4

LATTICES AND BOOLEAN ALGEBRA

INTRODUCTION

The statement algebra of Chap. 1 and the algebra of sets given in Chap. 2 provide a motivation for the study of an abstract algebraic system possessing all the essential properties of these algebras. Such an algebraic system was introduced by George Boole in 1854 and is known as Boolean algebra. Before we study Boolean algebra in this chapter, we consider a more general algebraic system called a lattice. A Boolean algebra is then introduced as a special lattice.

A basic difference between the algebraic systems studied in this chapter and those given in Chap. 3 is the fact that the ordering relation plays a significant role in the algebraic systems studied here. In order to emphasize the role of an ordering relation, a lattice is first introduced as a partially ordered set, followed by the definition of a lattice as an algebraic system.

Both lattices and Boolean algebra have important applications in the theory and design of computers. There are many other areas such as engineering and science to which Boolean algebra is applied.

4-1 LATTICES AS PARTIALLY ORDERED SETS

In this section we introduce a lattice as a partially ordered set satisfying certain properties. Partially ordered sets, their properties, and associated terminology given in Sec. 2-3.9 will be used throughout our discussion here. In particular, the notion of the least upper bound (LUB) and the greatest lower bound (GLB) of a subset of a partially ordered set will be used repeatedly.

4-1.1 Definition and Examples

Definition 4-1.1 A *lattice* is a partially ordered set $\langle L, \leq \rangle$ in which every pair of elements $a, b \in L$ has a greatest lower bound and a least upper bound.

The greatest lower bound of a subset $\{a, b\} \subseteq L$ will be denoted by $a * b$ and the least upper bound by $a \oplus b$. It is customary to call the GLB $\{a, b\} = a * b$ the *meet* or *product* of a and b, and the LUB $\{a, b\} = a \oplus b$ the *join* or *sum* of a and b. Other symbols such as $\wedge$ and $\vee$ or $\cdot$ and $+$ are also used to denote the meet and join of two elements respectively. When using the symbols $\cdot$ and $+$ it is not uncommon to supress the dot and write $a \cdot b$ simply as ab. In certain cases, the symbols $\cap$ and $\cup$ are also used to denote the meet and join respectively. It follows from the definition of a lattice that both $*$ and $\oplus$ are binary operations on L because of the uniqueness of the least upper bound and greatest lower bound of any subset of a partially ordered set.

A totally ordered set is trivially a lattice, but not all partially ordered sets are lattices, as can be seen from the Hasse diagrams of some of the partially ordered sets given in Figs. 4-1.1 and 4-1.2. For the sake of brevity, throughout this chapter we shall refer to the Hasse diagrams simply as the diagrams of partially ordered sets. Naturally, the diagram of a totally ordered set is a chain.

The following are some examples of lattices. These examples will be referred to frequently throughout this chapter.

EXAMPLE 1 Let S be any set and $\rho(S)$ be its power set. The partially ordered set $\langle \rho(S), \subseteq \rangle$ is a lattice in which the meet and join are the same as the operations $\cap$ and $\cup$ respectively. In particular, when S has a single element, the corresponding lattice is a chain containing two elements. When S has two and three elements, the diagrams of the corresponding lattices are as shown in Fig. 4-1.1*b* and *f* respectively.

EXAMPLE 2 Let $\mathbf{I}_+$ be the set of all positive integers, and let D denote the relation of "division" in $\mathbf{I}_+$ such that for any $a, b \in \mathbf{I}_+$, $a\ D\ b$ iff a divides b. Then $\langle \mathbf{I}_+, D \rangle$ is a lattice in which the join of a and b is given by the least common multiple (LCM) of a and b, that is, $a \oplus b$ = LCM of a and b, and the meet of a and b, that is, $a * b$ is the greatest common divisor (GCD) of a and b.

EXAMPLE 3 Let n be a positive integer and S_n be the set of all divisors of n; for example, $n = 6$, $S_6 = \{1, 2, 3, 6\}$ and for $n = 24$, $S_{24} = \{1, 2, 3, 4, 6, 8, 12, 24\}$. Let D denote the relation of "division" as defined in Example 2. The

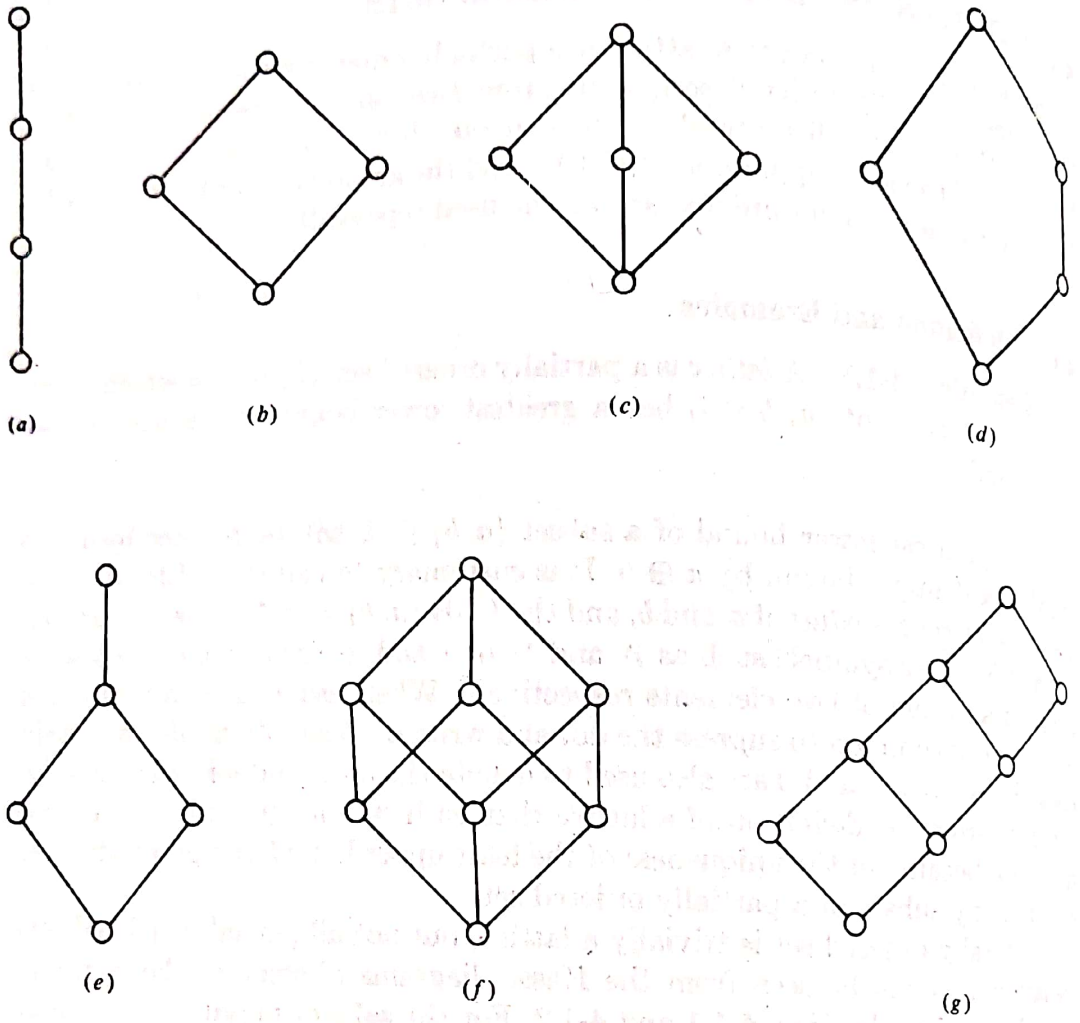

FIGURE 4-1.1 Lattices.

lattices $\langle S_6, D\rangle$, $\langle S_{24}, D\rangle$, $\langle S_8, D\rangle$, and $\langle S_{30}, D\rangle$ are given in Fig. 4-1.1*b*, *g*, *a*, and *f* respectively.

EXAMPLE 4 Let S be a nonempty set and $\Pi(S)$ be the set of all partitions of S. Two binary operations $*$ and $\oplus$ on $\Pi(S)$ were introduced in Sec. 3-2.1. We can also define a corresponding partial ordering relation $\leq$ on $\Pi(S)$ such that for $\Pi_1, \Pi_2 \in \Pi(S)$, $\Pi_1 \leq \Pi_2$ iff every block of Π_1 is a subset of some block of Π_2. It is easy to see that $\langle \Pi(S), \leq \rangle$ is a lattice in which the operations $*$ and $\oplus$ are the required meet and join respectively.

In particular, let $S = \{a, b, c\}$; then

$$\Pi(S) = \{\Pi_1, \Pi_2, \Pi_3, \Pi_4, \Pi_5\}$$

where

$$\Pi_1 = \{\overline{a, b, c}\} \qquad \Pi_2 = \{\overline{a, b}, \bar{c}\} \qquad \Pi_3 = \{\overline{a, c}, \bar{b}\}$$

$$\Pi_4 = \{\bar{a}, \overline{b, c}\} \qquad \text{and} \qquad \Pi_5 = \{\bar{a}, \bar{b}, \bar{c}\}$$

The diagram of $\langle \Pi(S), \leq \rangle$ is given in Fig. 4-1.1*c*.

One can show that there are 15 partitions of a set of four elements, 52 partitions of a set of five elements, and so on.

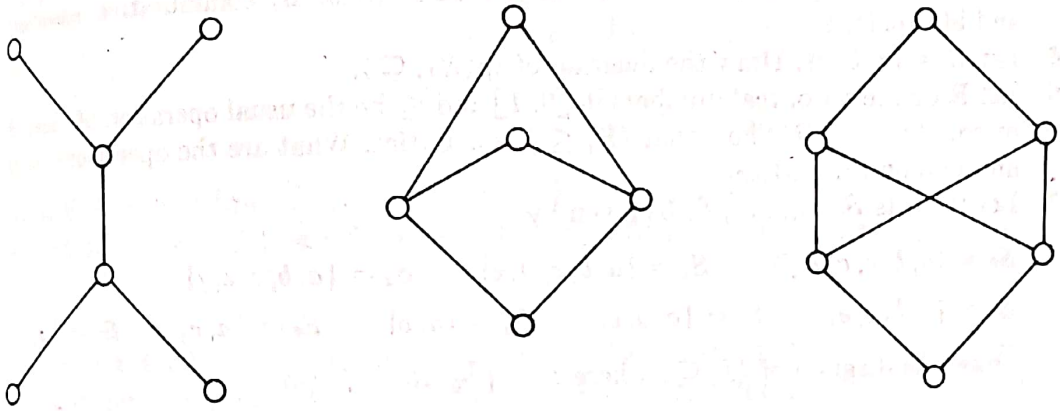

FIGURE 4-1.2 Partially ordered sets which are not lattices.

The previous examples show that different lattices can be represented by the same diagram except that the nodes have different labels. We show in Sec. 4-1.4 that different partially ordered sets may be represented by the same diagram if they are order-isomorphic.

Recall that for any partial ordering relation $\leq$ on a set S, the converse relation $\geq$ is also a partial ordering relation on S. The diagram of $\langle S, \geq \rangle$ can be obtained from that of $\langle S, \leq \rangle$ by simply turning it upside down. The partially ordered sets $\langle S, \leq \rangle$ and $\langle S, \geq \rangle$ are called duals of each other. If $A \subseteq S$, then LUB A with respect to the relation $\leq$ is the same as GLB A with respect to the relation $\geq$, and vice versa. In other words, the GLB and LUB are interchanged if we interchange the relations $\leq$ and $\geq$. In terms of lattices, we can say that the operations of meet and join on $\langle L, \leq \rangle$ become the operations of join and meet on $\langle L, \geq \rangle$. In any case, $\langle L, \geq \rangle$ is a lattice if $\langle L, \leq \rangle$ is a lattice. We may now formulate the *principle of duality* of lattices as follows.

Any statement about lattices involving the operations $*$ and $\oplus$ and the

relations $\leq$ and $\geq$ remains true if $*$ is replaced by $\oplus$, $\oplus$ by $*$, $\leq$ by $\geq$, and $\geq$ by $\leq$.

The operations $*$ and $\oplus$ are called *duals* of each other as are the relations $\leq$ and $\geq$. Similarly, the lattices $\langle L, \leq \rangle$ and $\langle L, \geq \rangle$ are called duals of each other.

EXERCISES 4-1.1

1 Explain why the partially ordered sets given in Fig. 4-1.2 are not lattices.
2 Draw the diagrams of lattices $\langle S_n, D \rangle$ given in Example 3 for $n = 4, 6, 10, 12, 15, 45,$ 60, 75, and 210. For what values of n do you expect $\langle S_n, D \rangle$ to be a chain?
3 Show that there are 15 partitions of a set of four elements. Draw the diagram of the corresponding lattice.
4 Show that the operations of meet and join on a lattice are commutative, associative, and idempotent.
5 Let $S = \{a, b, c\}$. Draw the diagram of $\langle \rho(S), \subseteq \rangle$.
6 Let $\mathbf{R}$ be the set of real numbers in $[0, 1]$ and $\leq$ be the usual operation of "less than or equal to" on $\mathbf{R}$. Show that $\langle \mathbf{R}, \leq \rangle$ is a lattice. What are the operations of meet and join on this lattice?
7 Let the sets $S_0, S_1, \ldots, S_7$ be given by

$S_0 = \{a, b, c, d, e, f\}$ $S_1 = \{a, b, c, d, e\}$ $S_2 = \{a, b, c, e, f\}$
$S_3 = \{a, b, c, e\}$ $S_4 = \{a, b, c\}$ $S_5 = \{a, b\}$ $S_6 = \{a, c\}$ $S_7 = \{a\}$

Draw the diagram of $\langle L, \subseteq \rangle$ where $L = \{S_0, S_1, \ldots, S_7\}$.

4-1.2 Some Properties of Lattices

We shall first list some of the properties of the two binary operations of meet and join denoted by $*$ and $\oplus$ on a lattice $\langle L, \leq \rangle$. For any $a, b, c \in L$, we have

(L-1)	$a * a = a$	**(L-1)′**	$a \oplus a = a$ *(Idempotent)*
(L-2)	$a * b = b * a$	**(L-2)′**	$a \oplus b = b \oplus a$ *(Commutative)*
(L-3)	$(a * b) * c = a * (b * c)$	**(L-3)′**	$(a \oplus b) \oplus c = a \oplus (b \oplus c)$ *(Associative)*
(L-4)	$a * (a \oplus b) = a$	**(L-4)′**	$a \oplus (a * b) = a$ *(Absorption)*

The identities **(L-1)** to **(L-4)** can be proved using the definitions of the operators $*$ and $\oplus$. The identities **(L-1)′** to **(L-4)′** then follow from the principle of duality. The latter identities can also be proved directly. We shall prove the identity **(L-4)**.

For any $a \in L$, $a \leq a$ and $a \leq a \oplus b$ by definition of $\oplus$; hence $a \leq a * (a \oplus b)$. On the other hand, $a * (a \oplus b) \leq a$ by the definition of $*$. Therefore, $a * (a \oplus b) = a$.

These identities along with the following theorem will be used in defining a lattice as an algebraic system in the next section.

Theorem 4-1.1 Let $\langle L, \leq \rangle$ be a lattice in which $*$ and $\oplus$ denote the operations of meet and join respectively. For any $a, b \in L$,

$$a \leq b \Leftrightarrow a * b = a \Leftrightarrow a \oplus b = b$$

PROOF We shall first prove that $a \leq b \Leftrightarrow a * b = a$. In order to do this, let us assume that $a \leq b$. We also know that $a \leq a$. Therefore $a \leq a * b$. But from the definition of $a * b$, we have $a * b \leq a$. Hence $a \leq b \Rightarrow a * b = a$. Next, assume that $a * b = a$; but it is only possible if $a \leq b$, that is, $a * b = a \Rightarrow a \leq b$. Combining these two results, we get the required equivalence.

