CHAPTER

12 Boolean Algebra

The circuits in computers and other electronic devices have inputs, each of which is either a 0 or a 1, and produce outputs that are also 0s and 1s. Circuits can be constructed using any basic element that has two different states. Such elements include switches that can be in either the on or the off position and optical devices that can be either lit or unlit. In 1938 Claude Shannon showed how the basic rules of logic, first given by George Boole in 1854 in his *The Laws of Thought*, could be used to design circuits. These rules form the basis for Boolean algebra. In this chapter we develop the basic properties of Boolean algebra. The operation of a circuit is defined by a Boolean function that specifies the value of an output for each set of inputs. The first step in constructing a circuit is to represent its Boolean function by an expression built up using the basic operations of Boolean algebra. We will provide an algorithm for producing such expressions. The expression that we obtain may contain many more operations than are necessary to represent the function. Later in the chapter we will describe methods for finding an expression with the minimum number of sums and products that represents a Boolean function. The procedures that we will develop, Karnaugh maps and the Quine–McCluskey method, are important in the design of efficient circuits.

12.1 Boolean Functions

Introduction

Boolean algebra provides the operations and the rules for working with the set $\{0, 1\}$. Electronic and optical switches can be studied using this set and the rules of Boolean algebra. The three operations in Boolean algebra that we will use most are complementation, the Boolean sum, and the Boolean product. The **complement** of an element, denoted with a bar, is defined by $\overline{0} = 1$ and $\overline{1} = 0$. The Boolean sum, denoted by $+$ or by *OR*, has the following values:

$$1 + 1 = 1, \qquad 1 + 0 = 1, \qquad 0 + 1 = 1, \qquad 0 + 0 = 0.$$

The Boolean product, denoted by $\cdot$ or by *AND*, has the following values:

$$1 \cdot 1 = 1, \qquad 1 \cdot 0 = 0, \qquad 0 \cdot 1 = 0, \qquad 0 \cdot 0 = 0.$$

When there is no danger of confusion, the symbol $\cdot$ can be deleted, just as in writing algebraic products. Unless parentheses are used, the rules of precedence for Boolean operators are: first, all complements are computed, followed by all Boolean products, followed by all Boolean sums. This is illustrated in Example 1.

EXAMPLE 1 Find the value of $1 \cdot 0 + \overline{(0 + 1)}$.

Solution: Using the definitions of complementation, the Boolean sum, and the Boolean product, it follows that

$$\begin{aligned} 1 \cdot 0 + \overline{(0 + 1)} &= 0 + \overline{1} \\ &= 0 + 0 \\ &= 0. \end{aligned}$$

◀

The complement, Boolean sum, and Boolean product correspond to the logical operators, $\neg$, $\vee$, and $\wedge$, respectively, where 0 corresponds to **F** (false) and 1 corresponds to **T** (true). Equalities in Boolean algebra can be directly translated into equivalences of compound propositions. Conversely, equivalences of compound propositions can be translated into equalities in Boolean algebra. We will see later in this section why these translations yield valid logical equivalences and identities in Boolean algebra. Example 2 illustrates the translation from Boolean algebra to propositional logic.

EXAMPLE 2 Translate $1 \cdot 0 + \overline{(0+1)} = 0$, the equality found in Example 1, into a logical equivalence.

Solution: We obtain a logical equivalence when we translate each 1 into a **T**, each 0 into an **F**, each Boolean sum into a disjunction, each Boolean product into a conjunction, and each complementation into a negation. We obtain

$$(\mathbf{T} \wedge \mathbf{F}) \vee \neg(\mathbf{T} \vee \mathbf{F}) \equiv \mathbf{F}. \qquad ◀$$

Example 3 illustrates the translation from propositional logic to Boolean algebra.

EXAMPLE 3 Translate the logical equivalence $(\mathbf{T} \wedge \mathbf{T}) \vee \neg\mathbf{F} \equiv \mathbf{T}$ into an identity in Boolean algebra.

