DISCRETE MATHEMATICS
（离散数学）

云南大学数学系

李 建 平

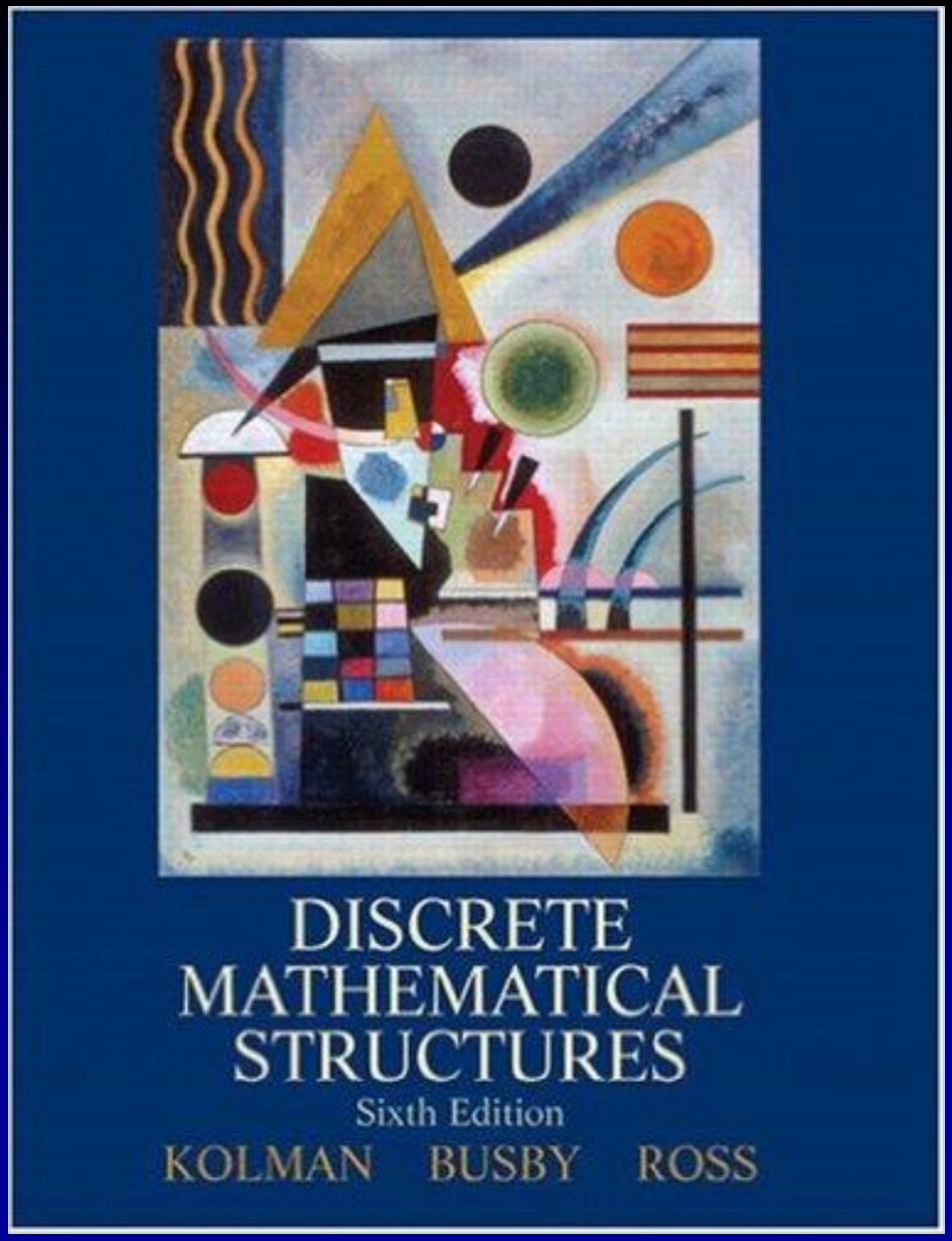
教材：Discrete Mathematical Structures

（第六版）

作者：B. Kolman，R.C. Busby and S.C. Ross

出版商：PRENTICE HALL,Inc.,

高等教育出版社影印版


DISCRETE
MATHEMATICAL
STRUCTURES
Sixth Edition
KOLMAN BUSBY ROSS

Contents

1. Fundamentals
2. Logic
3. Counting
4. Relations and Digraphs
5. Functions
6. Order Relations and Structures
7. Trees

8. Topics in Graph Theory
9 . Semigroups and Groups
10. Languages and Finite-State Machines
11. Groups and Coding

Chapter 1 FUNDAMENTALS

1.1 SETS AND SUBSETS

- Sets

A **set** is any well-defined collection of objects called the elements or members of the set.

One way of describing a set that has a finite number of elements is by listing the elements of the set between braces.

The **order** in which the elements of a set are listed is not important.

We use uppercase letters such as A,B,C to denote sets, and lowercase letters such as a, b, c, x, y, z, t to denote the members (or elements) of sets.

We indicate the fact that x is an element of the set A by writing $x \in A$, and we indicate the fact that x is not an element of A by writing $x \notin A$.

Another useful way to define a set is by specifying a **property** that the elements of the set have in common. We use the notation P (x) to denote a sentence or statement P concerning the variable object x. The set denoted by P (x), written {x| P (x)}, is just the collection of all objects for which P is sensible and true.

We introduce here several sets and their

Z^+

notations that will be used through this class.

Z^+={x| x is a positive integer}.

N ={x| x is a positive integer or zero}.

Z ={x| x is an integer}.

Q={x| x is a rational number}.

R={x| x is a real number}.

The set that has no elements in it is denoted either by { } or the symbol Ø, and is called the **empty set.**

We say two sets A and B are **equal** if they have the same elements, and we write A=B.

- Subsets

If every element of A is also an element of B, i.e., if whenever $x \in A$, then $x \in B$, we say that A is a **subset** of B or that A is **contained in** B, and we write $A \subseteq B$. If A is not a subset of B, we write $A \not\subset B$.

Diagrams, such as those in Figure 1.1, which are used to show relationships between sets, and called Venn diagrams (**文氏图**) after the British logician John Venn.

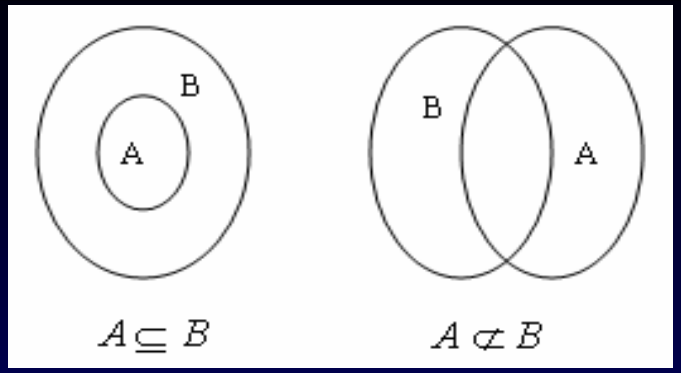


Figure 1.1

For any set A, since there are no elements of Ø that are not in A, we have Ø $\subseteq$ A.

It is easy to see that A = B if and only if A $\subseteq$ B and B $\subseteq$ A.

"universal set" U, other set mentioned in the discussion will automatically be assumed to be a subset of U.

In Venn diagrams, the universal set U will be denoted by a rectangle (矩形), while sets within U will be denoted by circles as shown in Figure 1.2.

A set A is called **finite** if it has n distinct elements, where $n \in N$, n is called the **cardinality** (**基数**) of A and is denoted by |A|.