It is possible to show that $a \leq b \Leftrightarrow a \oplus b = b$ in a similar manner. Alternatively, from $a * b = a$, we have

$$b \oplus (a * b) = b \oplus a = a \oplus b$$

but

$$b \oplus (a * b) = b$$

Hence $a \oplus b = b$ follows from $a * b = a$. By repeating similar steps, we can show that $a * b = a$ follows from $a \oplus b = b$, and hence these are equivalent.

////

Theorem 4-1.1 establishes a connection between the partial ordering relation $\leq$ and the two binary operations $*$ and $\oplus$ on the meet and join in a lattice $\langle L, \leq \rangle$. We shall use this result in Sec. 4-1.3 to show that a lattice can be defined as an algebraic system. We now prove some basic inequalities that hold between the elements of a lattice.

Theorem 4-1.2 Let $\langle L, \leq \rangle$ be a lattice. For any $a, b, c \in L$, the following properties called *isotonicity* hold.

$$b \leq c \quad \Rightarrow \quad \begin{cases} a * b \leq a * c \\ a \oplus b \leq a \oplus c \end{cases}$$

PROOF From Theorem 4-1.1,

$$b \leq c \Leftrightarrow b * c = b$$

To show that $a * b \leq a * c$, we shall show that

$$(a * b) * (a * c) = a * b$$

Note that

$$(a * b) * (a * c) = (a * a) * (b * c) = a * (b * c) = a * b$$

The second result can be proved in a similar manner.

////

We shall now list some implications which hold for any $a, b, c \in L$ where $\langle L, \leq \rangle$ is a lattice. These implications follow from the definitions of the opera-

tions $*$ and $\oplus$ on L. They can also be proved by using the properties of isotonicity.

$$a \leq b \wedge a \leq c \Rightarrow a \leq b \oplus c \tag{1}$$

$$a \leq b \wedge a \leq c \Rightarrow a \leq b * c \tag{2}$$

Of course (1) is obvious from the definition of $\oplus$. Implication (2) can also be proved from the definition of $*$ and from the fact that both b and c are comparable to a. It can also be proved by using Theorem 4-1.2. In a similar manner, we can write the duals of (1) and (2) as

$$a \geq b \wedge a \geq c \Rightarrow a \geq b * c \tag{3}$$

$$a \geq b \wedge a \geq c \Rightarrow a \geq b \oplus c \tag{4}$$

We shall frequently employ these implications in our proofs.

Theorem 4-1.3 Let $\langle L, \leq \rangle$ be a lattice. For any $a, b, c \in L$, the following inequalities, called the *distributive inequalities*, hold:

$$a \oplus (b * c) \leq (a \oplus b) * (a \oplus c)$$

$$a * (b \oplus c) \geq (a * b) \oplus (a * c)$$

PROOF From $a \leq a \oplus b$ and $a \leq a \oplus c$ we have, using (2),

$$a \leq (a \oplus b) * (a \oplus c) \tag{5}$$

$$b * c \leq b \leq a \oplus b$$

and

$$b * c \leq c \leq a \oplus c$$

Hence, by using (2) again we get

$$b * c \leq (a \oplus b) * (a \oplus c) \tag{6}$$

From (5) and (6) and by using (4), we get the required inequality

$$a \oplus (b * c) \leq (a \oplus b) * (a \oplus c)$$

The second distributive inequality can be proved in a similar manner or by using the principle of duality. ////

Theorem 4-1.4 Let $\langle L, \leq \rangle$ be a lattice. For any $a, b, c \in L$ the following holds:

$$a \leq c \Leftrightarrow a \oplus (b * c) \leq (a \oplus b) * c \tag{7}$$

PROOF Since $a \leq c \Leftrightarrow a \oplus c = c$ from Theorem 4-1.1, we get the required result by substituting c for $a \oplus c$ in the first distributive inequality. One could prove the above equivalence directly using an argument similar to the one given in the proof of Theorem 4-1.3. ////

The inequality given in Theorem 4-1.4 is called the *modular inequality*. There are other ways in which the modular inequalities are expressed:

$$(a * b) \oplus (a * c) \leq a * [b \oplus (a * c)] \tag{8}$$

$$(a \oplus b) * (a \oplus c) \geq a \oplus [b * (a \oplus c)] \tag{9}$$

The method of proof is similar to the one used in proving Theorem 4-1.3. We shall leave it as an exercise.

EXERCISES 4-1.2

1 Show that the identities (**L-1**) and (**L-1**)′ follow from the identities (**L-2**) to (**L-4**) and their duals.

2 Complete the proof of Theorem 4-1.1 by showing that in a lattice

$$a \leq b \Leftrightarrow a \oplus b = b$$

3 Show that in a lattice if $a \leq b \leq c$, then

$$a \oplus b = b * c$$

and

$$(a * b) \oplus (b * c) = b = (a \oplus b) * (a \oplus c)$$

4 Show that in a lattice if $a \leq b$ and $c \leq d$, then $a * c \leq b * d$.

5 In a lattice, show that

$$(a * b) \oplus (c * d) \leq (a \oplus c) * (b \oplus d)$$

$$(a * b) \oplus (b * c) \oplus (c * a) \leq (a \oplus b) * (b \oplus c) * (c \oplus a)$$

6 Show that a lattice with three or fewer elements is a chain.

7 Prove that every finite subset of a lattice has an LUB and a GLB. (*Hint:* Use th principle of mathematical induction.) What can you say about a finite lattice?

8 Prove inequalities (8) and (9).

9 Show that Theorem 4-1.4 is a self-dual.

4-1.3 Lattices as Algebraic Systems

In this section we define a lattice as an algebraic system on which it is possible to define a partial ordering relation. The advantage of considering a lattice as an algebraic system is that many concepts which are associated with algebraic systems can be applied to lattices as well. Thus it is possible to define sublattices, direct product of lattices, and also lattice homomorphisms.

Definition 4-1.2 A *lattice* is an algebraic system $\langle L, *, \oplus \rangle$ with two binary operations $*$ and $\oplus$ on L which are both (1) commutative and (2) associative and (3) satisfy the absorption laws. In other words, the operations $*$ and $\oplus$ satisfy the identities (**L-2**) to (**L-4**) and (**L-2**)′ to (**L-4**)′ given in Sec. 4-1.2.

The absence of the identities (**L-1**) and (**L-1**)′ in the definition here is due to the fact that (**L-4**) and its dual imply the identities (**L-1**) and (**L-1**)′ as follows. For any $a \in L$,

$$a * a = a * [a \oplus (a * a)] = a$$

where we have replaced the second a in $a * a$ by $a \oplus (a * a)$ and then from (**L-4**)′ obtained a in the second step. The identity $a \oplus a = a$ can be proved in a similar manner or by the principle of duality.

Note that Definition 4-1.2 does not assume the existence of any partial

ordering on L. We shall now show that a partial ordering relation on L follows as a consequence of the properties of the operations $*$ and $\oplus$.

Let us define a relation R on L such that for $a, b \in L$

$$a \, R \, b \Leftrightarrow a * b = a$$

Obviously, for any $a \in L$, $a * a = a$, so that $a \, R \, a$, or the relation R is reflexive. Now for some $a, b \in L$ let us assume that $a \, R \, b$ and $b \, R \, a$, so that $a * b = a$ and $b * a = b$. But $a * b = b * a$, and so $a = b$. The assumptions $a \, R \, b$ and $b \, R \, a$ imply $a = b$, or that the relation R is antisymmetric. Finally, let us assume that for some $a, b, c \in L$, $a \, R \, b$ and $b \, R \, c$. This requires that $a * b = a$ and $b * c = b$. Thus, $a * c = (a * b) * c = a * (b * c) = a * b = a$, or $a \, R \, c$. The last step shows that the relation R is transitive. From this we can conclude that R is a partial ordering relation.

It is easy to show that $a * b = a \Leftrightarrow a \oplus b = b$. Hence we could have defined the same partial ordering relation R on L as

$$a \, R \, b \Leftrightarrow a \oplus b = b \qquad \text{for any } a, b \in L$$

Our next step is to show that for any two elements $a, b \in L$, the greatest lower bound and the least upper bound of $\{a, b\} \subseteq L$ with respect to the partial ordering R are $a * b$ and $a \oplus b$, respectively.

From the absorption laws $a * (a \oplus b) = a$ and $b * (a \oplus b) = b$, we have $a \, R \, (a \oplus b)$ and $b \, R \, (a \oplus b)$. Let us now assume that there exists an element $c \in L$ such that $a \, R \, c$ and $b \, R \, c$. This means that

$$a \oplus c = c \qquad \text{and} \qquad b \oplus c = c$$

or

$$(a \oplus c) \oplus (b \oplus c) = (a \oplus b) \oplus c = c \oplus c = c$$

implying that $(a \oplus b) \, R \, c$. The last step shows that $a \oplus b$ is the least upper bound of a and b. In a similar manner, we can show that $a * b$ is the greatest lower bound of $\{a, b\}$ with respect to the partial ordering relation R. We can summarize the discussion by saying that on a lattice $\langle L, *, \oplus \rangle$ it is possible to define a partial ordering relation R such that for any $a, b \in L$

$$a \, R \, b \Leftrightarrow a * b = a \Leftrightarrow a \oplus b = b$$

and that LUB $\{a, b\} = a \oplus b$ and GLB $\{a, b\} = a * b$ with respect to the relation R on L.

On the other hand, it was shown earlier in Sec. 4-1.1 that in a lattice $\langle L, \leq \rangle$ defined as a partially ordered set, it is possible to define two binary operations $*$ and $\oplus$ such that for any $a, b \in L$

$$a * b = \text{GLB } \{a, b\} \qquad \text{and} \qquad a \oplus b = \text{LUB } \{a, b\}$$

and

$$a \leq b \Leftrightarrow a * b = a \Leftrightarrow a \oplus b = b$$

where the operations $*$ and $\oplus$ are both commutative and associative and satisfy the absorption laws. This establishes the equivalence of the two definitions where the relation R is the same as the relation $\leq$ on L.

4-1.4 Sublattices, Direct Product, and Homomorphism

The advantage of defining a lattice as an algebraic system is that we can introduce the concept of sublattices in a natural way.

Definition 4-1.3 Let $\langle L, *, \oplus \rangle$ be a lattice and let $S \subseteq L$ be a subset of L. The algebra $\langle S, *, \oplus \rangle$ is a *sublattice* of $\langle L, *, \oplus \rangle$ iff S is closed under both operations $*$ and $\oplus$.

From the definition it follows that a sublattice itself is a lattice. However, any subset of L which is a lattice need not be a sublattice, as will be shown by an example. Note that for a partially ordered set the situation is simpler in the sense that every subset of a partially ordered set is also a partially ordered set under the same partial ordering relationship. Thus, if $\langle P, \leq \rangle$ is a partially ordered set and $Q \subseteq P$, then $\langle Q, \leq \rangle$ is also a partially ordered set.

For a lattice $\langle L, *, \oplus \rangle$ and for any two elements $a, b \in L$ such that $a \leq b$, the closed *interval* $[a, b]$ consisting of all the elements $x \in L$ such that $a \leq x \leq b$ is a sublattice of L.

EXAMPLE 1 Let $\langle L, \leq \rangle$ be a lattice in which $L = \{a_1, a_2, \ldots, a_8\}$ and S_1, S_2, and S_3 be the subsets of L given by $S_1 = \{a_1, a_2, a_4, a_6\}$, $S_2 = \{a_3, a_5, a_7, a_8\}$, and $S_3 = \{a_1, a_2, a_4, a_8\}$. The diagram of $\langle L, \leq \rangle$ is given in Fig. 4-1.3. Observe that $\langle S_1, \leq \rangle$ and $\langle S_2, \leq \rangle$ are sublattices of $\langle L, \leq \rangle$, but $\langle S_3, \leq \rangle$ is not a sublattice, because $a_2, a_4, \in S_3$ but $a_2 * a_4 = a_6 \notin S_3$. Note that $\langle S_3, \leq \rangle$ is a lattice.

EXAMPLE 2 The lattice of divisors of any positive integer n given in Example 3, Sec. 4-1.1, and denoted by $\langle S_n, D \rangle$ is a sublattice of $\langle \mathbf{I}_+, D \rangle$ given in Example 2 of the same section.

EXAMPLE 3 Let S be any set and $\rho(S)$ be its power set. It was shown in Example 1, Sec. 4–1.1, that $\langle \rho(S), \subseteq \rangle$ is a lattice in which the meet and join are the usual operations of intersection and union respectively. A family of subsets of S such that for any two subsets A and B in this family both $A \cap B$ and $A \cup B$ are in the family, is obviously a sublattice of $\langle \rho(S), \subseteq \rangle$. Such a family is called a *ring of subsets* of S and is denoted by $\langle R(S), \subseteq \rangle$. The lattice $\langle R(S), \cap, \cup \rangle$ is not a ring in the sense of the definition of a ring given in Sec. 3-5.5; that is why some authors prefer to call it a *lattice of subsets*.

As a particular case of Example 3, let $S = \{p, q, r\}$. The diagram of the lattice $\langle \rho(S), \cap, \cup \rangle$ is the same as given in Fig. 4-1.3 in which $a_1 = S = \{p, q, r\}$, $a_2 = \{p, q\}$, $a_3 = \{p, r\}$, $a_4 = \{q, r\}$, $a_5 = \{p\}$, $a_6 = \{q\}$, $a_7 = \{r\}$, and $a_8 = \varnothing$. The sets S_1 and S_2 given in Example 1 are both examples of a ring of subsets of S and are sublattices of $\langle \rho(S), \cap, \cup \rangle$.

Definition 4-1.4 Let $\langle L, *, \oplus \rangle$ and $\langle S, \wedge, \vee \rangle$ be two lattices. The algebraic system $\langle L \times S, \cdot, + \rangle$ in which the binary operations $+$ and $\cdot$ on $L \times S$

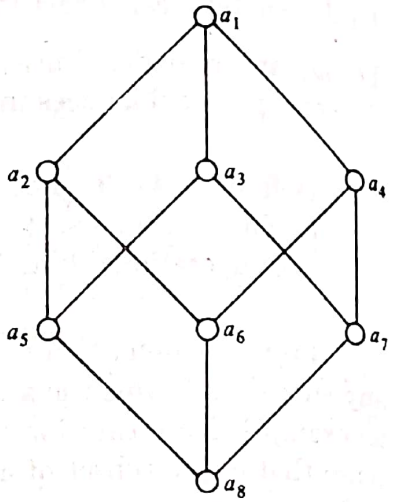

FIGURE 4-1.3

are such that for any $\langle a_1, b_1 \rangle$ and $\langle a_2, b_2 \rangle$ in $L \times S$

$$\langle a_1, b_1 \rangle \cdot \langle a_2, b_2 \rangle = \langle a_1 * a_2, b_1 \wedge b_2 \rangle$$
$$\langle a_1, b_1 \rangle + \langle a_2, b_2 \rangle = \langle a_1 \oplus a_2, b_1 \vee b_2 \rangle$$

is called the *direct product* of the lattices $\langle L, *, \oplus \rangle$ and $\langle S, \wedge, \vee \rangle$.

The operations $+$ and $\cdot$ on $L \times S$ are commutative and associative and satisfy the absorption laws because they are defined in terms of the operations $*$, $\oplus$ and $\wedge$, $\vee$. Therefore, the direct product is itself a lattice. Since $\langle L \times S, \cdot, + \rangle$ is a lattice, we can form a direct product of this lattice with another lattice, and so on. As before, we shall write $L \times L$ as L^2 and $L \times L \times L$ as L^3. The order of the lattice formed by the direct product of two lattices is equal to the product of the orders of the lattices appearing in the direct product. It should be noted that not all lattices can be written as a direct product of other lattices. The direct product of lattices can be used to construct large lattices from smaller ones.

EXAMPLE 4 Let $L = \{0, 1\}$ and the lattice $\langle L, \leq \rangle$ be as shown in Fig. 4-1.4. The lattices $\langle L^2, \leq_2 \rangle$, $\langle L^3, \leq_3 \rangle$ are shown in Fig. 4-1.4. In general, the diagram of $\langle L^n, \leq_n \rangle$ is an n cube.

