Chapter 9 Relations

9.1 Relations and Their Properties

9.2 n-ary Relations and Their Applications

9.3 Representing Relations

【Definition】A *binary relation* R from a set A to a set B is a subset of $A \times B$.

Note:

- A *binary relation* R is a set.
- $R \subseteq A \times B$
- $R = \{(a, b) | a \in A, b \in B, aRb\}$

1. Functions As Relations

The graph of function f from set A to set B is a relation from A to B.

2. Relations on a Set

【Definition】A *relation on the set* A is a relation from A to A.

Note:

- $R \subseteq A \times A$

3. Representing Relations

The methods of representing a relation:

- list its all ordered pairs
- using a set build notation/specification by predicates
- 2D table

  $R = \{(2,2), (2,4), (2,6), (3,3), (3,6), (4,4)\}$

	2	3	4	5	6
2	×		×		×
3		×			×
4			×		

- Connection matrix /zero-one matrix
- Directed graph/Digraph

4. Connection Matrices

【Definition】Let R be a relation from $A = \{a_1, a_2, \cdots, a_m\}$, $to\ B = \{b_1, b_2, \cdots, b_n\}$

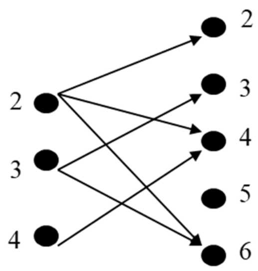
An $m \times n$ *connection matrix* $M_R = [m_{ij}]$ for R is defined by $m_{ij} = \begin{cases} 1 & if (a_i, b_j) \in R, \\ 0 & if (a_i, b_j) \notin R. \end{cases}$

5. Directed graph/Digraph

【Definition】A *directed graph* or a *digraph*, consists of a set V of *vertices* together with a set E of ordered pairs of elements of V called *edges(or arcs)*.

The *vertices a,b* is called the *initial* and *terminal* vertices of the edge *(a,b)*, respectively.

$A = \{2,3,4\}, B = \{2,3,4,5,6\}$ $\quad R = \{(x, y) | x \in A, y \in B, x|y\}$



6. Special Properties of Binary Relations

■ Reflexive

【Definition】A relation R on a set A is *reflexive* if $(x, x) \in R, for\ every\ element\ x \in A$.

$\forall x(x \in A \rightarrow (x, x) \in R)$

Matrices Representing：All the elements on the main diagonal of M_R must be 1s.

$$M_r = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot & \\ & & & & & 1 \end{bmatrix}$$

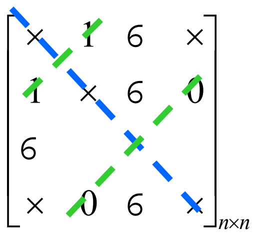
Digraphs Representing: There is a loop at every vertex of the directed graph.

■ Irreflexive

【Definition】A relation R on a set A is *irreflexive* if $\forall x(x \in A \rightarrow (x, x) \notin R)$

■ Symmetric

【Definition】A relation R on a set A is *symmetric* if $\forall x \forall y((x, y) \in R \rightarrow (y, x) \in R)$



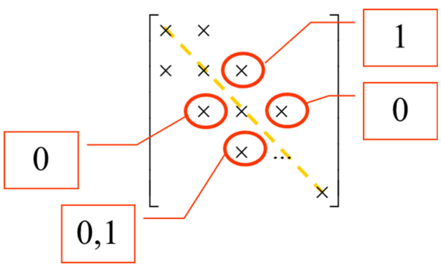
Digraphs Representing: If there is an arc (x, y) there must be an arc (y, x).

■ Antisymmetric

【Definition】A relation R on a set A is *antisymmetric* if $\forall x \forall y((x, y) \in R \wedge (y, x) \in R \rightarrow x = y)$

Note:

(1) $\forall x \forall y((x, y) \in R \wedge x \neq y \rightarrow (y, x) \notin R)$



(2)

(3) If there is an arc from x to y there cannot be one from y to x if $x \neq y$.

(4) The symmetric and antisymmetric relations are not opposites.

■ Transitive

【Definition】A relation R on a set A is *transitive* if whenever $\forall x \forall y \forall z((x, y) \in R \wedge (y, z) \in R \rightarrow$

$(x, z) \in R)$

Note:

(1) $\overline{(m_{ij} \wedge m_{jk})} \vee m_{ik} = 1$

(2) If there is an arc from *x* to *y* and one from *y* to *z* then there must be one from *x* to *z*.

