1

Elements of Mathematical Logic

1.1. INTRODUCTORY NOTES

Mathematical (symbolic) logic traces its origins to the so-called traditional formal logic, from which it emerged in response to a desire to formalize certain aspects of intellectual activity. This desire continued to influence its subsequent growth as an independent science, when it addressed itself to the task of providing the logical foundations of mathematics by tackling problems of consistency and completeness of axiomatic systems underlying this science, the problem of determining all the inferences derivable from these axioms, as well as a variety of similar questions. Eventually mathematical logic grew into a powerful research tool, but its use continued to be restricted to the domain of pure theory, even though there were men who recognized its potential in the field of applied science (as long ago as 1910 Paul Ehrenfest pointed out the possibility of using the constructs of mathematical logic to describe the operation of practical systems such as switching circuits). Be that as it may, it was only in the thirties that the engineering application of mathematical logic came into its own. It was during that time that V.I. Shestakov [111, 112] and C.E. Shannon [231] worked on the application of mathematical logic to switching networks and led the way for M.A. Gavrilov [21] and the independent theory of relay switching. Before long mathematical logic penetrated even deeper into the applied sciences. It was found that not only relay switching networks but also many other discrete-action systems were susceptible to description by its techniques.

Thus mathematical logic became an accepted tool in the development and design of a great variety of engineering systems, while at the same time maintaining its extreme importance in theoretical research. Its applied aspect proved especially valuable in recent years, in connection with the research into the general laws of control which govern both technology and Nature.

Since there are two aspects of mathematical logic—the theoretical and the applied—the subject can be developed in two distinct ways. In accordance with our main objective, we shall confine ourselves to the applied aspect, with the further restriction that we shall now discuss only these elements of logic which are needed for an understanding of later sections.

1.2. BASIC CONCEPTS

In discussing logic, we shall experience time and again the importance of a fundamental mathematical concept—the *functional relationship*. In its most general form this concept is associated with the idea of existence of two sets and of mapping of one set onto the other. Suppose we have sets X and Y consisting of elements x and y, respectively, that is,

$$X = \{x\}, \quad Y = \{y\}.$$

If, by virtue of some condition, each element x belonging to set X (this is written as $x \in X$) is matched with a specific element y of set Y ($y \in Y$), then the matching condition is said to define y as a function of x, or, alternatively, one says that set X maps into set Y. The function $y = y(x)$ is also said to be defined on the set X (called the domain of the function) and to have values in set Y (the range of the function); x is called the *independent variable or argument*, and y is called the *function*.

Every specific functional relationship is determined, on the one hand, by the characteristics of sets X and Y and, on the other hand, by the nature of elements x and y in these two sets.

Let us consider some basic characteristics of sets. A set is classed as either *finite* or *infinite*, depending upon the number of elements constituting it. For example, the set of letters of the alphabet is finite; the set of molecules in a finite body is also finite; but sets consisting of all positive integers, or of all rational numbers, or of all real numbers are infinite. The set of all points on a line segment and the set of all points in a plane figure are also infinite.

Sets may be compared according to their cardinality. Two sets are said to have the same cardinality if a one-to-one correspondence can be established between their elements. The concept of cardinality of a set allows us to distinguish two important classes of infinite sets. These are the *countable** and the *continuum* sets.

---

*Also called denumerable.

Countable sets are sets that have the cardinality of the set of all natural numbers, and continuum sets are sets that have the cardinality of the set of all real numbers.

In particular, the set of all even integers is countable, since the elements of this set can be easily placed in a one-to-one correspondence with the elements of the set of natural numbers. Indeed, by arranging the even integers and the natural numbers in ascending order, we can establish the following one-to-one correspondence between the elements of these two sets:

$$2,\ 4,\ 6,\ \ldots,\ 2n,\ \ldots$$
$$1,\ 2,\ 3,\ \ldots,\ \ n,\ \ldots$$

The set of all algebraic numbers, the set of all rational numbers, and so on, are also countable.

The continuum sets include the set of all irrational numbers, the set of all points in a line segment, the set of all points on a plane figure, and many others.

In some cases, comparison of infinite sets in terms of their cardinality leads to statements that may sound quite paradoxical. For instance, it would seem strange, at a first glance, that the set of points in a segment (AB in Fig. 1.1) and the set of points in a section of the same segment (AC in Fig. 1.1) should have the same cardinality. This, however, may easily be proven with the help of Fig. 1.1. Here, each point M in segment AB may be connected by a ray to an origin O; this ray intersects segment AC at a point M', which is seen to be in one-to-one correspondence with point M of segment AB, showing that our two sets do indeed have the same cardinality. Similarly, it may be demonstrated that the set of points in a plane figure, or even in a three-dimensional body, has the same cardinality as the set of points in a line segment, namely, that of the continuum.

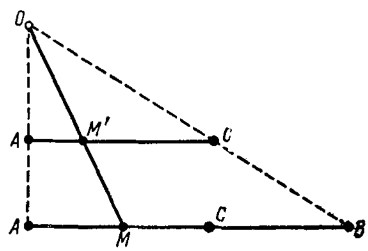

Fig. 1.1.

Let us now return to functional relationships. As already stated, such a relationship is specified by the nature of the elements in the sets on which it is defined, and by the characteristic properties of these sets. If a function is defined on the set X of all real numbers x and assumes values from a set Y which also consists of all real numbers y, then we have a real function y of a single real variable

x, or $y = y(x)$. If, however, the function assumes values from the same set of real numbers y, but each element of the set $Z = \{z\}$ on which it is defined is a sequence of n real numbers $x_1, x_2, \ldots, x_n$, then we are no longer dealing with a real function of a single real variable, but with a real function y of n real variables $x_1, x_2, \ldots, x_n$, that is, $y = y(x_1, x_2, \ldots, x_n)$.

The above functions are based on the set of real numbers, and it is this characteristic that unites them into a single class. The distinguishing feature of this class of functions is that both the values assumed by the function and the arguments of this function are defined on continuum sets.

The basic characteristic of functions of mathematical logic is that both their domain and their range (that is, the sets which participate in the mapping) consist of elements that, in general, have no connection with any defined quantities whatsoever. We are thus saying that we cannot distinguish between the elements of these sets by any other means than assigning to them symbols of some kind, for example, numerals.

The list of all symbols describing the elements of a given set is called the *alphabet* of this set; an undefined symbol, which may represent any element of the set, is called a ***logical variable***. Each specific symbol is then one of the values which the logical variable can assume.

Thus we have seen that, in terms of the properties of the elements of the mapped sets, logical functions are functions of the most general type. Moreover, they assume values from finite sets. In this they differ from many other functions (for example, functions of real variables, which are, in general, defined on continuum sets).

