2-1 SETS AND STRINGS

A programming language consists of a set of programs (or strings). As we will see shortly, it is convenient to use a grammar as a formal vehicle for generating these programs. Several relations can be defined on the rules of a grammar, and these relations can lead to efficient compilation algorithms for the language associated with that grammar. Consequently, the purpose of this section is to give a brief introduction to the basic elements of sets and strings. In particular, the concepts of sets and operations on sets are given in Sec. 2-1.1. The notion of a relation is introduced in Sec. 2-1.2. Other important aspects about relations which are introduced in this subsection include the graph and transitive closure of a relation. The section ends with an elementary treatment of strings.

2-1.1 Basic Concepts of Set Theory

By a *set* we mean a collection of objects of any sort. The word "object" is used here in a very broad sense to include even abstract objects. A fundamental concept of set theory is that of membership or belonging to a set. Any object belonging to a set is called a *member* or an *element* of that set.

If an element p belongs to a set A, we write $p \in A$, which is read as "p is an element of the set A" or "p belongs to the set A." If there exists an object q which does not belong to the set A, we express this fact as $q \notin A$.

There are many ways of specifying a set. For example, a set consisting of the decimal digits is generally written as

$$\{0,1,2,3,4,5,6,7,8,9\}$$

The names of the elements are enclosed in braces and separated by commas. If we wish to denote this set as D, we write

$$D = \{0,1,2,3,4,5,6,7,8,9\}$$

where the equality sign indicates that D is the set $\{0,1,2,3,4,5,6,7,8,9\}$.

This method of specifying a set is not always convenient. For example, the set D can be more easily described by using a *predicate* as follows:

$$D = \{x \mid x \text{ is a decimal digit}\}$$

where the symbol $|$ is read as "such that." In this case, the predicate is " is a decimal digit." If we let $P(x)$ denote any predicate, then $\{x \mid P(x)\}$ defines a set and is read "the set of all x such that $P(x)$ is true." An element a belongs to this set if $P(a)$ is true; otherwise a does not belong to the set. If we denote the set $\{x \mid P(x)\}$ by B, then $B = \{x \mid P(x)\}$.

Sets which are specified by listing their elements can also be characterized by means of a predicate. For example, the set $\{a, b, 1, 9\}$ can be defined as

$$\{x \mid (x = a) \vee (x = b) \vee (x = 1) \vee (x = 9)\}$$

where the symbol $\vee$ denotes a logical disjunction (or). A predicate can also contain the logical connectives $\wedge$ (logical conjunction or and), $\neg$ (logical negation), $\rightarrow$ (if then), $\Leftrightarrow$ (if and only if), and the relational operators $<$, $\leq$, $=$, $\neq$, $\geq$, and $>$. Furthermore, a predicate can also contain the *existential quantifier* ($\exists$, meaning "there exists") and the *universal quantifier* ($\forall$, representing "for every").

Although it is possible to characterize any set by a predicate, it is sometimes convenient to specify sets by another method, such as

$$C = \{1, 3, 5, \ldots\}$$

$$S = \{a, a^2, a^3, \ldots\}$$

In this representation the missing elements can be determined from the elements present and from the context.

The number of distinct elements present in a set may be finite or infinite. We call a set *finite* if it contains a finite number of distinguishable elements; otherwise a set is *infinite*.

Note that no restriction has been placed on the objects that can be members of a set. It is not unusual to have sets whose members are themselves sets, such as $A = \{0, \{a, b\}, 1, \{p\}\}$. It is important, however, to distinguish between the set $\{p\}$, which is an element of A, and the element p, which is a member of $\{p\}$ but not a member of A.

For any two sets A and B, if every element of A is an element of B, then A is called a *subset* of B, or A is said to be *included* in B, or B includes A. Symbolically, this relation is denoted by $A \subseteq B$, or equivalently by $B \supseteq A$. For any two sets A and B, note that $A \subseteq B$ does not necessarily imply that $B \subseteq A$ except for the following case. Two sets A and B are said to be *equal* iff (if and only if) $A \subseteq B$ and $B \subseteq A$ so that $A = B$. The set A is called a *proper subset* of a set B if $A \subseteq B$ and $A \neq B$. This relation is represented by $A \subset B$.

We now introduce two special sets; the first includes every set under discussion while the second is included in every set under discussion. A set is called a *universal set* (E) if it includes every set under discussion. For example, the universal set for natural numbers may be $E = \{0, 1, 2, \ldots\}$. A set which does not contain any element is called an *empty set* or a *null set*. An empty set will be denoted by ϕ.

Given any set A, we know that the null set ϕ and the set A are both subsets of A. Also for any element $p \in A$, the set $\{p\}$ is a subset of A. Similarly, we can consider other subsets of A. Rather than finding individual subsets of A, we would like to say something about the set of all subsets of A. For any set A, a collection or family of all subsets of A is called the *power set* of A. The power set of A is denoted by $\rho(A)$ or 2^A, so that $\rho(A) = 2^A = \{x | x \subseteq A\}$.

We now introduce the concept of an indexed set. Let $J = \{s_1, s_2, s_3, \ldots\}$ and A be a family of sets $A = \{A_{s_1}, A_{s_2}, A_{s_3}, \ldots\}$ such that for any $s_i \in J$ there corresponds a set $A_{s_i} \in A$, and also $A_{s_i} = A_{s_j}$ iff $s_i = s_j$. A is then called an *indexed set*, J the *index set*, and any subscript such as the s_i in A_{s_i} is called an *index*.

An indexed family of sets can also be written as $A = \{A_i\}_{i \in J}$. In particular, if $J = \{1, 2, 3, \dots\}$, then $A = \{A_1, A_2, A_3, \dots\}$. Also, if $J = \{1, 2, 3, \dots, n\}$, then $A = \{A_1, A_2, A_3, \dots, A_n\} = \{A_i\}_{i \in I_n}$ where $I_n = \{1, 2, \dots, n\}$. For a set S containing n elements, the power set $\rho(S)$ is written as the indexed set $\rho(S) = \{B_i\}_{i \in J}$ where $J = \{0, 1, 2, \dots, 2^n - 1\}$.