Note that in the lattice $\langle L^n, \leq_n \rangle$ any element can be written as $\langle a_1, a_2, \ldots, a_n \rangle$ in which a_i is either 0 or 1 for $i = 1, 2, \ldots, n$. The partial ordering relation $\leq_n$ on L^n can be defined for any $a, b \in L^n$, where $a = \langle a_1, a_2, \ldots, a_n \rangle$ and $b = \langle b_1, b_2, \ldots, b_n \rangle$, as

$$a \leq_n b \Leftrightarrow a_i \leq b_i \qquad \text{for all } i = 1, 2, \ldots, n$$

where $\leq$ means the relation of "less than or equal to" on $\{0, 1\}$. The operations $*$ and $\oplus$ on L^n can also be defined easily. The lattice $\langle L^n, \leq_n \rangle$ will be called the *lattice of n-tuples of 0 and 1*.

EXAMPLE 5 Consider the chains of divisors of 4 and 9, that is, $L_1 = \{1, 2, 4\}$ and $L_2 = \{1, 3, 9\}$, and the partial ordering relation of "division" on L_1 and L_2. The lattice $L_1 \times L_2$ is shown in Fig. 4-1.5. Notice that the diagram of the lattice

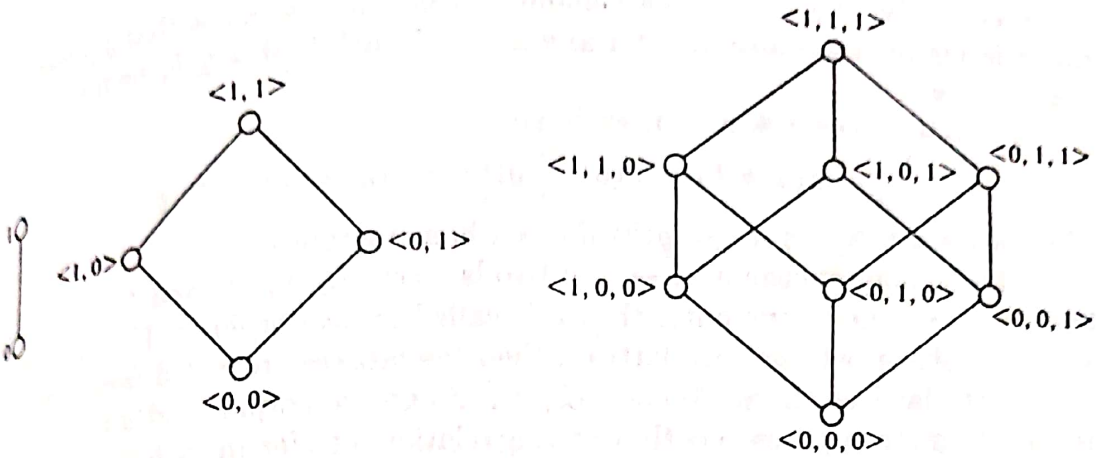

FIGURE 4-1.4

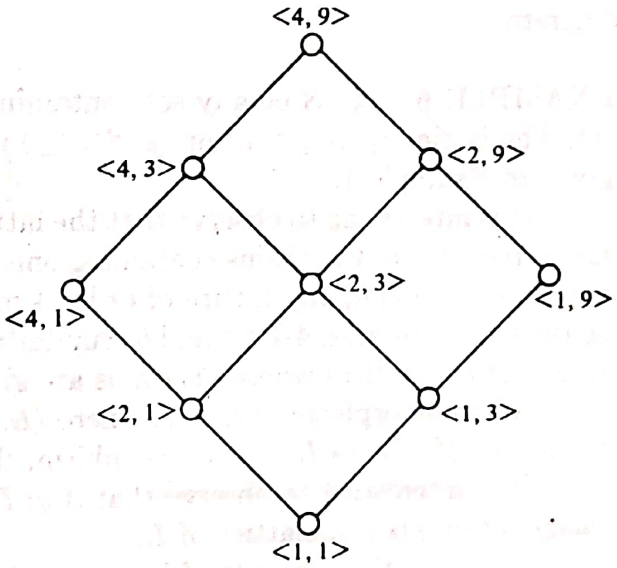

FIGURE 4-1.5

of divisors of 36 is the same as the one given in Fig. 4-1.5 except that the node $\langle a, b \rangle$ is replaced by the product ab.

Definition 4-1.5 Let $\langle L, *, \oplus \rangle$ and $\langle S, \wedge, \vee \rangle$ be two lattices. A mapping $g: L \to S$ is called a *lattice homomorphism* from the lattice $\langle L, *, \oplus \rangle$ to $\langle S, \wedge, \vee \rangle$ if for any $a, b \in L$,

$$g(a * b) = g(a) \wedge g(b) \qquad \text{and} \qquad g(a \oplus b) = g(a) \vee g(b)$$

Observe that both the operations of meet and join are preserved. There may be mappings which preserve only one of the two operations. Such mappings are not lattice homomorphisms.

Let $\langle L, *, \oplus \rangle$ and $\langle S, \wedge, \vee \rangle$ be two lattices and the partial ordering relations on L and S corresponding to the operations of meet and join be $\leq$ and

$\leq'$ respectively. If $g: L \to S$ is a homomorphism, then we show that g preserves the ordering relations also; i.e., for any $a, b \in L$ such that $a \leq b$, we must have $g(a) \leq' g(b)$.

From $a \leq b \Leftrightarrow a * b = a$, we have

$$g(a * b) = g(a) \wedge g(b) = g(a) \Leftrightarrow g(a) \leq' g(b)$$

This means $a \leq b \Rightarrow g(a) \leq' g(b)$ if g is a homomorphism.

If a homomorphism $g: L \to S$ of two lattices $\langle L, *, \oplus \rangle$ and $\langle S, \wedge, \vee \rangle$ is bijective, i.e., one-to-one onto, then g is called an *isomorphism*. If there exists an isomorphism between two lattices, then the lattices are called *isomorphic*.

If the lattices $\langle L, *, \oplus \rangle$ and $\langle S, \wedge, \vee \rangle$ are isomorphic and g denotes an isomorphism, then g preserves the ordering relation; i.e., for any $a, b \in L, a \leq b \Rightarrow g(a) \leq' g(b)$. In addition to this, we also have $g(a) \leq' g(b) \Rightarrow a \leq b$. This result also shows that the two lattices which are isomorphic can be represented by the same diagram in which the nodes are replaced by the images. This fact explains why we found that several different lattices could be represented by the same diagram.

EXAMPLE 6 Let S be any set containing n elements and $\rho(S)$ be its power set. The lattice $\langle \rho(S), \cap, \cup \rangle$ or $\langle \rho(S), \subseteq \rangle$ is isomorphic to the lattice $\langle L^n, \leq_n \rangle$ given in Example 4.

It is interesting to observe that the lattices with one, two, or three elements are isomorphic to the chains containing one, two, or three elements, respectively. On the other hand, any lattice of order 4 must be isomorphic to one of the two lattices given in Figs. 4-1.1 *a* and *b*. Similarly, any lattice of order 5 is isomorphic to one of the lattices whose diagrams are given in Fig. 4-1.6.

A homomorphism $g: L \to L$ where $\langle L, *, \oplus \rangle$ is a lattice is called an *endomorphism*. If $g: L \to L$ is an isomorphism, then g is called an *automorphism*.

It is interesting to observe that if $g: L \to L$ is an endomorphism, then the image set of g is a sublattice of L.

Although the concepts of homomorphism and isomorphism are associated with any algebraic system, we shall now show how these concepts can be applied to partially ordered sets also.

Definition 4-1.6 Let $\langle P, \leq \rangle$ and $\langle Q, \leq' \rangle$ be two partially ordered sets. A mapping $f: P \to Q$ is said to be *order-preserving* relative to the ordering $\leq$ in P and $\leq'$ in Q iff for any $a, b \in P$ such that $a \leq b, f(a) \leq' f(b)$ in Q.

If $\langle P, \leq \rangle$ and $\langle Q, \leq' \rangle$ are lattices and $g: P \to Q$ is a lattice homomorphism, then g is order-preserving.

Definition 4-1.7 Two partially ordered sets $\langle P, \leq \rangle$ and $\langle Q, \leq' \rangle$ are called *order-isomorphic* if there exists a mapping $f: P \to Q$ which is bijective and if both f and f^{-1} are order-preserving.

It may happen that a mapping $f: P \to Q$ is bijective and order-preserving, but that f^{-1} is not order-preserving (see Example 7). In such a case, P and Q are not order-isomorphic. For lattices $\langle L, \leq \rangle$ and $\langle S, \leq' \rangle$, an order isomorphism

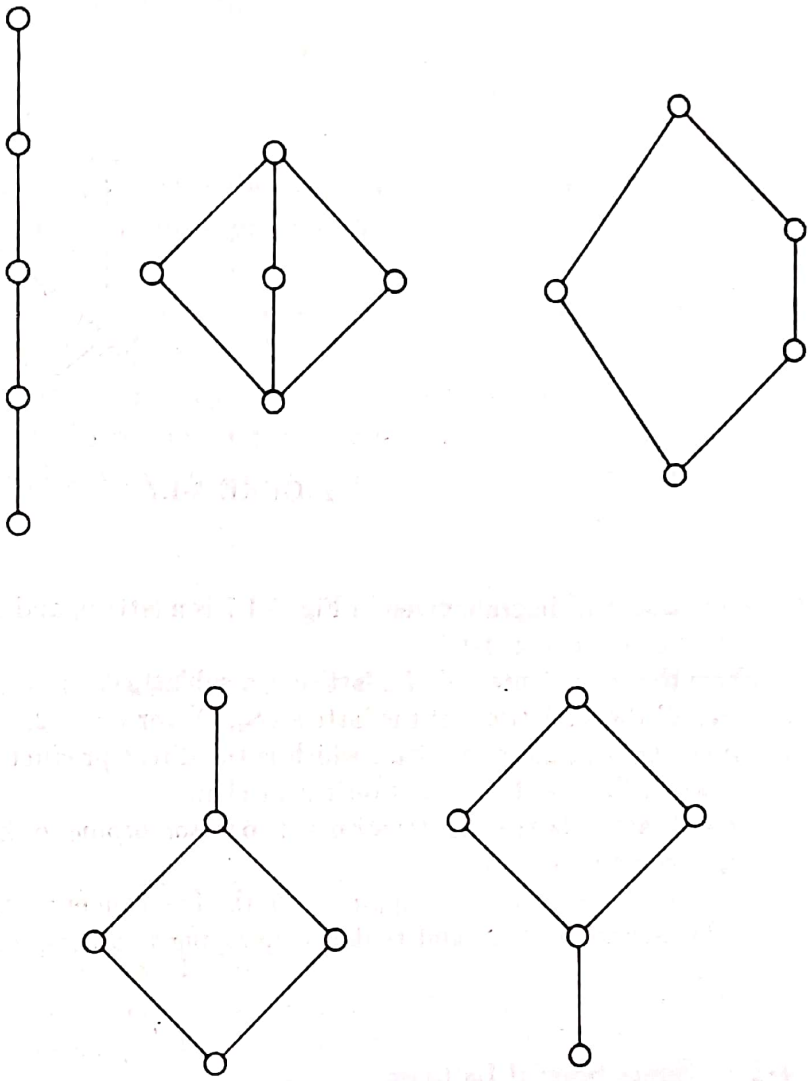

FIGURE 4-1.6 Lattices of order 5.

is equivalent to lattice isomorphism. Hence lattices which are order-isomorphic as partially ordered sets are isomorphic. The importance of order isomorphism lies in the fact that two partially ordered sets which are order-isomorphic can be represented by the same diagram.

EXAMPLE 7 Consider the lattice $\langle S_n, D \rangle$ for $n = 12$, that is, the lattice of divisors of 12 in which the partial ordering relation D means "division" as given in Example 3, Sec. 4-1.1. Consider another lattice $\langle S_n, \leq \rangle$ in which $\leq$ denotes the ordering relation "less than or equal to." A mapping $f: S_n \to S_n$ given by $f(x) = x$ is order-preserving and bijective, but f^{-1} is not order-preserving. Hence $\langle S_n, D \rangle$ and $\langle S_n, \leq \rangle$ are neither order-isomorphic nor isomorphic.

EXERCISES 4-1.4

1 For the lattice $\langle L, \subseteq \rangle$ given in Prob. 7 of Exercises 4-1.1, what are the operations of meet and join?

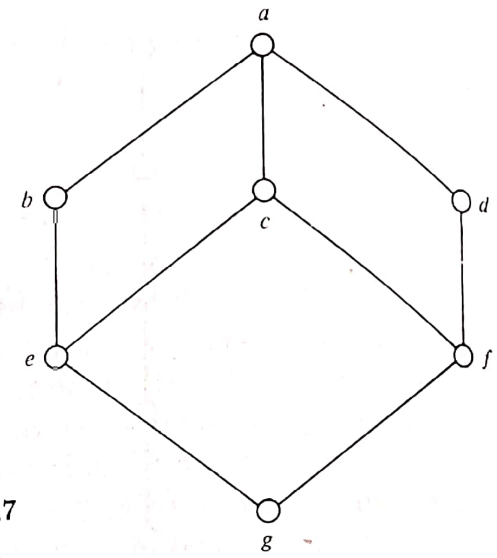

FIGURE 4-1.7

2 Show that the diagram given in Fig. 4-1.7 is a lattice, and it is not a sublattice of the lattice given in Fig. 4-1.1*f*.
3 Show that every interval of a lattice is a sublattice.
4 Find all the sublattices of the lattice $\langle S_n, D \rangle$ for $n = 12$.
5 Draw the diagram of a lattice which is the direct product of the five-element lattice shown in Fig. 4-1.1*c* and a two-element chain.
6 Show that the lattice $\langle S_n, D \rangle$ for $n = 216$ is isomorphic to the direct product of lattices for $n = 8$ and $n = 27$.
7 Show that there exists a mapping from the five-element lattice given in Fig. 4-1.1*c* to a three-element chain and that this mapping is order-preserving. Is it a homomorphism?

4-1.5 Some Special Lattices

In a lattice every pair of elements has a least upper bound and a greatest lower bound. As a consequence of this fact, one can show by using the principle of mathematical induction that every finite subset of a lattice has a least upper bound and a greatest lower bound. This, however, may not be the case for an infinite subset of a lattice. Consider, for example, the lattice $\langle \mathbf{I}_+, \leq \rangle$ in which $\mathbf{I}_+$ is the set of positive integers. The subset consisting of even positive integers has no least upper bound.

Let $\langle L, *, \oplus \rangle$ be a lattice and $S \subseteq L$ be a finite subset of L where $S = \{a_1, a_2, \ldots, a_n\}$. The greatest lower bound and the least upper bound of S can be expressed as

$$\text{GLB } S = \mathop{*}_{i=1}^{n} a_i \qquad \text{and} \qquad \text{LUB } S = \mathop{\oplus}_{i=1}^{n} a_i \tag{1}$$

where

$$\mathop{*}_{i=1}^{2} a_i = a_1 * a_2 \qquad \text{and} \qquad \mathop{*}_{i=1}^{k} a_i = \mathop{*}_{i=1}^{k-1} a_i * a_k \qquad k = 3, 4, \ldots$$

A similar representation can be given for $\mathop{\oplus}_{i=1}^{n} a_i$. Because of the associativity of

the operations $*$ and $\oplus$, we can write

$$\overset{n}{\underset{i=1}{*}} a_i = a_1 * a_2 * \cdots * a_n \qquad \text{and} \qquad \overset{n}{\underset{i=1}{\oplus}} a_i = a_1 \oplus a_2 \oplus \ldots \oplus a_n$$

Definition 4-1.8 A lattice is called *complete* if each of its nonempty subsets has a least upper bound and a greatest lower bound.

