Boolean functions

Irina Prosvirnina

- Boolean functions
- Boolean algebra
- The disjunctive normal form
- The conjunctive normal form
- Functional completeness

Boolean functions

- 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.

Boolean functions

- The **complement** of an element, denoted with a bar, is defined by

$$\bar{0} = 1, \qquad \bar{1} = 0$$

Boolean functions

- The **Boolean sum**, denoted by $+$ or by OR, has the following values:

$$1 + 1 = 1, 1 + 0 = 1, 0 + 1 = 1, 0 + 0 = 0$$

Boolean functions

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

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

Boolean functions

Example 1

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

Solution:

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

Boolean functions

- 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.

Boolean functions

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

$$(T \wedge F) \vee \neg(T \vee F) \equiv F.$$

Boolean functions

Example 3

Translate the logical equivalence $(T \wedge T) \vee \neg F \equiv T$ 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

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

Boolean expressions and Boolean functions

Let $B = \{0, 1\}$.

Then

$$B^n = \{(x_1, x_2, \ldots, x_n) | x_i \in B \text{ for } 1 \leq i \leq n\}$$

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**.

Boolean expressions and Boolean functions

Example 4

The function $F(x, y) = x\bar{y}$ from the set of ordered pairs of Boolean variables B^2 to the set $\{0,1\}$ is a Boolean function of degree 2 with

$$F(1, 1) = 0,$$
$$F(1, 0) = 1,$$
$$F(0, 1) = 0,$$
$$F(0, 0) = 0.$$

Boolean expressions and Boolean functions

Example 4

$F(x, y) = x\bar{y}$ is a Boolean function with
$F(1,1) = 0, F(1,0) = 1, F(0,1) = 0, F(0,0) = 0.$
We display these values of F in Table 1:

Boolean expressions and Boolean functionsии

- 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 $\overline{E_1}$, $(E_1 E_2)$ and $(E_1 + E_2)$ are Boolean expressions.

Boolean expressions and Boolean functions

- 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.

Boolean expressions and Boolean functions

Example 5

Find the values of the Boolean function represented by

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

Solution:

The values of this function are displayed in Table 2.

Example 5

Table 2					
1	1	1	1		
1	1	0	1		
1	0	1	0		
1	0	0	0		
0	1	1	0		
0	1	0	0		
0	0	1	0		
0	0	0	0		

Example 5

Table 2					
1	1	1	1	0	
1	1	0	1	1	
1	0	1	0	0	
1	0	0	0	1	
0	1	1	0	0	
0	1	0	0	1	
0	0	1	0	0	
0	0	0	0	1	

Example 5

Table 2					
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 expressions and Boolean functions

- 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**.

Boolean expressions and Boolean functions

Example 6

The Boolean expressions $xy + 0$ and $xy \cdot 1$ are equivalent.

Boolean expressions and Boolean functions

- The **complement** of the Boolean function F is the function $\bar{F}$:

$$\bar{F}(x_1, x_2, \ldots, x_n) = \overline{F(x_1, x_2, \ldots, x_n)}$$

Boolean expressions and Boolean functions

Let F and G be Boolean functions of degree n.

The **Boolean sum** $F + G$ and the **Boolean product** FG are defined by:

$(F + G)(x_1, \dots, x_n) = F(x_1, \dots, x_n) + G(x_1, \dots, x_n)$;

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

Boolean expressions and Boolean functions

A Boolean function of degree two is a function from a set with four elements, namely, pairs of elements from B^2, 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 $F_1 - F_{16}$.

The Boolean functions of degree two

Table 3. The 16 Boolean functions of degree two.

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

Boolean expressions and Boolean functions

- How many different Boolean functions of degree n are there?

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.

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 7.
- The proofs of the remaining properties are left as exercises.

Identities of Boolean algebra

Example 7

Show that the distributive law

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

is valid.

Solution:

The verification of this identity is shown in Table 4.

The identity holds because the last two columns of the table agree.

Example 7 $x(y+z) = xy + xz$

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

Identity	***Name***
	Identity laws
	Domination laws
	Idempotent laws
	Law of the double complement
	Commutative laws

Identity	***Name***
	Associative laws
	Distributive laws
	De Morgan’s laws
	Absorption laws
	Unit property Zero property

Identities of Boolean algebra

Compare these Boolean identities with the logical equivalences and the set identities!

Identities of Boolean algebra

- 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.