As an example, consider two sets. Set $X = \{x\}$ consists of all the *different* white keys of the piano. Let us denote these keys, from left to right, by symbols $x_1, x_2, \ldots, x_{50}$; the list of these symbols is alphabet of set $X = \{x_1, x_2, \ldots, x_{50}\}$. Set $Y = \{y\}$ consists of the seven different notes contained in an octave, and its alphabet is $\{y_1, y_2, \ldots, y_7\}$, where the symbols $y_1, y_2, y_3, y_4, y_5, y_6$, and y_7 denote the notes c, d, e, f, g, a, and b, respectively. In a well-tuned piano each symbol of the alphabet $\{x\}$ is in a one-to-one correspondence with a specific symbol of the alphabet $\{y\}$. This means that the variable y, which assumes the values $y_1, y_2, \ldots, y_7$, is a logical function of the independent variable x, which assumes the values $x_1, x_2, \ldots, x_{50}$. This function may be specified in several ways, for example, in the form of a table (see Table 1.1).

The first classification to which we may subject the functions of mathematical logic is that based on the number of different sets involved in the mapping of a given function. If only one set is involved, so that the set is mapped into itself, the corresponding logical function is said to be *homogeneous*. A function involving mapping of one set onto a different set is said to be *heterogeneous*. For example, the logical function given in Table 1.2 is homogeneous, while that of Table 1.3 is heterogeneous.

Table 1.1

x	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	...	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}	x_{49}	x_{50}
y	y_6	y_7	y_1	y_2	y_3	y	y_5	y_6	y_7	y_1	...	y_7	y_1	y_2	y_3	y_4	y_5	y_6

We said before that the set from which a logical function takes its values is finite; and since any homogeneous logical function represents the mapping of some set onto itself, it follows that the set constituting the range of a homogeneous logical function must be finite. The corresponding logical variable may be two-valued, three-valued, or, in general, m-valued.

Table 1.2

α_1	α_2	α_3	α_4
α_2	α_1	α_3	α_3

Table 1.3

α_1	α_2	α_3	α_4
β_2	β_1	β_1	β_3

Each of the values of the argument of a heterogeneous logical function is usually called an *object*, and the function itself is called a *predicate*. While the set of the argument values (the object set) may be infinite, the heterogeneous logical functions themselves—the predicates—may only be two-valued, three-valued, or, in general, m-valued (where m must be finite).

In the theory of real variables, we are accustomed to real functions of n real arguments. In the same way, the theory of logical variables admits of logical functions not only of one but also of n independent variables.

We shall divide functions of several variables into two classes. One of the classes shall include functions in which all the arguments, as well as the function itself, are logical variables assuming values from the same set. Again, we shall call such functions "homogeneous."* Our second class shall comprise all those logical functions of several variables which do not belong to the first class; again, we shall call such functions "heterogeneous."

As in the case of functions of one independent variable, the logical variables that are the arguments of the heterogeneous functions of several variables are called objects; the functions themselves are again called predicates.

Depending on the number of arguments in a given heterogeneous logical function, we have *one-place*, *two-place*, and, in general, *n-place predicates*. One-place predicates are sometimes called *properties*, while multiple-place predicates are called *relations*.

To illustrate these concepts and terminology, let us consider a few examples:

Suppose we examine the event: I shall meet a man whom I know. This event may or may not occur, depending on occurrence or nonoccurrence of the following elementary events constituting the composite event: One of the persons I shall meet will be someone I know, and this person shall also be a man. Here we have a homogeneous logical function with two arguments; it is homogeneous because both arguments and the function itself are events, that is, logical variables assuming values from the same binary set whose elements are "the event shall occur" and "the event shall not occur." By denoting one argument (event: meeting a person whom I know) by x_1, the other (event: meeting a man) by x_2, and the function (event: meeting a man whom I know) by y, we can represent their relationship in the form of Table 1.4. The characters 0 and 1 in the table are symbols corresponding to the elements "the event shall not occur" and "the event shall occur."

Table 1.4

$x_1 \backslash x_2$	0	1
0	0	0
1	0	1

In our previous example of the piano keys, the logical function was heterogeneous. There we had a seven-valued, one-place predicate whose object variable (the number of the key) assumed values from a fifty-element set.

The estimation of the truth value of a statement given by the algebraic expression

$$x_1 + x_2 > 10,$$

*The mathematicians who developed the theory of homogeneous logical functions worked with a set whose elements were called "true" and "false" propositions. For this reason this theory is referred to as "propositional calculus."

which is true for some numerical values of x_1 and x_2 and false for other, leads us to an example of a two-place, two-valued predicate; here we have two independent variables, and they assume values from a set of real numbers, which has the cardinality of the continuum.

Consider another example: it is no great trick to determine the day of the week corresponding to a certain date (day, month, year). The rules governing this problem constitute a heterogeneous logical function—a three-place, seven-valued predicate. The object variables in this case assume values from three sets: one of these contains 31 elements, another 12 elements, and the third a countable number of elements.

No single mathematical theory applicable to all the possible logical functions exists as yet. The theory which as of now has reached the highest state of development is that governing two-valued functions. This branch of mathematical logic (two-valued or binary logic) serves a dual function: on the one hand, it supports the entire edifice of mathematical logic; on the other hand, it is precisely this branch of the theory that is, at present, of the greatest applied value. The same, however, cannot be said about the theory of many-valued logic, which is still a long way from perfection. For this reason we shall not concern ourselves with it any further and shall proceed to the postulates of binary logic which includes the calculi of two-valued propositions and predicates.

1.3. PROPOSITIONAL CALCULUS

a) Definition of Logical Functions

We shall now discuss homogeneous binary logical functions

$$y = y(x_1, x_2, \ldots, x_n),$$

that is, functions in which all the independent variables $x_1, x_2, \ldots, x_n$, as well as the function y itself, assume values from the same binary set M. We shall denote the two elements of this set by symbols 0 and 1; these symbols shall then constitute the entire alphabet of all the logical variables which are arguments of these logical functions.

Now let us construct a table (see Table 1.5) of 2^n columns and n rows. The heading of each row shall be one of the n independent variables. The heading of each column will be a numeral from the set $0, 1, 2, \ldots, 2^n - 1$.

Next, let us fill each column with a sequence of symbols 0 and 1 such that this sequence, when read from bottom to top, shall form

Table 1.5

$r = 2^n$ columns

x \ k	0	1	2	3	4	5	6	7	...	2^n-2	2^n-1
x_1	0	1	0	1	0	1	0	1	...	0	1
x_2	0	0	1	1	0	0	1	1	...	1	1
x_3	0	0	0	0	1	1	1	1	...	1	1
...	...	...	...	...	...	...	...	...	...	...	...
x_n	0	0	0	0	0	0	0	0	...	1	1

n rows

the binary representation of the numeral in the heading of the column. The best way to complete such a table is as follows: We enter in the first row (row x_1) a string of pairs (01); in the second row (row x_2), a string of groups of four (0011); in the third row, a string of groups of eight (00001111), and so on. Now each column of the table shows one of the possible combinations of values which may be assumed by the n independent variables. Thus it may be said that each column corresponds to a point in an n-dimensional binary logical space (a space constructed on the basis of the two-element set M). The table as a whole (the aggregate of all the 2^n columns) is a complete description of the entire n-dimensional binary logical space which consists of $r = 2^n$ points; the numeral k heading a column is then a symbol denoting a point in this space.