7. Combining Relations

1) Set operations

(1) Set operations $\cup, \cap, \overline{\ \ }, -, \oplus$

(2) Boolean operations/logical operations

The Boolean Or $\vee$: $0 \vee 0 = 0, 0 \vee 1 = 1, 1 \vee 0 = 1, 1 \vee 1 = 1$

The Boolean And $\wedge$: $0 \wedge 0 = 0, 0 \wedge 1 = 0, 1 \wedge 0 = 0, 1 \wedge 1 = 1$

The Complement $-$: $\bar{0} = 1, \bar{1} = 0$

The logical operations of matrices：

$M_{R_1 \cup R_2} = [c_{ij} \vee d_{ij}] = M_{R_1} \vee M_{R_2}$

$M_{R_1 \cap R_2} = [c_{ij} \wedge d_{ij}] = M_{R_1} \wedge M_{R_2}$

$M_{\bar{R}_1} = [\bar{c}_{ij}]$

$M_{R_1 - R_2} = M_{R_1 \cap \bar{R}_2} = [c_{ij} \wedge \bar{d}_{ij}]$

2) Composition

$R = \{(a, b) | a \in A, b \in B, aRb\}$, $S = \{(b, c) | b \in B, c \in C, bSc\}$

The *composite of R and S*: $S \circ R$

$S \circ R = \{(a, c) | a \in A \wedge c \in C \wedge \exists b(b \in B \wedge aRb \wedge bSc\}$

How to computer *S*o*R*?

(1) Using the definition directly

(2) Using the connection matrix

$\mathbf{M}_R = [r_{ij}]_{m \times n}, \mathbf{M}_S = [s_{jk}]_{n \times l}$

$\mathbf{M}_{S \circ R} = \mathbf{M}_R \cdot \mathbf{M}_S = [w_{ik}]_{m \times l}$, $w_{ik} = \bigvee_{j=1}^{n} (r_{ij} \wedge s_{jk})$

【Definition】Let *R* be a relation on the set A. The powers $R^n, n = 1,2,3, \cdots$, are defined recursively by $R^1 = R, and\ R^{n+1} = R^n \circ R$

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

3) Inverse relation

$R = \{(a, b) | a \in A, b \in B, aRb\}$

The *inverse relation* from *B* to *A:* $R^{-1}(R^c)$ $\{(b, a) | (a, b) \in R, a \in A, b \in B\}$

How to get R^{-1}?

(1) Using the definition directly

(2) Reverse all the arcs in the digraph representation of *R*

(3) Take the transpose $M_R{}^T$ of the connection matrix M_R of *R*.

4) The properties of relation operations

Suppose that *R*, *S* are the relations from *A* to *B*, *T* is the relation from *B* to *C*, *P* is the relation from *C* to *D*, then

(1) $(R \cup S)^{-1} = R^{-1} \cup S^{-1}$

(2) $(R \cap S)^{-1} = R^{-1} \cap S^{-1}$

(3) $(\overline{R})^{-1} = \overline{R^{-1}}$

(4) $(R - S)^{-1} = R^{-1} - S^{-1}$

(5) $(A \times B)^{-1} = B \times A$

(6) $\overline{R} = A \times B - R$

(7) $(S \circ T)^{-1} = T^{-1} \circ S^{-1}$

(8) $(R \circ T) \circ P = R \circ (T \circ P)$

(9) $(R \cup S) \circ T = R \circ T \cup S \circ T$

9.4 Closures of Relations

【Definition】*The closure of a relation* R with respect to property P is the relation S with property P containing R such that S is a subset of every relation with property P containing R. (The smallest relation with property P containing R)

1. Reflexive Closure

【Theorem】Let R be a relation on A. The *reflexive closure* of R, denoted by $r(R)$, is $R \cup I_A$

The *diagonal relation* on A $I_A = \{(x, x) | x \in A\}$

【Corollary】$R = R \cup I_A \Leftrightarrow$ R is a reflexive relation .

Given R, how to obtain its reflexive closure?

- Add to R all ordered pairs of the form (a, a) with $a \in A$, not in R
- Add loops to all vertices on the digraph representation of R.
- Put 1's on the diagonal of the connection matrix of R.

2. Symmetric Closure

【Theorem】Let R be a relation on A. The *symmetric closure* of R, denoted by $s(R)$, *is* $R \cup R^{-1}$

Note:

- Add an edge from x to y whenever this edge is not already in directed graph but the edge from y to x is.
- Add all ordered pairs of the form (b,a) where (a,b) is in the relation, that are not already in R.
- $M_{s(R)} = M_R \vee {M_R}^T$

【Corollary】$R = R \cup R^{-1}$ $\Leftrightarrow R$ is a symmetric relation .

