Boolean Algebra

1 Boolean Algebra

A **Boolean algebra** is a set X equipped with two binary operations $\wedge, \vee$, one unary operation $'$, and two distinct elements $0, 1$, satisfying the following properties:

1) Commutative Laws

$$x \wedge y = y \wedge x, \quad x \vee y = y \vee x.$$

2) Associative Laws

$$x \wedge (y \wedge z) = (x \wedge y) \wedge z,$$
$$x \vee (y \vee z) = (x \vee y) \vee z.$$

3) Distributive Laws

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z),$$
$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z).$$

4) Identity Laws

$$x \wedge 1 = x, \quad x \vee 0 = x.$$

5) Complementation Laws

$$x \wedge x' = 0, \quad x \vee x' = 1.$$

The operation $\vee$ is called **join**, $\wedge$ is called **meet**, and the unary operation $'$ is called **complementation**.

Theorem 1.1 (Duality Principle)**.** *Let F be a formula on a Boolean algebra. Let F' be a formula obtained from F by interchanging $\wedge$ and $\vee$, 0 and 1, and keeping other variables unchanged, then F is a valid identity if and only if F' is a valid identity, i.e.,*

$$F \text{ is valid} \iff F' \text{ is valid}$$

Theorem 1.2. *The following properties hold in every Boolean algebra:*

a) Idempotent Laws

$$x \wedge x = x, \quad x \vee x = x.$$

b) More Identity Laws

$$x \wedge 0 = 0, \quad x \vee 1 = 1.$$

c) Absorbtion Laws

$$(x \wedge y) \vee x = x, \quad (x \vee y) \wedge x = x.$$

Proof. a) On the one hand $x \wedge x = (x \wedge x) \vee 0 = (x \wedge x) \vee (x \wedge x')$. On the other hand $x = x \wedge 1 = x \wedge (x \vee x') = (x \wedge x) \vee (x \wedge x')$. Likewise,

$$x \vee x = (x \vee x) \wedge 1 = (x \vee x) \wedge (x \vee x'),$$

$$x = x \vee 0 = x \vee (x \wedge x') = (x \vee x) \wedge (x \vee x').$$

b)

$$x \wedge 0 = x \wedge (x \wedge x') = (x \wedge x) \wedge x' = x \wedge x' = 0;$$

$$x \vee 1 = x \vee (x \vee x') = (x \vee x) \vee x' = x \vee x' = 1.$$

c)

$$(x \wedge y) \vee x = (x \wedge y) \vee (x \wedge 1) = x \wedge (y \vee 1) = x \wedge 1 = x;$$

$$(x \vee y) \wedge x = (x \vee y) \wedge (x \vee 0) = x \vee (y \wedge 0) = x \vee 0 = x.$$

□

Lemma 1.3 (Complementation Lemma)**.** *If $w \vee z = 1$ and $w \wedge z = 0$, then*

$$z = w'.$$

Proof.

$$\begin{aligned} z &= z \vee 0 = z \vee (w \wedge w') = (z \vee w) \wedge (z \vee w') \\ &= (w \vee z) \wedge (w' \vee z) = 1 \wedge (w' \vee z) = (w \vee w') \wedge (w' \vee z) \\ &= (w' \vee w) \wedge (w' \vee z) = w' \vee (w \wedge z) = w' \vee 0 = w'. \end{aligned}$$

□

Corollary 1.4. $(z')' = z$.

Proof. Let $w = z'$. Since $w \vee z = z' \vee z = 1$ and $w \wedge z = z' \wedge z = 0$, it follows that $z = w'$, i.e., $(z')' = z$. □

Theorem 1.5 (De Morgan Laws)**.**

$$(x \vee y)' = x' \wedge y', \quad (x \wedge y)' = x' \vee y'.$$

Proof. From the Complementation Lemma, for the first identity we only need to prove

$$(x \vee y) \vee (x' \wedge y') = 1, \quad (x \vee y) \wedge (x' \wedge y') = 0.$$

