BOOLEAN ALGEBRA

Learning Outcomes

At the end of this lesson students will be able to:

I. use the laws of Boolean algebra as they apply to sets;
II. manipulate Boolean-valued expressions
III. simplify Boolean expressions;
IV. investigate de Morgan's laws
V. apply mathematical knowledge and skills in a problem solving context.

Introduction

The most obvious way to simplify Boolean expressions is to manipulate them in the same way as normal algebraic expressions are manipulated. With regards to logic relations in digital forms, a set of rules for symbolic manipulation is needed in order to solve for the unknowns.

A set of rules formulated by the English mathematician *George Boole* describe certain propositions whose outcome would be either *true or false*.

Basic Definitions

Let B be a non empty set with two binary operations + and *, a unary operation ‘ ,and two distinct elements 0 and 1.Then B is called *Boolean Algebra* if it holds the basic *axioms* [B_1] to [B_4].

Sometimes we will designate a Boolean algebra by ‹B,+,*,’,0,1› when we want to emphasize its six parts.

Axioms

Where a, b, c are any elements in B:

$[B_1]$ Commutative laws:

(1a) $a + b = b + a$ (1b) $a * b = b * a$

$[B_2]$ Distributive laws:

(2a) $a + (b * c) = (a + b) * (a + c)$ (2b) $a * (b + c) = (a * b) + (a * c)$

$[B_3]$ Identity laws:

(3a) $a + 0 = a$ (3b) $a * 1 = a$

$[B_4]$ Complement laws:

(4a) $a + a' = 1$ (4b) $a * a' = 0$

Boolean Operations

- The *complement* is denoted by a '. It is defined by
- 0' = 1   and   1' = 0.
- The *Boolean sum*, denoted by + or by OR,
- has the following values:
- 1 + 1 = 1,   1 + 0 = 1,   0 + 1 = 1,   0 + 0 = 0
- The *Boolean product*, denoted by . or by * or by AND,
- has the following values:
- 1 * 1 = 1,   1* 0 = 0,   0 * 1 = 0,   0 * 0 = 0

Precedence

We adopt the usual convention that, unless we are guided by parenthesis, ' has precedence over *, and * has precedence over +.

Example:

$a + b * c$ means $a + (b * c)$ and not $(a + b) * c$

$a * b'$ means $a * (b')$ and not $(a * b)'$

Duality

The *dual* of any statement in a Boolean algebra B is the statement obtained by changing every AND(*) to OR(+), every OR(+) to AND(*) and all 1's to 0's and vice-versa in the original statement.

Example:

The dual of

$(1 + a) * (b + 0) = b$ is $(0 * a) + (b * 1) = b$

Principle of Duality

Theorem 1:

The dual of any theorem in a Boolean algebra is also a theorem.

i.e. if any statement is a consequence of the axioms of a Boolean algebra, then the dual is also a consequence of those axioms since the dual statement can be proven by using the dual of each step of the proof of the original statement.

Basic Theorems

Using the axioms [B_1] through [B_4], we prove the following theorems

Theorem 2: Let a, b, c be any elements in a Boolean Algebra B.

i. Idempotent laws:

(5a) $a + a = a$ (5b) $a * a = a$

ii. Boundedness laws:

(6a) $a + 1 = 1$ (6b) $a * 0 = 0$

iii. Absorption laws:

(7a) $a + (a * b) = a$ (7b) $a * (a + b) = a$

iv. Associative laws:

(8a) $(a + b) + c = a + (b + c)$ (8b) $(a * b) * c = a * (b * c)$

Basic Theorems

Theorem 3: Let a be any element of a Boolean algebra B.

i. (Uniqueness of Complement) If $a + x = 1$ and $a * x = 0$, then $x = a'$.

ii. (Involution law) $(a')' = a$.

iii. (9a) $0' = 1$ (9b) $1' = 0$.

Basic Theorems

Theorem 4 :

DeMorgan's laws :

(10a) $(a + b)' = a' * b'$

(10b) $(a * b)' = a' + b'$

Sum of Products Form

Consider a set of variables, say $x_1, x_2, \ldots\ldots, x_n$. A *Boolean expression* E in these variables, sometimes written $E(x_1, \ldots\ldots, x_n)$, is any variables or any expression built up from the variables using the Boolean operations +, *, and ‘.

Example:

$$E_1 = (x + y'z)' + (xyz' + x'y)' \qquad and \qquad E_2 = ((xy'z' + y)' + x'z)'$$

are Boolean expression in x, y and z.