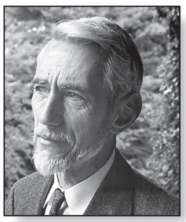
Solution: We obtain an identity in Boolean algebra when we translate each **T** into a 1, each **F** into a 0, each disjunction into a Boolean sum, each conjunction into a Boolean product, and each negation into a complementation. We obtain

$$(1 \cdot 1) + \overline{0} = 1. \qquad ◀$$

Boolean Expressions and Boolean Functions

Let $B = \{0, 1\}$. Then $B^n = \{(x_1, x_2, \ldots, x_n) \mid x_i \in B \text{ for } 1 \le i \le n\}$ is the set of all possible n-tuples of 0s and 1s. The variable x is called a **Boolean variable** if it assumes values only from B, that is, if its only possible values are 0 and 1. A function from B^n to B is called a **Boolean function of degree *n***.

Links



CLAUDE ELWOOD SHANNON (1916–2001) Claude Shannon was born in Petoskey, Michigan, and grew up in Gaylord, Michigan. His father was a businessman and a probate judge, and his mother was a language teacher and a high school principal. Shannon attended the University of Michigan, graduating in 1936. He continued his studies at M.I.T., where he took the job of maintaining the differential analyzer, a mechanical computing device consisting of shafts and gears built by his professor, Vannevar Bush. Shannon's master's thesis, written in 1936, studied the logical aspects of the differential analyzer. This master's thesis presents the first application of Boolean algebra to the design of switching circuits; it is perhaps the most famous master's thesis of the twentieth century. He received his Ph.D. from M.I.T. in 1940. Shannon joined Bell Laboratories in 1940, where he worked on transmitting data efficiently. He was one of the first people to use bits to represent information. At Bell Laboratories he worked on determining the amount of traffic that telephone lines can carry. Shannon made many fundamental contributions to information theory. In the early 1950s he was one of the founders of the study of artificial intelligence. He joined the M.I.T. faculty in 1956, where he continued his study of information theory.

Shannon had an unconventional side. He is credited with inventing the rocket-powered Frisbee. He is also famous for riding a unicycle down the hallways of Bell Laboratories while juggling four balls. Shannon retired when he was 50 years old, publishing papers sporadically over the following 10 years. In his later years he concentrated on some pet projects, such as building a motorized pogo stick. One interesting quote from Shannon, published in *Omni Magazine* in 1987, is "I visualize a time when we will be to robots what dogs are to humans. And I am rooting for the machines."

EXAMPLE 4 The function $F(x, y) = x\overline{y}$ from the set of ordered pairs of Boolean variables to the set $\{0, 1\}$ is a Boolean function of degree 2 with $F(1, 1) = 0$, $F(1, 0) = 1$, $F(0, 1) = 0$, and $F(0, 0) = 0$. We display these values of F in Table 1. ◀

TABLE 1

x	y	$F(x, y)$
1	1	0
1	0	1
0	1	0
0	0	0

Boolean functions can be represented using expressions made up from variables and Boolean operations. The **Boolean expressions** in the variables $x_1, x_2, \ldots, x_n$ are defined recursively as

$0, 1, x_1, x_2, \ldots, x_n$ are Boolean expressions;
if E_1 and E_2 are Boolean expressions, then $\overline{E}_1$, (E_1E_2), and $(E_1 + E_2)$ are Boolean expressions.

Each Boolean expression represents a Boolean function. The values of this function are obtained by substituting 0 and 1 for the variables in the expression. In Section 12.2 we will show that every Boolean function can be represented by a Boolean expression.

EXAMPLE 5 Find the values of the Boolean function represented by $F(x, y, z) = xy + \overline{z}$.

Solution: The values of this function are displayed in Table 2. ◀

TABLE 2

x	y	z	xy	$\overline{z}$	$F(x, y, z) = xy + \overline{z}$
1	1	1	1	0	1
1	1	0	1	1	1
1	0	1	0	0	0
1	0	0	0	1	1
0	1	1	0	0	0
0	1	0	0	1	1
0	0	1	0	0	0
0	0	0	0	1	1

Note that we can represent a Boolean function graphically by distinguishing the vertices of the n-cube that correspond to the n-tuples of bits where the function has value 1.