In fact,

$$\begin{aligned}(x \vee y) \vee (x' \wedge y') &= [(x \vee y) \vee x'] \wedge [(x \vee y) \vee y'] \\ &= (y \vee 1) \wedge (x \vee 1) = 1 \wedge 1 = 1;\end{aligned}$$

$$\begin{aligned}(x \vee y) \wedge (x' \wedge y') &= [x \wedge (x' \wedge y')] \vee [y \wedge (x' \wedge y')] \\ &= (0 \wedge y') \vee (0 \wedge x') = 0 \vee 0 = 0.\end{aligned}$$

For the second identity we need to prove

$$(x \wedge y) \vee (x' \vee y') = 1, \quad (x \wedge y) \wedge (x' \vee y') = 0.$$

In fact,

$$\begin{aligned}(x \wedge y) \vee (x' \vee y') &= [x \vee (x' \vee y')] \vee [y \vee (x' \vee y')] \\ &= (1 \vee y') \wedge (1 \vee x') = 1 \wedge 1 = 1;\end{aligned}$$

$$\begin{aligned}(x \wedge y) \wedge (x' \vee y') &= [(x \wedge y) \wedge x'] \vee [(x \wedge y) \wedge y'] \\ &= (y \wedge 0) \vee (x \wedge 0) = 0 \wedge 0 = 0.\end{aligned}$$

□

Example 1.1. The power set $\mathcal{P}(S)$ of a nonempty set S is a Boolean algebra whose

- binary operation $\wedge$ is the set intersection $\cap$,

- binary operation $\cup$ is the set union $\cup$,
- unary operation $'$ is the set complement $^{-}$,
- distinct element 0 is the empty set $\varnothing$, and
- distinct element 1 is the whole set S.

Example 1.2. The binary space $\mathbb{B} = \{0, 1\}$ is a Boolean algebra, where

$$x \wedge y = \min\{x, y\}, \quad x \vee y = \max\{x, y\}, \quad x' = 1 - x.$$

Example 1.3. The n**-dimensional binary space** is the Cartesian product

$$\mathbb{B}^n = \{0, 1\}^n = \mathbb{B} \times \cdots \times \mathbb{B} \ (n \text{ copies}),$$

and is a Boolean algebra under the Boolean operations

$$(x_1, \ldots, x_n) \wedge (y_1, \ldots, y_n) = (x_1 \wedge y_1, \ldots, x_n \wedge y_n),$$

$$(x_1, \ldots, x_n) \vee (y_1, \ldots, y_n) = (x_1 \vee y_1, \ldots, x_n \vee y_n),$$

$$(x_1, \ldots, x_n)' = (x_1', \ldots, x_n'),$$

whose two distinct elements $\mathbf{0}$ and $\mathbf{1}$ are

$$\mathbf{0} = (0, \ldots, 0), \quad \mathbf{1} = (1, \ldots, 1).$$

Example 1.4. The set of sentences, generated by some simple sentences $p, q, \ldots$ with logic connectives $\wedge, \vee, \neg$, forms a Boolean algebra, where $\neg$ is the unary operation $'$, 0 is the contradiction, and 1 is the tautology.

Example 1.5. The set of all functions from a nonempty set S to $\mathbb{B}$, denoted $\mathbb{B}^S$, forms a Boolean algebra, where for functions $f, g : S \to \mathbb{B}$,

$$(f \wedge g)(x) = f(x) \wedge g(x), \quad (f \vee g)(x) = f(x) \vee g(x),$$

$$f'(x) = 1 - f(x),$$

and 0 is the constant function having the value zero everywhere, 1 is the constant function having the value 1 everywhere.

Two Boolean algebras B_1, B_2 are said to be **isomorphic** if there is a bijection $\phi : B_1 \to B_2$ such that for $x, y \in B_1$,

$$\phi(x \wedge y) = \phi(x) \wedge \phi(y), \quad \phi(x \vee y) = \phi(x) \vee \phi(y),$$

$$\phi(x') = \phi(x)', \quad \phi(0) = 0, \quad \phi(1) = 1.$$

The map ϕ is known as an **isomorphism**.