In a more graphic representation of an n-dimensional binary logical space, 0 and 1 can be regarded as real numbers. Then the one-dimensional case may be represented in terms of two points on a real axis (Fig. 1.2). The two-dimensional case may be represented in terms of four vertices of a unit square (Fig. 1.3), and the three-dimensional case—by the vertices of a unit cube (Fig. 1.4). In general, the n-dimensional binary logical space may be represented in terms of the set of all the vertices of an n-dimensional unit cube.

Fig. 1.2.

To define a specific binary homogeneous logical function $y = y(x_1, x_2, \ldots, x_n)$ means to specify which of the two possible values

(0 or 1) will be assumed by the logical variable y at a point k of the corresponding binary n-dimensional logical space (or at a vertex of the n-dimensional cube). This information is furnished by a ***table of correspondences**** (Table 1.6), which specifies our function in the form $y = y(k)$. A table of correspondences $y = y(k)$, together with a table for the n-dimensional binary logical space $k = k(x_1, x_2, \ldots, x_n)$ completely specifies the homogeneous binary logical function $y = y(x_1, x_2, \ldots\ldots, x_n)$ of n arguments.

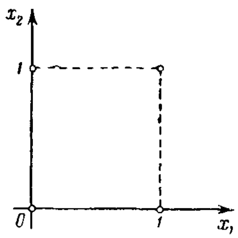

Fig. 1.3.

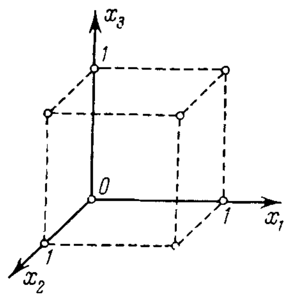

Fig. 1.4.

At any k, the function $y(k)$ can be either 0 or 1. It follows that each function of n arguments can be represented in a table of correspondences by some sequence of zeros and ones. The length of this sequence is $r = 2^n$, so that the total number of different functions that may possibly be constructed on the set of points of an n-dimensional binary space is $s = 2^r = 2^{(2^n)}$. All these functions can therefore be enumerated and hence designated by a numeral (numbered).

Table 1.6

k	0	1	2	3	...	$2^n - 2$	$2^n - 1$
y	$y(0)$	$y(1)$	$y(2)$	$y(3)$	...	$y(2^n - 2)$	$y(2^n - 1)$

There is a convenient method for scanning and numbering all these functions. This involves constructing a table such as 1.7 that combines all the possible correspondence tables; it contains $r = 2^n$ columns and $s = 2^r$ rows. We can complete this table as follows: We

*The term table of combinations is also used.

Table 1.7

$y \backslash k$	0	1	2	3	...	$r-1$
y_0	0	0	0	0	...	0
y_1	1	0	0	0	...	0
y_2	0	1	0	0	...	0
y_3	1	1	0	0	...	0
...	...	...	...	...	...	
y_{s-1}	1	1	1	1	...	1

enter pairs (01) into the first column, then we enter groups of four (0011) into the second column, groups of eight (00001111) into the third, and so on.* Then the string of zeros and ones in each row, when read from right to left, will be the binary representation of the number designating the function corresponding to this row. We shall refer to such a table as a *general correspondence table.*

In addition to this tabular method of defining homogeneous binary logical functions, there exists an analytical procedure which is widely used. This procedure is based on our ability to transform homogeneous functions into composite functions. Indeed, we know that both the homogeneous function and its argument assume values from the same set. This means that the logical variable that is the functional variable in one relation may be the argument in another relation. The transformation into composite functions allows us to express any homogeneous binary logical function in terms of certain simple functions. Naturally, the use of such functions and of the related notation entails some specific rules, that is, a special algebra.

b) Functions of One and Two Variables

Let us begin with the simplest case, where the function has only one argument ($n = 1$), and where the general correspondence table

*This table differs from 1.5 in that we fill in the columns and not the rows.

combined with the table of unidimensional binary logical space assumes the form of Table 1.8.

Table 1.8

k	0	1	Notation
x	0	1	
y_0	0	0	$y_0 = 0$
y_1	1	0	$y_1 = \bar{x}$
y_2	0	1	$y_2 = x$
y_3	1	1	$y_3 = 1$

The number of points in this logical space is then $r = 2^n = 2^1 = 2$, while the number of different functions is $s = 2^r = 2^2 = 4$. These four functions—y_0, y_1, y_2, and y_3— are shown in Table 1.8. There are no other functions of one argument.

The values of the functions y_0 and y_3 do not vary with the values of the argument, so that these functions are called *constant*. We shall denote them by $y_0 = 0$, $y_3 = 1$.

The function y_2, called the *identity function*, always assumes the same value as the argument x; the obvious notation is $y_2 = x$.

The function y_1 becomes 1 when $x = 0$, and 0 when $x = 1$. It is termed *negation*, and merits a special notation, $y_1 = \bar{x}$; this is read as "not x." Note that two of the above four functions can always be expressed as composite functions, using the symbolic notation for the two other functions. Thus,

$$\begin{aligned} y_3 &= \bar{y}_0 = \bar{0} = 1, \\ y_2 &= \bar{y}_1 = \bar{\bar{x}} = x. \end{aligned} \tag{1.1}$$

Therefore, we can define any homogeneous binary function of one argument in an analytical form by applying the special symbolic notation for negation to the two functions $y = 0$ and $y = \bar{x}$.

The general correspondence table for the case of functions of two arguments x_1 and x_2 $(n = 2)$, is Table 1.9. Here the number of points

Table 1.9

k	0	1	2	3	Notation
x_1	0	1	0	1	
x_2	0	0	1	1	
y_0	0	0	0	0	$y_0 = 0$
y_1	1	0	0	0	$y_1 = x_1 \downarrow x_2$
y_2	0	1	0	0	$y_2 = x_1 \leftarrow x_2$
y_3	1	1	0	0	$y_3 = \bar{x}_2$
y_4	0	0	1	0	$y_4 = x_2 \leftarrow x_1$
y_5	1	0	1	0	$y_5 = \bar{x}_1$
y_6	0	1	1	0	$y_6 = x_1 \nabla x_2$
y_7	1	1	1	0	$y_7 = x_1 / x_2$
y_8	0	0	0	1	$y_8 = x_1 \,\&\, x_2$
y_9	1	0	0	1	$y_9 = x_1 \sim x_2$
y_{10}	0	1	0	1	$y_{10} = x_1$
y_{11}	1	1	0	1	$y_{11} = x_2 \rightarrow x_1$
y_{12}	0	0	1	1	$y_{12} = x_2$
y_{13}	1	0	1	1	$y_{13} = x_1 \rightarrow x_2$
y_{14}	0	1	1	1	$y_{14} = x_1 \vee x_2$
y_{15}	1	1	1	1	$y_{15} = 1$

in the logical space is $r = 2^n = 2^2 = 4$, while the number of different functions is $s = 2^r = 2^4 = 16$. The column on the extreme right gives the notation used for these functions. We see that six of these sixteen functions were encountered among the functions of one argument. These include two constant functions ($y_0 = 0$ and $y_{15} = 1$), two identity functions ($y_{10} = x_1$ and $y_{12} = x_2$), and two negation functions ($y_3 = \bar{x}_2$ and $y_5 = \bar{x}_1$).