If A is a set, then the set of all subsets of A is called the **power set**（**幂集**）of A and is denoted by P(A).

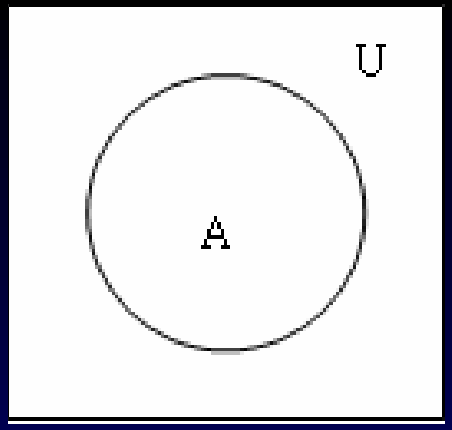


Figure 1.2

1.2 OPERATIONS ON SETS

If A and B are sets, we define their **union**（并集） as the set consisting of all elements that belong to A or B and denote it by $A \cup B$.

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}.$$

Observe that $x \in A \cup B$ if $x \in A$ or $x \in B$ or x belongs to both A and B.

We can illustrate the union of two sets with a Venn diagram as follows. If A and B are the sets in Figure 1.5(a), then $A \cup B$ is the set represented by the shaded region in Figure1.5(b).

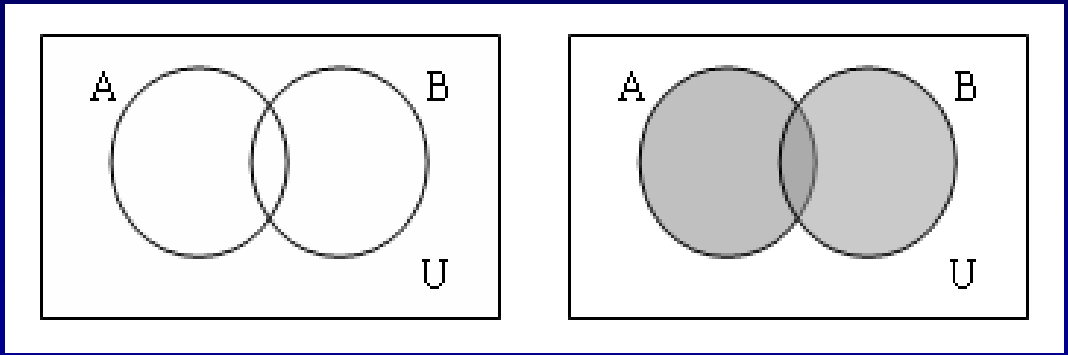


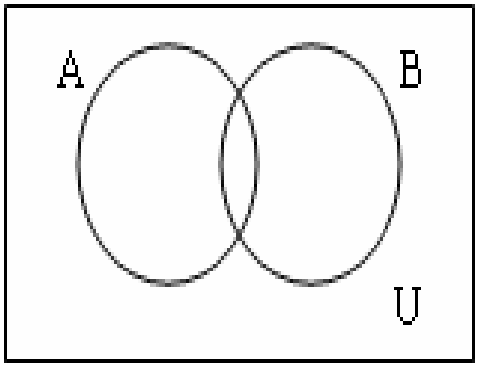
Figure 1.5 (a) (b) $A \cup B$

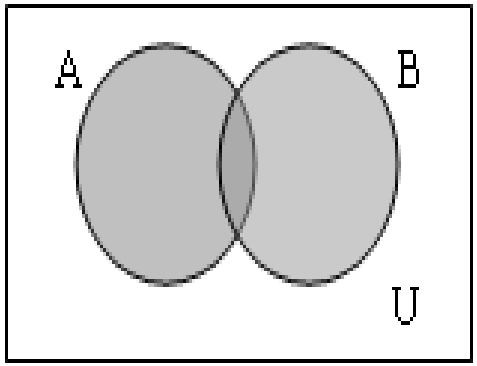
If A and B are sets, we define their **intersection** as the set consisting of all elements that belong to both A and B and denote it by A∩B. (**称 A∩B为A和B的交集合**). Thus

$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}.$$

Two sets that have no common elements, such as B and C in Example 2, are called disjoint sets.

We can illustrate the interaction of two sets by a Venn diagram as follows. If A and B are the sets given in Figure 1.6(a), then A∩B is the set represented by the shaded region in Figure 1.6(b).





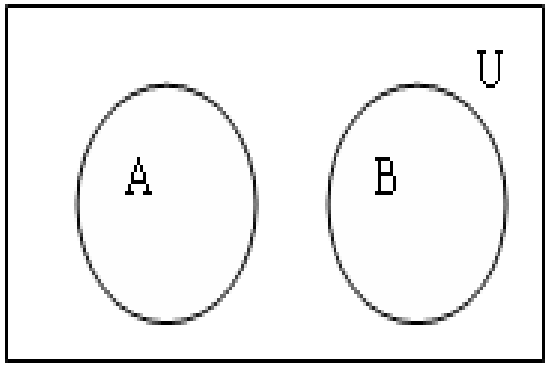


Figure 1.6 (a) (b) $A \cap B$ Figure 1.7

The operations of union and interaction can be defined for three or more sets in an obvious manner:

$$A \cup B \cup C = \{x \mid x \in A \text{ or } x \in B \text{ or } x \in C\}$$

and

$$A \cap B \cap C = \{x \mid x \in A \text{ and } x \in B \text{ and } x \in C\}$$

In general, if $A_1, A_2, \ldots, A_n$ are subsets of U, then $A_1 \cup A_2 \cup \ldots \cup A_n$ will be denoted by $\bigcup_{k=1}^{n} A_k$ and $A_1 \cap A_2 \cap \ldots \cap A_n$ will be denoted by $\bigcap_{k=1}^{n} A_k$

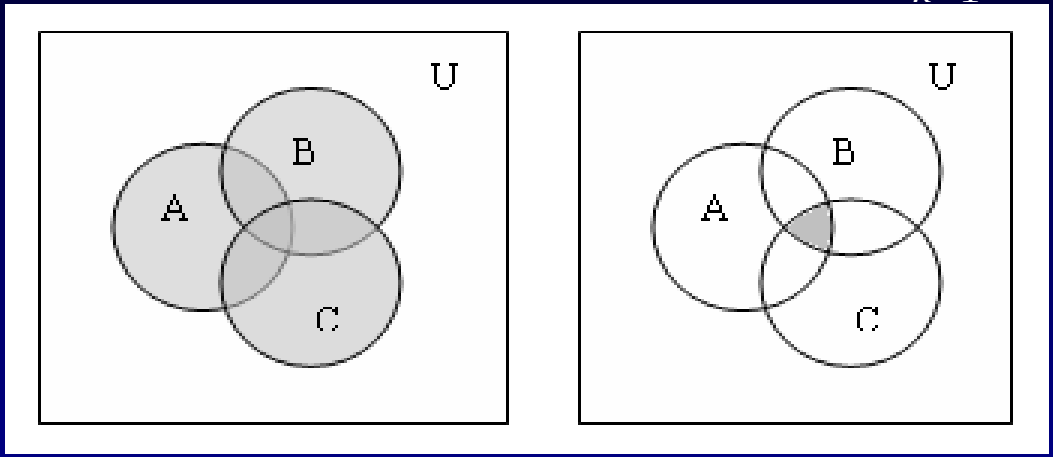


Figure 1.8 (b) $A \cup B \cup C$ (c) $A \cap B \cap C$

If A and B are two sets, we define the **complement** of B with respect to A (**B相对A的补集合**)

as the set of all elements that belong to A but not to B, and we denote it by A-B.

$$A\text{-}B=\{x \mid x \in A \text{ and } x \notin B\}.$$

If U is a universal set containing A, then U-A is called the **complement** of A and is denoted by $\overline{A}$. Thus $\overline{A}=\{x \mid x \notin A\}$.