Let us now consider some operations on sets. In particular, we emphasize the set operations of intersection, union, and complement. Using these operations, one can construct new sets by combining the elements of given sets. The *intersection* of any two sets A and B, written as $A \cap B$, is the set consisting of all elements which belong to both A and B. Symbolically,

$$A \cap B = \{x | x \in A \wedge x \in B\}$$

For any indexed set $A = \{A_i\}_{i \in J}$,

$$\bigcap_{i \in J} A_i = \{x | x \in A_i \text{ for every } i \in J\}$$

For $J = I_n = \{1, 2, \dots, n\}$, we can write

$$\bigcap_{i=1}^{n} A_i = \bigcap_{i \in I_n} A_i = A_1 \cap A_2 \cap \cdots \cap A_n$$

Two sets A and B are called *disjoint* iff $A \cap B = \phi$. A collection of sets is called a *disjoint collection* if, for every possible pair of sets in the collection, the two sets are disjoint. The elements of a disjoint collection are said to be *mutually disjoint*.

Let A be an indexed set $A = \{A_i\}_{i \in J}$. The set A is a disjoint collection iff $A_i \cap A_j = \phi$ for all $i, j \in J, i \neq j$.

For any two sets A and B, the *union* of A and B, written as $A \cup B$, is the set of all elements which are members of the set A or the set B or both. Symbolically, it is written as $A \cup B = \{x | x \in A \vee x \in B\}$.

For any indexed set $A = \{A_i\}_{i \in J}$,

$$\bigcup_{i \in J} A_i = \{x | x \in A_i \text{ for at least one } i \in J\}$$

For $J = I_n = \{1, 2, 3, \dots, n\}$, we may write

$$\bigcup_{i=1}^{n} A_i = A_1 \cup A_2 \cup \cdots \cup A_n$$

The *relative complement* of B in A (or of B with respect to A), written as $A - B$, is the set consisting of all elements of A which are not elements of B, that is,

$$A - B = \{x | x \in A \wedge x \notin B\}$$

The relative complement of B in A is also called the *difference* of A and B. The relative complement of A with respect to the universal set E, that is, $E - A$, is called the *absolute complement* of A.

So far we have been concerned with sets, their equality, and operations on sets to form new sets. We now introduce the notion of an ordered pair.

An *ordered pair* consists of two objects in a given fixed order. Note that an ordered pair is not a set consisting of two elements. The ordering of the two objects is important. We denote an ordered pair by (x, y).

The equality of two ordered pairs (x, y) and (u, v) is defined by

$$(x, y) = (u, v) \Leftrightarrow ((x = u) \wedge (y = v))$$

where the symbol "$\Leftrightarrow$" denotes *logical equivalence*; that is, $A \Leftrightarrow B$ means that A is equivalent to B. At this point we want to also define the term "imply." A is said to *imply* B written $A \Rightarrow B$, if and only if $A \rightarrow B$ is always *true*. In mathematics, the notations $A \rightarrow B$ and $A \Rightarrow B$ are often used interchangeably.

The idea of an ordered pair can be extended to define an ordered triple and, more generally, an n-tuple. We write an n-tuple as $(x_1, x_2, \ldots, x_n)$.

An important idea in set theory is the notion of a cartesian product. Given two sets A and B, the set of all ordered pairs such that the first member of the ordered pair is an element of A and the second member is an element of B is called the *cartesian product* of A and B and is written as $A \times B$. Accordingly

$$A \times B = \{(x, y) | (x \in A) \wedge (y \in B)\}$$

This notion of cartesian product can be extended to any finite number of sets. Let $A = \{A_i\}_{i \in I_n}$ be an indexed set and $I_n = \{1, 2, \ldots, n\}$. We denote the cartesian product of the sets $A_1, A_2, \ldots, A_n$ by

$$\bigtimes_{i \in I_n} A_i = A_1 \times A_2 \times \cdots \times A_n$$

which is defined recursively as

$$\bigtimes_{i \in I_1} A_i = A_1 \qquad \text{and} \qquad \bigtimes_{i \in I_m} A_i = \Bigl(\bigtimes_{i \in I_{m-1}} A_i\Bigr) \times A_m \qquad \text{for } m = 2, 3, \ldots, n$$

Our definition of cartesian product of n sets is related to the definition of n-tuples in the sense that

$$\begin{aligned} &A_1 \times A_2 \times \cdots \times A_n \\ &\quad = \{(x_1, x_2, \ldots, x_n) | (x_1 \in A_1) \wedge (x_2 \in A_2) \wedge \cdots \wedge (x_n \in A_n)\} \end{aligned}$$

As an example, consider the following sets:

$$A_1 = \{1, 3\}, \qquad A_2 = \{3, 4\}, \qquad A_3 = \{1, 3, 4, 6\}$$

Several set operations are exhibited in the following:

$$A_1 \cup A_2 = \{1, 3, 4\}$$

$$A_1 \cap A_3 = \{1, 3\}$$

$$\bigcup_{i=1}^{3} A_i = \{1, 3, 4, 6\}$$

$$\bigcap_{i=1}^{3} A_i = \{3\}$$

$$A_3 - A_1 = \{4, 6\}$$

$$\rho(A_1) = \{\phi, \{1\}, \{3\}, \{1, 3\}\}$$

$$A_1 \times A_2 = \{(1, 3), (1, 4), (3, 3), (3, 4)\}$$

In this section we have introduced the basic concepts of set theory. We now turn our attention to the notions of relations and orderings.

EXERCISES 2-1.1

1 Give another description of the following sets and indicate those which are infinite sets.
(*a*) $\{x \mid x$ is an integer and $5 \leq x \leq 12\}$
(*b*) $\{2, 4, 8, \ldots\}$
(*c*) All the countries of the world

2 Given $S = \{2, a, \{3\}, 4\}$ and $R = \{\{a\}, 3, 4, 1\}$, indicate whether the following are *true* or *false*.
(*a*) $\{a\} \in S$
(*b*) $\{a\} \in R$
(*c*) $\{a, 4, \{3\}\} \subseteq S$
(*d*) $\{\{a\}, 1, 3, 4\} \subset R$
(*e*) $R = S$
(*f*) $\{a\} \subseteq S$
(*g*) $\{a\} \subseteq R$
(*h*) $\phi \subset R$
(*i*) $\phi \subseteq \{\{a\}\} \subseteq R \subseteq E$
(*j*) $\{\phi\} \subseteq S$
(*k*) $\phi \in R$
(*l*) $\phi \subseteq \{\{3\}, 4\}$

3 Show that $(R \subseteq S) \wedge (S \subset Q) \rightarrow R \subset Q$ is always true. Is it correct to replace $R \subset Q$ by $R \subseteq Q$? Explain your answer.