Obviously, every finite lattice must be complete. Also every complete lattice must have a least element and a greatest element. The least and the greatest elements of a lattice, if they exist, are called the *bounds* (*units, universal bounds*) of the lattice and are denoted by 0 and 1 respectively. A lattice which has both elements 0 and 1 is called a bounded lattice. For the lattice $\langle L, *, \oplus \rangle$ with $L = \{a_1, \ldots, a_n\}$,

$$\overset{n}{\underset{i=1}{*}} a_i = 0 \qquad \text{and} \qquad \overset{n}{\underset{i=1}{\oplus}} a_i = 1 \tag{2}$$

The bounds 0 and 1 of a lattice $\langle L, *, \oplus, 0, 1 \rangle$ satisfy the following identities. For any $a \in L$,

$$a \oplus 0 = a \qquad a * 1 = a \tag{3}$$

$$a \oplus 1 = 1 \qquad a * 0 = 0 \tag{4}$$

Obviously, 0 is the identity of the operation $\oplus$, and 1 is the identity of the operation $*$. Similarly, 0 and 1 are zeros with respect to the operations $*$ and $\oplus$ respectively. In a bounded lattice, 1 and 0 are duals of each other, and the principle of duality can now be extended to include the interchanges of 0 and 1. The identities in (3) are duals of each other, and so also are the identities in (4).

For bounded lattices it is possible to introduce the notion of a complement of an element in the following manner.

Definition 4-1.9 In a bounded lattice $\langle L, *, \oplus, 0, 1 \rangle$, an element $b \in L$ is called a *complement* of an element $a \in L$ if

$$a * b = 0 \qquad \text{and} \qquad a \oplus b = 1$$

Note that the definition of a complement is symmetric in a and b, so that b is a complement of a if a is a complement of b. Any element $a \in L$ may or may not have a complement. Furthermore, an element of L may have more than one complement in L.

From the identities (3) and (4) we have

$$0 * 1 = 0 \qquad \text{and} \qquad 0 \oplus 1 = 1 \tag{5}$$

which show that 0 and 1 are complements of each other. It is easy to show that 1 is the only complement of 0. Let us assume that $c \neq 1$ is a complement of 0 and $c \in L$; then

$$0 * c = 0 \qquad \text{and} \qquad 0 \oplus c = 1$$

However, $0 \oplus c = c$ from (3), and $c \neq 1$ leads to a contradiction. In a similar manner we can show that 0 is the only complement of 1.

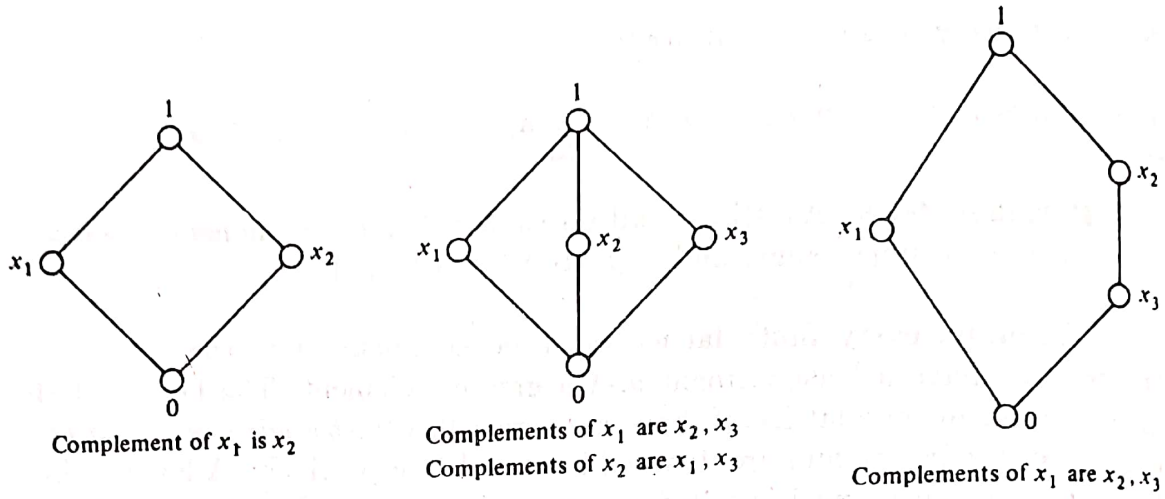

FIGURE 4-1.8 Complements in lattices.

Definition 4-1.10 A lattice $\langle L, *, \oplus, 0, 1 \rangle$ is said to be a *complemented* lattice if every element of L has at least one complement.

In Fig. 4-1.8 some lattices are shown, and the complements of some of the elements are noted below the diagrams.

EXAMPLE 1 Let $\langle L^n, \leq_n \rangle$ be the lattice of n-tuples of 0 and 1 given in Example 4, Sec. 4-1.4. This is a complemented lattice in which every element has a unique complement. The complement of an element of L^n can be obtained by interchanging 1 by 0 and 0 by 1 in the n-tuple representing the element. As a special case, let $n = 3$. The bounds of $\langle L^3, \leq_3 \rangle$ are $\langle 0, 0, 0 \rangle$ and $\langle 1, 1, 1 \rangle$. The complement of $\langle 1, 0, 1 \rangle$ is $\langle 0, 1, 0 \rangle$.

EXAMPLE 2 The lattice $\langle \rho(S), \subseteq \rangle$ of the power set of any set S is isomorphic to the lattice $\langle L^n, \leq_n \rangle$ provided S has n elements. The meet and join operations on $\rho(S)$ are $\cap$ and $\cup$ respectively, while the bounds are $\varnothing$ and S. The lattice $\langle \rho(S), \subseteq \rangle$ is a complemented lattice in which the complement of any subset A of S is the set $S - A$.

It was shown in Theorem 4-1.3 that the elements of any lattice satisfy the distributive inequalities. We shall now define a special class of lattices as follows.

Definition 4-1.11 A lattice $\langle L, *, \oplus \rangle$ is called a *distributive lattice* if for any $a, b, c \in L$,

$$a * (b \oplus c) = (a * b) \oplus (a * c) \tag{6}$$

and

$$a \oplus (b * c) = (a \oplus b) * (a \oplus c) \tag{7}$$

In other words, in a distributive lattice the operations $*$ and $\oplus$ distribute over each other.

It may be mentioned here that the equalities (6) and (7) are equivalent to one another (see Prob. 7, Exercises 4-1.5), and it is sufficient to verify any

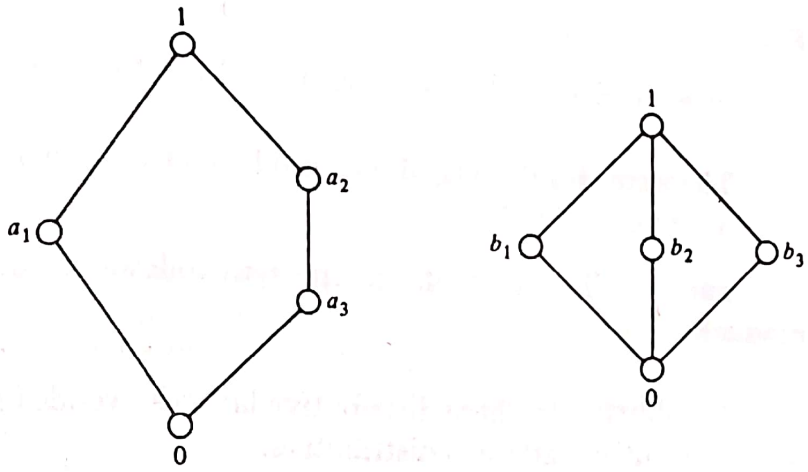

FIGURE 4-1.9 Lattices which are not distributive.

one of these two equalities for all possible combinations of the elements of a lattice. Note that the distributive equalities may be satisfied by some elements of a lattice, but this does not guarantee that the lattice is distributive (see Example 3).

The lattices given in Examples 1 and 2 are distributive lattices.

EXAMPLE 3 Show that the lattices given by the diagrams in Fig. 4-1.9 are not distributive.

SOLUTION

$$a_3 * (a_1 \oplus a_2) = a_3 * 1 = a_3 = (a_3 * a_1) \oplus (a_3 * a_2)$$

$$a_1 * (a_2 \oplus a_3) = 0 = (a_1 * a_2) \oplus (a_1 * a_3)$$

but

$$a_2 * (a_1 \oplus a_3) = a_2 * 1 = a_2$$

$$(a_2 * a_1) \oplus (a_2 * a_3) = 0 \oplus a_3 = a_3$$

Hence the lattice is not distributive. In the other case, $b_1 * (b_2 \oplus b_3) = b_1$ while $(b_1 * b_2) \oplus (b_1 * b_3) = 0$, which shows that the lattice is not distributive. ////

The two five-element lattices given in Fig. 4-1.9 are important because of a theorem which states that a lattice is distributive iff no sublattice is isomorphic to either of the two five-element lattices given there. We shall not prove this theorem.

The following theorems show that certain lattices are always distributive.

Theorem 4-1.5 Every chain is a distributive lattice.

PROOF Let $\langle L, \leq \rangle$ be a chain and $a, b, c \in L$. Consider the following possible cases: (1) $a \leq b$ or $a \leq c$, and (2) $a \geq b$ and $a \geq c$. We shall now show that the distributive law (6) is satisfied by a, b, c.

For (1),

$$a * (b \oplus c) = a \quad \text{and} \quad (a * b) \oplus (a * c) = a$$

For (2),

$$a * (b \oplus c) = b \oplus c \qquad \text{and} \qquad (a * b) \oplus (a * c) = b \oplus c \qquad ////$$

Theorem 4-1.6 The direct product of any two distributive lattices is a distributive lattice.

PROOF The proof of the theorem follows from the definition of direct product. ////

In addition to these distributive lattices, we also have that any sublattice of a distributive lattice is distributive.

Observe that the distributive laws as stated in Eqs. (6) and (7) are duals of each other; therefore, the principle of duality holds for all distributive lattices.

The following are some examples of distributive lattices.

EXAMPLE 4 The ring of subsets of a given set S defined in Example 3, Sec. 4-1.4, and denoted by $\langle R(S), \cap, \cup \rangle$ is a distributive lattice, because of the fact that both set union and set intersection satisfy the distributive laws.

EXAMPLE 5 The lattice $\langle I_+, D \rangle$ given in Example 2, Sec. 4-1.1, is a distributive lattice, and so also are the sublattices $\langle S_n, D \rangle$ for any positive integer n.

The following interesting theorem holds for a distributive lattice.

Theorem 4-1.7 Let $\langle L, *, \oplus \rangle$ be a distributive lattice. For any a, b, $c \in L$,

$$(a * b = a * c) \wedge (a \oplus b = a \oplus c) \Rightarrow b = c$$

PROOF

$$(a * b) \oplus c = (a * c) \oplus c = c$$

$$(a * b) \oplus c = (a \oplus c) * (b \oplus c) = (a \oplus b) * (b \oplus c)$$

$$= b \oplus (a * c) = b \oplus (a * b) = b \qquad ////$$

An important consequence of this theorem is that in a distributive lattice, if an element $a \in L$ has a complement, then it must be unique. Suppose that b and c are complements of a; then

$$a * b = a * c = 0 \qquad \text{and} \qquad a \oplus b = a \oplus c = 1$$

But from Theorem 4-1.7 this means $b = c$.

Recall that a lattice is called complemented if every element of the lattice has at least one complement. If we now consider those lattices which are complemented as well as distributive, then we are assured that every element of such a lattice has a unique complement, and we denote the complement of an element $a \in L$ by a'. Lattices which are complemented and distributive are called Boolean algebras. We shall study such lattices in detail in the next section.

It may be mentioned here that the converse of Theorem 4-1.7 also holds. We shall, however, omit the proof.

EXERCISES 4-1.5

1 Find the complements of every element of the lattice $\langle S_n, D\rangle$ for $n = 75$.
2 Show that in a lattice with two or more elements, no element is its own complement.
3 Show that a chain of three or more elements is not complemented.
4 Which of the two lattices $\langle S_n, D\rangle$ for $n = 30$ and $n = 45$ are complemented? Are these lattices distributive?
5 Show that De Morgan's laws, given by

$$(a * b)' = a' \oplus b' \quad \text{and} \quad (a \oplus b)' = a' * b'$$

hold in a complemented, distributive lattice.
6 Show that in a complemented, distributive lattice

$$a \leq b \Leftrightarrow a * b' = 0 \Leftrightarrow a' \oplus b = 1 \Leftrightarrow b' \leq a'$$

7 Show that Eqs. (6) and (7) are equivalent.
8 Show that a lattice is distributive iff

$$(a * b) \oplus (b * c) \oplus (c * a) = (a \oplus b) * (b \oplus c) * (c \oplus a)$$

9 Show that in a distributive lattice, the distributive laws can be generalized as

$$a * (\overset{n}{\underset{i=1}{\oplus}} b_i) = \overset{n}{\underset{i=1}{\oplus}} (a * b_i) \quad \text{and} \quad a \oplus (\overset{n}{\underset{i=1}{*}} b_i) = \overset{n}{\underset{i=1}{*}} (a \oplus b_i)$$

10 Show that in a bounded distributive lattice, the elements which have complements form a sublattice.
11 A lattice is said to be *modular* if

$$a \leq c \Rightarrow a \oplus (b * c) = (a \oplus b) * c$$

Show that every distributive lattice is modular, but not conversely.

4-2 BOOLEAN ALGEBRA

The example of the power set $\rho(S)$ of a nonempty set S appeared throughout our discussion of lattices. It is not accidental. In fact, we first started with a general algebraic system called a lattice and gradually imposed those conditions on lattices which are satisfied by the lattice of the power set. Our aim was to arrive at an algebraic system which has all the essential characteristics of the lattice of the power set. Once this is done, we arrive at an abstract algebraic system which will be shown to be isomorphic to the lattice of the power set of a set. Many other algebraic systems such as the statement algebra and switching algebra are also special cases of such an algebraic system called Boolean algebra. We shall be concerned with only finite Boolean algebras in this chapter.

4-2.1 Definition and Examples

Definition 4-2.1 A *Boolean algebra* is a complemented, distributive lattice.

A Boolean algebra will generally be denoted by $\langle B, *, \oplus, ', 0, 1\rangle$ in which $\langle B, *, \oplus\rangle$ is a lattice with two binary operations $*$ and $\oplus$ called the meet and join respectively. The corresponding partially ordered set will be denoted by $\langle B, \leq \rangle$. The bounds of the lattice are denoted by 0 and 1, where 0 is the least element and 1 the greatest element of $\langle B, \leq \rangle$. Since $\langle B, *, \oplus\rangle$ is complemented and because of the fact that it is a distributive lattice, each element of B has a unique complement. We shall denote the unary operation of complementation by $'$, so that for any $a \in B$, the complement of a is denoted by $a' \in B$.

Most of the properties of a Boolean algebra have been derived in the previous section. We shall list some of the important properties here. It may be mentioned that the properties listed here are not independent of each other. There are redundancies, but our list is chosen because of the importance of these properties.

A Boolean algebra $\langle B, *, \oplus, ', 0, 1\rangle$ satisfies the following properties in which a, b, and c denote any elements of the set B.

1 $\langle B, *, \oplus\rangle$ is a lattice in which the operations $*$ and $\oplus$ satisfy the following identities:

(L-1)	$a * a = a$	**(L-1)′**	$a \oplus a = a$
(L-2)	$a * b = b * a$	**(L-2)′**	$a \oplus b = b \oplus a$
(L-3)	$(a * b) * c = a * (b * c)$	**(L-3)′**	$(a \oplus b) \oplus c = a \oplus (b \oplus c)$
(L-4)	$a * (a \oplus b) = a$	**(L-4)′**	$a \oplus (a * b) = a$

(See Sec. 4-1.2 for these identities.)

2 $\langle B, *, \oplus\rangle$ is a distributive lattice and satisfies these identities:

(D-1) $a * (b \oplus c) = (a * b) \oplus (a * c)$

(D-2) $a \oplus (b * c) = (a \oplus b) * (a \oplus c)$

(D-3) $(a * b) \oplus (b * c) \oplus (c * a) = (a \oplus b) * (b \oplus c) * (c \oplus a)$

(D-4) $a * b = a * c$, and $a \oplus b = a \oplus c \Rightarrow b = c$

(See Definition 4-1.11, Theorem 4-1.7, and Prob. 8 of Exercises 4-1.5.)