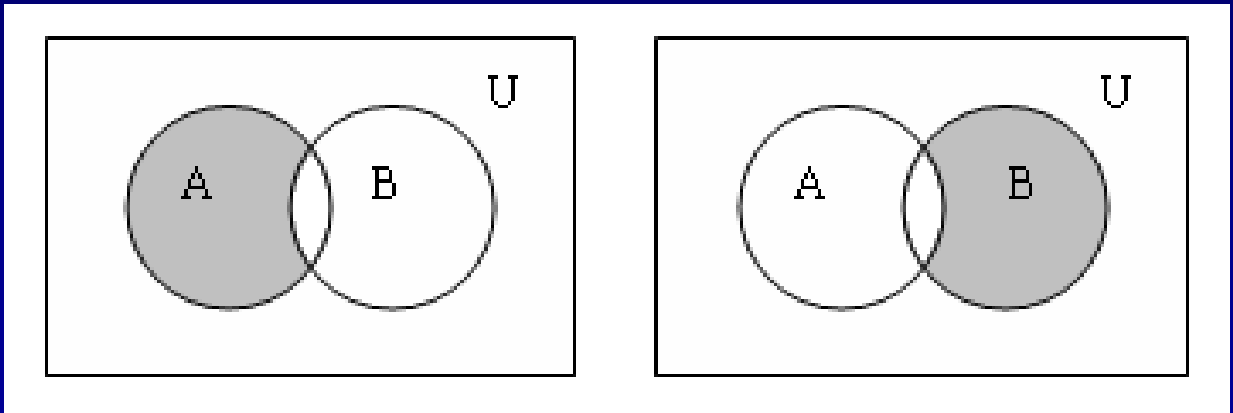


Figure 1.9 (b) (c)

If A and B are two sets, we define their **symmetric difference** (**对称差**) as the set of all elements that belong to A or to B, but not to both A and B, and we denote it by $A \oplus B$. Thus

$$A \oplus B = \{x \mid (x \in A \text{ and } x \notin B) \text{ or } (x \in B \text{ and } x \notin A)\}.$$

If A and B are as indicated in Figure 1.11(a), their symmetric difference is the shaded region shown in Figure 1.11(b).

It is easy to see that

$$A \oplus B = (A - B) \cup (B - A).$$

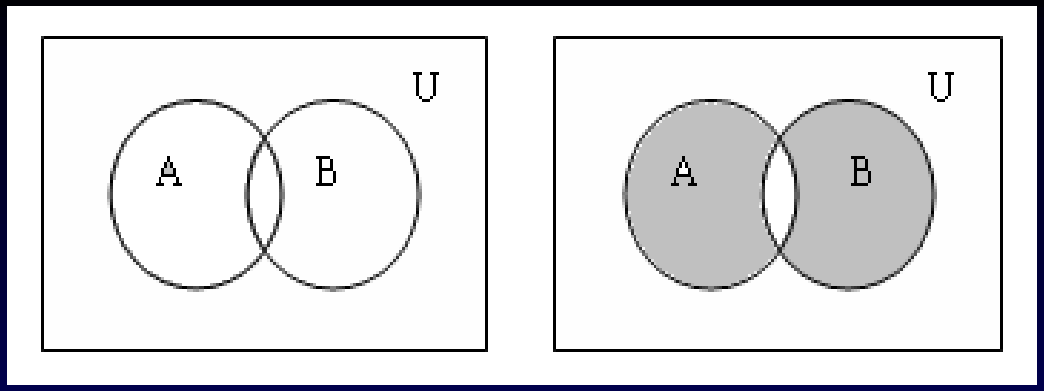


Figure 1.11 (a) (b) A⊕B

- Algebraic Properties of Set Operations

Theorem 1: The operations defined on sets satisfy the following properties:

Communicative Properties (**交换性**)

1. $A \cup B = B \cup A$
2. $A \cap B = B \cap A$

Associative Properties (结合性)

3. $A \cup (B \cup C)=(A \cup B) \cup C$

4. $A \cap (B \cap C)=(A \cap B) \cap C$

Distributive Properties (分配性)

5. $A \cap (B \cup C)=(A \cap B) \cup (A \cap C)$

6. $A \cup (B \cap C)=(A \cup B) \cap (A \cup C)$

Idempotent Properties (幂等性)

7. $A \cup A=A$

8. $A \cap A=A$

Properties of the Complement (互补性)

9. $(\overline{\overline{A}})=A$

10. $A \cup \bar{A} = U$
11. $A \cap \bar{A} = \varnothing$
12. $\bar{\varnothing} = U$
13. $\bar{U} = \varnothing$
14. $\overline{A \cup B} = \bar{A} \cap \bar{B}$
15. $\overline{A \cap B} = \bar{A} \cup \bar{B}$

Properties 14 and 15 are known as De Morgan's laws.

Properties of a Universal Set (全集性)

16. $A \cup U = U$
17. $A \cap U = A$

Properties of the Empty Set (空集性)

18. $A \cup \varnothing = A$

19. $A \cap \varnothing = \varnothing$

- The Addition Principle (加法原理)

Suppose now that A and B are finite subsets of a universal set U. If A and B are disjoint sets, that is, if $A \cap B = \varnothing$, then each element of $A \cup B$ appears in either A or B, but not in both; therefore, $|A \cup B| = |A| + |B|$. If A and B overlap, then elements in $A \cap B$ belong to both sets, and then sum $|A|+|B|$ counts these elements twice. To correct for this

double counting, we subtract $|A \cap B|$.

Theorem 2: If A and B are finite sets, then

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

- The Addition Principle for disjoint Sets

If a task T_1 can be performed in exactly n ways, and a different task T_2 can be performed in exactly m ways, then the number of ways of performing task T_1 or task T_2 is n + m.

Theorem 3: Let A, B, and C be finite sets. Then

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|.$$

1.3 SEQUENCES

A **sequence** (**序列**) is simply a list of objects in a definite order: a first element, second element, third element, and so on. The list may stop after n steps, $n \in N$, or it may go on forever. In the first case we say that the sequence is **finite**, and in the second case we say that it is **infinite**. The elements may all be different, or some may be repeated.

Two kinds of formulas are commonly used to describe sequences. In Example 2, a natural description of the sequence is that successive terms are produced by adding 5 to the previous

term. If we use a subscript to indicate a term's position in the sequence, we can describe the sequence in Example 2 as $a_1=3$, $a_n=a_{n-1}+5$.

A formula, like this one, that refers to previous terms to define the next term is called **recursive**（**递归**）. Every recursive formula must include a starting place.

Sequences of letters or other symbols, written without the commas, are also referred to as **Strings**（**串**）.

The **set corresponding to a sequence** is simply

the set of all distinct elements in the sequence.

The idea of a sequence is important in computer science, where a sequence is sometimes called a **linear array or list （线性数组，列表）**. If we have a sequence S: s_1,s_2,s_3, …, we think of all the elements of S as completely determined.

An array, on the other hand, may be viewed as a sequence of positions, which we represent in Figure 1.16 as boxes.