Identities of Boolean algebra

- 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.
- This result, called the **duality principle**, is useful for obtaining new identities.

Identities of Boolean algebra

Example 8

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.

Disjunctive normal form

We now show how any Boolean expression can be expressed in an equivalent standard form (called the disjunctive normal form).

Disjunctive normal form

A **minterm** is a Boolean function which has precisely one 1 in the final column of its truth table.

Example of a minterm

1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

Sinse $m(x, y, z) = 1$

if and only if

$x = 0, y = 1, z = 1,$

it follows that

$m(x, y, z) = \bar{x}yz.$

The expression $\bar{x}yz$ is called the **product representation** of the minterm m.

Disjunctive normal form

- Any Boolean function which does not equal 0 identically can be expressed uniquely as a disjunction of minterms.

This is called the disjunctive normal form (DNF) of the Boolean expression.

A procedure for constructing a Boolean expression representing a function with given values as DNF

By taking Boolean sums of distinct minterms we can build up a Boolean expression with a specified set of values.

In particular, a Boolean sum of minterms has the value 1 when exactly one of the minterms in the sum has the value 1.

It has the value 0 for all other combinations of values of the variables.

A procedure for constructing a Boolean expression representing a function with given values as DNF

Consequently, given a Boolean function, a Boolean sum of minterms can be formed that has the value 1 when this Boolean function has the value 1, and has the value 0 when the function has the value 0.

The minterms in this Boolean sum correspond to those combinations of values for which the function has the value 1.

Disjunctive normal form

Example 9

Find the disjunctive normal form of the Boolean expression $(x + y)\overline{z}$.

Solution of example 9

Table 7					
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

-

Table 7					
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

DNF: $xy\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z}$

Conjunctive normal form

A **maxterm** is a Boolean function which has precisely one 0 in the final column of its truth table.

Example of a maxterm

Table 8			
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	1

Sinse $m(x, y, z) = 0$

if and only if

$x = 0, y = 1, z = 1,$

it follows that

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

The expression $x + \bar{y} + \bar{z}$ is called the **sum representation** of the maxterm m.

Conjunctive normal form

- Any Boolean function which does not equal 1 identically can be expressed uniquely as a conjunction of maxterms.

This is called the **conjunctive normal form** (CNF) of the Boolean expression.

A procedure for constructing a Boolean expression representing a function with given values as CNF

By taking Boolean product of distinct maxterms we can build up a Boolean expression with a specified set of values.

In particular, a Boolean product of maxterms has the value 0 when exactly one of the maxterms in the product has the value 0.

It has the value 1 for all other combinations of values of the variables.

A procedure for constructing a Boolean expression representing a function with given values as CNF

Consequently, given a Boolean function, a Boolean product of maxterms can be formed that has the value 0 when this Boolean function has the value 0, and has the value 1 when the function has the value 1.

The maxterms in this Boolean sum correspond to those combinations of values for which the function has the value 0.

Conjunctive normal form

Example 10

Find the conjunctive normal form of the Boolean expression $(x + y)\overline{z}$.

Solution of example 10

Table 9					
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

Table 9					
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

CNF:

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

Functional completeness

- Every Boolean function can be expressed as a Boolean sum of minterms.
- Each minterm is the Boolean product of Boolean variables or their complements.
- This shows that every Boolean function can be represented using the Boolean operators $\cdot$, $+$, and $\bar{0}$.
- Because every Boolean function can be represented using these operators we say that the set $\{\cdot, +, \bar{0}\}$ is **functionally complete**.

Functional completeness

Can we find a smaller set of functionally complete operators?

We can do so if one of the three operators of this set can be expressed in terms of the other two.

Functional completeness

- The set $\{\cdot, \bar{0}\}$ is functionally complete:

$$x + y = \overline{\bar{x} \cdot \bar{y}}.$$

Functional completeness

- The set $\{+, \bar{0}\}$ is functionally complete:

$$x \cdot y = \overline{\bar{x} + \bar{y}}.$$

Functional completeness

We have found sets containing two operators that are functionally complete.

Can we find a smaller set of functionally complete operators, namely, a set containing just one operator? Such sets exist.

Functional completeness

- Define the $|$ or $NAND$ operator by

$$1|1 = 0 \text{ and } 1|0 = 0|1 = 0|0 = 1.$$

Define the $\downarrow$ or NOR operator by

$$1 \downarrow 1 = 1 \downarrow 0 = 0 \downarrow 1 = 0 \text{ and } 0 \downarrow 0 = 1.$$

Both of the sets $\{|\}$ and $\{\downarrow\}$ are functionally complete.