3. Transitive Closure

- *A path of length n in a digraph G:* A sequence of edges $(x_0, x_1),(x_1, x_2),. . . ,(x_{n-1}, x_n)$ Notation: $x_0, x_1, x_2,. . . , x_{n-1}, x_n$.
- *Cycle or circuit*: If $x_0 = x_n$ $(n \geqq 1)$
- *There is a path of length n from a to b in R:* $\exists a, x_1, x_2,. . . , x_{n-1}, b$ *such that* $(a, x_1) \in R$, $(x_1, x_2) \in R$, ..., $(x_{n-1}, b) \in R$

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

【Definition】 *The connectivity relation* denoted by R^*, is the set of ordered pairs (a, b) such that there is a path (in R) from a to b: $R^* = \bigcup_{n=1}^{\infty} R^n$

【Theorem】 $t(R) = R^*$.

【Theorem】 If $|A| = n$, then any path of length $> n$ must contain a cycle.

【Corollary】 If $|A| = n$, then $t(R) = R* = R \cup R^2 \cup \cdots \cup R^n$

【Corollary】 Let M_R be the zero-one matrix of the relation R on a set with n elements. The zero-one matrix of the transitive closure is $M_{t(R)} = M_R \vee M_R^{[2]} \vee \cdots \vee M_R^{[n]}$

Warshall's Algorithm

Warshall's algorithm is based on the construction of a sequence of zero-one matrices, such as $\boldsymbol{W_0, W_1, W_2, \cdots, W_n}$

$W_0 = M_R$

$W_k = [w_{ij}{}^{(k)}]$

$$w_{ij}{}^{(k)} = \begin{cases} 1 & \text{If there is a path from } v_i \text{ to } v_j \text{ such that all the interior vertices of this path} \\ & \qquad\qquad\qquad \text{are in the set } \{v_1, v_2, \ldots, v_k\} \\ 0 & \text{otherwise} \end{cases}$$

$W_n = M_{t(R)}$

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

$W := M_R = [w_{ij}]_{n \times n}$
for *k* :=1 to *n*
begin
  for *i* :=1 to *n*
  begin
    for *j* :=1 to *n*
      $w_{ij} = w_{ij} \vee (w_{ik} \wedge w_{kj})$;
  end
end{ $W = [w_{ij}]$ **is** $M_{t(R)}$ }

The complexity of algorithm: $2n^3$

9.5 Equivalence Relations

1. Equivalence relations and equivalence classes

【Definition】A relation *R* on a set *A* is an *equivalence relation* if *R* is

- reflexive
- symmetric
- transitive

Terminologies:

- *a and b are equivalent*: *R* is an equivalence relation, and $(a, b) \in R$
  Notation: $a \sim b$
- *the equivalence class of x*: The set of all elements that are related to an element *x* of *A*
  Notation: $[x]_R$ *or* $[x]$
  a representative of the equivalence class $[x]_R$: $b \in [x]_R$

【 Theorem 1】Let *R* be an equivalence relation on a set *A*. The following statements are equivalent:

(1) aRb

(2) $[a] = [b]$

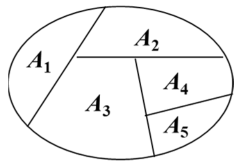
(3) $[a] \cap [b] \neq \phi$

2. Equivalence relations and partitions

【Definition】A *partition* of set *A* is a collection of *disjoint nonempty* subsets of *A* that have *A* as their union.

Let $\{A_i | i \in I\}$ be a collection of subsets of *A*. Then the collection forms a *partition* of *A* if and only if

- $A_i \neq \varphi$ for $i \in I$ (I is an index set)
- $A_i \cap A_j = \varphi$, when $i \neq j$
- $\forall a \in A, \exists i$ such that $a \in A_i (i = 1,2,\cdots)$



Notation: $pr(A) = \{A_i | i \in I\}$

【Theorem 2】 Let *R* be an equivalence relation on a set *A*. Then the equivalence classes of *R* form a partition of *A*. Conversely, given a partition $\{A_i | i \in I\}$ of the set *A*, there is an equivalence relation *R* that has the sets $A_i, i \in I$, as its equivalence classes.

(an equivalence relation on a set *A* ↔ a partition of *A*)

3. The operations of equivalence relations

【Theorem 3】 If R_1, R_2 are equivalence relations on *A*, then $R_1 \cap R_2$ is an equivalence relation on *A*.

【Theorem 4】 If R_1, R_2 are equivalence relations on *A*, then $R_1 \cup R_2$ is a reflexive and symmetric relation on A.

【Theorem 5】 If R_1, R_2 are equivalence relations on *A*, then $(R_1 \cup R_2)*$ is an equivalence relation on A.