EXAMPLE 6 The function $F(x, y, z) = xy + \overline{z}$ from B^3 to B from Example 5 can be represented by distinguishing the vertices that correspond to the five 3-tuples $(1, 1, 1)$, $(1, 1, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 0)$, where $F(x, y, z) = 1$, as shown in Figure 1. These vertices are displayed using solid black circles. ◀

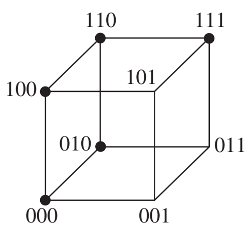


FIGURE 1

Boolean functions F and G of n variables are equal if and only if $F(b_1, b_2, \ldots, b_n) = G(b_1, b_2, \ldots, b_n)$ whenever $b_1, b_2, \ldots, b_n$ belong to B. Two different Boolean expressions that represent the same function are called **equivalent**. For instance, the Boolean expressions xy, $xy + 0$, and $xy \cdot 1$ are equivalent. The **complement** of the Boolean function F is the function $\overline{F}$, where $\overline{F}(x_1, \ldots, x_n) = \overline{F(x_1, \ldots, x_n)}$. Let F and G be Boolean functions of degree n. The **Boolean sum** $F + G$ and the **Boolean product** FG are defined by

$$(F + G)(x_1, \ldots, x_n) = F(x_1, \ldots, x_n) + G(x_1, \ldots, x_n),$$
$$(FG)(x_1, \ldots, x_n) = F(x_1, \ldots, x_n)G(x_1, \ldots, x_n).$$

A Boolean function of degree two is a function from a set with four elements, namely, pairs of elements from $B = \{0, 1\}$, to B, a set with two elements. Hence, there are 16 different Boolean functions of degree two. In Table 3 we display the values of the 16 different Boolean functions of degree two, labeled $F_1, F_2, \ldots, F_{16}$.

TABLE 3 **The 16 Boolean Functions of Degree Two.**

x	y	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

EXAMPLE 7 How many different Boolean functions of degree n are there?

Solution: From the product rule for counting, it follows that there are 2^n different n-tuples of 0s and 1s. Because a Boolean function is an assignment of 0 or 1 to each of these 2^n different n-tuples, the product rule shows that there are 2^{2^n} different Boolean functions of degree n. ◀

Table 4 displays the number of different Boolean functions of degrees one through six. The number of such functions grows extremely rapidly.

TABLE 4 **The Number of Boolean Functions of Degree n.**

Degree	*Number*
1	4
2	16
3	256
4	65,536
5	4,294,967,296
6	18,446,744,073,709,551,616

Identities of Boolean Algebra

There are many identities in Boolean algebra. The most important of these are displayed in Table 5. These identities are particularly useful in simplifying the design of circuits. Each of the identities in Table 5 can be proved using a table. We will prove one of the distributive laws in this way in Example 8. The proofs of the remaining properties are left as exercises for the reader.

EXAMPLE 8 Show that the distributive law $x(y + z) = xy + xz$ is valid.

Solution: The verification of this identity is shown in Table 6. The identity holds because the last two columns of the table agree. ◀

The reader should compare the Boolean identities in Table 5 to the logical equivalences in Table 6 of Section 1.3 and the set identities in Table 1 in Section 2.2. All are special cases of the same set of identities in a more abstract structure. Each collection of identities can be obtained by making the appropriate translations. For example, we can transform each of the identities in Table 5 into a logical equivalence by changing each Boolean variable into a propositional variable, each 0 into a **F**, each 1 into a **T**, each Boolean sum into a disjunction, each Boolean product into a conjunction, and each complementation into a negation, as we illustrate in Example 9.

TABLE 5 **Boolean Identities.**

Identity	*Name*
$\overline{\overline{x}} = x$	Law of the double complement
$x + x = x$ $x \cdot x = x$	Idempotent laws
$x + 0 = x$ $x \cdot 1 = x$	Identity laws
$x + 1 = 1$ $x \cdot 0 = 0$	Domination laws
$x + y = y + x$ $xy = yx$	Commutative laws
$x + (y + z) = (x + y) + z$ $x(yz) = (xy)z$	Associative laws
$x + yz = (x + y)(x + z)$ $x(y + z) = xy + xz$	Distributive laws
$\overline{(xy)} = \overline{x} + \overline{y}$ $\overline{(x + y)} = \overline{x}\,\overline{y}$	De Morgan's laws
$x + xy = x$ $x(x + y) = x$	Absorption laws
$x + \overline{x} = 1$	Unit property
$x\overline{x} = 0$	Zero property

Compare these Boolean identities with the logical equivalences in Section 1.3 and the set identities in Section 2.2.