Functional completeness

- To see that $\{|\}$ is functionally complete, because $\{\cdot, \bar{0}\}$ is functionally complete, all that we have to do is show that both of the operators $\cdot$ and $\bar{0}$ can be expressed using just the $|$ operator.

This can be done as

$$\bar{x} = x|x,$$
$$xy = (x|y)|(x|y).$$

Functional completeness

- To see that $\{\downarrow\}$ is functionally complete, because $\{\cdot, \bar{0}\}$ is functionally complete, all that we have to do is show that both of the operators $\cdot$ and $\bar{0}$ can be expressed using just the $\downarrow$ operator.

This can be done as

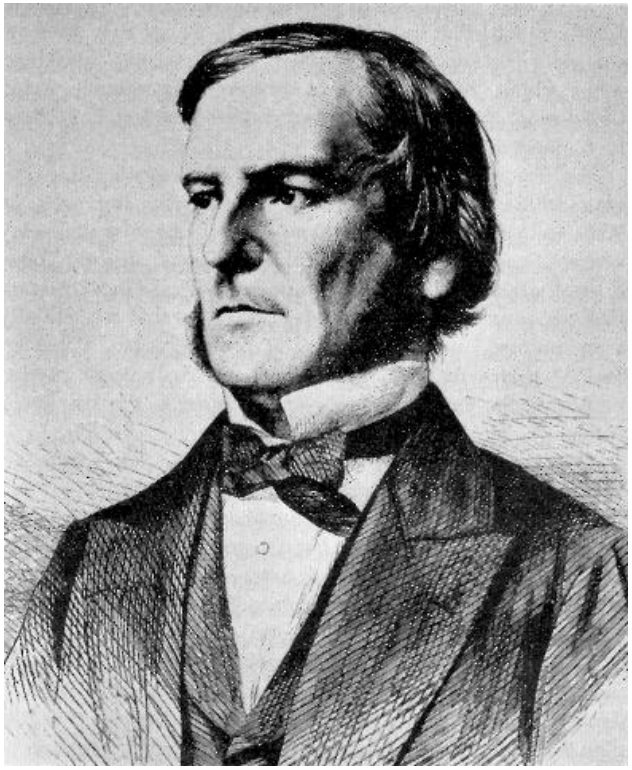
$$\bar{x} = x \downarrow x,$$

$$xy = (x \downarrow x) \downarrow (y \downarrow y).$$

Biography

George Boole, the son of a cobbler, was born in Lincoln, England, in November 1815. Because of his family's difficult financial situation, Boole struggled to educate himself while supporting his family.

Nevertheless, he became one of the most important mathematicians of the 1800s. Although he considered a career as a clergyman, he decided instead to go into teaching, and soon afterward opened a school of his own.

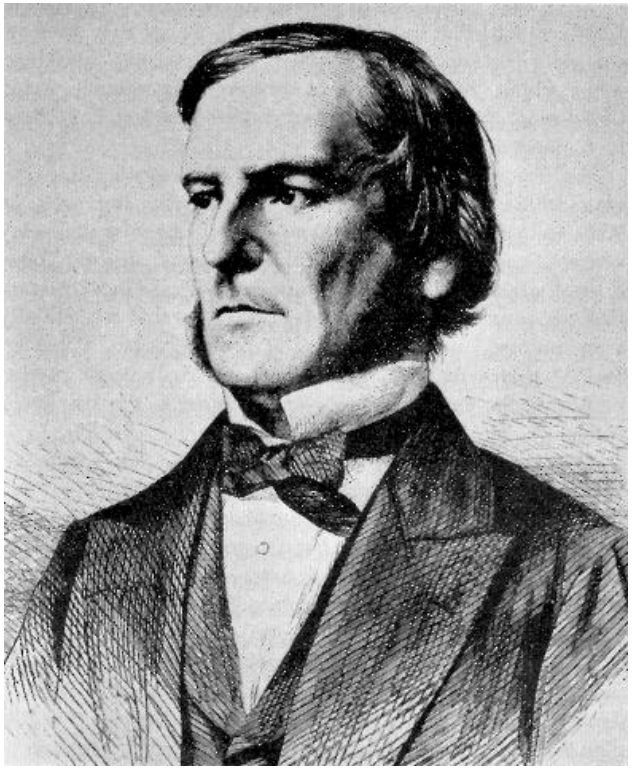


George Boole
1815 – 1864

Biography

In his preparation for teaching mathematics, Boole – unsatisfied with textbooks of his day – decided to read the works of the great mathematicians.

