Boolean Algebra

A *Boolean algebra* is a set *B* of values together with:

- two binary operations, commonly denoted by + and · ,
- a unary operation, usually denoted by ¯ or ~ or ′ ,
- two elements usually called *zero* and *one*, such that for every element *x* of *B*:

$$x + \overline{x} = 1 \text{ and } x \cdot \overline{x} = 0$$

In addition, certain axioms must be satisfied:

- *closure properties* for both binary operations and the unary operation
- *associativity* of each binary operation over the other,
- *commutativity* of each each binary operation,
- *distributivity* of each binary operation over the other,
- *absorption* rules,
- *existence of complements* with respect to each binary operation

We will assume that · has higher precedence than +; however, this is not a general rule for all Boolean algebras.

Axioms of Boolean Algebra

Associative Laws: for all *a*, *b* and *c* in *B*,

$$(a+b)+c=a+(b+c) \qquad (a\cdot b)\cdot c=a\cdot(b\cdot c)$$

Commutative Laws: for all *a* and *b* in *B*,

$$a+b=b+a \qquad a\cdot b=b\cdot a$$

Distributive Laws: for all *a*, *b* and *c* in *B*,

$$a+(b\cdot c)=(a+b)\cdot(a+c) \qquad a\cdot(b+c)=(a\cdot b)+(a\cdot c)$$

Absorption Laws: for all *a*, *b* and *c* in *B*,

$$a+(a\cdot b)=a \qquad a\cdot(a+b)=a$$

Existence of Complements: for each *a* in *B*, there exists an element *ā* in *B* such that

$$a+\overline{a}=1 \qquad a\cdot\overline{a}=0$$

Examples of Boolean Algebras

The classic example is B = {true, false} with the operations AND, OR and NOT.

An isomorphic example is B = {1, 0} with the operations +, · and ~ defined by:

a	*b*	*a + b*	*a * b*	*~a*
0	0	0	0	1
1	0	1	0	0
0	1	1	0	
1	1	1	1	

Given a set *S*, the power set of *S*, *P(S)* is a Boolean algebra under the operations union, intersection and relative complement.

Other, interesting examples exist…

More Properties

It's also possible to derive some additional facts, including:

- the elements 0 and 1 are unique
- the complement of an element *a* is unique
- 0 and 1 are complements of each other

DeMorgan's Laws & More

DeMorgan's Laws are useful theorems that can be derived from the fundamental properties of a Boolean algebra.

For all *a* and *b* in *B*, $\overline{a+b} = \overline{a} \cdot \overline{b}$ $\overline{a \cdot b} = \overline{a} + \overline{b}$

Of course, there's also a *double-negation law*: $\overline{\overline{a}} = a$

And there are *idempotency laws*: $a + a = a$ $a \cdot a = a$

Boundedness properties: $a + 0 = a$ $a \cdot 0 = 0$

$a + 1 = 1$ $a \cdot 1 = a$

Logic Expressions and Equations

A *logic expression* is defined in terms of the three basic Boolean operators and variables which may take on the values 0 and 1. For example:

$$z: \quad x_0 \cdot \overline{y_0} + \overline{x_0} \cdot y_0$$

$$y: \quad (x_1 \cdot x_2) + (x_1 \cdot \overline{x_2} \cdot x_3) + \overline{x_1} \cdot \overline{(\overline{x_2} + x_3)}$$

A *logic equation* is an assertion that two logic equations are *equal*, where equal means that the values of the two expressions are the same for all possible assignments of values to their variables. For example:

$$x_0 \cdot \overline{y_0} + \overline{x_0} \cdot y_0 = \overline{(\overline{x_0} + y_0) \cdot (x_0 + \overline{y_0})}$$

Of course, equations may be true or false. What about the one above?

Why do they call it "algebra"?

A Boolean expression can often be usefully transformed by using the theorems and properties stated earlier:

$$\overline{\left(\overline{x_0} + y_0\right)\cdot\left(x_0 + \overline{y_0}\right)} = \overline{\left(\overline{x_0} + y_0\right)} + \overline{\left(x_0 + \overline{y_0}\right)}$$

$$= \overline{\overline{x_0}} \cdot \overline{y_0} + \overline{x_0} \cdot \overline{\overline{y_0}}$$

$$= x_0 \cdot \overline{y_0} + \overline{x_0} \cdot y_0$$

That is a relatively simple example of a reduction.

Try showing the following expressions are equal:

$$\overline{x_0 + \left(y_0 \cdot z_0\right)} = \overline{x_0} \cdot \overline{y_0} + \overline{x_0} \cdot \overline{z_0}$$

Why do they call it "algebra"?

Here's another that happens to be related to binary addition:

$Z = \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} + A \cdot B \cdot C$	Given
$= \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} + A \cdot B \cdot C + A \cdot B \cdot C + A \cdot B \cdot C$	Idempotence
$= \left(\overline{A} \cdot B \cdot C + A \cdot B \cdot C\right) + \left(A \cdot \overline{B} \cdot C + A \cdot B \cdot C\right) + \left(A \cdot B \cdot \overline{C} + A \cdot B \cdot C\right)$	Commutativity, Associativity
$= \left(\overline{A} + A\right) \cdot B \cdot C + A \cdot C \cdot \left(\overline{B} + B\right) + A \cdot B \cdot \left(\overline{C} + C\right)$	Commutativity, Distributivity
$= 1 \cdot B \cdot C + A \cdot C \cdot 1 + A \cdot B \cdot 1$	Complements
$= A \cdot C + B \cdot C + A \cdot B$	Boundedness

Tautologies, Contradictions & Satisfiables

A *tautology* is a Boolean expression that evaluates to true (1) for all possible values of its variables.