EXAMPLE 9 Translate the distributive law $x + yz = (x + y)(x + z)$ in Table 5 into a logical equivalence.

Solution: To translate a Boolean identity into a logical equivalence, we change each Boolean variable into a propositional variable. Here we will change the Boolean variables x, y, and z into the propositional variables p, q, and r. Next, we change each Boolean sum into a disjunction and each Boolean product into a conjunction. (Note that 0 and 1 do not appear in this identity and

TABLE 6 **Verifying One of the Distributive Laws.**

x	y	z	$y + z$	xy	xz	$x(y + z)$	$xy + xz$
1	1	1	1	1	1	1	1
1	1	0	1	1	0	1	1
1	0	1	1	0	1	1	1
1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0

complementation also does not appear.) This transforms the Boolean identity into the logical equivalence

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r).$$

This logical equivalence is one of the distributive laws for propositional logic in Table 6 in Section 1.3. ◀

Identities in Boolean algebra can be used to prove further identities. We demonstrate this in Example 10.

EXAMPLE 10 Prove the **absorption law** $x(x + y) = x$ using the other identities of Boolean algebra shown in Table 5. (This is called an absorption law because absorbing $x + y$ into x leaves x unchanged.)

Extra Examples

Solution: We display steps used to derive this identity and the law used in each step:

$$\begin{aligned}
x(x + y) &= (x + 0)(x + y) && \text{Identity law for the Boolean sum} \\
&= x + 0 \cdot y && \text{Distributive law of the Boolean sum over the Boolean product} \\
&= x + y \cdot 0 && \text{Commutative law for the Boolean product} \\
&= x + 0 && \text{Domination law for the Boolean product} \\
&= x && \text{Identity law for the Boolean sum.}
\end{aligned}$$

◀

Duality

The identities in Table 5 come in pairs (except for the law of the double complement and the unit and zero properties). To explain the relationship between the two identities in each pair we use the concept of a dual. The **dual** of a Boolean expression is obtained by interchanging Boolean sums and Boolean products and interchanging 0s and 1s.

EXAMPLE 11 Find the duals of $x(y + 0)$ and $\overline{x} \cdot 1 + (\overline{y} + z)$.

Solution: Interchanging $\cdot$ signs and $+$ signs and interchanging 0s and 1s in these expressions produces their duals. The duals are $x + (y \cdot 1)$ and $(\overline{x} + 0)(\overline{y}z)$, respectively. ◀

The dual of a Boolean function F represented by a Boolean expression is the function represented by the dual of this expression. This dual function, denoted by F^d, does not depend on the particular Boolean expression used to represent F. An identity between functions represented by Boolean expressions remains valid when the duals of both sides of the identity are taken. (See Exercise 30 for the reason why this is true.) This result, called the **duality principle**, is useful for obtaining new identities.

EXAMPLE 12 Construct an identity from the absorption law $x(x + y) = x$ by taking duals.

Solution: Taking the duals of both sides of this identity produces the identity $x + xy = x$, which is also called an absorption law and is shown in Table 5. ◀

The Abstract Definition of a Boolean Algebra

In this section we have focused on Boolean functions and expressions. However, the results we have established can be translated into results about propositions or results about sets. Because of this, it is useful to define Boolean algebras abstractly. Once it is shown that a particular structure is a Boolean algebra, then all results established about Boolean algebras in general apply to this particular structure.

Boolean algebras can be defined in several ways. The most common way is to specify the properties that operations must satisfy, as is done in Definition 1.

