Boolean Algebra

Mathematical Methods in the Physical Sciences

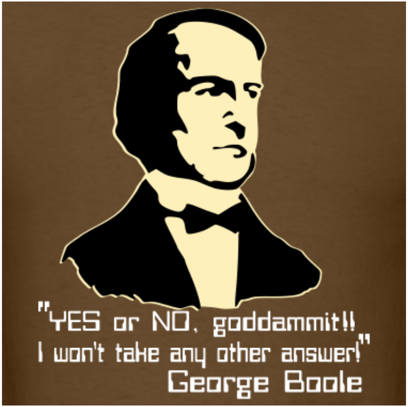
Steve Mazza

Naval Postgraduate School
Monterey, CA

SE3030, Winter/2014
Quantitative Methods of Systems Engineering

Introduction

In 1854 George Boole introduced a 2-state algebra designed to solve logic problems. Today this algebra is at the heart of network and computer science.

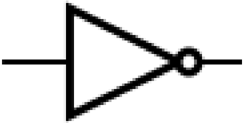


Receives input x and produces x' where

$$x' = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x = 1 \end{cases}$$

The output is the *compliment* of the input.

x	x'
1	0
0	1

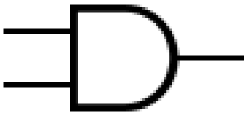


Receives input x_1 and x_2 and produces $(x_1 \wedge x_2)$ where

$$(x_1 \wedge x_2) = \begin{cases} 1 & \text{if } x_1 = x_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

There may be more than two inputs but the there is always one output.

x_1	x_2	$(x_1 \wedge x_2)$
0	0	0
0	1	0
1	0	0
1	1	1

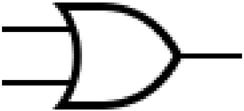


Receives input x_1 and x_2 and produces $(x_1 \vee x_2)$ where

$$(x_1 \vee x_2) = \begin{cases} 1 & \text{if } x_1 = 1 \text{ or } x_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

There may be more than two inputs but the there is always one output.

x_1	x_2	$(x_1 \wedge x_2)$
0	0	0
0	1	1
1	0	1
1	1	1

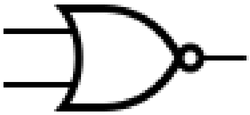


Receives input x_1 and x_2 and produces $(x_1 \vee x_2)'$ where

$$(x_1 \vee x_2)' = \begin{cases} 1 & \text{if } x_1 = x_2 = 0 \\ 0 & \text{otherwise} \end{cases}$$

There may be more than two inputs but the there is always one output.

x_1	x_2	$(x_1 \wedge x_2)'$
0	0	1
0	1	0
1	0	0
1	1	0

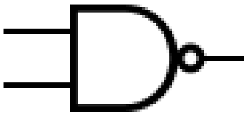


Receives input x_1 and x_2 and produces $(x_1 \wedge x_2)'$ where

$$(x_1 \wedge x_2)' = \begin{cases} 1 & \text{if } x_1 = 0 \text{ or } x_2 = 0 \\ 0 & \text{otherwise} \end{cases}$$

There may be more than two inputs but the there is always one output.

x_1	x_2	$(x_1 \wedge x_2)'$
0	0	1
0	1	1
1	0	1
1	1	0



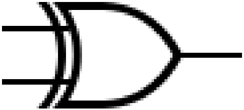
Negated and Exclusive Gates
XOR Gate

Receives input x_1 and x_2 and produces $(x_1 \oplus x_2)$ where

$$(x_1 \oplus x_2) = \begin{cases} 1 & \text{if only } x_1 = 0 \text{ or only } x_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

There may be more than two inputs but the there is always one output. The XNOR gate implements the logical expressions: $x_1 \wedge \overline{x_2} \vee \overline{x_1} \wedge x_2$ and $(x_1 \vee x_2) \wedge \overline{x_1 \wedge x_2}$.

x_1	x_2	$(x_1 \oplus x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

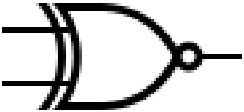


Receives input x_1 and x_2 and produces $(x_1 \oplus x_2)'$ where

$$(x_1 \oplus x_2)' = \begin{cases} 1 & \text{if } x_1 = x_2 \\ 0 & \text{otherwise} \end{cases}$$

There may be more than two inputs but the there is always one output. The XNOR gate implements the logical expression: $x_1 \wedge x_2 \vee \overline{x_1} \wedge \overline{x_2}$.

x_1	x_2	$(x_1 \oplus x_2)'$
0	0	1
0	1	0
1	0	0
1	1	1



Combinatorial Circuit

In digital circtuis, low voltages represent 0 and high voltages represent 1.

