M_S = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \qquad M_{R \cap S} = M_R \wedge M_S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Representing Relations Using Matrices

- Do you remember the **Boolean product** of two zero-one matrices?
- 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 $A \circ B$, is the $m \times n$ matrix with (i, j)th entry $[c_{ij}]$, where
- $c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2i}) \vee \ldots \vee (a_{ik} \wedge b_{kj})$.
- $c_{ij} = 1$ if and only if at least one of the terms $(a_{in} \wedge b_{nj}) = 1$ for some n; otherwise $c_{ij} = 0$.

Representing Relations Using Matrices

- Let us now assume that the zero-one matrices $M_A = [a_{ij}]$, $M_B = [b_{ij}]$ and $M_C = [c_{ij}]$ represent relations A, B, and C, respectively.
- **Remember:** For $M_C = M_A \odot M_B$ 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, $C = B \circ A$ (**composite** of A and B).

Representing Relations Using Matrices

- This gives us the following rule:
- $M_{B \circ A} = M_A \circ M_B$
- 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**:
- $M_{R^n} = M_R^{[n]}$ (n-th **Boolean power**).

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

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

Outline

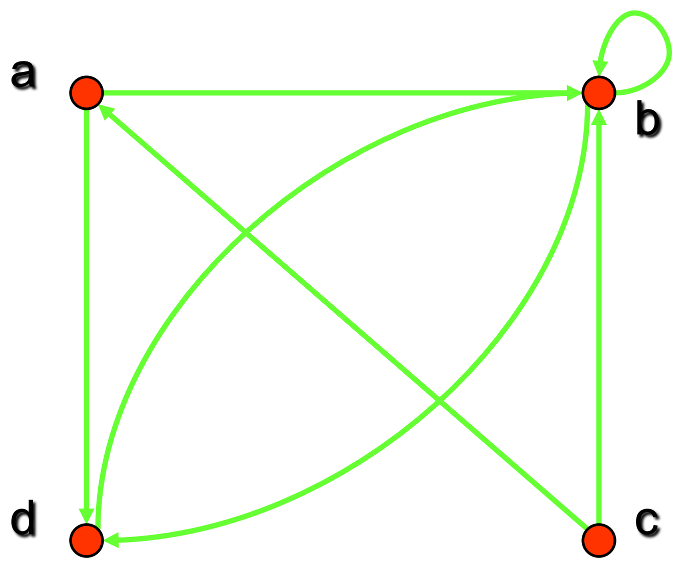
- Representing Relations Using Matrices
- **Representing Relations Using Digraph**
- Equivalence Relations
- Equivalence Classes
- Application Example

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

- 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.

Outline

- Representing Relations Using Matrices
- Representing Relations Using Digraph
- **Equivalence Relations**
- Equivalence Classes
- Application Example

Equivalence Relations

- **Equivalence relations** are used to relate objects that are similar in some way.
- **Definition:** A relation on a set A is called an equivalence relation if it is reflexive, symmetric, and transitive.
- Two elements 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 aRb if and only if l(a) = l(b), where l(x) is the length of the string x. Is R an equivalence relation?
- **Solution:**
  - R is reflexive, because l(a) = l(a) and therefore aRa for any string a.
  - R is symmetric, because if l(a) = l(b) then l(b) = l(a), so if aRb then bRa.
  - R is transitive, because if l(a) = l(b) and l(b) = l(c), then l(a) = l(c), so aRb and bRc implies aRc.
- **R is an equivalence relation.**

Outline

- Representing Relations Using Matrices
- Representing Relations Using Digraph
- Equivalence Relations
- **Equivalence Classes**
- Application Example

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*.
- 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.

Example

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

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

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

Equivalence Classes

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

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, k}, {r, u, m}, {c, 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.

Example

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

Outline

- Representing Relations Using Matrices
- Representing Relations Using Digraph
- Equivalence Relations
- Equivalence Classes
- **Application Example**

Databases and Relations

- Consider a relational 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 and 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 and 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 and 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 and Relations

- **Example:** What is $J_1(Y, R)$, 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 and 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 $J_1(Y, R)$ has 2 + 4 – 1 = 5 fields.

Next class

- Topic: Graph Theory (1)
- Pre-class reading: Chap 10.1-10.2