Of the remaining ten functions, two (y_4 and y_{11}) are not independent, since they differ from functions y_2 and y_{13} only in the relative position of the two arguments. We are thus left with eight new functions of two independent variables. They have the following special properties:

The function $y_{14} = x_1 \vee x_2$ is 0 if, and only if, both arguments are 0. It is called *disjunction* and is read "x_1 or x_2."

The function $y_{13} = x_1 \rightarrow x_2$ is called *implication*. It becomes 0 if, and only if, the first argument (x_1) is 1 and the second (x_2) is 0; it is read "if x_1 then x_2" or "from x_1 follows x_2."

The function $y_9 = x_1 \sim x_2$ is called *equivalence*. It becomes 1 if both arguments have the same value, and it is 0 if the arguments have different values. It is read "x_1 is equivalent to x_2," or "x_1 if, and only if, x_2."

The function $y_8 = x_1 \,\&\, x_2$ becomes 1 if, and only if, both arguments are equal to 1. It is called *conjunction* and is read "x_1 and x_2."

The function $y_7 = x_1/x_2$ is called the *Sheffer stroke*; it is 0 if, and only if, both arguments are 1.

The function $y_6 = x_1 \bigtriangledown x_2$ is called the *Exclusive OR*; it is 1 if either the first or the second argument is 1 (but not if both are equal to 1).

The function $y_2 = x_1 \leftarrow x_2$ is called, in technical applications, the *inhibit function*. It is equal to the first argument (x_1) if the second argument (x_2) is 0; if the second argument is 1, the function becomes 0, no matter what the first argument is.

The function $y_1 = x_1 \downarrow x_2$ is called the *Pierce stroke function*; it becomes 0 if, and only if, both arguments are 0.

Now, we should also note that any function in the upper part of the table (that is, one of the functions $y_0, y_1, \ldots, y_7$) is a negation of some function from the lower part of the table (that is, one of the functions $y_8, y_9, \ldots, y_{15}$).

Consider, for example, the functions y_6 and y_9. We see from the table that $y_6 = 0$ if (and only if) $y_9 = 1$ and, conversely, $y_6 = 1$ if $y_9 = 0$. Thus, the variable y_6 may itself be considered an argument whose values uniquely determine the values assumed by variable y_9. From our definition of negation, we have $y_6 = \overline{y_9}$. But $y_6 = x_1 \nabla x_2$ and $y_9 = x_1 \sim x_2$. Consequently, $x_1 \nabla x_2 = \overline{x_1 \sim x_2}$. The table also shows that this relationship holds for all pairs of functions which are arranged symmetrically around a line dividing the seventh and eighth rows. We can write this relationship as $y_{15-i} = \overline{y_i}$, where $i = 0, 1, 2, \ldots, 15$.

Thus, the table implies that exactly half (i.e., four) of the eight two-argument functions still under discussion, are not independent.

Indeed,

$$\left.\begin{array}{lll} y_7 = \bar{y}_8, & \text{i.e.,} & x_1/x_2 = \overline{x_1 \& x_2}, \\ y_6 = \bar{y}_9, & \text{i.e.,} & x_1 \nabla x_2 = \overline{x_1 \sim x_2}, \\ y_2 = \bar{y}_{13}, & \text{i.e.,} & x_1 \leftarrow x_2 = \overline{x_1 \rightarrow x_2}, \\ y_1 = \bar{y}_{14}, & \text{i.e.,} & x_1 \downarrow x_2 = \overline{x_1 \vee x_2}. \end{array}\right\} \quad (1.2)$$

Therefore we can now drop operations defined by $/$, ∇, $\leftarrow$, and $\downarrow$, and obtain a list of six simple logical functions

$$\begin{array}{ll} \text{constant} & y = 0, \\ \text{negation} & y = \bar{x}, \\ \text{conjunction} & y = x_1 \& x_2, \\ \text{disjunction} & y = x_1 \vee x_2, \\ \text{implication} & y = x_1 \rightarrow x_2, \\ \text{equivalence} & y = x_1 \sim x_2 \end{array} \quad (1.3)$$

which are sufficient, but by no means necessary, for expressing any function of one or two independent variables in an analytical form.

To prove this, consider the function $y = \bar{x}_1 \vee x_2$. Because it is a function of two independent variables, it must be equivalent to one of those shown in Table 1.9. We shall determine which one by finding its values at all four points of the corresponding two-dimensional binary logical space, that is, at all possible values of arguments x_1 and x_2. The process of finding these values is illustrated by Table 1.10, where we use the notation $y_1 = \bar{x}_1$; consequently $y = y_1 \vee x_2$. Our values of y show that $y = x_1 \rightarrow x_2$, which means that we are dealing with the identity

$$x_1 \rightarrow x_2 = \bar{x}_1 \vee x_2. \quad (1.4)$$

Similarly, it can be shown that

$$x_1 \rightarrow x_2 = \overline{x_1 \& \bar{x}_2}, \quad (1.5)$$

$$x_1 \sim x_2 = (\bar{x}_1 \vee x_2) \& (x_1 \vee \bar{x}_2). \quad (1.6)$$

Identities (1.5) and (1.6) show that functions of one or two independent variables may be completely described without employing implication and equivalence. Thus our set of simple functions may be further reduced to the following four:

$$\left.\begin{array}{ll} \text{constant} & y=0, \\ \text{negation} & y=\bar{x}, \\ \text{conjunction} & y=x_1 \,\&\, x_2, \\ \text{disjunction} & y=x_1 \vee x_2. \end{array}\right\} \quad (1.7)$$

As we shall see below, this is the most convenient set of simple functions and hence is the one most frequently employed. However, in principle, even this set can be still further reduced.

