Ceng 124
Discrete Structures

2018-2019 Spring Semester

by Sibel T. Özyer, Spring 2019

Topics

- 12 Boolean Algebra
- 12.1 Boolean Functions
- 12.2 Representing Boolean Functions

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 $\overline{1} = 0$.

Introduction (cont.)

- 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 · or by *AND*, has the following values:

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

Example1

- Find the value of $1 \cdot 0 + \overline{(0 + 1)}$.
- Solution: $$\begin{aligned} 1 \cdot 0 + \overline{(0 + 1)} &= 0 + \overline{1} \\ &= 0 + 0 \\ &= 0. \end{aligned}$$

Logical Equivalence

- The complement, Boolean sum, and Boolean product correspond to the logical operators, ¬,∨, and ∧, 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.

Example2

- 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:

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

Example3

- 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:

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

Boolean Expressions and Boolean Functions

- 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.

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

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

Example4

- Translate the distributive law $x + yz = (x + y)(x + z)$

  into a logical equivalence.
- Solution: 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.

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

12.2 Representing Boolean Functions

- Two important problems of Boolean algebra will be studied in this section.
- The first problem is: Given the values of a Boolean function, how can a Boolean expression that represents this function be found?
- The second problem is: Is there a smaller set of operators that can be used to represent all Boolean functions?

Sum-of-Product Expansions

- Question: Find Boolean expressions that represent the functions *F(x, y, z)* and *G(x, y, z)*, which are given in Table 1.

TABLE 1

x	*y*	*z*	*F*	*G*
1	1	1	0	0
1	1	0	0	1
1	0	1	1	0
1	0	0	0	0
0	1	1	0	0
0	1	0	0	1
0	0	1	0	0
0	0	0	0	0

- Solution: F is , $x\overline{y}z$, has the value 1 if and only if $x = y = z = 1$, which holds if and only if $x = z = 1$ and $y = 0$.
- The Boolean sum of these two products, $xy\overline{z} + \overline{x}y\overline{z}$, represents *G*, because it has the value 1 if and only if $x = y = 1$ and $z = 0$, or $x = z = 0$ and $y = 1$.

Example

- Find the sum-of-products expansion for the function *F (x, y, z)* = $(x + y)\overline{z}$.
- Solution: We will find the sum-of-products expansion of *F(x, y, z)* in two ways. First, we will use Boolean identities to expand the product and simplify. We find that

$$
\begin{aligned}
F(x, y, z) &= (x + y)\overline{z} \\
&= x\overline{z} + y\overline{z} && \text{Distributive law} \\
&= x1\overline{z} + 1y\overline{z} && \text{Identity law} \\
&= x(y + \overline{y})\overline{z} + (x + \overline{x})y\overline{z} && \text{Unit property} \\
&= xy\overline{z} + x\overline{y}\,\overline{z} + xy\overline{z} + \overline{x}y\overline{z} && \text{Distributive law} \\
&= xy\overline{z} + x\overline{y}\,\overline{z} + \overline{x}y\,\overline{z}. && \text{Idempotent law}
\end{aligned}
$$

Solution (cont.)

- Second, we can construct the sum-of-products expansion by determining the values of *F* for all possible values of the variables *x*, *y*, and *z*. These values are found in Table 2.

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

TABLE 2

x	y	z	$x + y$	$\overline{z}$	$(x + y)\overline{z}$
1	1	1	1	0	0
1	1	0	1	1	1
1	0	1	1	0	0
1	0	0	1	1	1
0	1	1	1	0	0
0	1	0	1	1	1
0	0	1	0	0	0
0	0	0	0	1	0

Solution (cont.)

- It is also possible to find a Boolean expression that represents a Boolean function by taking a Boolean product of Boolean sums.
- The resulting expansion is called the **conjunctive normal form** or **product-of-sums expansion** of the function.