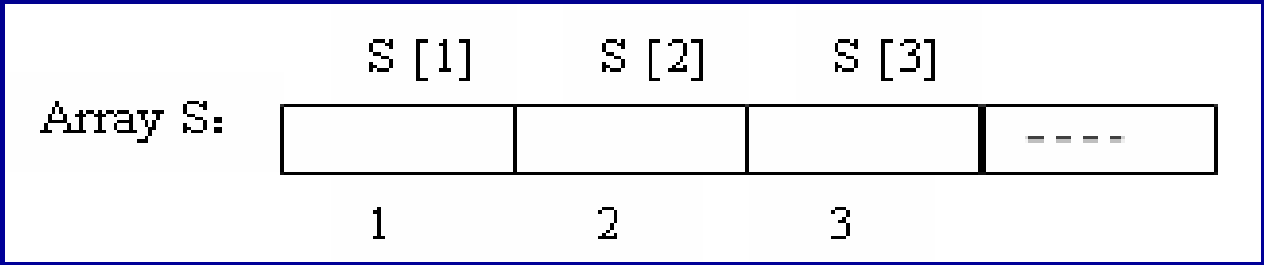


Figure 1.16

The element assigned to position n will be denoted by S[n], and the sequence S[1], S[2],…will be called the **sequence of values** of the array S.

- Characteristic Functions

If A is a subset of a universal set U, the characteristic function f_A （**特征函数**）of A is defined for each $x \in U$ as follows:

$$f_A = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A. \end{cases}$$

Theorem 1: Characteristic functions of subsets satisfy the following properties:

(a) $f_{A\cap B}=f_A f_B$; i.e., $f_{A\cap B}(x)=f_A(x)f_B(x)$ for all x.

(b) $f_{A\cup B}=f_A+f_B-f_A f_B$; i.e., $f_{A\cup B}(x)= f_A(x)+f_B(x)-f_A(x)f_B(x)$ for all x.

(c) $f_{A\oplus B}=f_A+f_B-2f_A f_B$; i.e., $f_{A\oplus B}(x)= f_A(x)+f_B(x)-2f_A(x)f_B(x)$ for all x.

- Computer Representation of Sets and Subsets

To represent a set in a computer, the elements of the set must be arranged in a sequence. When we list the set A={a, b, c, …, r}, we normally assume no particular ordering of the elements in A. Let us

identify for now the set A with the sequence a, b, c, …, r.

When a universal set U is finite ,say $U=\{x_1,x_2,\ldots, x_n\}$, and A is a subset of U, then the characteristic function assigns 1 to an element that belongs to A and 0 to an element that does not belong to A. Thus f_A can be represented by a sequence of 0's and 1's of length n.

Any set with n elements can be arranged in a sequence of length n, so each of its subsets corresponds to a sequence of zeros and ones of length n, representing the characteristic function of

that subset. This fact allows us to represent a universal set in a computer as an array A of length n.

A set is called **countable** （**可数的**）if it is the set corresponding to some sequence. Informally, this means that the members of the set can be arranged in a list, with a first, second, third, …, element, and the set can therefore be “counted.” A set that is not countable is called **uncountable.**

- Strings and Regular Expressions

Given a set A, we can construct the set A* consisting of all finite sequences of elements of A. The set A is called an **alphabet**（**字母表**）, and the finite sequences in A* are called **words**（**字**） from A, or sometimes strings from A. We assume that A*contains the **empty sequence** or **empty string**, we denote this string by ∧.

If $w_1=s_1s_2s_3\ldots s_n$ and $w_2=t_1t_2t_3\ldots t_k$ are elements of A* for some set A, we define the **catenation**（**联结**）of w_1 and w_2 as the sequence $s_1s_2s_3\ldots s_nt_1t_2t_3\ldots t_k$. The **catenation** of w_1 with w_2 is written as $w_1 \cdot w_2$ or w_1w_2, and is another element of A*. Note that if w belong to A*, then $w\cdot\wedge=w$ and $\wedge\cdot w=w$.

A **regular expression over A（A上的正规表达）** is a string constructed from the elements of A and the symbols (,), ∨, *, ∧, according to the following definition.

RE1. The symbol ∧ is a regular expression.

RE2. If x∈A, the symbol x is a regular expression.

RE3. If α and β are regular expressions, then the expression α·β （simply, αβ）is regular.

RE4. Ifα andβ are regular expressions, then the expression (α ∨β) is regular（ α 或者β ）.

RE5. If α is a regular expression, then the expression (α)* is regular.

The definition is recursive.

Associated with each regular expression over the set A, there is a corresponding subset of A*. Such sets are called **regular subsets** of A* or just **regular sets** if no reference to A is needed. We use the following correspondence rules.

1. The expression $\wedge$ corresponds to the set $\{\wedge\}$, where $\wedge$ is the empty string in A*.

2. If $x \in A$, then the regular expression x corresponding to the set {x}.

3. If α and β are regular expressions corresponding to the subsets M and N of A*, then corresponds to $M \cdot N = \{s \cdot t | s \in M \text{ and } t \in N\}$. Thus $M \cdot N$ is the set of all catenations of strings in M with strings in N.

4. If the regular expressions α and β correspond to the subsets M and N of A*, then $(\alpha \vee \beta)$ corresponds to $M \cup N$.

5. If the regular expression corresponds to the subset M of A*, then $(\alpha)^*$ corresponds to the set M*. Note that M is a set of strings from A.

Elements from M* are finite sequences of such strings, and thus may themselves be interpreted as strings from M* are finite sequences of such strings, and thus may themselves be interpreted as stings from A. Note also that we always have $\wedge \in M^*$.

1.4 Division in the Integers

If m is a **multiple** of n, say m=qn, then we can write m=qn+r, where r is 0. On the other hand (as shown in Figure 1.19), if m is not a multiple of n, we let qn be the first multiple of n lying to the left of m and let r be m-qn. Then r is the distance from qn to m, so clearly 0<r<n, and again we have m=qn+r.

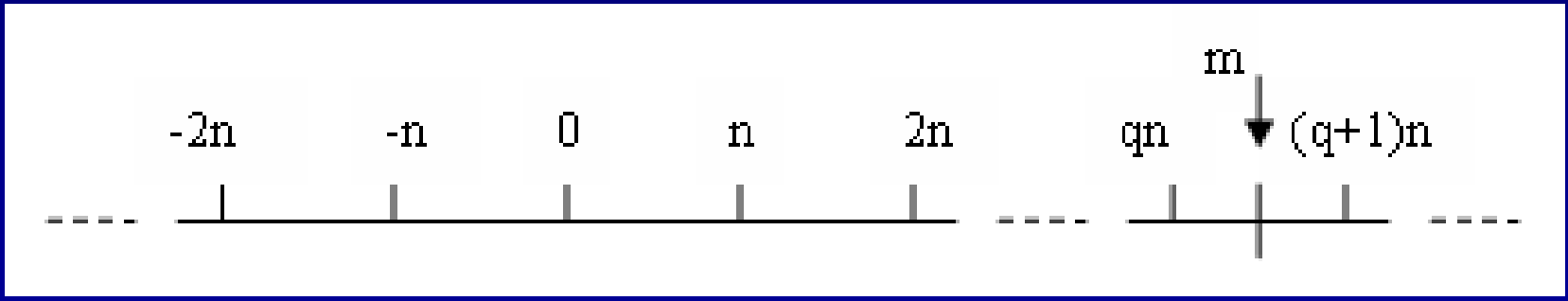


Figure 1.19

Theorem 1: If n and m are integers and $n>0$, we can write $m=qn+r$ for integers q and r with $0\le r<n$.