Indeed, the procedure used to establish identities (1.4), (1.5), and (1.6), which enables us to dispense with implication and equivalence, may be employed to show that the following identities also hold:

$$\left.\begin{array}{r} x_1 \vee x_2=\overline{\bar{x}_1 \,\&\, \bar{x}_2}, \\ x_1 \,\&\, x_2=\overline{\bar{x}_1 \vee \bar{x}_2}, \\ 0=x \,\&\, \bar{x}. \end{array}\right\} \quad (1.8)$$

Table 1.10

k	0	1	2	3
x_1	0	1	0	1
x_2	0	0	1	1
y_1	1	0	1	0
y	1	0	1	1

This means that either of the last two functions, as well as the first function of set (1.7) can also be dropped. We thus arrive at a set consisting of only two functions

$$\left.\begin{array}{ll} \text{negation} & y=\bar{x}, \\ \text{conjunction} & y=x_1 \,\&\, x_2 \\ \text{(or disjunction} & y=x_1 \vee x_2), \end{array}\right\} \quad (1.9)$$

by means of which we can express any function of one or two arguments.

We shall conclude this subsection by pointing out the special properties of the Sheffer stroke $y = x_1/x_2$ and the Pierce stroke $y = x_1 \downarrow x_2$. Either of these is sufficient for complete expression of any function of one or two independent variables by virtue of the fact that both functions of the previously described set (1.0) may be expressed by either of these forms. Thus

$$\left.\begin{array}{r} \bar{x}=x/x=x \downarrow x, \\ x_1 \,\&\, x_2=(x_1/x_2)/(x_1/x_2), \\ x_1 \vee x_2=(x_1 \downarrow x_2) \downarrow (x_1 \downarrow x_2). \end{array}\right\} \quad (1.10)$$

And since the set (1.9) is sufficient for complete description, so are the two special functions.

c) Functions of n Variables. Conjunctive and Disjunctive Normal Forms

The symbolism employed with one- and two-argument functions may be extended to functions of three, four, and, in general, n independent variables; for example,

$$y = (x_1 \rightarrow \bar{x}_2) \sim (\bar{x}_1 \,\&\, x_3). \tag{1.11}$$

We can construct a correspondence table for a function of n arguments. To complete it, we scan all possible combinations of the values $x_1, x_2, \ldots, x_n$ (that is, all the points $k_0, k_1, \ldots, k_{2n-1}$ in an n-dimensional logical binary space) and determine the values of y at these points. For instance, for the function (1.11) at the point $k = 2$, that is, at $x_1 = 0$, $x_2 = 1$, $x_3 = 0$, we have $y(0, 1, 0) = 0$. Similar computations at all other points give Table 1.11.

Table 1.11

k	0	1	2	3	4	5	6	7
x_1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1
y	0	0	0	1	1	0	1	1

We shall now show that the symbolism used for functions of one or two arguments also allows us to express in analytical form any function of any number of independent variables.

As an example, consider Table 1.11. We shall assume that this table is given but that its analytical expression (1.11) is unknown, and we shall derive that expression from the table. In so doing we shall employ a procedure that is applicable to any other similar table.

Let us first consider any column in Table 1.11 in which $y = 1$: for instance, column $k = 3$. In this column $x_1 = 1$, $x_2 = 1$, $x_3 = 0$. We therefore write $y_1 = x_1 \,\&\, x_2 \,\&\, \bar{x}_3$, which, as can easily be seen, becomes 1 if, and only if, $x_1 = 1$, $x_2 = 1$, and $x_3 = 0$, that is, precisely at point $k = 3$. In an analogous manner we derive the functions

$$\left.\begin{aligned} y_2 &= \bar{x}_1 \,\&\, \bar{x}_2 \,\&\, x_3, \\ y_3 &= \bar{x}_1 \,\&\, x_2 \,\&\, x_3, \\ y_4 &= x_1 \,\&\, x_2 \,\&\, x_3, \end{aligned}\right\} \tag{1.12}$$

that become 1 only at points numbered $k = 4$, $k = 6$, and $k = 7$, respectively, that is, at those points of Table 1.11 at which $y = 1$.

Function $y = y_1 \vee y_2 \vee y_3 \vee y_4$ becomes 0 if, and only if, $y_1 = 0$, $y_2 = 0$, $y_3 = 0$ and $y_4 = 0$; in all other cases, $y = 1$. Since these "other cases" are the points $k = 3$, $k = 4$, $k = 6$, and $k = 7$, this means that function

$$y = (x_1 \& x_2 \& \bar{x}_3) \vee (\bar{x}_1 \& \bar{x}_2 \& x_3) \vee (\bar{x}_1 \& x_2 \& x_3) \vee (x_1 \& x_2 \& x_3) \quad (1.13)$$

corresponds exactly to our starting Table 1.11. We have thus obtained an analytical expression for the function given by Table 1.11. However, our new expression is not in the form of Eq. (1.11), but in another, "standard," form. While there is a marked difference in the appearance of (1.11) and (1.13), both expressions represent the same function, defined by Table 1.11; that is, we have the identity

$$\begin{gathered}(x_1 \to \bar{x}_2) \sim (\bar{x}_1 \& x_3) = (x_1 \& x_2 \& \bar{x}_3) \vee \\ \vee (\bar{x}_1 \& \bar{x}_2 \& x_3) \vee (\bar{x}_1 \& x_2 \& x_3) \vee (x_1 \& x_2 \& x_3).\end{gathered} \quad (1.14)$$

The technique just illustrated is quite general. Indeed, any function of n arguments can be given in the form of Table 1.12. Let us now take any column in which $y = 1$ and, writing out the conjunction of all the n independent variables $x_1 \& x_2 \& x_3 \& \ldots \& x_n$, let us mark with the sign of negation those variables of this column that become 0. We then form such conjunctions for all the other columns where $y = 1$, and we join them together by disjunction signs. Now we shall have an expression containing several conjunctive terms joined by disjunction signs. Each such term contains all the variables $x_1, x_2, \ldots, x_n$, some or all of which carry negation signs [for example, we may have $x_1 \& x_2 \& \ldots \& x_n$ (no negated variables), as well as $\bar{x}_1 \& \bar{x}_2 \& \ldots \& \bar{x}_n$ (all the variables negated)]. The various functions derived from the table and represented in this form can differ only in the number of disjunctive terms and in the way in which the negation signs are distributed above the variables x_i of the component conjunctions.

Expressions of this type are very important in propositional calculus: the disjunctive expression, constructed of terms which are different conjunctions of all the independent variables of a logical function, or their negations, is called the complete (or full, or perfect) *disjunctive normal form* of the function.

Table 1.12

k	0	1	2	3	...	k	...	$2^n - 1$
x_1	0	1	0	1	...	$x_1(k)$	...	1
x_2	0	0	1	1	...	$x_2(k)$	...	1
...	...	...	...	...	...	...	...	...
x_n	0	0	0	0	...	$x_n(k)$	...	1
y	$y(0)$	$y(1)$	$y(2)$	$y(3)$	...	$y(k)$	...	$y(2^n - 1)$

The complete (or full, or perfect) *conjunctive normal form* is the conjunctive expression constructed of terms which are different disjunctions of all the independent variables of a logical function, or their negations.