Combinatorial Circuit

A combinatorial circuit iis a circuit which produces a unique output for every combination of inputs.

Negative feedback from an op-amp provides a counter-example since the feedback loop negates the uniqueness of the output for every combination of inputs.

Boolean Expression

Boolean Expression

A boolean expression is any expression built up from individual state variables (e.g., x_1, x_2) by applyting the operations $\wedge$, $\vee$, and $'$ a finite number of times.

The output of a combinatorial circuit is a boolean expression.

- Associative Laws: $\forall a, b, c \in 0, 1$
  - $(a \wedge b) \wedge c = a \wedge (b \wedge c)$
  - $(a \vee b) \vee c = a \vee (b \vee c)$
- Identity Laws: $\forall a \in 0, 1$
  - $(a \wedge 1) = a$
  - $(a \vee 0) = a$
- Commutative Laws: $\forall a, b \in 0, 1$
  - $(a \wedge b) = (b \wedge a)$
  - $(a \vee b) = (b \vee a)$
- Complement Laws: $\forall z \in 0, 1$
  - $(a \wedge a') = 0$
  - $(a \vee a') = 1$
- Distributive Laws: $\forall a, b, c \in 0, 1$
  - $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
  - $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

$$(x_1 \wedge x_2)' = x_1' \vee x_2'$$
$$(x_1 \vee x_2)' = x_1' \wedge x_2'$$

We demonstrate the first example.

x_1	x_2	$(x_1 \wedge x_2)'$	x_1'	x_2'	$x_1' \vee x_2'$
1	1	0	0	0	0
1	0	1	0	1	1
0	1	1	1	0	1
0	0	1	1	1	1

Equivalent Combinatorial Circuits

Two combinatorial circuits are said to be equivalent if they produce the same output for the same input.

Example

$$(x_1 \wedge x_2)' = y \tag{1}$$
$$(x_1' \vee x_2') = y \tag{2}$$

x_1	x_2	y
1	1	0
1	0	1
0	1	1
0	0	1

Table : (example 1)

x_1	x_2	y
1	1	0
1	0	1
0	1	1
0	0	1

Table : (example 2)

Boolean Algebra

A Boolean algebra B consists of a set S together with any two binary operations $\wedge$ and $\vee$, a singular operation $'$ and two specific elements 0 and 1 on S such that the following laws hold.

- Associative Laws: $\forall a, b, c \in S$
  - $(a \wedge b) \wedge c = a \wedge (b \wedge c)$
  - $(a \vee b) \vee c = a \vee (b \vee c)$
- Identity Laws: $\forall a \in S$
  - $(a \wedge 1) = a$
  - $(a \vee 0) = a$
- Commutative Laws: $\forall a, b \in S$
  - $(a \wedge b) = (b \wedge a)$
  - $(a \vee b) = (b \vee a)$
- Complement Laws: $\forall z \in S$
  - $(a \wedge a') = 0$
  - $(a \vee a') = 1$
- Distributive Laws: $\forall a, b, c \in S$
  - $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
  - $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

Dual of a Statement

Two Boolean expressions are said to be the dual of each other if one expression is obtained from the other by the following replacements:

- replace 0 by 1
- replace 1 by 0
- replace $\wedge$ by $\vee$
- replace $\vee$ by $\wedge$

Example 1

$(x \wedge y)' = x' \vee y'$ is the dual of $(x \vee y)' = x' \wedge y'$

Example 2

$(x \wedge 1) = x$ is the dual of $(x \vee 0) = x$

Boolean Function

Let $B = (S, \vee, \wedge, ', 0, 1)$ be a Boolean algebra and let $X(x_1, x_2, x_3, \ldots, x_n)$ be a Boolean expression in n variables. A function $f : B^n \to B$ is called a Boolean function if f is of the form

$$f(x_1, x_2, x_3, \ldots, x_n) = X(x_1, x_2, x_3, \ldots, x_n)$$

Various Normal Forms

- Disjunctive normal form: a Boolean function $f : B^n \to B$ consisting of a sum of elamantary products.
- Conjunctive normal form: a Boolean function $f : B^n \to B$ consisting of a product of elamantary sums.

Questions?