4 Give the power sets of the following.
(*a*) $\{a, \{b\}\}$
(*b*) $\{1, \phi\}$
(*c*) $\{X, Y, Z\}$

5 Given $A = \{x \mid x$ is an integer and $1 \leq x \leq 5\}$, $B = \{3, 4, 5, 17\}$, and $C = \{1, 2, 3, \ldots\}$, find $A \cap B$, $A \cap C$, $A \cup B$, and $A \cup C$.

6 Show that $A \subseteq A \cup B$ and $A \cap B \subseteq A$.

7 Show that $A \subseteq B \Leftrightarrow A \cup B = B$.

8 If $S = \{a, b, c\}$, find nonempty disjoint sets A_1 and A_2 such that $A_1 \cup A_2 = S$. Find other solutions to this problem.

9 The *symmetric difference* of two sets A and B is the set $A + B$ defined by $A + B = (A - B) \cup (B - A)$. Given $A = \{2, 3, 4\}$, $B = \{1, 2\}$, and $C = \{4, 5, 6\}$, find $A + B$, $B + C$, $A + B + C$, and $(A + B) + (B + C)$. Show that $A + B + C$ is associative.

10 Give examples of sets A, B, C such that $A \cup B = A \cup C$, but $B \neq C$.

11 Write the members of $\{a, b\} \times \{1, 2, 3\}$.

12 Write $A \times B \times C$, B^2, A^3, $B^2 \times A$, and $A \times B$ where $A = \{1\}$, $B = \{a, b\}$, and $C = \{2, 3\}$.

13 Show by means of an example that $A \times B \neq B \times A$ and $(A \times B) \times C \neq A \times (B \times C)$.

14 Show that for any two sets A and B

$$\rho(A) \cup \rho(B) \subseteq \rho(A \cup B)$$

$$\rho(A) \cap \rho(B) = \rho(A \cap B)$$

Show by means of an example that

$$\rho(A) \cup \rho(B) \neq \rho(A \cup B)$$

15 Prove the identities

$$A \cap A = A \qquad A \cap \phi = \phi \qquad A \cap E = A \qquad \text{and} \qquad A \cup E = E$$

16 Show that $A \times (B \cap C) = (A \times B) \cap (A \times C)$.

17 Show that $A \times B = B \times A \Leftrightarrow (A = \phi) \vee (B = \phi) \vee (A = B)$

18 Show that $(A \cap B) \cup C = A \cap (B \cup C)$ iff $C \subseteq A$

19 Show that $(A - B) - C = (A - C) - (B - C)$

20 Prove that $(A \cap B) \times (C \cap D) = (A \times C) \cap (B \times D)$

2-1.2 Relations

The concept of a relation is a basic concept in mathematics as well as in everyday life. Associated with a relation is the act of comparing objects which are related to one another. The ability of a computer to perform different tasks based upon the result of a comparison is an important attribute used several times during the execution of a typical program. In this subsection we first formalize the concept of a relation and then discuss methods of representing a relation by using a matrix or its graph. The relation matrix is useful in determining the properties of a relation and also in representing a relation on a computer. Various basic properties of a relation are given, and certain important classes of relations are introduced.

The word "relation" suggests some familiar examples of relations such as the relation of father to son, sister to brother, or uncle to nephew. Familiar examples in arithmetic are relations such as less than, greater than, or that of equality between two real numbers. We also know the relation between the area of an equilateral triangle and the length of one of its sides and the area of a square and the length of a side. These examples suggest relationships between two objects.

Throughout the discussion we consider relations, called binary relations, between a pair of objects. Any set of ordered pairs defines a *binary relation*. We call a binary relation simply a relation. It is sometimes convenient to express a particular ordered pair, say, $(x, y) \in R$, where R is a relation, as xRy.

In mathematics, relations are often denoted by special symbols rather than by capital letters. A familiar example is the relation "less than" for real numbers. This relation is denoted by $<$. In fact, $<$ should be considered as the name of a set whose elements are ordered pairs. More precisely the relation $<$ is

$$< \; = \{(x, y) \mid x, y \text{ are real numbers and } x \text{ is less than } y\}$$

Let S be a binary relation. The set $D(S)$ of all objects x such that for some y, $(x, y) \in S$ is called the *domain* of S, that is,

$$D(S) = \{x \mid (\exists y)((x, y) \in S)\}$$

where the symbol $\exists$ denotes existential quantification. Similarly, the set $R(S)$ of all objects y such that for some x, $(x, y) \in S$ is called the *range* of S, that is,

$$R(S) = \{y \mid (\exists x)((x, y) \in S)\}$$

Since a relation has been defined as a set of ordered pairs, it is therefore possible to apply the usual operations of sets to relations as well. The resulting sets will also be ordered pairs and will define some relations. If R and S denote two relations, then $R \cap S$ defines a relation such that

$$x(R \cap S)y \Leftrightarrow xRy \wedge xSy$$

Similarly, $R \cup S$ is a relation such that

$$x(R \cup S)y \Leftrightarrow xRy \vee xSy$$

Also

$$x(R - S)y \Leftrightarrow xRy \wedge x \not{S} y$$

where $x \not{S} y$ denotes that x is not related to y in relation S.

A binary relation R in a set X is *reflexive* if, for every $x \in X$, xRx, that is, $(x, x) \in R$. The relation R is *symmetric* if, for every x and y in X, whenever xRy, then yRx. R is said to be *transitive* if, for every x, y, and z in X, whenever xRy and yRz, then xRz. R is *irreflexive* if, for every $x \in X$, $(x, x) \notin R$. Finally, R is said to be *antisymmetric* if, for every x and y in X, whenever xRy and yRx, then $x = y$.

Several important classes of relations having one or more of the properties given here will be discussed later in this subsection.

A relation R from a finite set X to a finite set Y can also be represented by a matrix called the *relation matrix* of R.

Let $X = \{x_1, x_2, \ldots, x_m\}$, $Y = \{y_1, y_2, \ldots, y_n\}$, and R be a relation from X to Y. The relation matrix can be obtained by first constructing a table whose rows are labeled by successive elements of X and whose columns are labeled by successive elements of Y. If $x_i R y_j$, then we enter a 1 in the ith row and jth column. If $x_p \not{R} y_q$, then we enter a zero in the pth row and the qth column. As a special case, consider $m = 2$, $n = 3$, and R given by

$$R = \{(x_1, y_1), (x_2, y_1), (x_2, y_3)\}$$

The required table for R is Table 2-1.