The term *complete* is usually omitted, that is, we speak of a *disjunctive normal* or a *conjunctive normal* form whenever it is not required that *each* term of such a form be a conjunction or a disjunction (as the case may be) of all the variables of a logical function.

Let us now consider the following property of normal forms. If a function y is expressed by a normal (either simple or full) disjunctive (or conjunctive) form, and if all the $\vee$ signs in this expression are replaced by & and all the & signs are replaced by $\vee$, and if a negation sign is placed above each variable (if the variable already carries such a sign, another identical sign is added to it; this is equivalent to a removal of negation), then we obtain function $\bar{y}$ written in normal (either simple or full) conjunctive (or disjunctive) form. This property is a direct consequence of identities (1.8).*

In contrast to simple normal forms, the full normal forms are unique in the sense that there is only one way in which each function can be represented as a full normal disjunctive or conjunctive form (that is, if we disregard permutations of disjunctive or conjunctive terms and of independent variables).

We shall illustrate the importance of these concepts by two problems.

*This is referred to as Duality or De Morgan's Law.

Problem 1. Determine whether a function n arguments $y = y(x_1, x_2, \ldots, x_n)$ can be reduced to a constant function $y = 0$.

This problem is solved by reducing the given function to its disjunctive normal form. Then, if one finds that each disjunctive term contains at least one variable in conjunction with its negation (that is, $x_i \,\&\, \bar{x}_i$), the function is of the form $y = 0$. If this is not the case, than we can always find values at which $y = 1$; that is, the function is not a constant $y = 0$.

This problem has a dual in which it is required to determine whether a given function can be reduced to the form $y = \bar{0} = 1$. The solution is obtained by reducing the given function to its conjunctive normal form. If one then finds that each conjunctive term contains the expression $x_i \vee \bar{x}_i$, then in this case (and only in this case) the given function reduces to the form $y = 1$.

The question whether some function $y = y(x_1, x_2, \ldots, x_n)$ can be reduced to the form $y = 1$ or $y = 0$ is called the *decision problem*. Within this problem, functions that reduce to the form $y = 1$ (or $y = 0$) are called *identically true* (or *false*), whereas functions that do not reduce to either $y = 1$ or $y = 0$ are called *feasible*.

Problem 2. Given a logical function of n arguments $y = y(x_1, x_2, \ldots, x_n)$, find all sets of values of arguments at which $y = 1$.

The problem would be solved if the given function could be reduced to its full disjunctive normal form.

The required number of sets of argument values is exactly equal to the number of disjunctive terms in the full disjunctive normal form of the function. The specific values of all the arguments in each set is determined in the following manner. Each set of values x_i (where $i = 1, 2, \ldots, n$) at which $y = 1$ (the values are defined by the jth parentheses) has the form

$$x_1 = x_{1j}, \quad x_2 = x_{2j}, \ \ldots, \ x_n = x_{nj},$$

where x_{ij} is equal to 0 or 1, depending on whether the corresponding ith independent variable appears in the jth conjunctive parenthesis with or without a negation sign.

d) Functions of n Variables. The Algebra of Propositional Calculus

The full disjunctive normal form (1.13) of our example defined a function for which we already had a shorter expression. In other cases, too, there exist functional expressions that are shorter and more convenient to use than the full disjunctive normal forms. In other words, there are other cases in which we can establish identities similar to (1.14).

Thus far, we have proved all identities of functions of one, two, or more variables by a test involving substitution of all the possible values of these variables. This method has two major disadvantages: it does not afford any opportunities for deriving new identities and in this sense is passive; in addition, it becomes more and more laborious as the number of variables increases. Fortunately, however, we have at our disposal another method based on the use of certain rules for identical transformations. Thus the collection of simple functions

$$y=0, \quad y=\bar{x}, \quad y=x_1 \mathbin{\&} x_2, \quad y=x_1 \vee x_2, \\ y=x_1 \rightarrow x_2, \quad y=x_1 \sim x_2 \tag{1.15}$$

may be operated upon by means of a system of rules, usually referred to as the ***algebra of logic*** or ***Boolean algebra***, which consists of the following identities:

$$\bar{\bar{x}}=x, \tag{1.16}$$

$$x_1 \rightarrow x_2 = \bar{x}_1 \vee x_2, \tag{1.17}$$

$$x_1 \sim x_2 = (x_1 \rightarrow x_2) \mathbin{\&} (x_2 \rightarrow x_1), \tag{1.18}$$

$$\text{(a) } x \mathbin{\&} x = x, \qquad \text{(b) } x \vee x = x, \tag{1.19}$$

$$\text{(a) } x \mathbin{\&} \bar{x} = 0 \qquad \text{(b) } x \vee \bar{x} = 1, \tag{1.20}$$

$$\text{(a) } x \mathbin{\&} 1 = x, \qquad \text{(b) } x \vee 1 = 1, \tag{1.21}$$

$$\text{(a) } x \mathbin{\&} 0 = 0, \qquad \text{(b) } x \vee 0 = x, \tag{1.22}$$

$$\text{(a) } \overline{x_1 \mathbin{\&} x_2} = \bar{x}_1 \vee \bar{x}_2, \qquad \text{(b) } \overline{x_1 \vee x_2} = \bar{x}_1 \mathbin{\&} \bar{x}_2, \tag{1.23}$$

$$\text{(a) } x_1 \mathbin{\&} x_2 = x_2 \mathbin{\&} x_1, \qquad \text{(b) } x_1 \vee x_2 = x_2 \vee x_1, \tag{1.24}$$

$$\text{(a) } x_1 \mathbin{\&} (x_2 \mathbin{\&} x_3) = \\ = (x_1 \mathbin{\&} x_2) \mathbin{\&} x_3, \qquad \text{(b) } x_1 \vee (x_2 \vee x_3) = \\ = (x_1 \vee x_2) \vee x_3, \tag{1.25}$$

$$\text{(a) } x_1 \mathbin{\&} (x_2 \vee x_3) = \\ = (x_1 \mathbin{\&} x_2) \vee (x_1 \mathbin{\&} x_3), \qquad \text{(b) } x_1 \vee (x_2 \mathbin{\&} x_3) = \\ = (x_1 \vee x_2) \mathbin{\&} (x_1 \vee x_3). \tag{1.26}$$

Each of these identities may be proved by direct substitution of all the possible values of the variables appearing in the left and the right sides of the identity.

The OR and AND operations of this algebra have much in common with addition and multiplication of ordinary algebra. Thus, they obey the first and second commutative laws [identities (1.24)], as well as the first and second associative laws [identities (1.25)]. However, in contrast to ordinary algebra, they obey two distributive laws rather than one [identities (1.26)]; and "reduction of like terms" or "multiplication of a variable by itself" are accomplished via identities (1.19), without introducing any factors or exponents.