9.6 Partial Orderings

1. Partial Orderings

【Definition】Let *R* be a relation on a set *S*. Then *R* is a *partial ordering or partial order* if *R* is

- reflexive
- antisymmetric
- transitive

Notation: (*S, R*) ---- partially ordered set or a *poset*

Example:

(1) $R_1 = \{(a,b) \mid a \leq b, a, b \in Z\}$ $(Z, \leq)$

(2) $R_2 = \{(a,b) \mid a \mid b, a, b \in Z^+\}$ $(Z^+, |)$

(3) $R_3 = \{(s_1, s_2) \mid s_1 \subseteq s_2, s_1, s_2 \in P(S)\}$ $(P(S), \subseteq)$

Notation:

$$a \preccurlyeq b \quad ----(S, R) \quad is \quad a \quad poset, \quad (a,b) \in R$$

【Definition】*comparable/ incomparable*

The elements *a* and *b* of a poset (*S*,≼) are *comparable* if either *a* ≼ *b* or *b* ≼ *a*. When *a* and *b* are elements of *S* so that neither *a* ≼ *b* nor *b* ≼ *a*, then *a* and *b* are called i*ncomparable*.

【Definition】If (*S*,≼) is a poset and every two elements of *S* are comparable, *S* is called a *totally ordered* or *linearly ordered set*, ≼ is called a *total order* or *linear order*. In this case (*S*,≼) is also called a *chain*.

2. Lexicographic Order

【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), that is, (a_1, a_2) ≺ (b_1,b_2), either if $a_1 \prec_1 b_1$ or if $a_1 = b_1$ and $a_2 \prec_2 b_2$.

- A *lexicographic ordering* on the Cartesian product of two posets is a partial ordering.

lexicographic ordering of string

The string $a_1a_2 \cdots a_m$ is less then $b_1b_2 \cdots b_n$ if and only if

$(a_1, a_2, \cdots, a_t) < (b_1, b_2, \cdots, b_t), or$

$(a_1, a_2, \cdots, a_t) = (b_1, b_2, \cdots, b_t)\ and\ m < n$

Where $t = min(m, n)$

3. Hasse Diagrams

Hasse Diagrams ---- A method used to represent a partial ordering

To construct a Hasse diagram:

1) Construct a digraph representation of the poset (*A, R*) so that all arcs are pointed upward (except the loops).
2) Eliminate all loops
3) Eliminate all arcs that are redundant because of transitivity
4) Eliminate the arrows at the ends of arcs since everything points up.

4. chain and antichain

【Definition】$(A, \leq)$ is a poset. $B \subseteq A$，if $(B, \leq)$ is a totally ordered set，then *B* is called a *chain* of $(A, \leq)$.

$B \subseteq A$，if $\forall a, b \in B(a \neq b), (a, b) \notin R, (b, a) \notin R$ then *B* is called a *antichain* of $(A, \leq)$.

5. Maximal and Minimal Elements

【Definition】Let $(A, \leq)$ be a poset. $a \in A$, then *a* is a *maximal element* if there does not exist an element *b* in *A* such that $a \prec b$.

Similar defintion for a *minimal element.*

Note:

1. Maximal and minimal elements are the "top" and "bottom" elements in the Hasse diagram.
2. There can be more than one minimal and maximal element in a poset.

6. Greatest and Least Element

【Definition】Let $(A, \leq)$ be a poset. Then an element *a* in *A* is a *greatest element* of *A* if $b \leq a$ for every *b* in *A*, and *a* is a *least element* of *A* if $a \leq b$ for every *b* in *A*.

【Theorem】 The greatest and least element of the poset $(A, \leq)$ are unique when they exist.

7. Upper and Lower Bounds

【Definition】Let *A* be a subset of *S* in the poset $(S, \leq)$. If there exists an element *a* in *S* such that $b \leq a$ for all *b* in *A*, then *a* is called an *upper bound* of *A*.

Similar definition for *lower bounds*.

8. Least Upper and Greatest Lower Bounds

【Definition】If *a* is an upper bound of the subset *A* which is less than every other upper bound of *A,* then *a* is the *least upper bound*, denoted by lub(*A*).

Similarly for the *greatest lower bound*, glb(*A*).

9. Well-ordered Sets

【Definition】A poset *(A, R)* is *well-ordered set* if every nonempty subset of *A* has a least element.

Note: A well-ordered set is a totally ordered set.

10. Lattices

【Definition】A poset is called a *lattice* if every pair of elements has a lub and a glb.

11. Topological Sorting

We impose a total ordering ≤ on a poset (A,R) *compatible* with the partial order if $a \leq b$ whenever aRb.

Constructing a compatible total ordering from a partial ordering is called *topological sorting*.

【Lemma 1】 Every finite nonempty poset (S, ≤) has a minimal element.

Algorithm: To sort a poset (S, R).

- Select a (any) minimal element and put it in the list. Delete it from S.
- Continue until all elements appear in the list (and S is void).