Example 1.6. Given a nonempty set S, the Boolean algebras $\mathcal{P}(S)$ and $\mathbb{B}^S$ are isomorphic by the isomorphism

$$\phi : \mathcal{P}(S) \to \mathbb{B}^S, \quad A \mapsto \phi(A) = 1_A,$$

where $A \in \mathcal{P}(S)$, 1_A is the characteristic function of A, defined by

$$1_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \in A^c = S \smallsetminus A. \end{cases}$$

Theorem 1.6 (Principle of Duality)**.** *Let f, g be two expressions in a Boolean algebra B. If there is a formula*

$$f(x_1, \ldots, x_n, \wedge, \vee, 0, 1) = g(x_1, \ldots, x_n, \wedge, \vee, 0, 1),$$

then we have the following corresponding formula

$$f(x_1, \ldots, x_n, \vee, \wedge, 1, 0) = g(x_1, \ldots, x_n, \vee, \wedge, 1, 0),$$

which is obtained from the given formula by interchanging $\wedge$ and $\vee$, and interchanging 0 and 1.

Proof. Let us write $y_i = x_i'$ for $i = 1, \ldots, n$. Note that $f' = g'$.

$$\begin{aligned} [f(x_1, \ldots, x_n, \wedge, \vee, 0, 1)]' &= f(x_1', \ldots, x_n', \vee, \wedge, 1, 0) \\ &= f(y_1, \ldots, y_n, \vee, \wedge, 1, 0), \end{aligned}$$

$$\begin{aligned} [g(x_1, \ldots, x_n, \wedge, \vee, 0, 1)]' &= g(x_1', \ldots, x_n', \vee, \wedge, 1, 0) \\ &= g(y_1, \ldots, y_n, \vee, \wedge, 1, 0). \end{aligned}$$

We have $f(y_1, \ldots, y_n, \vee, \wedge, 1, 0) = g(y_1, \ldots, y_n, \vee, \wedge, 1, 0)$. $\square$

For instance, consider the following Boolean formulas

$$f(x_1, x_2, x_3, \wedge, \vee, 0, 1) = (x_1 \vee x_2') \wedge (x_3 \wedge 1)' \wedge (x_1 \vee 0),$$
$$g(x_1, x_2, x_3, \wedge, \vee, 0, 1) = (x_1' \wedge x_2)' \wedge (x_3' \vee 0) \wedge (x_1 \vee 0).$$

Note that $f(x_1, x_2, x_3, \wedge, \vee, 0, 1) = g(x_1, x_2, x_3, \wedge, \vee, 0, 1)$. We see that

$$\begin{aligned} f(x_1, x_2, x_3, \vee, \wedge, 1, 0) &= (x_1 \wedge x_2') \vee (x_3 \vee 0)' \vee (x_1 \wedge 1) \\ &= (x_1 \wedge x_2') \vee x_3' \vee x_1, \end{aligned}$$

$$\begin{aligned} g(x_1, x_2, x_3, \vee, \wedge, 1, 0) &= (x_1' \vee x_2)' \vee (x_3' \wedge 1) \vee (x_1 \wedge 1) \\ &= (x_1 \wedge x_2') \vee x_3' \vee x_1. \end{aligned}$$

2 Boolean Functions and Boolean Expressions

It is convenient to denote the truth values T by 1 and F by 0, and write $\mathbb{B} = \{0, 1\}$. The product set $\mathbb{B}^n$ is called the n-dimensional **Boolean algebra**.

A **Boolean function** of n variables $x_1, \ldots, x_n \in \mathbb{B}$ is a map $f : \mathbb{B}^n \to \mathbb{B}$. The variables $x_1, \ldots, x_n$ are called **Boolean variables** and can be viewed as simple statements. The variables $x_1, \ldots, x_n$ and their negations $\neg x_1, \ldots, \neg x_n$ are called **literals**. For convenience, we write $\neg x_1, \ldots, \neg x_n$ as $\bar{x}_1, \ldots, \bar{x}_n$. A **Boolean expression** is a sentence consisting of literals and connectives $\wedge$ and $\vee$, where

$$a \wedge b = \min\{a, b\}, \quad a \vee b = \max\{a, b\}.$$

We usually write $a \wedge b = ab$ and $\bar{a} = a'$.