If we assume that the elements of X and Y appear in a certain order, then the relation R can be represented by a matrix whose elements are 1s and 0s. This matrix can be written down from the table constructed or can be defined in the

Table 2-1

	y_1	y_2	y_3
x_1	1	0	0
x_2	1	0	1

following manner:

$$r_{ij} = \begin{cases} 1, & \text{if } x_i R y_j \\ 0, & \text{if } x_i \not R y_j \end{cases}$$

where r_{ij} is the element in the ith row and the jth column. The matrix obtained this way is the relation matrix. If X and Y have m and n elements, respectively, then the matrix is an $m \times n$ matrix. For the relation R just given, the relation matrix is

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

One not only can write a relation matrix when a relation R is given but also obtain the relation if the relation matrix is given.

Throughout the remainder of this subsection, unless otherwise stated, we will assume $X = Y$; that is, the relations are defined in a set X. Thus the relation matrices are square. A relation matrix reflects some of the properties of a relation in a set. If a relation is reflexive, all the diagonal entries must be 1. If a relation is symmetric, the relation matrix is symmetric. If a relation is antisymmetric, its matrix is such that if $r_{ij} = 1$, then $r_{ji} = 0$ for $i \neq j$.

A relation can also be represented pictorially by drawing its *graph*. Although we shall introduce some of the concepts of graph theory which are discussed in later chapters, here we shall use graphs only as a tool to represent relations. Let R be a relation in a set $X = \{x_1, x_2, \ldots, x_n\}$. The elements of X are represented by points or circles called *nodes*. The nodes corresponding to x_i and x_j are labeled x_i and x_j, respectively. These nodes may also be called vertices. If $x_i R x_j$, that is, if $(x_i, x_j) \in R$, then we connect nodes x_i and x_j by means of an arc and put an arrow on this arc in the direction from x_i to x_j. When all the nodes corresponding to the ordered pairs in R are connected by arcs with proper arrows, we get a *directed graph* of the relation R. If $x_i R x_j$ and $x_j R x_i$, then we draw two arcs between x_i and x_j. For the sake of simplicity, we may replace the two arcs by one arc with arrows pointing in both directions. If $x_i R x_i$, we get an arc which starts from node x_i and returns to x_i. Such an arc is called a *sling*.

From the graph of a relation it is possible to observe some of its properties. Several examples are given in Fig. 2-1.

Another very important notion which is used in later chapters is that of a partition. Let S be a given set and $A = \{A_1, A_2, \ldots, A_n\}$ where each A_i, $i = 1, 2, \ldots, n$, is a subset of S and $\bigcup_{i=1}^{n} A_i = S$, then the set A is called a *covering* of S, and the sets $A_1, A_2, \ldots, A_n$ are said to *cover* S. If, in addition, the elements of A, which are subsets of S, are mutually disjoint, then A is called a *partition* of S, and the sets $A_1, A_2, \ldots, A_n$ are called the *blocks* of the partition.

We now proceed to define an important class of relations which partitions a set. A relation R in a set X is called an *equivalence relation* if it is reflexive, symmetric, and transitive. If R is an equivalence relation in a set X, then $D(R)$, the domain of R, is X itself. Therefore, R will be called a relation on X. Some

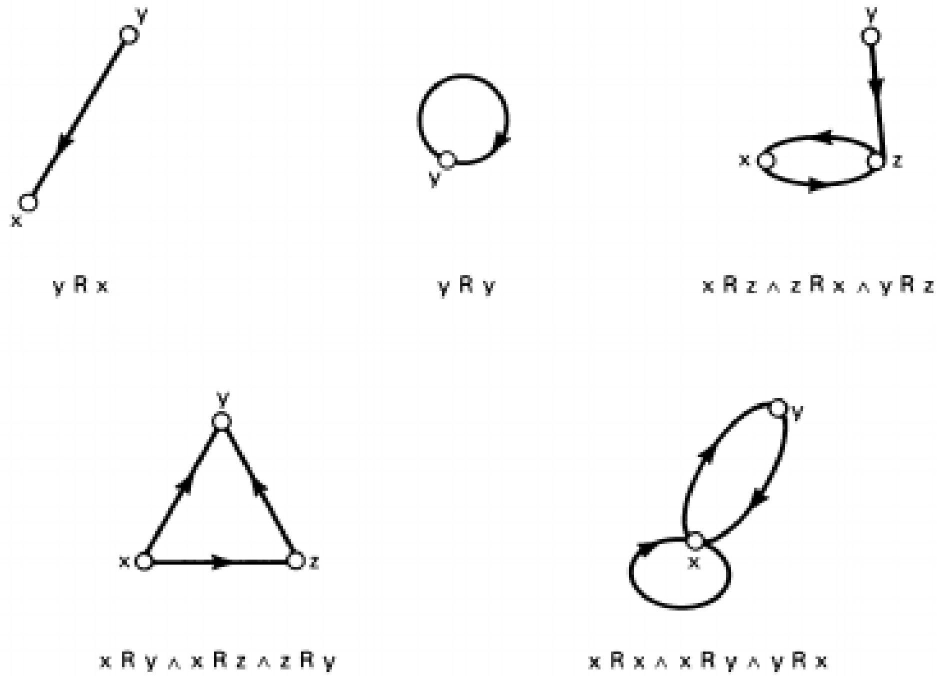

Figure 2-1 Graphs of relations.

examples of equivalence relations are equality of numbers on a set of real numbers, similarity of triangles on a set of triangles, and the relation of lines being parallel on a set of lines in a plane. Another example of an equivalence relation which will be used in the discussion on hashing functions (see Chap. 8) is that of equality in a modular number system. Let I denote the set of all positive integers, and let n be a positive integer. For $x \in I$ and $y \in I$, we define R as

$$R = \{(x, y) | x - y \text{ is divisible by } n\}$$

Note that "$x - y$ is divisible by n" is equivalent to the statement that both x and y have the same remainder when each is divided by n.

Perhaps the most important idea about relations that is used in formal parsing techniques involves taking the "transitive closure" of a relation. We now develop this idea.

Since a binary relation is a set of ordered pairs, the usual operations such as union and intersection on these sets produce other relations. We now consider another operation on relations—relations which are formed in two or more stages. Familiar examples of such relations are the relation of being a nephew or a brother's or sister's son, the relation of an uncle or a father's or mother's brother, and the relation of being a grandfather or a father's or mother's father. These relations can be produced in the following manner.

Let R be a relation from X to Y and S be a relation from Y to Z. Then a relation written as $R \circ S$ is called a *composite relation* of R and S where

$$R \circ S = \{(x, z) | x \in X \wedge z \in Z \wedge (\exists y)(y \in Y \wedge (x, y) \in R \wedge (y, z) \in S)\}$$

The operation of obtaining $R \circ S$ from R and S is called *composition* of relations.