$$a + \overline{a}$$

$$a \cdot b + a \cdot \overline{b} + \overline{a} \cdot b + \overline{a} \cdot \overline{b}$$

A *contradiction* is a Boolean expression that evaluates to false (0) for all possible values of its variables.

$$a \cdot \overline{a}$$

A Boolean expression is *satisfiable* if there is at least one assignment of values to its variables for which the expression evaluates to true (1).

$$a \cdot b + \overline{a} \cdot \overline{b}$$

Truth Tables

A Boolean expression may be analyzed by creating a table that shows the value of the expression for all possible assignments of values to its variables:

a	b	$a \cdot b$	$\overline{a} \cdot \overline{b}$	$a \cdot b + \overline{a} \cdot \overline{b}$
0	0	0	1	1
1	0	0	0	0
0	1	0	0	0
1	1	1	0	1

Proving Equations with Truth Tables

Boolean equations may be proved using truth tables (dull and mechanical):

$$a + 1 = 1$$

a	*a+1*
0	1
1	1

$$\overline{a \cdot b \cdot c} = \overline{a} + \overline{b} + \overline{c}$$

a	b	c	~(a*b*c)	~a*~b*~c
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Proving Equations Algebraically

Boolean equations may be proved using truth tables, which is dull and boring, or using the algebraic properties:

$$\forall a \in B, a \cdot 1 = a$$

$$a = a \cdot \left(a + \overline{a}\right) \qquad \text{absorption, with } b = \overline{a}$$

$$= a \cdot 1 \qquad \text{law of complements}$$

Note the duality

$$\forall a \in B, a + 0 = a$$

$$a = a + \left(a \cdot \overline{a}\right) \qquad \text{absorption, with } b = \overline{a}$$

$$= a + 0 \qquad \text{law of complements}$$

Proving Equations Algebraically

$\forall a \in B, a + a = a$

$a = a + a \cdot 1$ absorption, with $b = 1$

$= a + a \cdot (a + \overline{a})$ law of complements

$= a + a$ absorption, with $b = \overline{a}$

Sum-of-Products Form

A Boolean expression is said to be in *sum-of-products form* if it is expressed as a sum of terms, each of which is a product of variables and/or their complements:

$$a \cdot b + \overline{a} \cdot \overline{b}$$

It's relatively easy to see that every Boolean expression can be written in this form.

Why?

The summands in the sum-of-products form are called *minterms*.

- each minterm contains each of the variables, or its complement, exactly once
- each minterm is unique, and therefore so is the representation (aside from order)

Example: Truth Table to Function

Given a truth table for a Boolean function, construction of the sum-of-products representation is trivial:

- for each row in which the function value is 1, form a product term involving all the variables, taking the variable if its value is 1 and the complement if the variable's value is 0
- take the sum of all such product terms

x	y	z	F	
0	0	0	0	
0	0	1	1	$\overline{x} \cdot \overline{y} \cdot z$
0	1	0	1	$\overline{x} \cdot y \cdot \overline{z}$
0	1	1	0	
1	0	0	1	$x \cdot \overline{y} \cdot \overline{z}$
1	0	1	0	
1	1	0	0	
1	1	1	1	$x \cdot y \cdot z$

$$F = \overline{x} \cdot \overline{y} \cdot z + \overline{x} \cdot y \cdot \overline{z} + x \cdot \overline{y} \cdot \overline{z} + x \cdot y \cdot z$$

Product-of-Sums Form

A Boolean expression is said to be in *product-of-sums form* if it is expressed as a product of terms, each of which is a sum of variables:

$$(a+b)\cdot\left(\overline{a}+\overline{b}\right)$$

Every Boolean expression can also be written in this form, as a product of *maxterms*.

Facts similar to the sum-of-products form can also be asserted here.

The product-of-sums form can be derived by expressing the complement of the expression in sum-of-products form, and then complementing.

Example

Given a truth table for a Boolean function, construction of the product-of-sums representation is trivial:

- for each row in which the function value is 0, form a product term involving all the variables, taking the variable if its value is 1 and the complement if the variable's value is 0
- take the sum of all such product terms; then complement the result

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Row 000 → $\bar{x}\cdot\bar{y}\cdot\bar{z}$

Row 011 → $\bar{x}\cdot y\cdot z$

Row 101 → $x\cdot\bar{y}\cdot z$

Row 110 → $x\cdot y\cdot\bar{z}$

$$\overline{F} = \bar{x}\cdot\bar{y}\cdot\bar{z} + \bar{x}\cdot y\cdot z + x\cdot\bar{y}\cdot z + x\cdot y\cdot\bar{z}$$

$$F = \overline{\bar{x}\cdot\bar{y}\cdot\bar{z}}\cdot\overline{\bar{x}\cdot y\cdot z}\cdot\overline{x\cdot\bar{y}\cdot z}\cdot\overline{x\cdot y\cdot\bar{z}}$$

$$= (x+y+z)\cdot(x+\bar{y}+\bar{z})\cdot$$

$$(\bar{x}+y+\bar{z})\cdot(\bar{x}+\bar{y}+z)$$

Boolean Functions

A *Boolean function* takes n inputs from the elements of a Boolean algebra and produces a single value also an element of that Boolean algebra.

For example, here are all possible 2-input Boolean functions on the set {0, 1}:

A	B	zero	and		A		B	xor	or
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

A	B	nor	eq	B'		A'		nand	one
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Universality

Any Boolean function can be expressed using:

- only AND, OR and NOT
- only AND and NOT
- only OR and NOT
- only AND and XOR
- only NAND
- only NOR

The first assertion should be entirely obvious.

The remaining ones are obvious if you consider how to represent each of the functions in the first set using only the relevant functions in the relevant set.