A Boolean expression (formula) is a concrete form of a Boolean function.

Theorem 2.1. *Every Boolean function can be expressed as a Boolean formula.*

Proof. We proceed by induction on n, the number of Boolean variables. For $n = 1$, there are exactly four Boolean functions $f : \mathbb{B} \to \mathbb{B}$ of one variable x as follows:

$$\begin{aligned} f(0) = 0,\ f(1) = 1 &\Leftrightarrow f(x) = x; \\ f(0) = 1,\ f(1) = 0 &\Leftrightarrow f(x) = \bar{x}; \\ f(0) = 1,\ f(1) = 1 &\Leftrightarrow f(x) = x \vee \bar{x}; \\ f(0) = 0,\ f(1) = 0 &\Leftrightarrow f(x) = x \wedge \bar{x}. \end{aligned}$$

Assume it is true for Boolean functions of $n-1$ variables. Consider a Boolean function $f(x_1, x_2, \ldots, x_n)$ of n variables. Note that

$$g_0 = f(0, x_2, \ldots, x_n), \quad g_1 = f(1, x_2, \ldots, x_n)$$

are Boolean functions of $n-1$ variables $x_2, \ldots, x_n$. By induction hypothesis, g_0 and g_1 can be expressed as Boolean formulas.

We now claim the identity

$$f(x_1, x_2, \ldots, x_n) = (x_1 \wedge g_1) \vee (\bar{x}_1 \wedge g_0). \tag{1}$$

Recall that for any statement p,

$$0 \wedge p = 0, \quad 1 \wedge p = p, \quad 0 \vee p = p, \quad 1 \vee p = 1.$$

For $x_1 = 0$ and arbitrary $x_2, \ldots, x_n$,

$$\begin{aligned} \text{RHS} &= (0 \wedge g_1) \vee (1 \wedge g_0) \\ &= 0 \vee g_0 = g_0 \\ &= f(0, x_2, \ldots, x_n) = \text{LHS}. \end{aligned}$$

Likewise, for $x_1 = 1$ and arbitrary $x_2, \ldots, x_n$,

$$\begin{aligned} \text{RHS} &= (1 \wedge g_1) \vee (0 \wedge g_0) \\ &= g_1 \vee 0 = g_1 \\ &= f(1, x_2, \ldots, x_n) = \text{LHS}. \end{aligned}$$

The identity (1) shows that f can be expressed as a Boolean expression. $\square$

We also see from the proof that

$$f(x_1, \ldots, x_n) = (f(x_1, \ldots, x_{n-1}, 1) \wedge x_n) \vee (f(x_1, \ldots, x_{n-1}, 0) \wedge \bar{x}_n).$$

Example 2.1. Express the Boolean function

(x_1, x_2)	$f(x_1, x_2)$
(0,0)	1
(0,1)	0
(1,0)	0
(1,1)	1

as a Boolean formula.

Write the function $f(x_1, x_2)$ as

$$f(x_1, x_2) = (x_1 \wedge f(1, x_2)) \vee (\bar{x}_1 \wedge f(0, x_2)).$$

Since

$$\begin{aligned} f(1,0) = 0,\ f(1,1) = 1 \ &\Leftrightarrow\ f(1, x_2) = x_2, \\ f(0,0) = 1,\ f(0,1) = 0 \ &\Leftrightarrow\ f(0, x_2) = \bar{x}_2, \end{aligned}$$

we have

$$f(x_1, x_2) = (x_1 \wedge x_2) \vee (\bar{x}_1 \wedge \bar{x}_2).$$

Example 2.2. Express the Boolean function

(x_1, x_2, x_3)	$f(x_1, x_2, x_3)$
(0,0,0)	1
(0,0,1)	1
(0,1,0)	0
(0,1,1)	0
(1,0,0)	0
(1,0,1)	1
(1,1,0)	0
(1,1,1)	1

as a Boolean expression.
Solution. By Theorem 1,

$$f(x_1, x_2, x_3) \ = \ (x_1 \wedge f(1, x_2, x_3)) \vee (\bar{x}_1 \wedge f(0, x_2, x_3)).$$