This system of identities permits a purely analytical solution of a great variety of problems. Moreover, standard methods may be used for some of these solutions. For instance, any analytical function may be transformed directly into a normal disjunctive form, a procedure illustrated by the following example.

EXAMPLE. Let the starting function be

$$y = [\overline{x_1 \to (\bar{x}_1 \sim x_3)} \,\&\, (x_2 \to \bar{x}_3)] \vee (\overline{x_1 \to x_3}). \tag{1.27}$$

First, let us eliminate the $\to$ and $\sim$ signs. Applying identities (1.17) and (1.18), we obtain

$$y = [\overline{\bar{x}_1 \vee [(\bar{\bar{x}}_1 \vee x_3) \,\&\, (\bar{x}_3 \vee \bar{x}_1)]} \,\&\, (\bar{x}_2 \vee \bar{x}_3)] \vee (\overline{\bar{x}_1 \vee x_3}). \tag{1.28}$$

Next, let us eliminate those negation signs that relate not just to a single variable but to an entire aggregation of such variables. We do this by means of identities (1.23) and (1.16) to obtain

$$\begin{aligned} y &= [\bar{\bar{x}}_1 \,\&\, \overline{(\bar{\bar{x}}_1 \vee x_3) \,\&\, (\bar{x}_3 \vee \bar{x}_1)} \,\&\, (\bar{x}_2 \vee \bar{x}_3)] \vee (\bar{\bar{x}}_1 \,\&\, \bar{x}_3) = \\ &= [x_1 \,\&\, (\overline{x_1 \vee x_3}) \vee (\overline{\bar{x}_3 \vee \bar{x}_1}) \,\&\, (\bar{x}_2 \vee \bar{x}_3)] \vee (x_1 \,\&\, \bar{x}_3) = \\ &= \{x_1 \,\&\, [(\bar{x}_1 \,\&\, \bar{x}_3) \vee (x_3 \,\&\, x_1)] \,\&\, (\bar{x}_2 \vee \bar{x}_3)\} \vee (x_1 \,\&\, \bar{x}_3). \end{aligned} \tag{1.29}$$

Now, in order to arrive at a disjunctive normal form it is sufficient to expand the expression in the braces as specified by identity (1.26a). Simplifying the intermediate results by means of (1.16) and (1.19) as we go along, we obtain

$$\begin{aligned} y &= \{[(x_1 \,\&\, \bar{x}_1 \,\&\, \bar{x}_3) \vee (x_1 \,\&\, x_3 \,\&\, x_1)] \,\&\, (\bar{x}_2 \vee \bar{x}_3)\} \vee (x_1 \,\&\, \bar{x}_3) = \\ &= (x_1 \,\&\, \bar{x}_1 \,\&\, \bar{x}_3 \,\&\, \bar{x}_2) \vee (x_1 \,\&\, x_3 \,\&\, \bar{x}_2) \vee (x_1 \,\&\, \bar{x}_1 \,\&\, \bar{x}_3 \,\&\, \bar{x}_3) \vee \\ &\qquad \vee (x_1 \,\&\, x_3 \,\&\, \bar{x}_3) \vee (x_1 \,\&\, \bar{x}_3). \end{aligned} \tag{1.30}$$

The first, third, and fourth disjunctive terms of the above disjunctive normal form are 0, since they contain expressions of the form

$x \,\&\, \bar{x}$. The second and last terms lack such expressions, and therefore our function does not reduce to $y = 0$; it may therefore be written as

$$y = (x_1 \,\&\, \bar{x}_2 \,\&\, x_3) \vee (x_1 \,\&\, \bar{x}_3). \tag{1.31}$$

Thus,

$$\begin{aligned} y &= [\overline{x_1 \to (\bar{x}_1 \sim x_3)} \,\&\, (x_2 \to \bar{x}_3)] \vee \overline{(x_1 \to x_3)} = \\ &= (x_1 \,\&\, \bar{x}_2 \,\&\, x_3) \vee (x_1 \,\&\, \bar{x}_3). \end{aligned} \tag{1.32}$$

To reduce a given function to its complete disjunctive normal form, it must first be reduced to some normal disjunctive form by the methods already discussed. Let us follow the remainder of the procedure on our example.

Our normal form (1.32) is not full because its second disjunctive term does not comprise all the variables: x_2 (or $\bar{x}_2$) is missing. However, it is readily seen that the following identity is true:

$$x_1 \,\&\, \bar{x}_3 = x_1 \,\&\, \bar{x}_3 \,\&\, (x_2 \vee \bar{x}_2) = (x_1 \,\&\, x_2 \,\&\, \bar{x}_3) \vee (x_1 \,\&\, \bar{x}_2 \,\&\, \bar{x}_3). \tag{1.33}$$

Substituting the disjunction given by (1.33) for the second disjunctive term of the normal form (1.32), we obtain

$$y = (x_1 \,\&\, \bar{x}_2 \,\&\, x_3) \vee (x_1 \,\&\, x_2 \,\&\, \bar{x}_3) \vee (x_1 \,\&\, \bar{x}_2 \,\&\, \bar{x}_3) \tag{1.34}$$

which is the full disjunctive normal form of the given function. Of course, if these transformations would have given an expression containing several identical disjunctive terms, we would have retained only one of these.

Reduction to a conjunctive normal form differs from the above technique only in the last step where, instead of expanding the expression derived in the preceding steps in accordance with identity (1.26a), we use the second distributive law, that is, identity (1.26b).

EXAMPLE. Let the conjunctive normal form of a function of three variables be

$$y = x_1 \,\&\, (\bar{x}_1 \vee \bar{x}_2 \vee x_3). \tag{1.35}$$

To transform this into a complete normal form, we use the identities

$$\left.\begin{aligned} x_1 &= x_1 \vee (x_2 \,\&\, \bar{x}_2) = (x_1 \vee x_2) \,\&\, (x_1 \vee \bar{x}_2), \\ x_1 \vee x_2 &= x_1 \vee x_2 \vee (x_3 \,\&\, \bar{x}_3) = (x_1 \vee x_2 \vee x_3) \,\&\, (x_1 \vee x_2 \vee \bar{x}_3), \\ x_1 \vee \bar{x}_2 &= x_1 \vee \bar{x}_2 \vee (x_3 \,\&\, \bar{x}_3) = (x_1 \vee \bar{x}_2 \vee x_3) \,\&\, (x_1 \vee \bar{x}_2 \vee \bar{x}_3). \end{aligned}\right\} \tag{1.36}$$

and we get

$$\begin{aligned} y = (x_1 \vee x_2 \vee x_3) \,\&\, (x_1 \vee x_2 \vee \bar{x}_3) \,\&\, (x_1 \vee \bar{x}_2 \vee x_3) \,\&\, \\ \&\, (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \,\&\, (\bar{x}_1 \vee \bar{x}_2 \vee x_3), \end{aligned} \tag{1.37}$$

which is the complete conjunctive normal form of the starting function.