DEFINITION 1 A *Boolean algebra* is a set B with two binary operations $\vee$ and $\wedge$, elements 0 and 1, and a unary operation $\overline{}$ such that these properties hold for all x, y, and z in B:

$$\left.\begin{aligned} x \vee 0 &= x \\ x \wedge 1 &= x \end{aligned}\right\} \quad \text{Identity laws}$$

$$\left.\begin{aligned} x \vee \overline{x} &= 1 \\ x \wedge \overline{x} &= 0 \end{aligned}\right\} \quad \text{Complement laws}$$

$$\left.\begin{aligned} (x \vee y) \vee z &= x \vee (y \vee z) \\ (x \wedge y) \wedge z &= x \wedge (y \wedge z) \end{aligned}\right\} \quad \text{Associative laws}$$

$$\left.\begin{aligned} x \vee y &= y \vee x \\ x \wedge y &= y \wedge x \end{aligned}\right\} \quad \text{Commutative laws}$$

$$\left.\begin{aligned} x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \\ x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) \end{aligned}\right\} \quad \text{Distributive laws}$$

Using the laws given in Definition 1, it is possible to prove many other laws that hold for every Boolean algebra, such as idempotent and domination laws. (See Exercises 35–42.)

From our previous discussion, $B = \{0, 1\}$ with the *OR* and *AND* operations and the complement operator, satisfies all these properties. The set of propositions in n variables, with the $\vee$ and $\wedge$ operators, **F** and **T**, and the negation operator, also satisfies all the properties of a Boolean algebra, as can be seen from Table 6 in Section 1.3. Similarly, the set of subsets of a universal set U with the union and intersection operations, the empty set and the universal set, and the set complementation operator, is a Boolean algebra as can be seen by consulting Table 1 in Section 2.2. So, to establish results about each of Boolean expressions, propositions, and sets, we need only prove results about abstract Boolean algebras.

Boolean algebras may also be defined using the notion of a lattice, discussed in Chapter 9. Recall that a lattice L is a partially ordered set in which every pair of elements x, y has a least upper bound, denoted by $\text{lub}(x, y)$ and a greatest lower bound denoted by $\text{glb}(x, y)$. Given two elements x and y of L, we can define two operations $\vee$ and $\wedge$ on pairs of elements of L by $x \vee y = \text{lub}(x, y)$ and $x \wedge y = \text{glb}(x, y)$.

For a lattice L to be a Boolean algebra as specified in Definition 1, it must have two properties. First, it must be **complemented**. For a lattice to be complemented it must have a least element 0 and a greatest element 1 and for every element x of the lattice there must exist an element $\overline{x}$ such that $x \vee \overline{x} = 1$ and $x \wedge \overline{x} = 0$. Second, it must be **distributive**. This means that for every x, y, and z in L, $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ and $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$. Showing that a complemented, distributive lattice is a Boolean algebra has been left as Supplementary Exercise 39 in Chapter 9.

Exercises

1. Find the values of these expressions.
   a) $1 \cdot \overline{0}$ **b)** $1 + \overline{1}$ **c)** $\overline{0} \cdot 0$ **d)** $\overline{(1+0)}$
2. Find the values, if any, of the Boolean variable x that satisfy these equations.
   a) $x \cdot 1 = 0$ **b)** $x + x = 0$
   c) $x \cdot 1 = x$ **d)** $x \cdot \overline{x} = 1$
3. **a)** Show that $(1 \cdot 1) + (\overline{0 \cdot 1 + 0}) = 1$.
   b) Translate the equation in part (a) into a propositional equivalence by changing each 0 into an **F**, each 1 into a **T**, each Boolean sum into a disjunction, each Boolean product into a conjunction, each complementation into a negation, and the equals sign into a propositional equivalence sign.
4. **a)** Show that $(\overline{1} \cdot \overline{0}) + (1 \cdot \overline{0}) = 1$.
   b) Translate the equation in part (a) into a propositional equivalence by changing each 0 into an **F**, each 1 into a **T**, each Boolean sum into a disjunction, each Boolean product into a conjunction, each complementation into a negation, and the equals sign into a propositional equivalence sign.
5. Use a table to express the values of each of these Boolean functions.
   a) $F(x, y, z) = \overline{x}y$
   b) $F(x, y, z) = x + yz$
   c) $F(x, y, z) = x\overline{y} + \overline{(xyz)}$
   d) $F(x, y, z) = x(yz + \overline{y}\,\overline{z})$
6. Use a table to express the values of each of these Boolean functions.
   a) $F(x, y, z) = \overline{z}$
   b) $F(x, y, z) = \overline{x}y + \overline{y}z$
   c) $F(x, y, z) = x\overline{y}z + \overline{(xyz)}$
   d) $F(x, y, z) = \overline{y}(xz + \overline{x}\,\overline{z})$
7. Use a 3-cube Q_3 to represent each of the Boolean functions in Exercise 5 by displaying a black circle at each vertex that corresponds to a 3-tuple where this function has the value 1.
8. Use a 3-cube Q_3 to represent each of the Boolean functions in Exercise 6 by displaying a black circle at each vertex that corresponds to a 3-tuple where this function has the value 1.
9. What values of the Boolean variables x and y satisfy $xy = x + y$?
10. How many different Boolean functions are there of degree 7?
11. Prove the absorption law $x + xy = x$ using the other laws in Table 5.
12. Show that $F(x, y, z) = xy + xz + yz$ has the value 1 if and only if at least two of the variables x, y, and z have the value 1.
13. Show that $x\overline{y} + y\overline{z} + \overline{x}z = \overline{x}y + \overline{y}z + x\overline{z}$.