3 $\langle B, *, \oplus, 0, 1\rangle$ is a bounded lattice in which for any $a \in B$, the following hold:

(B-1)	$0 \leq a \leq 1$		
(B-2)	$a * 0 = 0$	**(B-2)′**	$a \oplus 1 = 1$
(B-3)	$a * 1 = a$	**(B-3)′**	$a \oplus 0 = a$

[See identities (3) and (4) in Sec. 4-1.5.]

4 $\langle B, *, \oplus, ', 0, 1\rangle$ is a uniquely complemented lattice in which the complement of any element $a \in B$ is denoted by $a' \in B$ and satisfies the follow-

ing identities:

(C-1) $a * a' = 0$ (C-1)′ $a \oplus a' = 1$

(C-2) $0' = 1$ (C-2)′ $1' = 0$

(C-3) $(a * b)' = a' \oplus b'$ (C-3)′ $(a \oplus b)' = a' * b'$

(See Definition 4-1.9 and Prob. 5, Exercises 4-1.5.)

5 There exists a partial ordering relation $\leq$ on B such that

(P-1) $a * b = \text{GLB } \{a, b\}$ (P-1)′ $a \oplus b = \text{LUB } \{a, b\}$

(P-2) $a \leq b \Leftrightarrow a * b = a \Leftrightarrow a \oplus b = b$

(P-3) $a \leq b \Leftrightarrow a * b' = 0 \Leftrightarrow b' \leq a' \Leftrightarrow a' \oplus b = 1$

(See Theorem 4-1.1 and Prob. 6, Exercises 4-1.5.)

As pointed out earlier, not all the identities given here are independent of one another. These identities arose by looking at a Boolean algebra as a special lattice. It is possible to define a Boolean algebra as an abstract algebraic system satisfying certain properties which are independent of each other. In fact, even the two binary operations $*$ and $\oplus$, the unary operation $'$, and the two distinguished elements are not all independent. One can define a Boolean algebra in terms of the operations $*$ and $'$ and a set of independent properties satisfied by these operations. We shall not, however, concern ourselves with this approach.

EXAMPLE 1 Let $B = \{0, 1\}$ be a set. The operations $*$, $\oplus$, and $'$ on B are given by Table 4-2.1. The algebra $\langle B, *, \oplus, ', 0, 1\rangle$ satisfies all the properties listed here and is one of the simplest examples of a two-element Boolean algebra. A two-element Boolean algebra is the only Boolean algebra whose diagram is a chain.

EXAMPLE 2 Let S be a nonempty set and $\rho(S)$ be its power set. The set algebra $\langle \rho(S), \cap, \cup, \sim, \varnothing, S\rangle$ is a Boolean algebra in which the complement of any subset $A \subseteq S$ is $\sim A = S - A$, the relative complement of the set A. If S has n elements, then $\rho(S)$ has 2^n elements and the diagram of the Boolean algebra is an n cube. The partial ordering relation on $\rho(S)$ corresponding to the operations $\cap$ and $\cup$ is the subset relation $\subseteq$. The diagrams for the Boolean algebra $\langle \rho(S), \cap, \cup\rangle$ when S has 1, 2, and 3 elements are given in Fig. 4-2.1. If S is an empty set, then $\rho(S)$ has only one element, viz., $\varnothing$, so that $\varnothing = 0 = 1$, and the corresponding Boolean algebra is a degenerate Boolean algebra. We shall consider nondegenerate Boolean algebras only.

Table 4-2.1

$*$	0	1
0	0	0
1	0	1

$\oplus$	0	1
0	0	1
1	1	1

x	x'
0	1
1	0

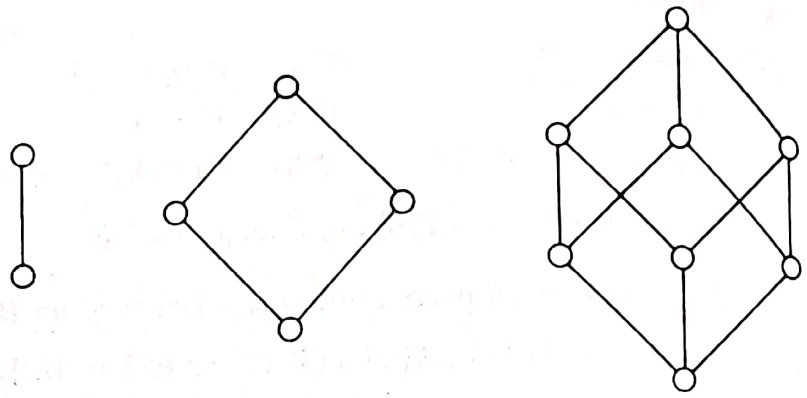

FIGURE 4-2.1 Boolean algebra of power sets.

EXAMPLE 3 Let S denote the set of statement formulas involving n statement variables. The algebraic system $\langle S, \wedge, \vee, \neg, \mathbf{F}, \mathbf{T}\rangle$ is a Boolean algebra in which $\wedge$, $\vee$, and $\neg$ denote the operations of conjunction, disjunction, and negation respectively. The elements **F** and **T** denote the formulas which are contradictions and tautologies respectively. Two statement formulas which are equivalent to one another are considered as equal. The partial ordering relation corresponding to the operations $\wedge$ and $\vee$ is the implication $\Rightarrow$ defined in Chap. 1.

EXAMPLE 4 Let B_n be the set of n-tuples whose members are either 0 or 1. Thus $a \in B_n$ iff $a = \langle a_1, a_2, \ldots, a_n\rangle$ where $a_i = 0$ or 1 for $i = 1, 2, \ldots, n$. Let us define for any $a = \langle a_1, a_2, \ldots, a_n\rangle$, $b = \langle b_1, b_2, \ldots, b_n\rangle$, and $a, b \in B_n$

$$a * b = \langle a_1 \wedge b_1, a_2 \wedge b_2, \ldots, a_n \wedge b_n\rangle$$
$$a \oplus b = \langle a_1 \vee b_1, a_2 \vee b_2, \ldots, a_n \vee b_n\rangle$$
$$a' = \langle \neg a_1, \neg a_2, \ldots, \neg a_n\rangle$$

where $\wedge$, $\vee$, and $\neg$ are the usual logical operations on $\{0, 1\}$. The algebra $\langle B_n, *, \oplus, ', 0_n, 1_n\rangle$ is a Boolean algebra in which 0_n and 1_n are n-tuples whose members are all 0s and 1s respectively. This algebra is known as a *switching algebra* and represents a switching network with n inputs and one output.

In a Boolean algebra it is possible to show that the associativity laws (**L-3**) and (**L-3**)′, the distributive laws (**D-1**) and (**D-2**), and De Morgan's laws (**C-3**) and (**C-3**)′ can be generalized over any finite number of elements by using the principle of mathematical induction. We shall simply state these generalized laws. For this purpose, let $S = \{a_1, a_2, \ldots, a_n\}$ and $T = \{b_1, b_2, \ldots, b_m\}$ and let $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m$ be the elements of a Boolean algebra; then

$$(\underset{S}{*}\, a_i) * (\underset{T}{*}\, b_j) = \underset{S \cup T}{*}\, c_k$$

where

$$\underset{S}{*}\, a_i = a_1 * a_2 * \cdots * a_n$$

$$\underset{T}{*}\, b_j = b_1 * b_2 * \cdots * b_m$$

$$\underset{S \cup T}{*}\, c_k = a_1 * a_2 * \cdots * a_n * b_1 * b_2 * \cdots * b_m$$

The order in which the elements $a_1, a_2, \ldots, b_1, b_2, \ldots$ appear in these expressions is unimportant. Similarly, the generalized distributive laws are

$$(\underset{S}{*}\, a_i) \oplus (\underset{T}{*}\, b_j) = \underset{S\times T}{*}\, (a_i \oplus b_j)$$

$$(\underset{S}{\oplus}\, a_i) * (\underset{T}{\oplus}\, b_j) = \underset{S\times T}{\oplus}\, (a_i * b_j)$$

The generalized De Morgan's laws are

$$(\underset{S}{*}\, a_i)' = \underset{S}{\oplus}\, a_i' \quad \text{and} \quad (\underset{S}{\oplus}\, a_i)' = \underset{S}{*}\, a_i'$$

Using the above results, we can also write

$$[(\underset{S}{*}\, a_i) \oplus (\underset{T}{*}\, b_j)]' = \underset{S\times T}{\oplus}\, (a_i' * b_j')$$

$$[(\underset{S}{\oplus}\, a_i) * (\underset{T}{\oplus}\, b_j)]' = \underset{S\times T}{*}\, (a_i' \oplus b_j')$$

4-2.2 Subalgebra, Direct Product, and Homomorphism

Definition 4-2.2 Let $\langle B, *, \oplus, ', 0, 1\rangle$ be a Boolean algebra and $S \subseteq B$. If S contains the elements 0 and 1 and is closed under the operations $*$, $\oplus$, and $'$, then $\langle S, *, \oplus, ', 0, 1\rangle$ is called a *sub-Boolean algebra.*

In practice it is not necessary to check for closure with respect to all three operations $*$, $\oplus$, and $'$, nor is it necessary to check whether 0 and 1 are in S. Only closure with respect to the set of operations $\{*, '\}$ or $\{\oplus, '\}$ is enough to guarantee that S is a subalgebra. This argument follows from the fact that these sets of operations are functionally complete in a Boolean algebra, because for any $a, b \in B$

$$a \oplus b = (a' * b')' \qquad \text{also} \qquad 1 = (a * a')' \qquad \text{and} \qquad 0 = a * a'$$

so that closure with respect to $*$ and $'$ guarantees closure with respect to $\oplus$ as well as the existence of 0 and 1 in the subalgebra, and similarly for the set $\{\oplus, '\}$.

Our definition of a sub-Boolean algebra implies that it is a Boolean algebra. A subset of a Boolean algebra can be a Boolean algebra; however, it may not be a sub-Boolean algebra because it is not closed with respect to the operations in B. We shall show this fact by means of an example (see Example 1 in this subsection).

For any Boolean algebra $\langle B, *, \oplus, ', 0, 1\rangle$, the subsets $\{0, 1\}$ and the set B are both sub-Boolean algebras. In addition to these sub-Boolean algebras, consider now any element $a \in B$ such that $a \neq 0$ and $a \neq 1$ and consider the set $\{a, a', 0, 1\}$. Obviously this set is a sub-Boolean algebra of the given Boolean algebra. Every element of B generates a sub-Boolean algebra. More generally, any subset of B generates a sub-Boolean algebra. We shall study Boolean algebras generated by a set of elements later in Sec. 4-3.1.

EXAMPLE 1 Consider the Boolean algebra given in Fig. 4-2.2. Let the subsets be

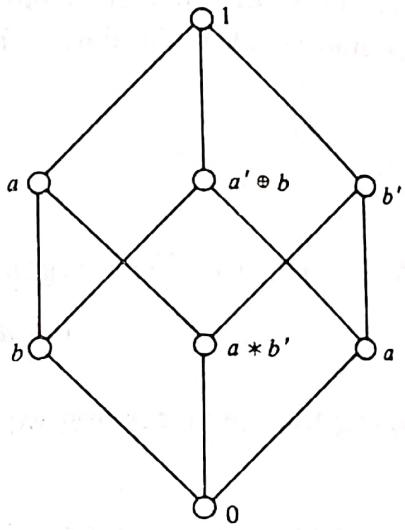

FIGURE 4-2.2

$$S_1 = \{a, a', 0, 1\}$$
$$S_2 = \{a' \oplus b, a * b', 0, 1\}$$
$$S_3 = \{a * b', b', a, 1\}$$
$$S_4 = \{b', a * b', a', 0\}$$
$$S_5 = \{a, b', 0, 1\}$$

The subsets S_1 and S_2 are sub-Boolean algebras. The subsets S_3 and S_4 are Boolean algebras, but not sub-Boolean algebras of the given algebra. The subset S_5 is not even a Boolean algebra.

Let $\langle B_1, *_1, \oplus_1, ', 0_1, 1_1\rangle$ and $\langle B_2, *_2, \oplus_2, '', 0_2, 1_2\rangle$ be two Boolean algebras. The *direct product* of the two Boolean algebras is defined to be a Boolean algebra that is given by $\langle B_1 \times B_2, *_3, \oplus_3, ''', 0_3, 1_3\rangle$ in which the operations are defined for any $\langle a_1, b_1\rangle$ and $\langle a_2, b_2\rangle \in B_1 \times B_2$ as

$$\langle a_1, b_1\rangle *_3 \langle a_2, b_2\rangle = \langle a_1 *_1 a_2, b_1 *_2 b_2\rangle$$
$$\langle a_1, b_1\rangle \oplus_3 \langle a_2, b_2\rangle = \langle a_1 \oplus_1 a_2, b_1 \oplus_2 b_2\rangle$$
$$\langle a_1, b_1\rangle''' = \langle a_1', b_1''\rangle$$
$$0_3 = \langle 0_1, 0_2\rangle \quad \text{and} \quad 1_3 = \langle 1_1, 1_2\rangle$$

The direct product of Boolean algebras enables us to generate new Boolean algebras. Thus from the 2-element Boolean algebra given in Example 1, Sec. 4-2.1, we can generate $B \times B = B^2$, $B \times B \times B = B^3$, etc., and finally the Boolean algebra of n-tuples given in Example 4, Sec. 4-2.1, which is B^n.

Let $\langle B, *, \oplus, ', 0, 1\rangle$ and $\langle P, \cap, \cup, -, \alpha, \beta\rangle$ be two Boolean algebras. A mapping $f: B \to P$ is called a *Boolean homomorphism* if all the operations of the Boolean algebra are preserved, i.e., for any $a, b \in B$

$$f(a * b) = f(a) \cap f(b) \qquad f(a \oplus b) = f(a) \cup f(b)$$
$$f(a') = \overline{f(a)} \qquad f(0) = \alpha \quad \text{and} \quad f(1) = \beta$$

As before, the above definition of homomorphism can be simplified by as-

serting that $f: B \to P$ preserves either the operations $*$ and $'$ or the operations $\oplus$ and $'$. It is easy to see that this definition implies the previous definition.

Suppose that, instead of preserving the operations which are functionally complete, we now consider a mapping $g: B \to P$ in which the operations $*$ and $\oplus$ are preserved. In other words, let g be a lattice homomorphism. Naturally, g preserves the order, and hence it maps the bounds 0 and 1 into the least and the greatest elements respectively of the image set $g(B) \subseteq P$. It is, however, not necessary that $g(0) = \alpha$ and $g(1) = \beta$. The complements, if defined in terms of $g(0)$ and $g(1)$ in $g(B)$, are preserved, and $\langle g(B), \cap, \cup, -, g(0), g(1) \rangle$ is a Boolean algebra. Note that $g: B \to P$ is not a Boolean homomorphism, although $g: B \to g(B)$ is. In any case, for any mapping from a Boolean algebra which preserves the operations $*$ and $\oplus$, the image set is a Boolean algebra.

We shall now consider some properties of a Boolean algebra that will eventually lead us to an important theorem in Boolean algebra. This theorem states that any Boolean algebra is isomorphic to a power set algebra $\langle \rho(S), \cap, \cup, \sim, \emptyset, S \rangle$ for some set S. This theorem is known as *Stone's representation theorem* and is valid for any Boolean algebra. We shall, however, restrict our discussion to finite Boolean algebras.

Let $\langle B, *, \oplus, ', 0, 1 \rangle$ be a Boolean algebra and $a_1, a_2, \ldots$ denote the elements of B. It is possible to write well-formed expressions involving the elements of B, the operations $*$, $\oplus$, and $'$, and the parentheses wherever necessary. The following are some of the examples of such expressions:

$$a_1 \qquad a_1 * a_2' \qquad (a_1 * a_3)' \oplus (a_3' \oplus a_4)' \qquad (a_1 \oplus a_2) * (a_1 \oplus a_3) * a_2'$$

Because of the fact that B is closed under the operations $*$, $\oplus$, and $'$, each expression represents a particular element of the Boolean algebra B. Every such expression represents a definite element, although various different expressions may represent the same element. We study the general problem of equality of expressions in Sec. 4-3.1. For the present, let us confine our attention to only those elements of B which can be expressed as the join of two or more elements of B. Naturally, there are elements of B which cannot be expressed as the join of two or more elements. Such elements are called *join-irreducible*. In fact the idea of join-irreducible is more general, and we define it as follows.