Since

$$\begin{aligned} &f(1, x_2, x_3) = (x_2 \wedge f(1, 1, x_3)) \vee (\bar{x}_2 \wedge f(1, 0, x_3)), \\ &f(1,1,0) = 0,\ f(1,1,1) = 1 \ \Leftrightarrow\ f(1, 1, x_3) = x_3, \\ &f(1,0,0) = 0,\ f(1,0,1) = 1 \ \Leftrightarrow\ f(1, 0, x_3) = x_3, \end{aligned}$$

then

$$\begin{aligned} f(1, x_2, x_3) &= (x_2 \wedge x_3) \vee (\bar{x}_2 \wedge x_3) \\ &= (x_2 \vee \bar{x}_2) \wedge x_3 \\ &= 1 \wedge x_3 = x_3. \end{aligned}$$

Similarly,

$$\begin{array}{l} f(0,x_2,x_3) = (x_2 \wedge f(0,1,x_3)) \vee (\bar{x}_2 \wedge f(0,0,x_3)), \\ f(0,1,0)=0,\ f(0,1,1)=0 \ \Leftrightarrow\ f(0,1,x_3) = x_3 \wedge \bar{x}_3, \\ f(0,0,0)=1,\ f(0,0,1)=1 \ \Leftrightarrow\ f(0,0,x_3) = x_3 \vee \bar{x}_3, \end{array}$$

then

$$\begin{aligned} f(0,x_2,x_3) &= (x_2 \wedge x_3 \wedge \bar{x}_3) \vee (\bar{x}_2 \wedge (x_3 \vee \bar{x}_3)) \\ &= 0 \vee (\bar{x}_2 \wedge 1) \\ &= \bar{x}_2 \wedge 1 \\ &= \bar{x}_2. \end{aligned}$$

Thus

$$f(x_1,x_2,x_3) = (x_1 \wedge x_3) \vee (\bar{x}_1 \wedge \bar{x}_2) = x_1x_3 \vee x_1'x_2'.$$

A Boolean expression of the form

$$f(x_1,\ldots,x_n) = \bigvee_{k=1}^{m} f_k(x_1,\ldots,x_n),$$

is said to be in **disjunctive normal form** or **sum-of-product form** if each $f_k(x_1,\ldots,x_k)$ is a conjunction (product) of some of the literals

$$x_1,\ldots,x_n,\bar{x}_1,\ldots,\bar{x}_n.$$

The conjunction of literals is also known as minterm, and the disjunctive normal form is known as the sum of minterms. The minterms of 3-variables are given as follows:

(a,b,c)	Minterm with value 1 at (a,b,c)
$(0,0,0)$	$x'y'z'$
$(0,0,1)$	$x'y'z$
$(0,1,0)$	$x'yz'$
$(0,1,1)$	$x'yz$
$(1,0,0)$	$xy'z'$
$(1,0,1)$	$xy'z$
$(1,1,0)$	xyz'
$(1,1,1)$	xyz

Every Boolean expression can be written in disjunctive normal form. You may define **conjunctive normal form** in a similar way.

Example 2.3. Find a Boolean express for the Boolean function $f : \mathbb{B}^3 \to \mathbb{B}$ given by the table

(a,b,c)	$f(a,b,c)$	Minterm with value 1 at (a,b,c)
$(0,0,0)$	0	$x'y'z'$
$(0,0,1)$	1	$x'y'z$
$(0,1,0)$	1	$x'yz'$
$(0,1,1)$	1	$x'yz$
$(1,0,0)$	0	$xy'z'$
$(1,0,1)$	1	$xy'z$
$(1,1,0)$	1	xyz'
$(1,1,1)$	1	xyz