Exercises 14–23 deal with the Boolean algebra {0, 1} with addition, multiplication, and complement defined at the beginning of this section. In each case, use a table as in Example 8.

14. Verify the law of the double complement.
15. Verify the idempotent laws.
16. Verify the identity laws.
17. Verify the domination laws.
18. Verify the commutative laws.
19. Verify the associative laws.
20. Verify the first distributive law in Table 5.
21. Verify De Morgan's laws.
22. Verify the unit property.
23. Verify the zero property.

The Boolean operator $\oplus$, called the *XOR* operator, is defined by $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$, and $0 \oplus 0 = 0$.

24. Simplify these expressions.
    a) $x \oplus 0$ **b)** $x \oplus 1$
    c) $x \oplus x$ **d)** $x \oplus \overline{x}$
25. Show that these identities hold.
    a) $x \oplus y = (x + y)\overline{(xy)}$
    b) $x \oplus y = (x\overline{y}) + (\overline{x}y)$
26. Show that $x \oplus y = y \oplus x$.
27. Prove or disprove these equalities.
    a) $x \oplus (y \oplus z) = (x \oplus y) \oplus z$
    b) $x + (y \oplus z) = (x + y) \oplus (x + z)$
    c) $x \oplus (y + z) = (x \oplus y) + (x \oplus z)$
28. Find the duals of these Boolean expressions.
    a) $x + y$ **b)** $\overline{x}\,\overline{y}$
    c) $xyz + \overline{x}\,\overline{y}\,\overline{z}$ **d)** $x\overline{z} + x \cdot 0 + \overline{x} \cdot 1$

*29. Suppose that F is a Boolean function represented by a Boolean expression in the variables $x_1, \ldots, x_n$. Show that $F^d(x_1, \ldots, x_n) = \overline{F(\overline{x}_1, \ldots, \overline{x}_n)}$.

*30. Show that if F and G are Boolean functions represented by Boolean expressions in n variables and $F = G$, then $F^d = G^d$, where F^d and G^d are the Boolean functions represented by the duals of the Boolean expressions representing F and G, respectively. [*Hint:* Use the result of Exercise 29.]

*31. How many different Boolean functions $F(x, y, z)$ are there such that $F(\overline{x}, \overline{y}, \overline{z}) = F(x, y, z)$ for all values of the Boolean variables x, y, and z?

*32. How many different Boolean functions $F(x, y, z)$ are there such that $F(\overline{x}, y, z) = F(x, \overline{y}, z) = F(x, y, \overline{z})$ for all values of the Boolean variables x, y, and z?

33. Show that you obtain De Morgan's laws for propositions (in Table 6 in Section 1.3) when you transform De Morgan's laws for Boolean algebra in Table 6 into logical equivalences.
34. Show that you obtain the absorption laws for propositions (in Table 6 in Section 1.3) when you transform the absorption laws for Boolean algebra in Table 6 into logical equivalences.