Definition 4-2.3 Let $\langle L, *, \oplus \rangle$ be a lattice. An element $a \in L$ is called *join-irreducible* if it cannot be expressed as the join of two distinct elements of L. In other words, $a \in L$ is join-irreducible if for any $a_1, a_2 \in L$

$$a = a_1 \oplus a_2 \Rightarrow (a = a_1) \vee (a = a_2)$$

In the case of a Boolean algebra, it can be shown that the only elements which are join-irreducible are those which cover the least element 0. Such elements are called the *atoms* of the Boolean algebra. It is also possible to show that except for the least element 0 and the atoms, every other element of a Boolean algebra can be represented as the join of two or more atoms of the algebra and that this representation is unique. We shall demonstrate this idea by some examples of Boolean algebras whose diagrams are given in Fig. 4-2.3.

Let us consider the set $S = \{a_1, a_2, \ldots, a_n\}$ of the atoms of a Boolean

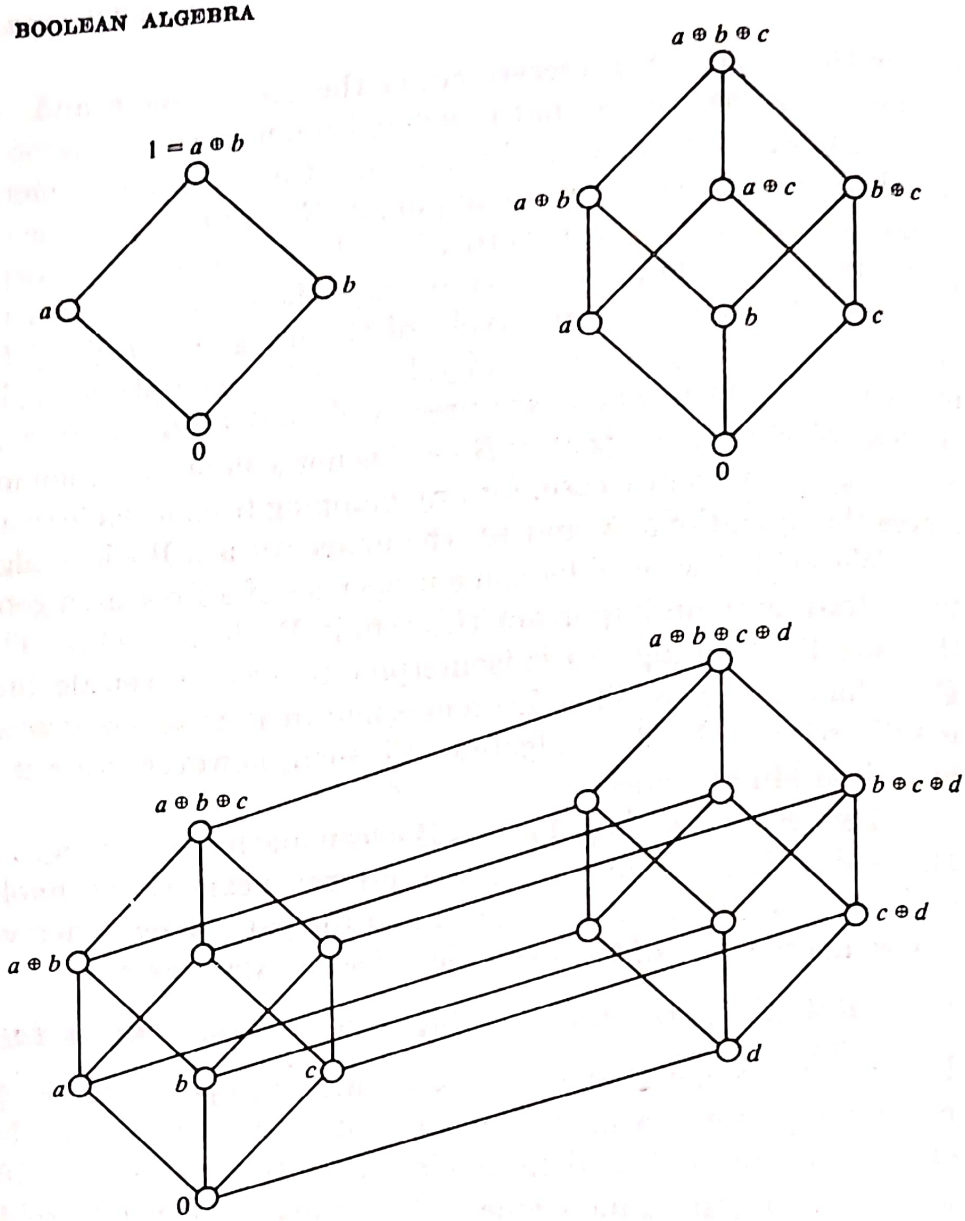

FIGURE 4-2.3

algebra $\langle B_n, *, \oplus, ', 0, 1\rangle$. Note that every element of B_n, except the atoms themselves and the least element 0, can be expressed as the join of some of its atoms. In fact, every such join represents an element of B_n. Consider all possible subsets of S and associate with each subset of S a Boolean expression consisting of the join of the elements of the subset. This means that we associate an element of B_n with every subset consisting of two or more elements of S. Let us now associate the element $a_i \in B_n$ with the subsets such as $\{a_i\} \subseteq S$ and the element $0 \in B_n$ with the subset $\varnothing \subseteq S$. In this way, we have associated with each subset of S a Boolean expression consisting of the join of the elements of the subset, and each such expression is distinct and represents an element of B_n. This association establishes a one-to-one correspondence between the subsets of S and the elements of B_n. It is possible to show that such a one-to-one correspondence preserves the operations $*$, $\oplus$, and $'$; that is, the Boolean algebra $\langle B_n, *, \oplus, ', 0, 1\rangle$ is isomorphic to the power set algebra $\langle \rho(S), \cap, \cup, \sim, \varnothing, S\rangle$. This result explains why the diagrams of Boolean algebras given so far are all n cubes containing 2^n elements for $n = 1, 2, \ldots$. This also shows that a Boolean algebra is completely specified by its atoms.

Let us now determine those atoms which are present in the join representation of a particular element of the Boolean algebra $\langle B_n, *, \oplus, ', 0, 1\rangle$ for which the set of atoms is $S = \{a_1, a_2, \ldots, a_n\}$. Let $a \in B_n$ be any element of B_n such that $a = a_i \oplus a_j \oplus a_k$ for some $a_i, a_j, a_k \in S$. Obviously, $a_i \leq a$, $a_j \leq a$, and $a_k \leq a$. Next, let $a_h \in S$ be such that $a_h \leq a$ and such that $a_h \neq a_i$, $a_h \neq a_j$, and $a_h \neq a_k$. This means

$$a_h = a_h * a = a_h * (a_i \oplus a_j \oplus a_k) = (a_h * a_i) \oplus (a_h * a_j) \oplus (a_h * a_k) = 0$$

But $a_h \neq 0$ because it is an atom; hence we have a contradiction, showing that any atom which is $\leq a$ must appear in the join representation of the element a. Also no atom is $\geq a$ unless $a = 0$. This result shows that for any element of B_n only those atoms appear in the join representation which are $\leq$ the element. Those atoms which are incomparable with the element do not appear in such a join representation.

The atoms of a Boolean algebra are also called its *minterms*. It follows from the previous discussion that every element of a Boolean algebra except the element 0 can be expressed as the join of its minterms. The use of the term "minterm" will become clear in Sec. 4-3.1 when we discuss the Boolean forms.

Instead of representing the elements of a Boolean algebra in terms of the join of their atoms, we could also represent the elements in terms of the meet of their *antiatoms*, where the antiatoms which are also called the *maxterms* are those elements of the Boolean algebra which are covered by the greatest element 1. In fact, the antiatoms are the complements of the atoms. Because of the principle of duality, the whole discussion can be repeated by interchanging $*$ by $\oplus$, 0 by 1, maxterm by minterm, and $\leq$ by $\geq$. We shall not pursue this idea any further.

EXERCISES 4-2

1 Prove the following Boolean identities:
   (a) $a \oplus (a' * b) = a \oplus b$
   (b) $a * (a' \oplus b) = a * b$
   (c) $(a * b) \oplus (a * b') = a$
   (d) $(a * b * c) \oplus (a * b) = a * b$

2 It is conventional in switching theory to use the symbols $+$ and $\cdot$ in place of the symbols $\oplus$ and $*$. Further simplifications of the Boolean expressions are introduced by assuming the order of precedence of the operators $'$, $\cdot$, and $+$ and also by suppressing the dot and writing $a \cdot b$ as ab. Write all the identities of Sec. 4-2.1 and those given in Prob. 1 by using these conventions and notations.

3 In any Boolean algebra, show that
   (a) $a = b \Leftrightarrow ab' + a'b = 0$
   (b) $a = 0 \Leftrightarrow ab' + a'b = b$
   (c) $(a + b')(b + c')(c + a') = (a' + b)(b' + c)(c' + a)$
   (d) $(a + b)(a' + c) = ac + a'b = ac + a'b + bc$
   (e) $a \leq b \Rightarrow a + bc = b(a + c)$

4 Simplify the following Boolean expressions:
   (a) $(a * b)' \oplus (a \oplus b)'$
   (b) $(a' * b' * c) \oplus (a * b' * c) \oplus (a * b' * c')$

(c) $(a * c) \oplus c \oplus [(b \oplus b') * e]$
(d) $(1 * a) \oplus (0 * a')$

5 Let $\langle \rho(S), \cap, \cup, \sim, \varnothing, S \rangle$ be the algebra of the subsets of $S = \{a, b, c\}$, and let $g: \rho(S) \rightarrow B$ be a mapping onto the two-element Boolean algebra given in Example 1 of Sec. 4-2.1, such that $g(x) = 1$ if x contains the element b, otherwise $g(x) = 0$. Show that g is a Boolean homomorphism.

6 Show that a mapping from one Boolean algebra to another which preserves the operations $\oplus$ and $'$ also preserves the operation $*$.

7 Show that a lattice homomorphism on a Boolean algebra which preserves 0 and 1 is a Boolean homomorphism.

8 Let $\langle B, *, \oplus, ', 0, 1 \rangle$ be a Boolean algebra. Define the operations $+$ and $\cdot$ on the elements of B by

$$a + b = (a * b') \oplus (a' * b)$$
$$a \cdot b = a * b$$

Show that $\langle B, +, \cdot, 1 \rangle$ is a Boolean ring with identity 1.

9 For the operation $+$ defined in Prob. 8, show that
(a) $(a + b) + b = a$
(b) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
(c) $a + a = 0$
(d) $a + 0 = a$
(e) $a + 1 = a'$

4-3 BOOLEAN FUNCTIONS

Several identities and expressions involving the elements of a Boolean algebra were given in Sec. 4-2. In this section we shall first introduce well-formed expressions involving Boolean variables. It will be shown that these expressions form a Boolean algebra called a free Boolean algebra. An equivalence relation on the set of Boolean expressions in a certain number of variables is then introduced. As a next step, it is shown that every Boolean expression is equivalent to another expression which possesses a certain standard, or normal, form. The idea of the equivalence of Boolean expressions is effectively used in the design of economical switching circuits, as is shown later in Secs. 4-4 and 4-5. In this section, we next introduce the concept of the value of a Boolean expression and a valuation process over a given Boolean algebra. It is then shown that if certain statements about Boolean expressions are true for a Boolean algebra, then they are true for all Boolean algebras. This finally permits us to define Boolean functions associated with Boolean expressions.

4-3.1 Boolean Forms and Free Boolean Algebra

In our discussion so far, we have assumed that we are given a Boolean algebra $\langle B, *, \oplus, ', 0, 1 \rangle$ in which the elements 0 and 1 are the distinguished elements, or the bounds, and the operations $*$, $\oplus$, and $'$ are defined on the elements of B and satisfy the identities given in Sec. 4-2.1. Let us now consider a set of n variables or literals $x_1, x_2, \ldots, x_n$ and operator symbols $*$, $\oplus$, and $'$. With the help of these symbols and variables we form strings according to certain rules which will now be given.

Definition 4-3.1 A *Boolean expression, form,* or *formula* in n variables $x_1, x_2, \ldots, x_n$ is any finite string of symbols formed in the following manner:

1 0 and 1 are Boolean expressions.

2 $x_1, x_2, \ldots, x_n$ are Boolean expressions.

3 If α_1 and α_2 are Boolean expressions, then $(\alpha_1) * (\alpha_2)$ and $(\alpha_1) \oplus (\alpha_2)$ are also Boolean expressions.

4 If α is a Boolean expression, then $(\alpha)'$ is also a Boolean expression.

5 No strings of symbols except those formed in accordance with rules 1 to 4 are Boolean expressions.

We shall generally denote a Boolean expression by $\alpha, \beta, \gamma, \ldots$, or more explicitly as $\alpha(x_1, x_2, \ldots, x_n)$. The only exception that we shall make in practice from that given in the definition is to drop some of the parentheses, wherever possible. With this convention, the following are examples of Boolean expressions in three variables x_1, x_2, and x_3.

$$x_1 \qquad x_1' \oplus x_2 \qquad (x_2' \oplus x_1)' * (x_3 \oplus x_1) \qquad (x_1' \oplus x_1) * x_2 * x_3'$$

Notice that a Boolean expression in n variables may or may not contain all the n variables.

Obviously one can construct an infinite number of Boolean expressions in n variables. However, if we assume that the operations $*$, $\oplus$, and $'$ satisfy all the identities of a Boolean algebra, then it is possible to define an equivalence relation on the set of Boolean expressions in n variables.

Definition 4-3.2 Two Boolean forms $\alpha(x_1, x_2, \ldots, x_n)$ and $\beta(x_1, x_2, \ldots, x_n)$ are called *equivalent* (or *equal*) if one can be obtained from the other by a finite number of applications of the identities of a Boolean algebra.

It is easy to see that the relation given by Definition 4-3.2 is an equivalence relation on the set of Boolean expressions in n variables and therefore partitions the set into equivalence classes. All those Boolean expressions which are in the same equivalence class are equivalent, or equal, to one another. We shall now show that the number of these equivalence classes is finite. Let us first consider certain special Boolean expressions called minterms. It will be convenient to call the operations $*$ and $\oplus$ "product" and "sum" respectively in the rest of our discussion.

Definition 4-3.3 A Boolean form in n variables $x_1, x_2, \ldots, x_n$ consisting of the product of n terms such as

$$x_1^{a_1} * x_2^{a_2} * \cdots * x_n^{a_n} = \mathop{*}_{i=1}^{n} x_i^{a_i}$$

in which a_i is either 0 or 1, x_i^0 stands for x_i', and x_i^1 stands for x_i for $i = 1, 2, \ldots, n$ is called a *minterm, complete product,* or a *fundamental product* of the n variables.

Note the similarity in the definition of "minterm" given here and that given

in Sec. 1-3.3. In fact, most of our discussion that follows will be similar to the discussion in Sec. 1-3.5. We shall denote a particular minterm by $\min_j$ or $\min_j^n$ or simply as m_j where j is the decimal representation of the binary number $a_1a_2 \cdots a_n$. Since for each i $(i = 1, 2, \ldots, n)$, a_i can be either 0 or 1, we have 2^n minterms which we denote as $m_0, m_1, \ldots, m_{2^n-1}$. These minterms satisfy the following properties:

$$\min_i * \min_j = 0 \qquad \text{for } i \neq j \tag{1}$$

$$\bigoplus_{i=0}^{2^n-1} \min_i = 1 \tag{2}$$

For $i \neq j$, there is at least one variable and its complement that appear in the product $\min_i * \min_j$ in Eq. (1), and hence it is equal to 0. Equation (2) can be proved by the principle of mathematical induction. Note also that any two minterms $\min_i$ and $\min_j$ for $i \neq j$ cannot be equivalent to each other.