The Boolean function f can be expressed as

$$f(x,y,z) = x'y'z \vee x'yz' \vee x'yz \vee xy'z \vee xyz' \vee xyz.$$

However, $x'y'z \vee xy'z$ is equivalent to $y'z$; $x'yz' \vee x'yz$ is equivalent to $x'y$; and $xyz' \vee xyz$ is equivalent to xy. Thus $f(x,y,z)$ is simplified to

$$f(x,y,z) = y'z \vee x'y \vee xy = y'z \vee y.$$

3 Logic Networks

Computer science at hardware level includes design of devices to produce appropriate outputs from given inputs. For the inputs and outputs that are 0's and 1's, the problem is to design circuitry that transforms input data into required output data. Mathematically, the transform is a Boolean function, which has Boolean expressions. These Boolean expressions are build up from the literals with logic connectives $\wedge, \vee, \neg$, which can be realized by **logic gates**. We only use the following six gates, which are ANSI/IEEE standard.

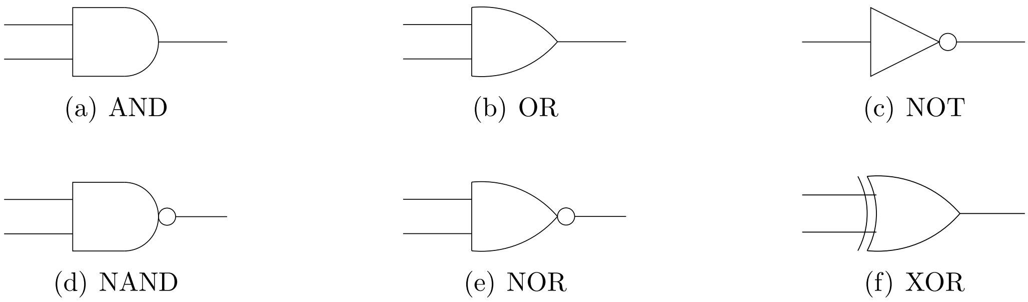
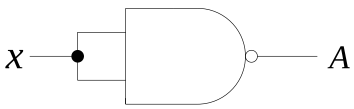


Figure 1: Six symbols of elementary logic gates

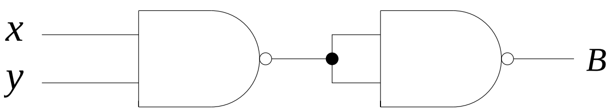
		NOT	AND	OR	NAND	NOR	XOR
x	y	x'	$x \wedge y$	$x \vee y$	$(x \wedge y)'$	$(x \vee y)'$	$x \oplus y$
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	1	0	0	0

Sometimes it is desirable or convenient to have all gates expressed in terms of one or two types of gates. For instance, every Boolean function or Boolean expression can be expressed by the same type gate or by two types of gates. For instance, all gates can be expressed by the gate NAND.

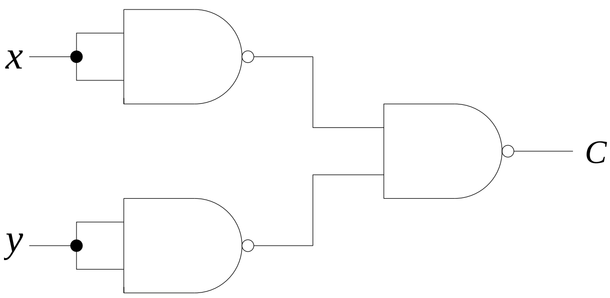
The negation $x' = (x \wedge x)' = A$ can be realized by



The conjunction $x \wedge y = (x \wedge y)'' = B$ can be realized by



The disjunction $x \vee y = (x' \wedge y')' = C$ can be realized by



Example 3.1. How to compute $28+15$? We write the two integers in binary numbers

$$43 = 32+8+2+1 = 1\cdot 2^5 + 0\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0,$$

$$27 = 16+8+2+1 = 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0.$$

We are to add binary the numbers 101011 (= 43 in base ten) and 11011 (= 27 in base ten). We perform

$$\begin{array}{lr} & 101011 \\ + & 11011 \\ \hline & 1000110 \end{array}$$

Binary addition is similar to ordinary decimal addition. Working from right to left, we can add the digits in each column. If the sum is 0 or 1, we write the sum in the answer line and carry a digit 0 to the column on the left. If the sum is 2 or 3 (in the case of $1+1$ and plus 1 carried from the right column), we write 0 or 1 respectively, and go to the next column on the left with a carry digit 1.