While reading papers of the great French mathematician Lagrange, Boole made discoveries in the calculus of variations, the branch of analysis dealing with finding curves and surfaces by optimizing certain parameters.



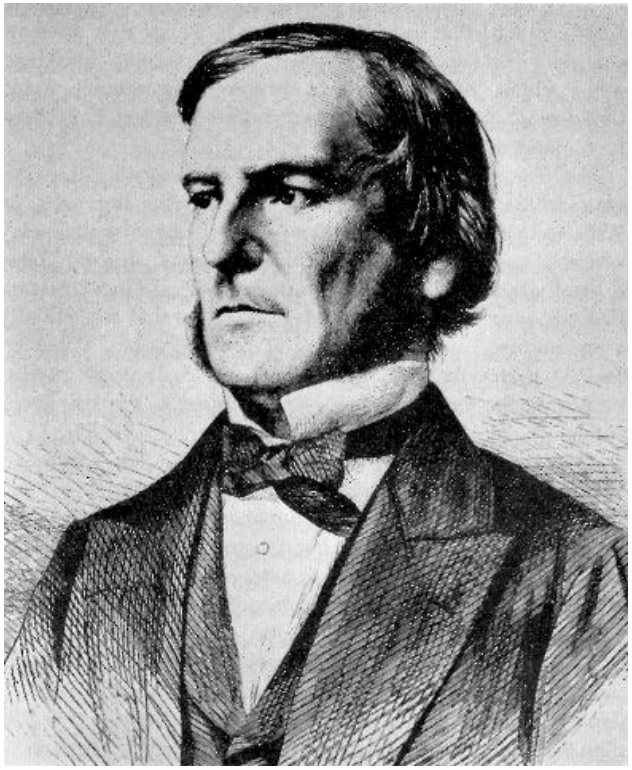
George Boole
1815 – 1864

Biography

In 1848 Boole published The Mathematical Analysis of Logic, the first of his contributions to symbolic logic.

In 1849 he was appointed professor of mathematics at Queen's College in Cork, Ireland.

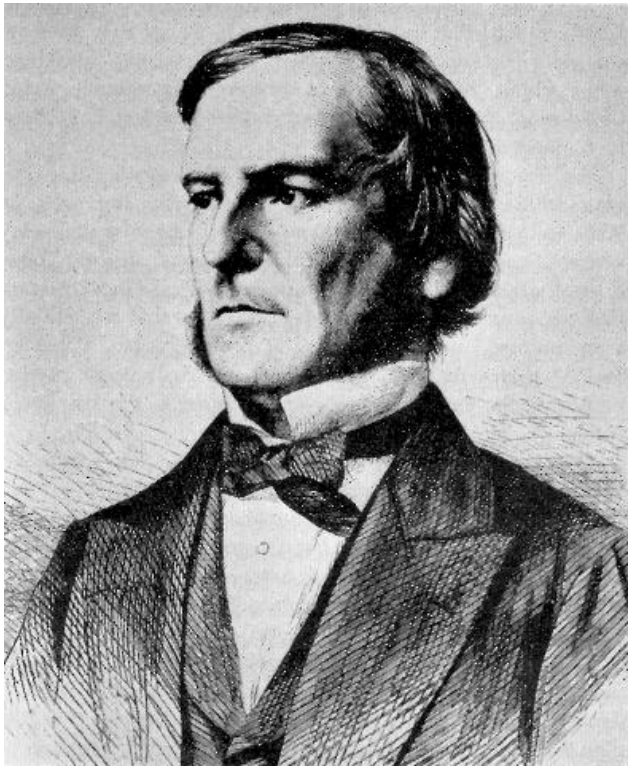
In 1854 he published The Laws of Thought, his most famous work. In this book, Boole introduced what is now called Boolean algebra in his honor.



George Boole
1815 – 1864

Biography

Boole wrote textbooks on differential equations and on difference equations that were used in Great Britain until the end of the nineteenth century.
Boole married in 1855; his wife was the niece of the professor of Greek at Queen's College.
In 1864 Boole died from pneumonia, which he contracted as a result of keeping a lecture engagement even though he was soaking wet from a rainstorm.



George Boole
1815 – 1864