If the r in Theorem 1 is zero, so that m is a multiple of n, we write $n|m$, which is read "n divides m." If $n|m$, then $m=qn$ and $n\le m$. If m is not a multiple of n, we write $n \nmid m$, which is read "n does not divide m."

Theorem 2: Let a, b, and c be integers.

(a) If $a|b$ and $a|c$, then $a|(b+c)$.

(b) If $a|b$ and $a|c$, where $b>c$, then $a|(b-c)$.

(c) If $a|b$ or $a|c$, then $a|bc$.

(d) If $a|b$ and $b|c$, then $a|c$.

Definition 1 An algorithm is complete steps necessary to perform a task or computation. The steps in an algorithm may be general descriptions, leaving much detail to be filled in, or they may be totally precise descriptions of every detail.

If some steps were all included, the algorithms would be much more detailed, but long. If the added detail is necessary, one possible solution is to group collections of related steps into other algorithm that is called subroutine and simply refer to these subroutines at appropriate points in the main algorithm.

A number p>1 in Z^+ is called **prime**（**素数、质数**）if the only positive integers that divide p are p and 1.

It is easy to write a set of steps, or an **algorithm**, to determine if a positive integer n>1 is a prime number.

ALGORITHM: PRIME

Input: an integer N>1

Output: whether an integer N is prime or not.

Begin

Step 1: Check whether N is 2. If so, N is prime. If not, proceed to

Step 2: Check whether 2|N. If so, N is not prime; otherwise, proceed to

Step 3: Compute the largest integer K≤ $\sqrt{N}$, where $\sqrt{N}$ is the arithmetic square root of N. Then

Step 4: Check whether D|N, where D is any odd number such that 1<D≤K. If D|N, then N is not prime; otherwise, N is prime.

End

Theorem 3: Every positive integer n>1 can be written uniquely as $p_1^{k1}p_2^{k2}\ldots p_s^{ks}$, where $p_1<p_2<\cdots<p_s$ are distinct primes that divide n and the k's are positive integers giving the number of times each prime occurs as a factor of n.

- Greatest Common Divisor （**最大公因数**）

If a, b, and k are in Z^+, and k|a and k|b, we say that k is a **common divisor** of a and b. If d is the largest such k, d is called the **greatest common divisor**, or GCD, of a and b, and we write

d=GCD(a,b).

Theorem 4: If d is GCD(a,b), then

(a) d=sa+tb for some integers s and t. (These are not necessarily positive.)

(b) If c is any other common divisor of a and b, then c|d.

From the definition of greatest common divisor and Theorem 4(b), we have the following result: Let a, b, and d be in Z^+. The integer d is the greatest common divisor of a and b if and only if

(a) d|a and d|b.

(b) Whenever c|a and c|b, then c|d.

We now present a procedure, called the **Euclidean algorithm**, to find GCD(a,b).

We now continue using Theorem 1 as follows:

$$\begin{aligned}
&\text{divide } a \text{ by } b: && a=k_1b+r_1 && 0\le r_1<b\\
&\text{divide } b \text{ by } r_1: && b=k_2r_1+r_2 && 0\le r_2<r_1\\
&\text{divide } r_1 \text{ by } r_2: && r_1=k_3r_2+r_3 && 0\le r_3<r_2\\
&\text{divide } r_2 \text{ by } r_3: && b=k_4r_3+r_4 && 0\le r_4<r_3 \qquad (2)\\
&\qquad\vdots && \qquad\vdots && \qquad\vdots\\
&\text{divide } r_{n-2} \text{ by } r_{n-1}: && r_{n-2}=k_nr_{n-1}+r_n && 0\le r_n<r_1\\
&\text{divide } r_{n-1} \text{ by } r_n: && r_{n-1}=k_{n+1}r_n+r_{n+1} && 0\le r_{n+1}<r_n.
\end{aligned}$$

Since $a>b>r_1>r_2>r_3>r_4>\ldots$, the remainder will eventually become zero, so at some point we have

$r_{n+1}=0$.

Upon continuing, we have

$GCD(a,b)=GCD(b,r_1)=GCD(r_1,r_2)=\cdots=GCD(r_{n-1},r_n)$.

We observed that if $d=GCD(a,b)$, we can find integers s and t such that $d=sa+tb$. Solve the next-to-last equation in (2) for r_n:

$$r_n=r_{n-2}-k_n r_{n-1}. \qquad (3)$$

Now solve the second-to-last equation in (2), $r_{n-3}=k_{n-1}r_{n-2}+r_{n-1}$ for r_{n-1}:

$$r_{n-1}=r_{n-3}-k_{n-1}r_{n-2}$$

and substitute this expression in (3):

$$r_n=r_{n-2}-k_n[r_{n-3}-k_{n-1}r_{n-2}].$$

Continue to work up through the equations in (2) and (1), replacing r_i by an expression involving r_{i-1} and r_{i-2}, and finally arriving at an expression involving only a and b.

Theorem 5: If a and b are in Z^+, then

$$GCD(a,b)=GCD(b,b\pm a).$$

- Least Common Multiple (**最小公倍数**)

If a, b, and k are in Z+, and a|k, b|k, we say k is a **common multiple** of a and b. The smallest such k, call it c, is called the **least common multiple**, or LCM, of a and b, and we write c=LCM(a,b).

Theorem 6: If a and b are two positive integers, then $GCD(a,b)\cdot LCM(a,b)=ab$.

Proof: Let $p_1,p_2,\ldots,p_k$ be all the prime factors of either a or b. Then we can write

$a=p_1^{a1}p_2^{a2}\ldots p_k^{ak}$ and $b=p_1^{b1}p_2^{b2}\ldots p_k^{bk}$

where some of the a_i and b_i may be zero. It then follows that

$$GCD(a,b)=p_1^{\min(a1,b1)}p_2^{\min(a2,b2)}\ldots p_k^{\min(ak,bk)}.$$

$$LCM(a,b)=p_1^{\max(a1,b1)}p_2^{\max(a2,b2)}\ldots p_k^{\max(ak,bk)}.$$

Hence

$$\begin{aligned} GCD(a,b)LCM(a,b)&=p_1^{a1+b1}p_2^{a2+b2}\ldots p_k^{ak+bk}\\ &=(p_1^{a1}p_2^{a2}\ldots p_k^{ak})(p_1^{b1}p_2^{b2}\ldots p_k^{bk})\\ &=ab. \end{aligned}$$

- Pseudocode Versions （**谓码形式**）

We use a **pseudocode** language to represent specific contents of an algorithm.

In the pseudocode for the algorithm to determine if an integer is prime, we assume the existence of functions SQR and INT, where SQR(N) returns the greatest integer not exceeding the arithmetic square root of N, saying $\sqrt{N}$, and INT(X) returns the greatest integer not exceeding X.

```
SUBROUTINE PRIME(N)
1. IF(N=2) THEN
    a. PRINT('PRIME')
    b. RETURN
2. ELSE
   a. IF(N/2=INT(N/2) )THEN
        1. PRINT('NOT PRIME')
        2. RETURN
    b. ELSE
        1. FOR D=3 THROUGH SQR(N) BY 2
            a.  IF(N/D=INT(N/D)) THEN
                1. PRINT('NOT PRIME')
                2. RETURN
        2. PRINT('PRIME')
        3. RETURN
```

The following gives a pseudocode program for finding the greatest common divisor of two positive integers.

```
FUNCTION GCD(X,Y)
1. WHILE(X≠Y)
    a. IF(X>Y)THEN
        1. X←X-Y
    b. ELSE
        1. Y←Y-X
2. RETURN(X)
END OF FUNCTION GCD
```

1.5 MATRICES

A **matrix** (**矩阵**)is a rectangular array of numbers arranged in m horizontal **rows** and n vertical **columns**:

$$A=\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

The ith row of A is $[a_{i1}\ a_{i2}\ \dots\ a_{in}]$, $1 \le i \le m$, and the jth column of **A** is $\begin{bmatrix} a_{1j} \\ a_{2j} \\ \dots \\ a_{mj} \end{bmatrix}$, $1 \le j \le n$.