Sum of Products Form

A *literal* is a variable or complemented variable, such as x, x', y, y', and so on. A *fundamental product* is a literal or a product of two or more literals in which no two literals involve the same variable.

Thus

$$xz', \quad xy'z, \quad x, \quad y', \quad x'yz$$

are fundamental products but $xyx'z$ and $xyzy$ are not. Any product of literals can be reduced to either 0 or a fundamental product, eg., $xyx'z=0$ since $xx'=0$ (complement law), and $xyzy=xyz$ since $yy=y$ (idempotent law).

Sum of Products Form

A fundamental product P_1 is said to be *contained in* (or *included in*) another fundamental product P_2 if the literals of P_1 are also literals of P_2. For example, $x'z$ is contained in $x'yz$, but $x'z$ is not contained in $xy'z$ since x' is not a literal of $xy'z$. Observe that if P_1 is contained in P_2, say $P_2=P_1*Q$, then, by the absorption law,

$$P_1+P_2= P_1+P_1*Q=P_1$$

Thus for instance, $x'z + x'yz = x'z$.

Sum of Products Form

Definition:

A Boolean expression E is called a *sum-of-products* expression if E is a fundamental product or the sum of two or more fundamental products none of which is contained in another.

Definition:

Let E be any Boolean expression. A *sum-of-products form* of E is an equivalent Boolean sum-of-products expression.

Sum-of-Products Form

Algorithm 1:

Input : Boolean Expression E.

Output : sum-of-products expression equivalent to E.

Step 1 : Use DeMorgan's laws and involution to move the complement operation into any parenthesis until finally the complement operation only applies to variables. Then E will consist only of sums and products of literals.

Sum-of-Products Form

Step 2 : Use the distributive operation to next transform E into a sum-of-products.

Step 3 : Use the commutative, idempotent, and complement laws to transform each product in E into 0 or a fundamental product.

Step 4 : Use the absorption and identity laws to finally transform E into a sum-of-products expression.

Example 1

Express the following Boolean expression $E(x,y,z)$ as a sum-of-products form:

$$E = ((xy)'z)'((x'+z)(y'+z'))'$$

Using DeMorgan's laws and involution

Step 1 : $E = (xy'' + z')((x'+z)' + (y'+z')')$

$$= (xy + z')(xz' + yz)$$

Using the distributive laws

Step 2 : $E = xyxz' + xyyz + xz'z' + yzz'$

Example 1(cont.)

Using the commutative, idempotent and complement laws

Step 3 : $E = xyz' + xyz + xz' + 0$

Using absorption law

Step 4 : $xz' + (xz' * y) = xz'$

Using identity law

$$E = xyz + xz'$$

Complete

Sum-of-Products Forms

A Boolean expression $E=E(x_1,x_2,\ldots,x_n)$ is said to be a *complete sum-of products* expression if E is a sum-of-products expression where each product P involves all the n variables. Such a fundamental product P which involves all the variables is called *minterm*, and there is a maximum of 2^n such products for n variables.

Theorem 5: Every non zero Boolean expression $E=E(x_1,x_2,\ldots,x_n)$ is equivalent to a complete sum-of-products expression and such a representation is unique.

Complete

Sum-of-Products Forms

Algorithm 2:

Input : Boolean sum-of-products Expression

$E=E(x_1,x_2,\ldots,x_n)$.

Output : Complete sum-of-products expression equivalent to E.

Step 1 : Find a product P in E which does not involve the variables x_i, and then multiply P by x_i+x_i', deleting any repeated products.(This is possibe since $x_i+x_i'=1$ and $P+P=P$.)

Step 2 : Repeat step1 until every product P in E is a minimum, i.e., every product P involves all the variables.

Example 2

Express $E(x,y,z)=x(y'z)'$ in its complete sum-of-products form

a) Apply algorithm1 to E to obtain

$$E = x(y'z)' = x(y+z') = xy + xz'$$

Now E is represented by a sum-of-products expression.

b) Apply algorithm2 to obtain

$$E = xy(z+z') + xz'(y+y') = xyz + xyz' + xyz' + xy'z'$$

$$= xyz + xyz' + xy'z'$$

Now E is represented by its complete sum-of-products form.

Minimal Boolean Expressions, Prime Implicants

There are many ways of representing the same Boolean expression E. Here we define and investigate a minimal sum-of-products form for E. We must also define and investigate prime implicants of E since the minimal sum-of-products involves such prime implicants.

Minimal

Sum-of-Products

E is a Boolean sum-of-products expression. E_L denote the number of literals in E and E_S denote the number of summands in E.