Half-adder. The rightmost column contains exactly two digits x, y [in the example $x = y = 1$]. The answer digit in this column is $x \oplus y$, and the carry digit for the next column is xy. Note that $S := x \oplus y = (x \vee y)(xy)'$ is the digit of output in this column, and $C := xy$ is the carry digit to the column in the left. See Figure

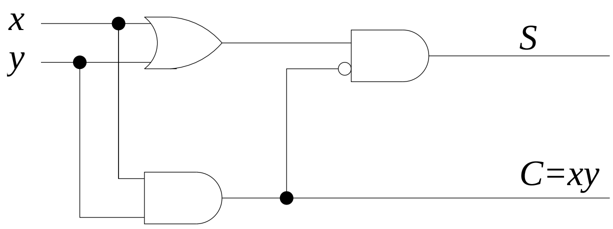


Figure 2: Half-adder: $S = x \oplus y$, $C = xy$

Full-adder. For more general case with a carry input C_I and a carry output C_o, we can combine two half-adders and an OR gate to have the following network: $S = x \oplus y \oplus C_I$, $C_o = xy \vee (x \oplus y)C_I$.

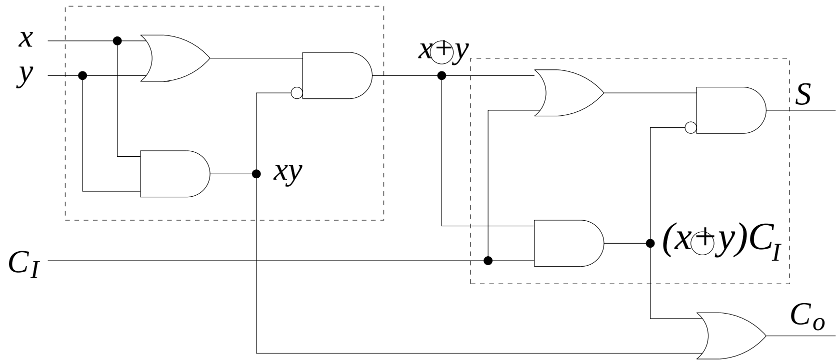


Figure 3: Full-adder: $S = x \oplus y \oplus C_I$, $C_o = xy \vee (x \oplus y)C_I$

4 Karnaugh Maps

The **Karnaugh map**, also known as the **K-map**, is a method to simplify Boolean expressions. The Karnaugh map reduces the need for extensive calculations by taking advantage of humans' pattern-recognition capability.

A required Boolean function can be transferred from a truth table onto a two-dimensional grid, where the cells are ordered in Gray code, and each cell position represents one combination of input conditions, while each cell value represents the corresponding output value. Optimal groups of 1s or 0s are identified, which represent the terms of a canonical form of the logic in the original truth table. These terms can be used to write a minimal Boolean expression representing the Boolean function.

Example 4.1. Given a Boolean function $f : \mathbb{B}^3 \to \mathbb{B}$.

(a) The Karnaugh map is

	yz	yz'	$y'z'$	$y'z$
x	0	1	0	0
x'	0	0	1	1

We see that

$$\begin{aligned} f(x,y,z) &= xyz' \vee x'y'z' \vee x'y'z \\ &= x'y' \vee xyz'. \end{aligned}$$

(b) The Karnaugh maps is

	yz	yz'	$y'z'$	$y'z$
x	0	1	1	0
x'	0	1	1	0

$$\begin{aligned} f(x,y,z) &= xyz' \vee xy'z' \vee x'yz' \vee x'y'z' \\ &= xz' \vee x'z' = z'. \end{aligned}$$