We have assumed that the operations $*$, $\oplus$, and $'$ along with the bounds 0 and 1 satisfy all the identities given in Sec. 4-2.1 when the operations are applied to the variables $x_1, x_2, \ldots, x_n$. Therefore, one can show that every Boolean expression except 0 can be expressed in an equivalent form consisting of the sums of minterms. Such an equivalent form is called the *sum-of-products canonical* form. For every Boolean expression in n variables, a canonical form exists and is unique in some sense. This statement can be proved in the same manner as was done in Sec. 1-3.3.

Observe that in a sum-of-products canonical form any particular minterm may or may not be present. Since there are 2^n minterms, we can have only 2^{2^n} different sum-of-products canonical forms. These canonical forms include the sum-of-products canonical form of 0 in which no minterm is present in the sum and also the sum-of-products canonical form of 1 where all the minterms are present in the sum. In any case, every Boolean expression in n variables is equivalent to exactly one of the 2^{2^n} Boolean expressions which have the sum-of-products canonical form. This fact allows us to partition the set of all the Boolean expressions into 2^{2^n} equivalence classes.

Equations (1) and (2) suggest that the 2^n minterms behave as the atoms of a Boolean algebra. Let us denote this Boolean algebra by $\langle B_{2^n}, *, \oplus, ', 0, 1\rangle$. All the elements of B_{2^n} can be obtained from the minterms by the operation of join $(\oplus)$. This means that the 2^{2^n} sum-of-products canonical forms are the elements of this Boolean algebra. In fact, any Boolean expression in the variables $x_1, x_2, \ldots, x_n$ is equal to one of the elements of B_{2^n} and hence represents an element of this algebra. For this reason, $\langle B_{2^n}, *, \oplus, ', 0, 1\rangle$ is called a *free Boolean algebra* generated by $x_1, x_2, \ldots, x_n$. The order of B_{2^n} is clearly 2^{2^n}, and the equality of two elements of B_{2^n} is understood to mean their equivalence as Boolean expressions.

EXAMPLE 1 Write the following Boolean expressions in an equivalent sum-of-products canonical form in three variables x_1, x_2, and x_3: (*a*) $x_1 * x_2$; (*b*) $x_1 \oplus x_2$; and (*c*) $(x_1 \oplus x_2)' * x_3$.

SOLUTION

$$(a)\quad x_1 * x_2 = x_1 * x_2 * (x_3 \oplus x_3')$$
$$= (x_1 * x_2 * x_3) \oplus (x_1 * x_2 * x_3')$$
$$= \min_6 \oplus \min_7 = \oplus\, 6, 7$$

$$(b)\quad x_1 \oplus x_2 = [x_1 * (x_2 \oplus x_2')] \oplus [x_2 * (x_1 \oplus x_1')]$$
$$= (x_1 * x_2) \oplus (x_1 * x_2') \oplus (x_2 * x_1) \oplus (x_1' * x_2)$$
$$= (x_1 * x_2) \oplus (x_1 * x_2') \oplus (x_1' * x_2)$$
$$= [(x_1 * x_2) * (x_3 \oplus x_3')] \oplus (x_1 * x_2') * (x_3 \oplus x_3')$$
$$\oplus [(x_1' * x_2) * (x_3 \oplus x_3')]$$
$$= (x_1 * x_2 * x_3) \oplus (x_1 * x_2 * x_3') \oplus (x_1 * x_2' * x_3)$$
$$\oplus (x_1 * x_2' * x_3') \oplus (x_1' * x_2 * x_3) \oplus (x_1' * x_2 * x_3')$$
$$= \min_7 \oplus \min_6 \oplus \min_5 \oplus \min_4 \oplus \min_3 \oplus \min_2$$
$$= \oplus\, 2, 3, 4, 5, 6, 7$$

$$(c)\quad (x_1 \oplus x_2)' * x_3 = (x_1' * x_2') * x_3 = \min_1$$ ////

EXAMPLE 2 Show that

$$(x_1' * x_2' * x_3' * x_4') \oplus (x_1' * x_2' * x_3' * x_4) \oplus (x_1' * x_2' * x_3 * x_4)$$
$$\oplus (x_1' * x_2' * x_3 * x_4') = x_1' * x_2'$$

SOLUTION

$$(x_1' * x_2' * x_3' * x_4') \oplus (x_1' * x_2' * x_3' * x_4) = x_1' * x_2' * x_3'$$
$$(x_1' * x_2' * x_3 * x_4) \oplus (x_1' * x_2' * x_3 * x_4') = x_1' * x_2' * x_3$$

Hence the given formula is equal to

$$(x_1' * x_2' * x_3') \oplus (x_1' * x_2' * x_3) = x_1' * x_2'$$ ////

Examples 1 and 2 show that for a given Boolean expression its equivalent sum-of-products canonical form is not necessarily simpler or shorter as a string. However, if we wish to determine whether any two given Boolean expressions are equivalent to one another, it may be easier to obtain their sum-of-products canonical forms and compare these forms, rather than trying to reduce one Boolean form into another. The process of obtaining the sum-of-products canonical form of a Boolean expression can be made mechanical.

In Sec. 1-2.15 it was shown that a switching circuit can be built to perform a certain task, which is first described by a number of statements. These statements are replaced by switches and gates, and the description of the circuit is obtained in terms of Boolean expressions. A circuit based upon such an expression may not be the most economical in a certain sense. Therefore, one may seek equivalent Boolean expressions which may be simpler (or more economical as a circuit from the point of view of the components used). Therefore, one is generally interested in obtaining either equivalent Boolean expressions which are simpler

in that either the number of variables is smaller or the number of operations is fewer, or an expression in which only certain types of operations appear but not others. The final expression to be obtained will depend upon the criteria used. We shall discuss some such methods in Sec. 4-4.

It is possible to repeat our discussion of canonical form using maxterms in place of minterms and showing that every Boolean expression in n variables is equivalent to a Boolean expression consisting of the product of maxterms only. Such a canonical form is known as the *product-of-sums canonical form*. The representation of maxterms and the method of obtaining the product-of-sums canonical form of a Boolean expression follow on the same lines as given in Secs. 1-3.3 to 1-3.5. One can also obtain these results by using the principle of duality.

EXAMPLE 3 Obtain the product-of-sums canonical forms of the Boolean expressions given in Example 1.

SOLUTION

$$\begin{aligned} x_1 * x_2 &= [x_1 \oplus (x_2 * x_2')] * [x_2 \oplus (x_1 * x_1')] \\ &= (x_1 \oplus x_2) * (x_1 \oplus x_2') * (x_1 \oplus x_2) * (x_1' \oplus x_2) \\ &= (x_1 \oplus x_2) * (x_1 \oplus x_2') * (x_1' \oplus x_2) \\ &= (x_1 \oplus x_2 \oplus x_3) * (x_1 \oplus x_2 \oplus x_3') * (x_1 \oplus x_2' \oplus x_3) \\ &\quad * (x_1 \oplus x_2' \oplus x_3') * (x_1' \oplus x_2 \oplus x_3) * (x_1' \oplus x_2 \oplus x_3') \\ &= \max_0 * \max_1 * \max_2 * \max_3 * \max_4 * \max_5 \\ &= * 0, 1, 2, 3, 4, 5 \end{aligned}$$

One could obtain this result directly from the sum-of-products form given in Example 1. It is easier to obtain the sum-of-products form than the product-of-sums canonical form in this case. On the other hand, for $x_1 \oplus x_2$, one can obtain the product-of-sums form directly as

$$\begin{aligned} x_1 \oplus x_2 &= x_1 \oplus x_2 \oplus (x_3 * x_3') = (x_1 \oplus x_2 \oplus x_3) * (x_1 \oplus x_2 \oplus x_3') \\ &= \max_0 * \max_1 \end{aligned}$$

In any case, if one of the canonical forms is known, then the other canonical form can be obtained directly. ////

4-3.2 Values of Boolean Expressions and Boolean Functions

Let $\alpha(x_1, x_2, \ldots, x_n)$ be a Boolean expression in n variables and $\langle B, *, \oplus, ', 0, 1\rangle$ be any Boolean algebra whose elements are denoted by $a_1, a_2, \ldots$. Let $\langle a_1, a_2, \ldots, a_n\rangle$ be an n-tuple of B^n. If we replace x_1 by a_1, x_2 by a_2, ..., and x_n by a_n in the Boolean expression $\alpha(x_1, x_2, \ldots, x_n)$, we obtain an expression which represents an element of B. We shall denote the resulting expression by $\alpha(a_1, a_2, \ldots, a_n) \in B$ and call it the *value* of the Boolean expression $\alpha(x_1, x_2, \ldots, x_n)$ for the n-tuple $\langle a_1, a_2, \ldots, a_n\rangle \in B^n$. It is possible to determine the values of the Boolean expression $\alpha(x_1, x_2, \ldots, x_n)$ for every n-tuple of B^n. The process of determining all such values is called a *valuation process* over the Boolean algebra

$\langle B, *, \oplus, ', 0, 1\rangle$. In particular, if $B = \{0, 1\}$, the valuation process over the two-element Boolean algebra is called a *binary valuation process.*

We shall now interpret the equivalence of two Boolean expressions in terms of the valuation process over any Boolean algebra. Recall that two Boolean expressions are equivalent to one another if one can be obtained from the other by using the identities of a Boolean algebra. Let $\alpha(x_1, x_2, \ldots, x_n)$ and $\beta(x_1, x_2, \ldots, x_n)$ be two Boolean expressions which are equivalent to one another, and let $\langle B, *, \oplus, ', 0, 1\rangle$ be a Boolean algebra over which we evaluate the two expressions. Let $\langle a_1, a_2, \ldots, a_n\rangle$ be any n-tuple of B^n and $\alpha(a_1, a_2, \ldots, a_n)$ be the value of $\alpha(x_1, x_2, \ldots, x_n)$ for this n-tuple. Since all the identities of a Boolean algebra hold for the given Boolean algebra, it is possible to transform α'- $a_2, \ldots, a_n)$ into $\beta(a_1, a_2, \ldots, a_n)$ by the use of the identities. This implies t the values of the two Boolean expressions which are equivalent will be eq for any n-tuple of B^n, or that the valuation process over any Boolean algeb results in identical values of the expressions. We shall see later in this section that the converse result also holds.

This discussion suggests that the values of a given Boolean expression can be obtained either by directly replacing the variables with the members of the n-tuple or by first obtaining an equivalent expression which is simpler in some sense and then replacing the variables. These two procedures will be demonstrated by examples.

EXAMPLE 1 Find the value of

$$x_1 * x_2 * [(x_1 * x_4) \oplus x_2' \oplus (x_3 * x_1')]$$

for $x_1 = a$, $x_2 = 1$, $x_3 = b$, and $x_4 = 1$, where $a, b, 1 \in B$ and the Boolean algebra $\langle B, *, \oplus, ', 0, 1\rangle$ is shown in Fig. 4-3.1.

SOLUTION By substituting the required elements in the expression, we get

$$a * 1 * [(a * 1) \oplus 1' \oplus (b * a')] = a * [a \oplus 0 \oplus (b * b)]$$
$$= a * (a \oplus b) = a * 1 = a$$

Alternatively, the given expression can be written in an equivalent form as

$$(x_1 * x_2 * x_1 * x_4) \oplus (x_1 * x_2 * x_2') \oplus (x_1 * x_2 * x_3 * x_1')$$
$$= (x_1 * x_2 * x_4) \oplus 0 \oplus 0 = x_1 * x_2 * x_4$$

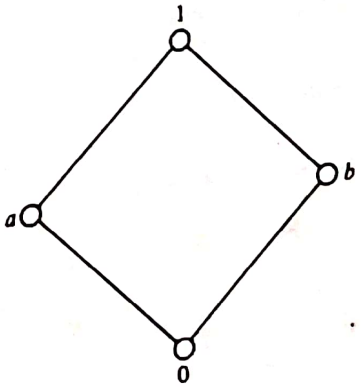

FIGURE 4-3.1

Now, replacing the variables, we get

$$a * 1 * 1 = a \qquad ////$$

EXAMPLE 2 Obtain the values of the Boolean forms

$$x_1 * (x_1' \oplus x_2) \qquad x_1 * x_2 \qquad \text{and} \qquad x_1 \oplus (x_1 * x_2)$$

over the ordered pairs of the two-element Boolean algebra.

SOLUTION Let $B = \{0, 1\}$. The elements of B^2 are listed in the first column of Table 4-3.1, and the values of the given Boolean expressions are given in the remaining columns. Observe that

$$x_1 * (x_1' \oplus x_2) = (x_1 * x_1') \oplus (x_1 * x_2) = 0 \oplus (x_1 * x_2) = x_1 * x_2$$

and

$$x_1 \oplus (x_1 * x_2) = x_1$$

This explains why the values of $x_1 * x_2$ and $x_1 * (x_1' \oplus x_2)$ are identical. Similarly, the values of x_1 and $x_1 \oplus (x_1 * x_2)$ are also identical. ////

We shall now examine the binary valuation process of a minterm. For this purpose let us consider the minterm

$$x_1^{m_1} * x_2^{m_2} * \cdots * x_n^{m_n}$$

in which $m_i = 0$ or 1 for $i = 1, 2, \ldots, n$. For the n-tuple $\langle m_1, m_2, \ldots, m_n \rangle \in \{0, 1\}^n$, the value of the minterm is 1, while for every other n-tuple of $\{0, 1\}^n$, its value is 0. For every minterm this is the case; viz., the value of the minterm is equal to 1 for exactly one n-tuple of $\{0, 1\}^n$ and 0 for all other n-tuples. Distinct minterms have the value 1 for distinct n-tuples of $\{0, 1\}^n$, and one can establish a one-to-one correspondence between the minterms and the n-tuples of $\{0, 1\}^n$ by associating with each minterm the n-tuple for which its value is 1.

Since the set $\{0, 1\}$ is a sub-Boolean algebra of any Boolean algebra, the values of a Boolean expression for any n-tuple of $\{0, 1\}^n$ must be either 0 or 1. Furthermore, the binary valuation process of a Boolean expression determines the minterms that are present in its sum-of-products canonical form. More precisely, appearing in the sum-of-products canonical form of a Boolean expression are only those minterms which are associated with the n-tuples of $\{0, 1\}^n$ for which the value of the Boolean expression is 1. This result shows that for a given Boolean expression, its equivalent sum-of-products canonical form can be completely determined by its binary valuation process.

Table 4-3.1

$\langle x_1, x_2 \rangle$	$x_1 * x_2$	x_1'	$x_1' \oplus x_2$	$x_1 * (x_1' \oplus x_2)$	$x_1 \oplus (x_1 * x_2)$
$\langle 0, 0 \rangle$	0	1	1	0	0
$\langle 0, 1 \rangle$	0	1	1	0	0
$\langle 1, 0 \rangle$	0	0	0	0	1
$\langle 1, 1 \rangle$	1	0	1	1	1

We know that any two Boolean expressions in n variables which are equivalent have the same sum-of-products canonical form. It is therefore possible to determine the equivalence of two Boolean expressions simply by their binary valuation process. However, once the equivalence is established, we can say that the values of two equivalent Boolean expressions must be equal for every n-tuple of B^n, where $\langle B, *, \oplus, ', 0, 1\rangle$ is any Boolean algebra. Thus, from a binary valuation process alone it is possible to make a statement about the valuation process over any Boolean algebra. It is easy to generalize this statement by saying that if two Boolean expressions have equal values for every n-tuple of a Boolean algebra, then they must have equal values for every n-tuple of any other Boolean algebra. This result follows from the fact that the equality of the values of two Boolean expressions over the n-tuples of any Boolean algebra implies their equality over the n-tuples of a two-element Boolean algebra which is a sub-Boolean algebra. These remarks can be rephrased by saying that any Boolean identity which holds for every n-tuple of one Boolean algebra holds for every Boolean algebra.

EXAMPLE 3 Show that the following Boolean expressions are equivalent to one another. Obtain their sum-of-products canonical form.