We say that **A** is **m by n**, written $m \times n$. If m=n, we say A is a **square matrix** of order n and that the numbers $a_{11}, a_{22}, \ldots, a_{nn}$ form the **main diagonal** of **A**. We refer to the number a_{ij}, which is in the ith row and jth column of **A** as the i, jth **element of A** or as the (i,j) **entry of A**, and we often write (1) as $\mathbf{A}=[a_{ij}]$ or $\mathbf{A}=[a_{ij}]_{m \times n}$.

A square matrix $\mathbf{A}=[a_{ij}]$ for which every entry off the main diagonal is zero, that is, $a_{ij}=0$ for $i \neq j$, is called a **diagonal matrix**.

If $\mathbf{A}=[a_{ij}]$ and $\mathbf{B}=[b_{ij}]$ are $m \times n$ matrices, then the **sum** of **A** and **B** is the matrix $\mathbf{C}=[c_{ij}]$ defined by

$c_{ij}=a_{ij}+b_{ij}$, $1\le i\le m, 1\le j\le n$.

A matrix all of whose entries are zero is called a **zero matrix** and is denoted by **0**.

Theorem 1:

(a) $\mathbf{A}+\mathbf{B}=\mathbf{B}+\mathbf{A}$.

(b) $(\mathbf{A}+\mathbf{B})+\mathbf{C}=\mathbf{A}+(\mathbf{B}+\mathbf{C})$.

(c) $\mathbf{A}+\mathbf{0}=\mathbf{0}+\mathbf{A}=\mathbf{A}$.

If $\mathbf{A}=[a_{ij}]$ is an $m\times p$ matrix and $\mathbf{B}=[b_{ij}]$ is a $p\times n$ matrix, then the **product** of **A** and **B**, denoted **AB**, is the $m\times n$ matrix $\mathbf{C}=[c_{ij}]$ defined by

$$c_{ij}=a_{i1}b_{1j}+a_{i2}b_{2j}+\ldots+a_{ip}b_{pj} \quad 1\le i\le m,\ 1\le j\le n.$$

The elements a_{i1}, a_{i2}, …, a_{ip} form the ith row of **A**, and the elements b_{1j}, b_{2j}, b_{pj} form the jth column of **B**. The element c_{ij} of **C**=**AB** can be computed in the following way, illustrated in Figure 1.20.

1. Select row i of **A** and column j of **B**, and place them side by side.
2. Multiply corresponding entries and add all the products.

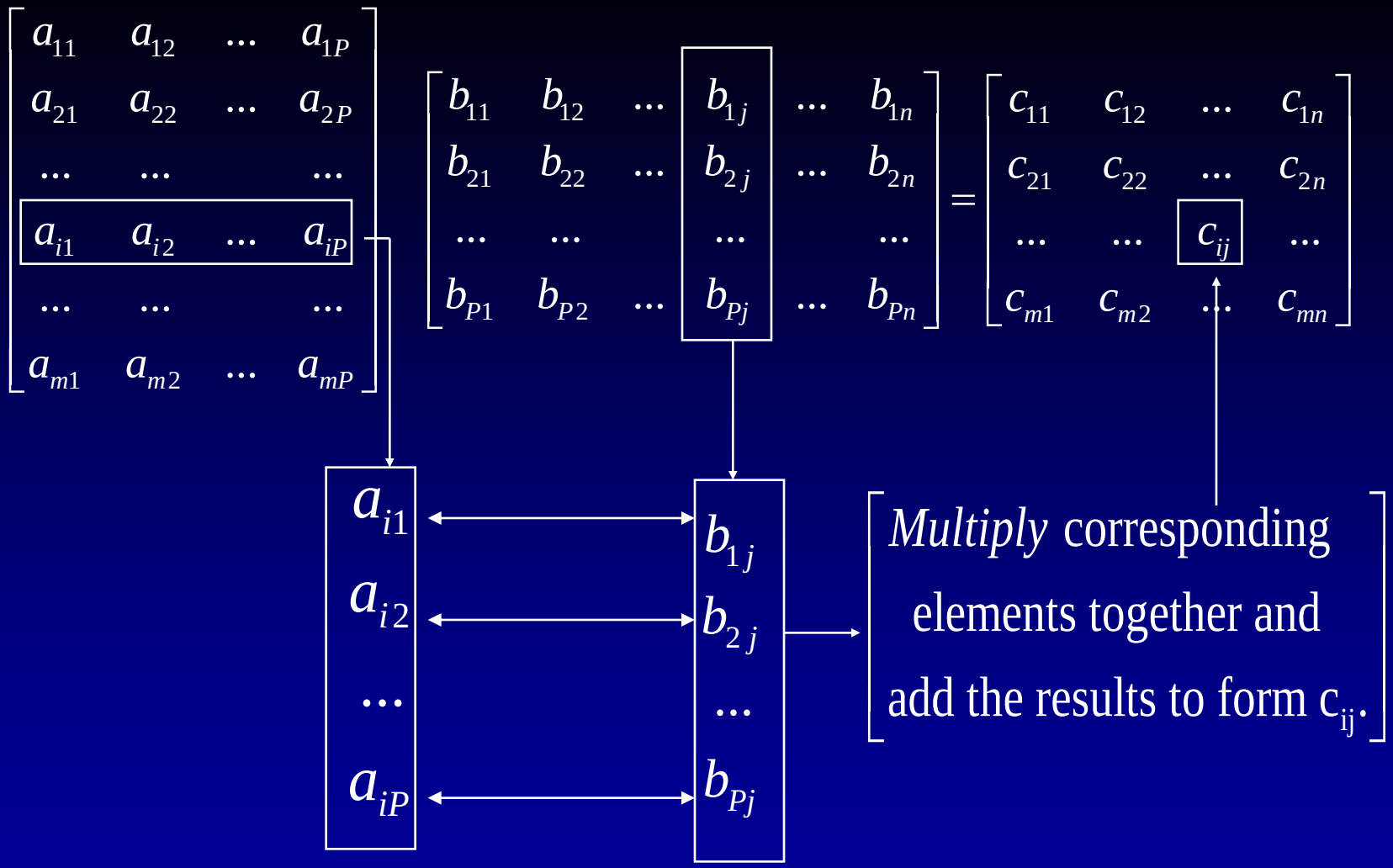

$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1P} \\ a_{21} & a_{22} & \dots & a_{2P} \\ \dots & \dots & & \dots \\ a_{i1} & a_{i2} & \dots & a_{iP} \\ \dots & \dots & & \dots \\ a_{m1} & a_{m2} & \dots & a_{mP} \end{bmatrix}$
$\begin{bmatrix} b_{11} & b_{12} & \dots & b_{1j} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2j} & \dots & b_{2n} \\ \dots & \dots & & \dots & & \dots \\ b_{P1} & b_{P2} & \dots & b_{Pj} & \dots & b_{Pn} \end{bmatrix}$
$=$
$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & c_{ij} & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix}$
a_{i1} a_{i2} $\dots$ a_{iP}
b_{1j} b_{2j} $\dots$ b_{Pj}
Multiply corresponding elements together and add the results to form c_{ij}.