Exaple:

$$E = xyz' + x'y't + xy'z't + x'yzt$$

Then E_L=3+3+4+4=14 and E_S=4

Minimal

Sum-of-Products

Suppose E and F are equivalent Boolean sum-of-products expressions. We say E is *simpler* than F if:

(i) $E_L < F_L$ and $E_S \leq F_L$ or (ii) $E_L \leq F_L$ and $E_S < F_L$

We say E is *minimal* if there is no equivalent sum-of-products expression which is simpler than E.

Prime Implicants

A fundamental product P is called a *prime implicant* of a Boolean expression E if $P + E = E$

But no other fundamental product contained in P has this property.

Example: $E = xy' + xyz' + x'yz'$

One can show that:

$xz' + E = E$ but $x + E \neq E$ and $z' + E \neq E$

Thus xz' is a prime implicant of E.

Prime Implicants

Theorem 6: A minimal sum-of-products form for a Boolean expression E is a sum of prime implicants of E.

Consensus of Fundamental Products

Let P_1 and P_2 be fundamental products such that exactly one variable, say x_k, appears uncomplemented in one of P_1 and P_2 and complemented in the other. Then the *consensus* of P_1 and P_2 is the product (without repetition) of the literals of $P1$ and the literals *of* P_2 after x_k and x_k' are deleted. (We do not define the consensus of $P_1=x$ and $P_2=x'$.)

Lemma: Suppose Q is the consensus of P_1 and P_2.

Then $P_1+P_2+Q=P_1+P_2$

Example 3

Find the consensus Q of P_1 and P_2 Where:

a) $P_1 = xyz'$ and $P_2 = xy't$.

Delete y and y' and then multiply the literals of P_1 and P_2 (without repetition) to obtain $Q = xz'st$.

b) $P_1 = xy'$ and $P_2 = y$.

Deleting y and y' yields $Q = x$.

Example 3 (cont.)

c) $P1=x'yz$ and $P2=x'yt$.

No variables appears uncomplemented in one of the products and complemented in the other. Hence $P1$ and $P2$ have no variables.

d) $P1=x'yz$ and $P2=xyz'$.

Each of x and z appear complemented in one of the products and uncomplemented in the other. Hence $P1$ and $P2$ have no consnsus.

Consensus Method for Finding Prime Implicants

Algorithm 3 (Consensus Method) :

Input : Boolean sum-of-products Expression $E=P_1+P_2+.....+P_m$ where the Ps are fundamental products.

Output : E as a sum of its prime implicants.

Step 1 : Delete any fundamental product P_i which includes any other fundamental product P_j.(permissible by the absoption law.)

Consensus Method for Finding Prime Implicants

Step 2 : Add the consensus of any P_i and P_j providing Q does not include any of the Ps.(permissible by Lemma.)

Step 4 : Repeat step1 and/or step 2 until neither can be applied.

Theorem 7: The consensus method will eventually stop, and then E will be the sum of its prime implicants.

Example 4

Let $E = xyz + x'z' + xyz' + x'y'z + x'yz'$.

Then:

$E = xyz + x'z' + xyz' + x'y'z$ ($x'yz'$ includes $x'z'$)

$= xyz + x'y' + xyz' + x'y'z + xy$ (consensus of xyz and xyz')

$= x'z' + x'y'z + xy$ (xyz and xyz' includes xy)

$= x'z' + x'y'z + xy + x'y'$ (consensus of $x'z'$ and $x'y'z$)

$= x'z' + xy + x'y'$ ($x'y'z$ includes $x'y'$)

$= x'z' + xy + x'y' + yz'$ (consensus of $x'z'$ and xy)

Minimal Sum-of-Products Form

Algorithm 4 :

Input : Boolean Expression $E=P_1+P_2+.....+P_m$ where the Ps are the prime implicants of E.

Output : E as a minimal sum-of-products.

Step 1 : Express each prime implicant P as a complete sum-of-products.

Step 2 : Delete one by one those prime implicants whose summands appear among the summands of the remaining prime implicants.

Example 5

Let $E = x'z' + xy + x'y' + yz'$

Where E is expressed as the sum of all its prime implicants.

Step 1:

$$x'z' = x'z'(y + y') = x'yz' + x'y'z'$$

$$xy = xy(z + z') = xyz + xyz'$$

$$x'y' = x'y'(z + z') = x'y'z + x'y'z'$$

$$yz' = yz'(x + x') = xyz' + x'yz'$$

Step 2: the summands of $x'z'$ are $x'yz$ and $x'y'z'$ which appear among the other summands. Thus delete $x'z'$ to obtain

$$E = xy + x'y' + yz'$$

The End