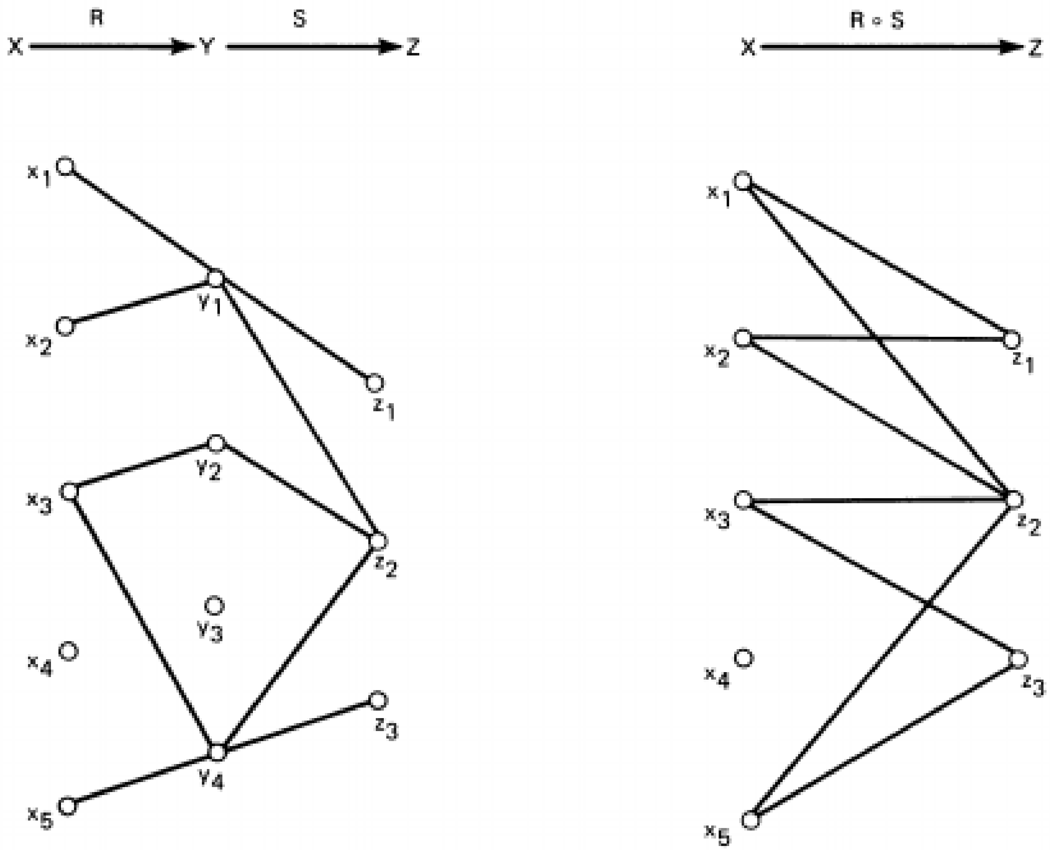

Figure 2-2 Relations R, S, and $R \circ S$.

Note that $R \circ S$ is empty if the intersection of the range of R and the domain of S is empty. $R \circ S$ is nonempty if there is at least one ordered pair $(x, y) \in R$ such that the second member $y \in Y$ of the ordered pair is a first member in an ordered pair in S. For the relation $R \circ S$, the domain is a subset of X and the range is a subset of Z. In fact, the domain is a subset of the domain of R, and its range is a subset of the range of S. From the graphs of R and S one can easily construct the graph of $R \circ S$. As an example, see Fig. 2-2.

The operation of composition is a binary operation on relations, and it produces a relation from two relations. The same operations can be applied again to produce other relations. For example, let R be a relation from X to Y, S a relation from Y to Z, and P a relation from Z to W. Then $R \circ S$ is a relation from X to Z. We can form $(R \circ S) \circ P$, which is a relation from X to W. Similarly, we can also form $R \circ (S \circ P)$, which again is a relation from X to W.

Let us assume that $(R \circ S) \circ P$ is nonempty, and let $(x, y) \in R$, $(y, z) \in S$, and $(z, w) \in P$. This assumption means $(x, z) \in R \circ S$ and $(x, w) \in (R \circ S) \circ P$. Of course, $(y, w) \in S \circ P$ and $(x, w) \in R \circ (S \circ P)$, which shows that

$$(R \circ S) \circ P = R \circ (S \circ P)$$

This result states that the operation of composition on relations is associative. We may delete the parentheses in writing $(R \circ S) \circ P$, so that

$$(R \circ S) \circ P = R \circ (S \circ P) = R \circ S \circ P$$

We know that the relation matrix of a relation R from a set $X = \{x_1, x_2, \ldots, x_m\}$ to a set $Y = \{y_1, y_2, \ldots, y_n\}$ is given by a matrix having m rows and n columns. We shall denote the relation matrix of R by M_R. M_R has entries which are 1s and 0s. Similarly the relation matrix M_S of a relation S from the set Y to a set $Z = \{z_1, z_2, \ldots, z_p\}$ is an $n \times p$ matrix. The relation matrix of $R \circ S$ can be obtained from the matrices M_R and M_S in the following manner.

From the definition it is clear that $(x_i, z_k) \in R \circ S$ if there is at least one element of Y, say, y_j, such that $(x_i, y_j) \in R$ and $(y_j, z_k) \in S$. There may be more than one element of Y which has properties similar to those of y_j, for example, $(x_i, y_r) \in R$ and $(y_r, z_k) \in S$. In all such cases, $(x_i, z_k) \in R \circ S$. Thus when we scan the ith row of M_R and kth column of M_S, and we come across at least one j, such that the entries in the jth location of the row as well as the column under consideration are 1s, then in this instance, the entry in the ith row and kth column of $M_{R \circ S}$ is also 1; otherwise it is 0. Scanning a row of M_R along with every column of M_S gives one row of $M_{R \circ S}$. In this way, we can obtain all the rows of $M_{R \circ S}$.

In general, let the relations A and B be represented by $n \times m$ and $m \times r$ matrices, respectively. Then the composition $A \circ B$ which we denote by the relation matrix C is expressed as

$$c_{ij} = \bigvee_{k=1}^{m} a_{ik} \wedge b_{kj} \qquad i = 1, 2, \ldots, n; \qquad j = 1, 2, \ldots, r$$

where $a_{ik} \wedge b_{kj}$ and $\bigvee_{k=1}^{m}$ indicate bit-ANDing (i.e., $1 \wedge 0 = 0 \wedge 1 = 0 \wedge 0 = 0, 1 \wedge 1 = 1$) and bit-ORing (i.e., $1 \vee 1 = 1 \vee 0 = 0 \vee 1 = 1, 0 \vee 0 = 0$), respectively.