Figure 1.20

By an **m×n array A** we will mean an m×n matrix A of **m×n** positions. This is a model for two-dimensional storage of information in a computer. The number assigned to row i and column j of an array **A** will be denoted **A**[i,j] or a_{ij}.

As for BA, we have the following four possibilities:

1. **BA** may not be defined; we may have n≠m.
2. **BA** may be defined, and then **BA** is p×p, while **AB** is m×n and p≠m. Thus **AB** and **BA** are not equal.
3. **AB** and **BA** may both be the same size, but not be equal as matrices.
4. **AB=BA**.

Theorem 2: Suppose that A,B and C are three matrices. Then

(a) $\mathbf{A}(\mathbf{BC})=(\mathbf{AB})\mathbf{C}$.

(b) $\mathbf{A}(\mathbf{B}+\mathbf{C})=\mathbf{AB}+\mathbf{AC}$.

(c) $(\mathbf{A}+\mathbf{B})\mathbf{C}=\mathbf{AC}+\mathbf{BC}$.

The n×n diagonal matrix

$$I=\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

all of whose diagonal elements are 1, is called the **identify matrix** of order n. If A is an m×n matrix, it is easy to verify that $\mathbf{I_m A}=\mathbf{A I_n}=\mathbf{A}$. If **A** is an n×n matrix and p is a positive integer, we define

$A_p=A\cdot A\cdot\cdot\cdot A$ (p factors) and $A^0=I_n$.

If p and q are nonnegative integers, we can prove the following laws of exponents for matrices:

$$\mathbf{A}^p\mathbf{A}^q=\mathbf{A}^{p+q} \text{ and } (\mathbf{A}^p)^q=\mathbf{A}^{pq}.$$

Observe that the rule $(\mathbf{AB})^p=\mathbf{A}^p\mathbf{B}^p$ does not hold for square matrices. However, if $\mathbf{AB}=\mathbf{BA}$, then $(\mathbf{AB})^p=\mathbf{A}^p\mathbf{B}^p$.

If $\mathbf{A}=[a_{ij}]$ is an $m\times n$ matrix, then the $n\times m$ matrix $\mathbf{A}^T=[a_{ij}^T]$, where $a_{ij}^T=a_{ji}$, $1\le i\le m$, $1\le j\le n$, is called the **transpose (转置) of A**.

Theorem 3: If A and B are matrices, then

(a) $(A^T)^T = A$.

(b) $(A+B)^T = A^T + B^T$.

(c) $(AB)^T = B^TA^T$.

- Boolean Matrix Operations

A **Boolean matrix** (**布尔矩阵**) is an m×n matrix whose entries are either zero or one.

Let **A**=[a_{ij}] and **B**=[b_{ij}] be m×n Boolean matrices. We define **A**∨**B**=**C**=[c_{ij}], the **join** (**或取**)of **A** and **B**, by

$$c_{ij}=\begin{cases}1 & \text{if } a_{ij}=1 \text{ or } b_{ij}=1\\ 0 & \text{if } a_{ij} \text{ and } b_{ij} \text{ are both } 0\end{cases}$$

and **A**∧**B**=**D**=[d_{ij}], the **meet** (**吸取**) of **A** and **B**, by

$$d_{ij}=\begin{cases}1 & \text{if } a_{ij} \text{ and } b_{ij} \text{ are both } 1\\ 0 & \text{if } a_{ij}=0 \text{ or } b_{ij}=0\end{cases}$$

Example Let $\mathbf{A}=\begin{bmatrix}1&0&1\\0&1&1\\1&1&0\\0&0&0\end{bmatrix}$ and $\mathbf{B}=\begin{bmatrix}1&1&0\\1&0&1\\0&0&1\\1&1&0\end{bmatrix}$.

(a) Compute $\mathbf{A}\vee\mathbf{B}$. (b) Compute $\mathbf{A}\wedge\mathbf{B}$.

Solution:

(a) Let $A\vee B=[c_{ij}]$. $A\vee B=\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\\1&1&0\end{bmatrix}$

(b) Let $A\wedge B=[d_{ij}]$. $A\wedge B=\begin{bmatrix}1&0&0\\0&0&1\\0&0&0\\0&0&0\end{bmatrix}$

Suppose that **A**=[a_{ij}] is an m×p Boolean matrix and **B**=[b_{ij}] is a p×n Boolean matrix.

The **Boolean product** of **A** and **B**, denoted **A**⊙**B**, is the m×n Boolean matrix **C**=[c_{ij}] defined by

$$c_{ij} = \begin{cases} 1 & \text{if } a_{ik} = 1 \text{ and } b_{kj} = 1 \text{ for some } k,\ 1 \le k \le p \\ 0 & \text{otherwise} \end{cases}$$

The preceding formula states that for any i and j the element c_{ij} of **C**=**A**⊙**B** can be computed in the following way, as illustrated in Figure 1.21.

1. Select row i of **A** and column j of **B**, and arrange them side by side.
2. Compare corresponding entries. If even a single

pair of corresponding entries consists of two 1's, then $c_{ij}=1$. If this is not the case, then $c_{ij}=0$.