(a) $(x \oplus y) * (x' \oplus z) * (y \oplus z)$
(b) $(x * z) \oplus (x' * y) \oplus (y * z)$
(c) $(x \oplus y) * (x' \oplus z)$
(d) $(x * z) \oplus (x' * y)$

SOLUTION The binary valuations of the expressions are given in Table 4-3.2. Since the values of the given Boolean expressions are equal over every triple of the two-element Boolean algebra, they are equivalent.

We can show the equivalence of the Boolean expression alternatively as follows.

$$(c) = [(x \oplus y) * x'] \oplus [(x \oplus y) * z] = (x * x') \oplus (y * x') \oplus (x * z) \oplus (y * z) = (b)$$

From the principle of duality or in a similar manner we can show that $(a) = (d)$.

Table 4-3.2

x	y	z	$x \oplus y$	$x' \oplus z$	$y \oplus z$	(a)	(c)	$x * z$	$x' * y$	$y * z$	(b)	(d)
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	0	0	0
0	1	0	1	1	1	1	1	0	1	0	1	1
0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	0	0	0	1	1
1	1	0	1	0	1	0	0	1	0	0	0	0
1	1	1	1	1	0	1	1	0	0	1	1	1

Finally, we show that

$$\begin{aligned}(a) &= (x \oplus y) * [(x' * y) \oplus z] \\ &= [(x \oplus y) * (x' * y)] \oplus [(x \oplus y) * z] \\ &= \{[(x \oplus y) * y] * x'\} \oplus [(x * z) \oplus (y * z)] \\ &= (y * x') \oplus (x * z) \oplus (y * z) = (b)\end{aligned}$$

All the four Boolean forms have been shown to be equivalent to one another. They all have the same sum-of-products canonical form which can be obtained either directly from the table as $\min_2 \oplus \min_3 \oplus \min_5 \oplus \min_7$ or from (d) as

$$\begin{aligned}(d) &= [(x * z) * (y \oplus y')] \oplus [(x' * y) * (z \oplus z')] \\ &= (x * z * y) \oplus (x * z * y') \oplus (x' * y * z) \oplus (x' * y * z') \qquad ////\end{aligned}$$

Given a Boolean expression $\alpha(x_1, x_2, \ldots, x_n)$ and a Boolean algebra $\langle B, *, \oplus, ', 0, 1\rangle$, we can obtain the values of the Boolean expression for every n-tuple of B^n. Let us now consider a function $f_{\alpha,B}: B^n \to B$ such that for any n-tuple $\langle a_1, a_2, \ldots, a_n\rangle \in B^n$, the value of $f_{\alpha,B}$ is equal to the value of the Boolean expression $\alpha(x_1, x_2, \ldots, x_n)$, that is,

$$f_{\alpha,B}(a_1, a_2, \ldots, a_n) = \alpha(a_1, a_2, \ldots, a_n)$$

for all $\langle a_1, a_2, \ldots, a_n\rangle \in B^n$. We shall call $f_{\alpha,B}$ the *function associated with* (or *described by*) the Boolean expression $\alpha(x_1, x_2, \ldots, x_n)$.

It follows from the definition that any two Boolean expressions which are equivalent to one another describe the same function irrespective of the Boolean algebra under consideration; that is, if $\alpha(x_1, x_2, \ldots, x_n) = \beta(x_1, x_2, \ldots, x_n)$, then $f_{\alpha,B} = f_{\beta,B}$ for every Boolean algebra B. Conversely, if for any two Boolean expressions $\alpha(x_1, x_2, \ldots, x_n)$ and $\beta(x_1, x_2, \ldots, x_n)$, the functions described by them over any Boolean algebra are equal, that is, if $f_{\alpha,B} = f_{\beta,B}$, then $\alpha(x_1, x_2, \ldots, x_n) = \beta(x_1, x_2, \ldots, x_n)$. This result shows that a function associated with a Boolean expression is defined over any Boolean algebra, if it is defined over a Boolean algebra. In such cases, it is most convenient to define the function over a two-element Boolean algebra.

Let us see how we can determine the values of such a function which is associated with a Boolean expression, say $\alpha(x_1, x_2, \ldots, x_n)$, and whose values are given over the n-tuples of a two-element Boolean algebra. To determine its values for an n-tuple of some other Boolean algebra, all we need to do is write the sum-of-products canonical form of the Boolean expression from its binary valuation, which is given in this case. The values of the given expression are the same as the values of its sum-of-products canonical form, which in turn are the values of the function. This idea is demonstrated by the following example.

EXAMPLE 4 Find the value of the function $f_{\alpha,B}: B^3 \to B$ for $x_1 = a$, $x_2 = 1$, and $x_3 = b$ where a, b, 1 are the elements of the Boolean algebra given in Example 1 and $\alpha(x_1, x_2, x_3)$ is the expression whose binary valuation is given in Table 4-3.3.

Table 4-3.3

$\langle x_1, x_2, x_3 \rangle$	$\alpha(x_1, x_2, x_3)$
$\langle 0, 0, 0 \rangle$	1
$\langle 0, 0, 1 \rangle$	0
$\langle 0, 1, 0 \rangle$	1
$\langle 0, 1, 1 \rangle$	1
$\langle 1, 0, 0 \rangle$	0
$\langle 1, 0, 1 \rangle$	1
$\langle 1, 1, 0 \rangle$	0
$\langle 1, 1, 1 \rangle$	0

SOLUTION From the table we can immediately obtain the sum-of-products form of $\alpha(x_1, x_2, x_3)$ by selecting for each value 1 a corresponding minterm

$$(x_1' * x_2' * x_3') \oplus (x_1' * x_2 * x_3') \oplus (x_1' * x_2 * x_3) \oplus (x_1 * x_2' * x_3)$$

The value of $\alpha(a, 1, b)$ is the same as the value of the corresponding canonical form, viz.,

$$\alpha(a, 1, b) = (b * 0 * a) \oplus (b * 1 * a) \oplus (b * 1 * b) \oplus (a * 0 * b) = b$$

////

Definition 4-3.4 Let $\langle B, *, \oplus, ', 0, 1 \rangle$ be a Boolean algebra. A function $f: B^n \to B$ which is associated with a Boolean expression in n variables is called a *Boolean function.*

Observe that not every function $g: B^n \to B$ is a Boolean function. If we assume that the Boolean algebra B is of order 2^m for $m > 1$, then it is easy to see that the number of functions from B^n to B is greater than 2^{2^n} showing that there are functions from B^n to B which are not Boolean functions. On the other hand, for $m = 1$, that is, for a two-element Boolean algebra, the number of functions from B^n to B is 2^{2^n}, which is the same as the number of distinct Boolean expressions in n variables. Hence every function from B^n to B in this case is a Boolean function.

From the definition of a Boolean function it is clear that there exists a one-to-one correspondence between the set of Boolean functions and the elements of a free Boolean algebra. Let us denote the set of Boolean functions in n variables by F_n. For any $f \in F_n$ associated with a Boolean expression α in n variables, let us denote the Boolean function associated with $(\alpha)'$ by $\bar{f}$. Clearly $\bar{f} \in F_n$. Next, for any two Boolean functions $f, g \in F_n$ associated with Boolean expressions α and β respectively, denote the Boolean functions corresponding to $(\alpha) * (\beta)$ and $(\alpha) \oplus (\beta)$ by $f \wedge g$ and $f \vee g$. Again, $f \wedge g, f \vee g \in F_n$. Finally, let us denote the Boolean functions associated with the Boolean expressions 1 and 0 by f_1 and f_0 respectively. Since the operations $\wedge$, $\vee$, and $-$ as well as the elements f_0 and f_1 on F_n are defined in terms of the operations $*$, $\oplus$, $'$ and the elements 0 and 1 of the free Boolean algebra, it is easy to see that $\langle F_n, \wedge, \vee, -, f_0, f_1 \rangle$ is a Boolean algebra of order 2^{2^n}. Furthermore, if we define a mapping

$g: F_n \rightarrow B_{2^n}$ such that for any $f \in F_n$, $g(f) = \alpha$, where α is the Boolean expression associated with the function f, then clearly g is an isomorphism. The two Boolean algebras $\langle F_n, \wedge, \vee, -, f_0, f_1 \rangle$ and $\langle B_{2^n}, *, \oplus, ', 0, 1 \rangle$ are isomorphic to one another. Because of this isomorphism the terms "Boolean expression" and "Boolean function" are often used interchangeably.

In the remaining part of this section we discuss a special class of Boolean expressions or functions which are symmetric in some of the variables. Recognition of such symmetries permits considerable simplification in the design of circuits which such functions may represent. We shall discuss only some basic notions of symmetries here.

A Boolean expression in n variables $x_1, x_2, \ldots, x_n$ is said to be *pairwise symmetric* with respect to the variables x_i and x_j, if by interchanging the variables x_i and x_j throughout the expression we obtain an equivalent expression. Accordingly, a Boolean expression $\alpha(x_1, x_2, \ldots, x_n)$ is symmetric in x_i and x_j if

$$\alpha(x_1, x_2, \ldots, x_i, \ldots, x_j, \ldots, x_n) = \alpha(x_1, x_2, \ldots, x_j, \ldots, x_i, \ldots, x_n)$$

For example, the expression

$$(x_1 * x_2) \oplus (x_2' * x_3') \oplus (x_1 * x_3)$$

is symmetric in x_2 and x_3 but not symmetric in x_1 and x_3. Similarly, the expression

$$(x_1 * x_2) \oplus (x_3 * x_4')$$

is symmetric in x_1 and x_2. It is also symmetric in x_3 and x_4'. We shall, however, restrict ourselves to the consideration of symmetries in the variables and not in their complements.

The definition of symmetry can be generalized to include symmetry with respect to all the variables.

Definition 4-3.5 A Boolean expression in n variables $x_1, x_2, \ldots, x_n$ is called *symmetric* if interchanging any two variables results in an equivalent expression.

Note that the definition permits the interchange of any number of variables, and the resulting expression will still remain equivalent to the original symmetric expression. The following are some examples of symmetric Boolean expressions in two and three variables:

(a) $(x_1 * x_2') \oplus (x_1' * x_2)$

(b) $(x_1' * x_2') \oplus (x_1 \oplus x_2)$

(c) $(x_1 * x_2' * x_3') \oplus (x_1' * x_2 * x_3') \oplus (x_1' * x_2' * x_3) \oplus (x_1 * x_2 * x_3)$

(d) $(x_1 * x_2 * x_3') \oplus (x_1 * x_2' * x_3) \oplus (x_1' * x_2 * x_3)$

Let us now consider the binary valuation of a symmetric expression in n variables. Clearly, its value will not change if the number of variables which are assigned the value 1 remains fixed. This means that its value remains independent of the particular variables which are assigned the value 1 because of the fact that the value of a symmetric function remains unchanged with inter-

changes of variables. This leads to a simple criterion which will now be stated as a theorem.

Theorem 4-3.1 A necessary and sufficient condition that a Boolean expression in n variables is symmetric is that there exists a set of numbers $n_1, n_2, \ldots, n_k$ such that the value of the expression is 1 if any number n_i of the variables for $i = 1, 2, \ldots, k$ is assigned the value 1 in a binary valuation process.

The numbers $n_1, n_2, \ldots, n_k$ are called the *characteristic numbers* of the symmetric function. The proof of the theorem depends upon the definition of symmetry and is straightforward. We shall, however, omit it. Observe that the characteristic numbers of the symmetric functions are 1 in (a), 0 and 2 in (b), 1 and 3 in (c), and 2 in (d).

An important consequence of the theorem is the fact that in the sum-of-products canonical form of a symmetric function only certain combinations of minterms must appear. If there is a minterm in the sum-of-products form of a symmetric expression whose value is 1, if m of the variables are assigned the value 1, then all those minterms must also be present whose values are 1 for exactly m variables. Obviously, m is one of the characteristic numbers in this case. Using this information, one can easily write symmetric functions with a given set of characteristic numbers. For example, with $n = 3$, the expression with characteristic number 0 and 2 is given by

$$(x_1' * x_2' * x_3') \oplus (x_1 * x_2 * x_3') \oplus (x_1 * x_2' * x_3) \oplus (x_1' * x_2 * x_3)$$

EXERCISE 4-3

1 Obtain the sum-of-products canonical forms of the following Boolean expressions.
   (a) $x_1 \oplus x_2$
   (b) $x_1 \oplus (x_2 * x_3')$
   (c) $(x_1 \oplus x_2)' \oplus (x_1' * x_3)$
   (d) $(x_1 * x_2') \oplus x_4$ (assuming that this is an expression in four variables x_1, x_2, x_3, and x_4)

2 Obtain the sum-of-products and product-of-sums canonical forms of the following expressions.
   (a) $x_1x_2' + x_3$
   (b) $[(x_1 + x_2)(x_3x_4)']'$
   (c) $x_2' + [x_3' + x_1 + (x_2x_3)']'](x_3 + x_1'x_2)$

3 If $\beta(x_1, x_2, \ldots, x_n)$ is the dual of $\alpha(x_1, x_2, \ldots, x_n)$, then show that

$$[\alpha(x_1, x_2, \ldots, x_n)]' = \beta(x_1', x_2', \ldots, x_n')$$

Here we have used the notations and conventions given in Prob. 2, Exercises 4-2. Show how this result is used in obtaining the product-of-sums canonical form of a given expression from its sum-of-products canonical form.

4 Show that
   (a) $[a * (b' \oplus c)]' * [b' \oplus (a * c')']' = a * b * c'$
   (b) $a' * [(b' \oplus c)' \oplus (b * c)] \oplus [(a \oplus b')' * c] = a' * b$

5 Given an expression $\alpha(x_1, x_2, x_3)$ defined to be $\sum 0, 3, 5, 7$, determine the value of

$\alpha(a, b, 1)$ where $a, b, 1 \in B$ and $\langle B, *, \oplus, 0, 1\rangle$ is the Boolean algebra given in Fig. 4-1.1*b*.

6 Obtain simplified Boolean expressions which are equivalent to these expressions:
(*a*) $m_0 + m_7$
(*b*) $m_0 + m_1 + m_2 + m_3$
(*c*) $m_5 + m_7 + m_9 + m_{11} + m_{13}$
where m_j are the minterms in the variables x_1, x_2, x_3, and x_4.

7 Let B be a Boolean algebra with 2^n elements. Show that the number of sub-Boolean algebras of B is equal to the number of partitions of a set with n elements.

8 Let n be an integer $(n > 1)$ and B be the set of divisors of n. For any $a, b \in B$, let $a \oplus b = \text{LCM}\ (a, b)$, $a * b = \text{GCD}\ (a, b)$, and $a' = n/a$. Show that $\langle B, *, \oplus, ', 0, 1\rangle$ is a Boolean algebra if n is not divisible by a square greater than 1. What are the sub-Boolean algebras of this algebra? Draw the diagram of the Boolean algebras for $n = 30$.

9 Show that the symmetric functions form a Boolean algebra.

10 Determine whether the following functions are symmetric.
(*a*) $a'bc' + a'c'd + a'bcd + abc'd$
(*b*) $a'bcd + a'c' + b'c'd' + ad'$
(*c*) $abc' + ab'c + a'bc + ab'c' + a'bc' + a'b'c$

4-4 REPRESENTATION AND MINIMIZATION OF BOOLEAN FUNCTIONS

In this section we are primarily concerned with the problem of obtaining a Boolean function which is minimal according to some criterion, such as the minimum numbers of gates and/or inputs, and which is equivalent to a given Boolean function. Such problems arise in the design of switching circuits. In a typical situation, the operational requirements of the circuit are stated in verbal form. These statements are subsequently transformed into a Boolean equation for the required output. The next step is to obtain an equation which is logically equivalent to the output equation which will result in a least expensive physical realization. The desired circuit is finally obtained by replacing the Boolean connectives in the minimal equation by appropriate logic blocks such as *OR* gates, *AND* gates, inverters, etc. The ideas of physical realization and logic blocks were introduced in Sec. 1-2.15. Furthermore, the notion of equivalent Boolean functions was introduced in Sec. 4-3.1 where examples of algebraic simplification were also given.