So far, we have demonstrated the use of Boolean algebra by reducing a given logical function of several variables to its full normal (or any normal) disjunctive (or conjunctive) form. However, the function may be given by means of a table. In this case, we can obtain a unique complete disjunctive normal form via the method already described. We can then transform this expression by means of Boolean identities and thus arrive at other analytic expressions of the same function. Given the variety of possible forms of a function, we are faced with the problem of determining which of these is optimum for our purposes. We shall return to this somewhat later.

1.4. TWO-VALUED PREDICATE CALCULUS

We shall now return to a subject which we have briefly considered at the end of Section 1.2. Thus we shall consider two-valued predicates, that is, logical functions which themselves assume only the values of 0 or 1, but whose arguments may take on values from any set whatsoever.

Predicate functions are denoted by capital letters. This permits us to distinguish visually between a predicate (a nonhomogeneous logical function) and a complex proposition (a homogeneous logical function). Thus, an n-place predicate may be written as

$$y = P(x_1, \ldots, x_n),$$

where $x_1 = \{x_{11}, \ldots, x_{1p}\}, \ldots, x_n = \{x_{n1}, \ldots, x_{nq}\}$ are the object variables and their alphabets.

Since two-valued predicates assume values from a binary set 0 and 1, they may themselves be the arguments of two-valued homogeneous logical functions; for this reason, we can apply to them the symbolism of propositional calculus. Thus, suppose we have the predicates

$$\left.\begin{array}{l} y_1 = P(x_1), \text{ where } x_1 = \{x_{11}, \ldots, x_{1p}\}, \\ y_2 = Q(x_2), \text{ where } x_2 = \{x_{21}, \ldots, x_{2q}\}. \end{array}\right\} \quad (1.38)$$

We can subject these predicates to any one of the operations of propositional calculus to obtain a new predicate; for example,

$$R(x_1, x_2) = P(x_1) \vee Q(x_2). \quad (1.39)$$

This use of operations of propositional calculus permits us to achieve several ends. To begin with, we can relate several simple predicates to each other and form a compound predicate, as in the above example. Also, we can relate predicates to any and all simple propositions, as well as to the compound propositions that can be formed from the simple ones by the same operations of propositional calculus. Thus from the predicates (1.38) and the binary logical variables

$$x_3 = \{0, 1\}, \quad x_4 = \{0, 1\}$$

we can construct a composite function, for example,

$$z = \{P(x_1) \rightarrow [Q(x_2) \vee (x_3 \,\&\, \bar{x}_4)]\} \sim x_3 \quad (1.40)$$

where Z can only be two-valued.

The only variables which we have encountered in the compound function of propositional calculus were the simple prepositions. In the predicate calculus, however, not only simple propositions, but also the object variables of the predicates, as well as *variable predicates* can act as variables. The presence of these elements constitutes the main characteristic of this calculus, and necessitates new operations that are qualitatively different from those employed in propositional calculus. The operators corresponding to these new operations are called *quantifiers*.

There are two types of quantifiers: the *universal* and the *existential*.

The universal quantifier is an operator that matches any one-place predicate $y = P(x)$ with the binary logical variable Z which becomes 1 if, and only if, $y = 1$ at all values of x. This is written

$$z = (\forall x) P(x),$$

where "$\forall x$" is the universal quantifier. The above expression is then read as "for all x there is $P(x)$."

The existential quantifier is an operator that matches a one-place predicate $y = P(x)$ with a binary logical variable z which becomes 0 if, and only if, $y = 0$ at all values of x. This is written

$$z = (\exists x) P(x),$$

where "$\exists x$" is the existential quantifier. The above expression is then read as "there is an x such that $y = P(x)$."

Let us discuss some general properties of these operators. In accordance with the definitions of quantifiers, the logical variable z in

$$\left.\begin{aligned} z &= (\forall x) P(x), \\ z &= (\exists x) P(x) \end{aligned}\right\} \tag{1.41}$$

is not a function of the object variable x; here, z is an "integral" characteristic of the predicate $P(x)$. To underscore the absence of functional dependence of z on x, the object variable x in such cases is said to be ***bound***. Object variables that are not bound are said to be *free*. Of course, the universal and existential quantifiers may also be applied to functions of propositional calculus. But if we do that, then they degenerate into finite conjunctions and disjunctions. Indeed, suppose we have a function $y = y(x_1, \ldots, x_n)$ in which both the variables and the function are two-valued logical variables. The same function may be given in the form $y = y(k)$, where k is a numeral denoting a point in an n-dimensional binary logical space. From the definition of quantifiers, we have

$$(\forall k) y(k) = y(0) \& y(1) \& \ldots \& y(k) \& \ldots \& y(2^n - 1),$$
$$(\exists k) y(k) = y(0) \vee y(1) \vee \ldots \vee y(k) \vee \ldots \vee y(2^n - 1).$$

For this reason we can consider the universal and existential quantifiers as generalized conjunction and generalized disjunction, respectively. And because of the analogy between conjunction or disjunction and the summation of real numbers, one can draw an analogy between the operations specified by quantifiers and the integration of functions of a real variable. If one applies a quantifier (either universal or existential) to an m-place (rather than a one-place) predicate, the result is again a predicate; this time it is, however, an $(m - 1)$-place predicate since one object variable becomes bound.

Thus, in dealing with predicates, we employ not only the operations of propositional calculus, but also operations involving binding of object variables by universal and existential quantifiers. The calculus in which the above operations are used to construct compound functions is called *restricted predicate calculus.*

This new operation of binding by quantifiers introduces identities which differ from those of the Section 1.3. Examples of such identities are

$$\overline{(\forall x) P(x)} = (\exists x) \overline{P(x)}, \tag{1.42a}$$

$$\overline{(\exists x) P(x)} = (\forall x) \overline{P(x)}. \tag{1.42b}$$

The identities of propositional calculus, supplemented by identities (1.42), comprise a mechanism useful for solving a variety of problems. As in propositional calculus, the most important problem of predicate calculus is that of decision, but because the independent variables are different, the manner in which this problem posed is also somewhat different.

Thus, the decision problem of propositional calculus in determining whether a given compound function is identically true, feasible, or identically false. However, the following must be asked in predicate calculus: (a) Is a given compound function identically true; that is, does it assume the value of 1 with any object variable and any predicate? Or (b) Is it identically true only over a certain set of object variables; that is, does it assume the value of 1 only over a certain set of object variables and for any predicate from this set? Or (c) Is it feasible; that is, does it assume the value of 1 at some values of object variables and at some predicates? And, finally, (d) Is it identically false, that is, unfeasible? In contrast to the case of propositional calculus, the decision problem of predicate calculus can be solved only for special kinds of compound functions.