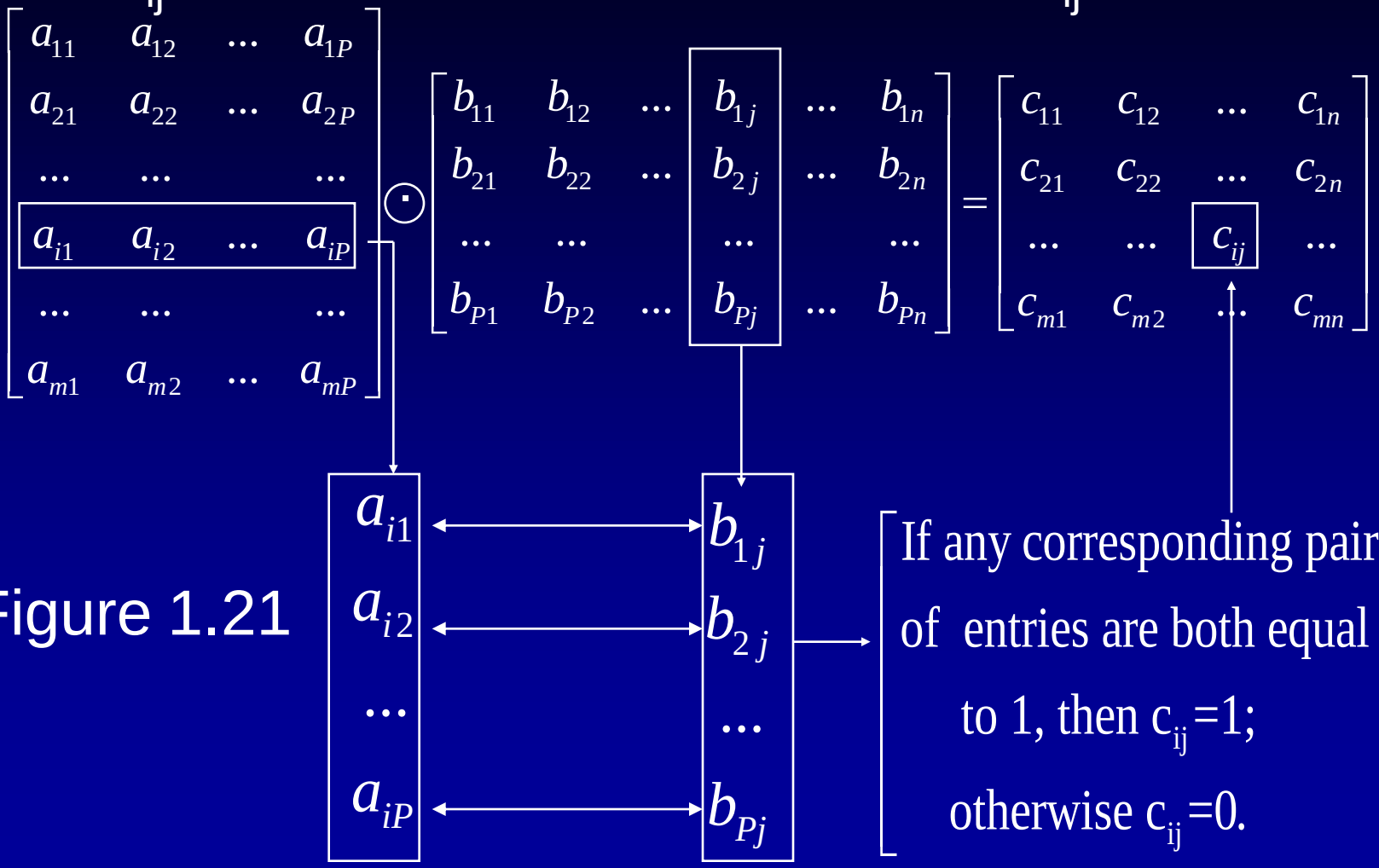


Figure 1.21

Example Let $\mathbf{A}=\begin{bmatrix}1&1&0\\0&1&0\\1&1&0\\0&0&1\end{bmatrix}$ and $\mathbf{B}=\begin{bmatrix}1&0&0&0\\0&1&1&0\\1&0&1&1\end{bmatrix}$.

Compute $\mathbf{A}\odot\mathbf{B}$.

Solution: Let $\mathbf{A}\odot\mathbf{B}=[e_{ij}]$.

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

Theorem 4: If **A**, **B**, and **C** are Boolean matrices of compatible sizes, then

1. (a) $\mathbf{A} \vee \mathbf{B} = \mathbf{B} \vee \mathbf{A}$.
   (b) $\mathbf{A} \wedge \mathbf{B} = \mathbf{B} \wedge \mathbf{A}$.
2. (a) $\mathbf{A} \vee (\mathbf{B} \vee \mathbf{C}) = (\mathbf{A} \vee \mathbf{B}) \vee \mathbf{C}$.
   (b) $\mathbf{A} \wedge (\mathbf{B} \wedge \mathbf{C}) = (\mathbf{A} \wedge \mathbf{B}) \wedge \mathbf{C}$.
3. (a) $\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C}) = (\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{C})$.
   (b) $\mathbf{A} \vee (\mathbf{B} \wedge \mathbf{C}) = (\mathbf{A} \vee \mathbf{B}) \wedge (\mathbf{A} \vee \mathbf{C})$.
4. $(\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C} = \mathbf{A} \odot (\mathbf{B} \odot \mathbf{C})$.

1.6 MATHEMATICAL STRUCTURES

- The collection of sets with the operations of union, intersection, and complement and their accompanying properties is a (discrete) **mathematical structure** or **system**.

  We denote this structure by (sets, $\cup$, $\cap$, $\overline{\ \ }$).

- The collection of 3X3 matrices with the operations of addition, multiplication, and transpose is a mathematical structure denoted by (3×3 matrices, $+$, $*$, ^{T}).

  A structure is **closed with respect to** an operation if that operation always produces another member of the collection of objects.

- The structure (5×5 matrices, +, *, ^{T}) is closed with respect to addition because the sum of two 5×5 matrices is another 5×5 matrix.
- The structure (odd integers, +, *) is not closed with respect to addition. The sum of two odd integer. This structure does have the closure property for multiplication, since the product of two odd numbers is an odd number.

An operation that combines two objects is a **binary operation**. An operation that requires only one object is a **unary operation**.

- (a) Set intersection is a binary operation since it combines two sets to produce a new set.
  (b) Producing the transpose of matrix is a unary operation.
- We say that the operation □ is **commutative**, if x□y= y□x, where □ is a binary operation.
  (a) Join and meet for Boolean matrices are commutative operations.

$$\mathbf{A} \vee \mathbf{B} = \mathbf{B} \vee \mathbf{A} \text{ and } \mathbf{A} \wedge \mathbf{B} = \mathbf{B} \wedge \mathbf{A}.$$

  (b) Ordinary matrix multiplication is not a commutative operation. $\mathbf{AB} \neq \mathbf{BA}$.

If □ is binary operation, then □ is **associative** or **has the associative property** (结合性) if (x□y)□z=x□(y□z).

- Set union is an associative operation, since $(A \cup B) \cup C = A \cup (B \cup C)$ is always true.

If a mathematical structure has two binary operations, say $\square$ and $\triangledown$, a **distributive property** (分配性) has the following pattern:

$$x \square (y \triangledown z) = (x \square y) \triangledown (x \square z).$$

(a) We are familiar with the distributive property for real numbers; if a, b, and c are real numbers, then $a \cdot (b+c) = a \cdot b + a \cdot c$. Note that because we have an agreement about real number arithmetic to multiply before adding, parentheses are not needed.

(b) The structure (sets, $\cup$, $\cap$, $\overline{}$) has two distributive properties:

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

and

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

If the unary operation is $*$ and the binary operation are $\square$ and $\triangledown$, then **De Morgan's laws** are

$$(x \square y)^* = x^* \triangledown y^* \text{ and } (x \triangledown y)^* = x^* \square y^*.$$

(a) As we saw in Section 1.2, sets satisfy De Morgan's laws for union, intersection, and complement: $\overline{(A \cup B)} = \bar{A} \cap \bar{B}$ and $\overline{(A \cap B)} = \bar{A} \cup \bar{B}$.

(b) The structure (real numbers, $+, \sqrt{*},$) does not satisfy De Morgan's laws.

We call an element e as an **identity** for $\square$, if $x \square e = e \square x = x$ for all x in the collection.

Theorem 1 If e is an indentity for a binary operation $\square$, then e is unique.

For ($n \times n$ matrices, $+$, $*$, ^{T}), $\mathbf{I}_n$ is the identity for matrix multiplication and the $n \times n$ zero matrix is the identity for matrix addition.

If a binary operation $\square$ has an identity e, i.e., $x \square e = e \square x = x$ for all x in the collection, we say y is a $\square$–**inverse** of x if $x \square y = y \square x = e$.

Theorem 2: If □ is an associative operation and x has a □–inverse y, then y is unique.

For example:

(a) In the structure (3×3 matrices, +, *, ^{T}), each matrix $\mathbf{A}=[a_{ij}]$ has a +-inverse, or additive inverse, i.e., $-\mathbf{A}=[-a_{ij}]$.

(b) In the structure (integers, +, *), only the integers 1 and -1 have multiplicative inverses.

Definition 1 An algorithm is complete steps necessary to perform a task or computation. The steps in an algorithm may be general descriptions, leaving much detail to be filled in, or they may be totally precise descriptions of every detail.

If some steps were all included, the algorithms would be much more detailed, but long. If the added detail is necessary, one possible solution is to group collections of related steps into other algorithm that is called subroutine and simply refer to these subroutines at appropriate points in the main algorithm.

Pseudocode

In pseudocode, successive steps are usually labeled with consecutive numbers

The end !