Let us now consider some distinct relations R_1, R_2, R_3, R_4 in a set $X = \{a, b, c\}$ given by

$$R_1 = \{(a,b), (a,c), (c,b)\}$$
$$R_2 = \{(a,b), (b,c), (c,a)\}$$
$$R_3 = \{(a,b), (b,c), (c,c)\}$$
$$R_4 = \{(a,b), (b,a), (c,c)\}$$

Denoting the composition of a relation by itself as

$$R \circ R = R^2 \qquad R \circ R \circ R = R \circ R^2 = R^3 \qquad \cdots \qquad R \circ R^{m-1} = R^m \qquad \cdots$$

let us write the powers of the given relations. Clearly

$$R_1^2 = \{(a,b)\} \qquad R_1^3 = \phi \qquad R_1^4 = \phi \qquad \cdots$$
$$R_2^2 = \{(a,c), (b,a), (c,b)\} \qquad R_2^3 = \{(a,a), (b,b), (c,c)\}$$
$$R_2^4 = R_2 \qquad R_2^5 = R_2^2 \qquad R_2^6 = R_2^3 \qquad \cdots$$
$$R_3^2 = \{(a,c), (b,c), (c,c)\} = R_3^3 = R_3^4 = R_3^5 \qquad \cdots$$
$$R_4^2 = \{(a,a), (b,b), (c,c)\} \qquad R_4^3 = R_4 \qquad R_4^5 = R_4^2 \qquad \cdots$$

Given a finite set X, containing n elements, and a relation R in X, we can interpret R^m ($m = 1, 2, \ldots$) in terms of its graph. This interpretation is done for a number of applications throughout the text. With the help of such an interpretation or from the examples given here, it is possible to say that there are at most n distinct powers of R, for R^m, $m > n$, that can be expressed in terms of $R, R^2, \ldots, R^n$. Our next step is to construct the relation in X given by

$$R^+ = R \cup R^2 \cup R^3 \cup \cdots$$

Naturally, this construction will require only a finite number of powers of R to be calculated, and these calculations can easily be performed by using the matrix representation of the relation R and the Boolean multiplication of these matrices. Let us now see what the corresponding relations R_1^+, R_2^+, R_3^+, and R_4^+ are

$$R_1^+ = R_1 \cup R_1^2 \cup R_1^3 \cdots = R_1$$

$$R_2^+ = R_2 \cup R_2^2 \cup R_2^3 \cdots = R_2 \cup R_2^2 \cup R_2^3$$

$$= \{(a,b), (b,c), (c,a), (a,c), (b,a), (c,b), (a,a), (b,b), (c,c)\}$$

$$R_3^+ = \{(a,b), (b,c), (c,c), (a,c)\}$$

$$R_4^+ = \{(a,b), (b,a), (c,c), (a,a), (b,b)\}$$

Observe that the relations $R_1^+, R_2^+, \ldots, R_4^+$ are all transitive and that $R_1 \subseteq R_1^+, R_2 \subseteq R_2^+, \ldots, R_4 \subseteq R_4^+$. From the graphs of these relations one can easily see that R_i^+ is obtained from R_i ($i = 1, 2, 3, 4$) by adding only those ordered pairs to R_i such that R_i^+ is transitive. We now define R^+ in general.

Definition 2-1 Let X be any finite set and R be a relation in X. The relation $R^+ = R \cup R^2 \cup R^3 \cup \cdots$ in X is called the *transitive closure* of R in X.

Theorem 2-1 The transitive closure R^+ of a relation R in a finite set X is transitive. Also for any other transitive relation P in X such that $R \subseteq P$, we have $R^+ \subseteq P$. In this sense, R^+ is the smallest transitive relation containing R.

Based on this theorem, the transitive closure of the relation matrix for some relation A can easily be computed by using the following algorithm due to Warshall.

Procedure WARSHALL (A, n, P). Given the relation matrix A with n rows, the following steps produce the transitive closure of A, which is denoted by P. The variables i, j, and k are local integer variables.

1. [Initialize]
   P ← A
2. [Perform a pass]
   Repeat through step 4 for k = 1, 2,...,n
3. [Process rows]
   Repeat step 4 for i = 1, 2,...,n

4. [Process columns]
   Repeat for $j = 1, 2, \ldots, n$
   $p_{ij} \leftarrow p_{ij} \vee (p_{ik} \wedge p_{kj})$.
5. [Finished]
   Exit □

To show that this algorithm produces the required matrix, note that step 1 produces a matrix in which $p_{ij} = 1$ if there is a path of length 1 from v_i to v_j. Assume that for a fixed k, the intermediate matrix P produced by steps 3 and 4 of the algorithm is such that the element in the ith row and jth column in this matrix is 1 if and only if there is a path from v_i to v_j which includes only nodes from $v_1, v_2, \ldots, v_k$ as intermediate nodes. Now with an updated value of k, we find that $p_{ij} = 1$ either if $p_{ij} = 1$ in an earlier step or if there is a path from v_i to v_j which traverses through v_{k+1}. This means that $p_{ij} = 1$ if and only if there is a path from v_i to v_j which includes only nodes from $v_1, v_2, \ldots, v_{k+1}$ as intermediate nodes.

Instances of transitive closures of relations will be given in Chaps. 6 and 7, where such relations are derived from a grammar and used in the parsing phase. Also, the transitive closure of a graph of a program can be used to perform certain optimizations in that program. This aspect of compiler writing is the topic of Chaps. 12 and 13.

In terminating this subsection we introduce the notion of the converse of a relation. Given a relation R from X to Y, a relation $\tilde{R}$ from Y to X is called the *converse* of R, where the ordered pairs of $\tilde{R}$ are obtained by interchanging the members in each of the ordered pairs of R. This means, for $x \in X$ and $y \in Y$, that $xRy \Leftrightarrow y\tilde{R}x$.

From the definition of $\tilde{R}$ it follows that $\tilde{\tilde{R}} = R$. The relation matrix $M_{\tilde{R}}$ of $\tilde{R}$ can be obtained by simply interchanging the rows and columns of M_R. Such a matrix is called the *transpose* of M_R. Therefore,

$$M_{\tilde{R}} = \text{transpose of } M_R$$

The graph of $\tilde{R}$ is also obtained from that of R by simply reversing the arrows on each arc.