(c) The Karnaugh maps is

	yz	yz'	$y'z'$	$y'z$
x	1	0	0	1
x'	1	1	0	1

$$\begin{aligned} f(x,y,z) &= xyz \vee xy'z \vee x'yz \vee x'yz' \vee x'y'z \\ &= yz \vee x'y \vee y'z \\ &= z \vee x'y. \end{aligned}$$

Example 4.2. Given a Boolean function $f : \mathbb{B}^4 \to \mathbb{B}$ of four variables by the Karnaugh map

	zw	zw'	$z'w'$	$z'w$
xy	0	1	0	0
xy'	0	0	1	1
$x'y'$	1	0	1	1
$x'y$	1	1	1	0

$$\begin{aligned} f(x,y,z,w) &= xyzw' \vee xy'z'w' \vee xy'z'w \vee x'y'zw \vee \\ & \quad x'y'z'w' \vee z'y'z'w \vee x'yzw \vee x'yzw' \vee x'yz'w' \\ &= y'z' \vee x'zw \vee yzw' \vee x'yw'. \end{aligned}$$

Example 4.3. Given a Boolean function $f : \mathbb{B}^4 \to \mathbb{B}$ of four variables by the Karnaugh map

	yz	yz'	$y'z'$	$y'z$
wx	1	1	1	0
wx'	0	1	1	1
$w'x'$	1	1	1	1
$w'x$	1	1	1	0

$$
\begin{aligned}
f(x,y,z,w) &= wxyz \vee wxyz' \vee wxy'z' \vee \\
&\quad wx'yz' \vee wx'z'y'z' \vee wx'y'z \vee \\
&\quad w'x'yz \vee w'x'yz' \vee w'x'y'z' \vee w'x'y'z \vee \\
&\quad w'xyz \vee w'xyz' \vee w'xy'z' \\
&= z' \vee xy \vee x'y' \vee w'x' \\
&= z' \vee xy \vee x'y' \vee w'y.
\end{aligned}
$$

The Boolean function can be also written as

$$
\begin{aligned}
f(x,y,z,w) &= (wxy'z \vee wx'yz \vee w'xy'z)' \\
&= (w' \vee x' \vee y \vee z')(w' \vee x \vee x' \vee z')(w \vee x' \vee y \vee z').
\end{aligned}
$$

Example 4.4. Given a Boolean function $f : \mathbb{B}^4 \to \mathbb{B}$ of four variables by the Karnaugh map

	yz	yz'	$y'z'$	$y'z$
wx	1	1	1	0
wx'	1	0	1	1
$w'x'$	0	1	1	0
$w'x$	0	0	1	1

$$
\begin{aligned}
f(x,y,z,w) &= wxyz \vee wxyz' \vee wxy'z' \vee \\
&\quad wx'yz \vee wx'y'z' \vee wx'y'z \vee \\
&\quad w'x'yz' \vee w'x'y'z' \vee w'xy'z' \vee w'xy'z \\
&= wxy \vee wx'z \vee y'z' \vee w'xy'z.
\end{aligned}
$$

Example 4.5. $f(x,y,z) = w'y \vee y'z \vee xy'z' \vee wx'z'$ for the following table.

	yz	yz'	$y'z'$	$y'z$
wx	0	0	1	1
wx'	0	1	1	1
$w'x'$	1	1	0	1
$w'x$	1	1	1	1

Example 4.6. Find the function the Boolean formulas $f(x,y,z)$ for the fol-

lowing tables.

	yz	yz'	$y'z'$	$y'z$
wx	1	0	0	1
wx'	0	1	1	0
$w'x'$	0	1	1	0
$w'x$	1	0	0	1

	yz	yz'	$y'z'$	$y'z$
wx	0	1	1	0
wx'	1	1	1	1
$w'x'$	1	1	1	1
$w'x$	0	1	1	0

	yz	yz'	$y'z'$	$y'z$
wx	1	1	1	1
wx'	1	0	0	1
$w'x'$	1	0	0	1
$w'x$	1	1	1	1

	yz	yz'	$y'z'$	$y'z$
wx	0	1	1	0
wx'	1	0	0	1
$w'x'$	1	0	0	1
$w'x$	0	1	1	0