We now consider the converse of a composite relation. For this purpose, let R be a relation from X to Y and S be a relation from Y to Z. Obviously, $\tilde{R}$ is a relation from Y to X, $\tilde{S}$ from Z to Y; $R \circ S$ is a relation from X to Z, and $R \,\tilde{\circ}\, S$ is a relation from Z to X. Also the relation $\tilde{S} \circ \tilde{R}$ is from Z to X. We now show that

$$R \,\tilde{\circ}\, S = \tilde{S} \circ \tilde{R}$$

If xRy and ySz, then $x(R \circ S)z$ and $z(R \,\tilde{\circ}\, S)x$. But $z\tilde{S}y$ and $y\tilde{R}x$, so that $z(\tilde{S} \circ \tilde{R})x$. This is true for any $x \in X$ and $z \in Z$; hence the required result.

The same rule can be expressed in terms of the relation matrices by saying that the transpose of $M_{R \circ S}$ is the same as the matrix $M_{\tilde{S} \circ \tilde{R}}$. The matrix $M_{\tilde{S} \circ \tilde{R}}$ can be obtained from the matrices $M_{\tilde{S}}$ and $M_{\tilde{R}}$, which in turn can be obtained from the matrices M_S and M_R.

Later in the text, the concept of a relation will be applied to a grammar. In fact there are several interesting relations which can be obtained from a grammar. Many of the operations which are to be performed on these relations have been discussed in this subsection.

EXERCISES 2-1.2

1 Give an example of a relation which is neither reflexive nor irreflexive.

2 Give an example of a relation which is both symmetric and antisymmetric.

3 If relations R and S are both reflexive, show that $R \cup S$ and $R \cap S$ are also reflexive.

4 If relations R and S are reflexive, symmetric, and transitive, show that $R \cap S$ is also reflexive, symmetric, and transitive.

5 Determine whether the following relations are transitive:

$$R_1 = \{(1,1)\} \qquad R_2 = \{(1,2),(2,2)\}$$

$$R_3 = \{(1,2),(2,3),(1,3),(2,1)\}$$

6 Given $S = \{1,2,3,4\}$ and a relation R on S defined by

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

show that R is not transitive. Find a relation $R_1 \supseteq R$ such that R_1 is transitive. Can you find another relation $R_2 \supseteq R$ which is also transitive?

7 Given $S = \{1,2,\ldots,10\}$ and a relation R on S where

$$R = \{(x,y) \mid x + y = 10\}$$

what are the properties of the relation?

8 Let R be a relation on the set of positive real numbers so that its graphical representation consists of points in the first quadrant of the cartesian plane. What can we expect if R is (a) reflexive, (b) symmetric, and (c) transitive?

9 Let L denote the relation "less than or equal to" and D denote the relation "divides," where xDy means "x divides y." Both L and D are defined on the set $\{1,2,3,6\}$. Write L and D as sets, and find $L \cap D$.

10 Show that the relations L and D given in Exercise 9 are both reflexive, antisymmetric, and transitive. Give another example of such a relation. Draw the graphs of these relations.

11 Let R denote a relation on the set of ordered pairs of positive integers such that $(x,y)R(u,v)$ iff $xv = yu$. Show that R is an equivalence relation.

12 Given a set $S = \{1,2,3,4,5\}$, find the equivalence relation on S which generates the partition whose equivalence classes are the sets $\{1,2\}$, $\{3\}$, and $\{4,5\}$. Draw the graph of the relation.

13 Prove that the relation "congruence modulo m" given by

$$\equiv \; = \{(x,y) \mid x - y \text{ is divisible by } m\}$$

over the set of positive integers is an equivalence relation. Show also that if $x_1 \equiv y_1$ and $x_2 \equiv y_2$, then $(x_1 + x_2) \equiv (y_1 + y_2)$.

14 Prove the following equivalences and equalities:

(a) $\tilde{\tilde{R}} = R$
(b) $R = S \Leftrightarrow \tilde{R} = \tilde{S}$
(c) $R \subseteq S \Leftrightarrow \tilde{R} \subseteq \tilde{S}$
(d) $R \tilde{\cup} S = \tilde{R} \cup \tilde{S}$
(e) $R \tilde{\cap} S = \tilde{R} \cap \tilde{S}$

15 Show that if a relation R is reflexive, then $\tilde{R}$ is also reflexive. Show also that similar remarks hold if R is transitive, irreflexive, symmetric, or antisymmetric.

16 What nonzero entries are there in the relation matrix of $R \cap \tilde{R}$ if R is an antisymmetric relation in a set X?

17 Given the relation matrix M_R of a relation R on the set $\{a, b, c\}$, find the relation matrices of $\tilde{R}$, $R^2 = R \circ R$, $R^3 = R \circ R \circ R$, and $R \circ \tilde{R}$.

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

18 Two equivalence relations R and S are given by their relation matrices M_R and M_S. Show that $R \circ S$ is not an equivalence relation.

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

Obtain equivalence relations R_1 and R_2 on $\{1,2,3\}$ such that $R_1 \circ R_2$ is also an equivalence relation.

19 Using Warshall's algorithm, obtain the transitive closure of the relation whose graph is given in Fig. 2-3.

20 For the graph of the relation R given in Fig. 2-4, determine $M_{\tilde{R}}$, $M_R \wedge M_{\tilde{R}}$, and $M_{\tilde{R}} \wedge M_R$.

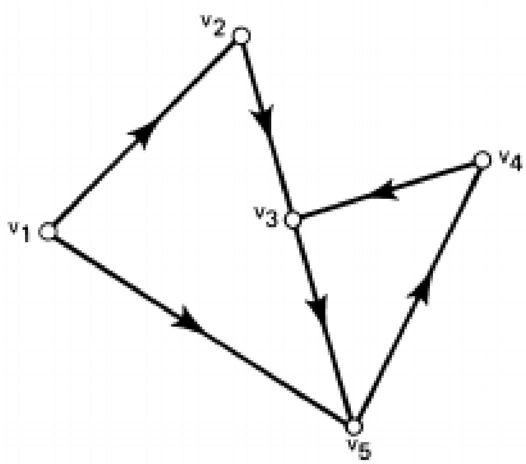

Figure 2-3

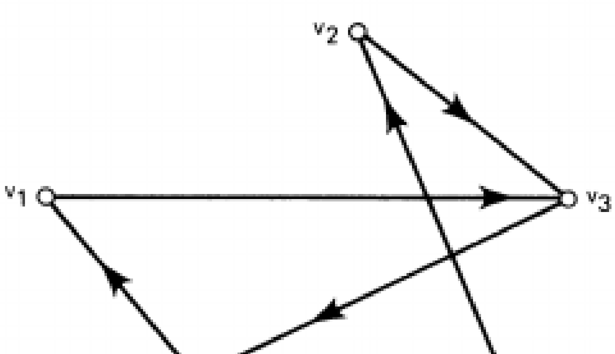

Figure 2-4

2-1.3 Strings

In this subsection we are concerned with some elementary properties of strings. There are many interesting properties exhibited by string operations, just as there are interesting properties for arithmetic operations over the natural numbers. To refamiliarize ourselves with some of the properties associated with operations, let us consider the operation of addition on the natural numbers. This operation can be represented, in general, by a functional system in two variables:

$$f(x, y) = x + y$$

where x and y are natural numbers. This system exhibits certain interesting properties. First, the sum of any two natural numbers is a natural number. This property is called *closure*. Closure is a necessary property for a system (i.e., a set and an operation on that set) to be classified as an algebraic system. Second, $(x + y) + z = x + (y + z) = x + y + z$ when x, y, and z are natural numbers; accordingly the operation of addition is said to be *associative*. Third, there exists a number i such that for every natural number x, $x + i = x$. This number is zero and is called the unit element or *identity* of the additive system. Many other important properties, such as distributivity and commutativity, exist when arithmetic operations such as addition and multiplication are applied to the set of natural numbers.

We begin a discussion of strings by formally defining a string. To do so, we must introduce the notion of an alphabet and the operation of concatenation. Simply stated, an alphabet V is a finite nonempty set of symbols. The set $V = \{a, b, c, \ldots, z\}$ is a familiar example of an alphabet and $\{\alpha, \beta, \gamma, \varepsilon\}$ is a four-character alphabet (which is a subalphabet of the Greek alphabet).

The *concatenation* of two alphabetic characters, say 'a' and 'b', is said to form a sequence of characters, namely, 'ab'. (Note that henceforth when we refer to a character from an alphabet or a sequence of such characters, they are enclosed in single quote marks.) The operation of concatenation also applies to sequences of characters. For example, 'ab' concatenated with 'ab' is '$abab$'. We denote the concatenation operator by the special symbol $\circ$. This allows us to write expressions such as 'ab' $\circ$ 'a', which is identical in value to 'aba'.

A *string* (or *sentence*) over an alphabet V is either a letter from the alphabet V or a sequence of letters derived from the concatenation of zero or more characters from the alphabet V. Examples of strings over an alphabet $V = \{a, b, c\}$ are 'a', 'ca', '$ccba$', and 'bbb'.

Let $V \circ V = V^2$ designate all strings of length 2 on V, $V \circ V \circ V = V^2 \circ V = V^3$ designate all strings of length 3 on V, and, in general, $V \circ V \circ \cdots \circ V = V^n$ designate all strings of length n on V. Then the closure of V, denoted as V^+, is defined as

$$V^+ = V \cup V^2 \cup V^3 \cup \cdots$$

For completeness, a special string ε called the empty (or null) string is often combined with V^+ to form the closure set V^* of V. That is, $V^* = \{\varepsilon\} \cup V \cup V^2 \cup V^3 \cup \cdots = \{\varepsilon\} \cup V^+$. The string ε has the identity property (i.e., $x \circ \varepsilon =$

$\varepsilon \circ x = x$ for any string x which is an element of V^*), and it is called the identity element in the system formed by the set V^* and the operation of concatenation. Associativity is another property of this system (that is, $(x \circ y) \circ z = x \circ (y \circ z) = x \circ y \circ z$ for $x, y, z \in V^*$).

As an example, consider the set of strings V^* that can be generated from an alphabet $V = \{x, y\}$. Some subsets of V^* are

$$V^2 = \{\text{'}xx\text{'}, \text{'}xy\text{'}, \text{'}yx\text{'}, \text{'}yy\text{'}\}$$

$$V^3 = \{\text{'}xxx\text{'}, \text{'}xxy\text{'}, \text{'}xyx\text{'}, \text{'}xyy\text{'}, \text{'}yxx\text{'}, \text{'}yxy\text{'}, \text{'}yyx\text{'}, \text{'}yyy\text{'}\}$$

$$V^4 = \{\text{'}xxxx\text{'}, \text{'}xxxy\text{'}, \text{'}xxyx\text{'}, \text{'}xxyy\text{'}, \text{'}xyxx\text{'}, \text{'}xyxy\text{'}, \text{'}xyyx\text{'}, \text{'}xyyy\text{'},$$
$$\text{'}yxxx\text{'}, \text{'}yxxy\text{'}, \text{'}yxyx\text{'}, \text{'}yxyy\text{'}, \text{'}yyxx\text{'}, \text{'}yyxy\text{'}, \text{'}yyyx\text{'}, \text{'}yyyy\text{'}\}$$

$$\vdots$$

We will, on many occasions, refer to the closure set V^* of an alphabet.

As another example of a string, let us examine FORTRAN. The FORTRAN alphabet consists of 26 letters, 10 digits, and a set of special characters, such as '(', ')', ',', '=', and '+'. It is only these characters that are used in writing a FORTRAN program. Hence a program can be viewed as the concatenation of characters over an alphabet to yield an arbitrarily long string. In Chap. 4, we see that the problem of ensuring that only the proper set of characters appears in a program is handled by the scanning phase of a compiler.

Let x, y, and z be strings over an alphabet where $z = xy$. The string x is called a *prefix* or *head* of z. If $y \neq \varepsilon$, then x is called a *proper prefix* or *proper head*. Similarly, y is called a *suffix* or *tail* of z, and if $x \neq \varepsilon$, then y is called a *proper suffix* or *proper tail*.

Some of the concepts introduced here are used in the next section, where the notions of grammars and languages are introduced.

2-2 DISCUSSION OF GRAMMARS

Programming languages must be precisely defined. Unfortunately, for some of the earlier programming languages the existence of a particular compiler finally provided the precise definition of such a language. The proper specification of a programming language involves the definition of the following:

1. The set of symbols (or alphabet) that can be used to construct correct programs
2. The set of all syntactically correct programs
3. The "meaning" of all syntactically correct programs

In this section we shall be concerned with the first two items in the specification of programming languages.

A language L can be considered a subset of the closure (including the empty string) of an alphabet. The language consisting of this closure set is not particu-