11.8 Basic properties of Boolean algebra

Some first properties

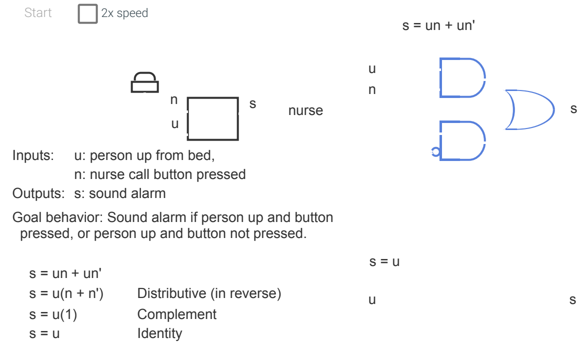
The benefit of building circuits from logic gates, rather than directly from transistors, becomes clear after learning some basic properties of Boolean algebra.

Table 11.8.1: A few basic properties of Boolean algebra.

Property	Name	Description
a(b + c) = ab + ac	***Distributive (for AND)***	Same as multiplication in regular algebra
a + a' = 1	***Complement***	Clearly one of a, a' must be 1 1 + 0 = 1 0 + 1 = 1
a · 1 = a	***Identity***	Result of a · 1 is always a's value *0* · 1 = *0* *1* · 1 = *1*

PARTICIPATION ACTIVITY 11.8.1: The properties of Boolean algebra are useful to simplify an equation, yielding a simpler circuit: Out-of-bed alarm system.

Start 2x speed



Applying Boolean algebra properties led to a simpler expression and thus a simpler circuit. Simplifying expressions is a common use of Boolean algebra.

PARTICIPATION ACTIVITY 11.8.2: Simplifying an expression using Boolean algebra.

Original expression: (d')(e + f)(d + d')

(d')(e + f) (d')(e + f)(1) d'e + d'f (d')(e + f)(d + d')

Original expression

Complement

Identity

Distributive

Reset

More properties

Below are more properties of Boolean algebra.

Table 11.8.2: More properties.

Property	Name	Description
ab = ba	***Commutative (for AND)***	Same as multiplication for regular algebra
a + b = b + a	***Commutative (for OR)***	Same as addition for regular algebra
a + 1 = *1*	***Null elements***	OR only needs one 1 to evaluate to 1 a = 0 0 + 1 = *1*

		a = 1 1 + 1 = 1
a + a = a aa = a	***Idempotent***	0 + 0 = 0 1 + 1 = 1 0 · 0 = 0 1 · 1 = 1

PARTICIPATION ACTIVITY

11.8.3: Simplifying an expression using more Boolean algebra properties.

Original expression: (e + 1)(e'f + fe' + d')

(e + 1)(e'f + fe' + d')   (1)(e'f + fe' + d')   e'f + e'f + d'   e'f + fe' + d'   e'f + d'

Original expression

Null elements

Identity

Commutative (for AND)

Idempotent

Reset

CHALLENGE ACTIVITY

11.8.1: Simplify the equation using Boolean algebra properties.

Start

Simplify z(x+x') to z

Properties

Distributive

ab+ac ≡ a(b+c)

Commutative

a+b ≡ b+a

Complement

a+a' ≡ 1

Identity

a·1 ≡ a

1   2

Check   Next

Example: Motion-sensing light equation

A designer may initially write an equation that matches his/her natural thinking of desired behavior, as below. The designer can then apply Boolean algebra properties to obtain a simpler equation (and thus a simpler eventual circuit).

PARTICIPATION ACTIVITY

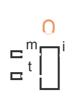
11.8.4: Simplifying an equation using Boolean algebra properties: Motion-sensing light.

Start   2x speed

Inputs: m: motion sensed

t: test mode

Outputs: i: illuminate lamp



Goal: Illuminate lamp if motion and not test mode,
or if test mode and no motion, or if test mode and motion

i = mt' + tm' + tm	Original equation
i = mt' + m't + mt	Commutative (for AND)
i = mt' + m't + mt + mt	Idempotent
i = mt' + mt + m't + mt	Commutative (for OR)
i = m(t' + t) + (m' + m)t	Distributive (twice)
i = m(1) + (1)t	Complement (twice)
i = m(1) + t(1)	Commutative (for AND)
i = m + t	Identity (twice)

PARTICIPATION ACTIVITY

11.8.5: Motion-sensing light example.

Consider the above motion-sensing light example.

1) The designer captured the desired behavior little-by-little as an equation, resulting in _____ terms on the right side.
   - 1
   - 2
   - 3

2) The first modification (commutative) just _____ literals within terms.
   - rearranged
   - eliminated

3) The next modification (idempotent) _____ the number of terms.
   - decreased
   - did not change
   - increased

4) Subsequent modifications resulted in a final equation having _____ terms on the right side.
   - 2
   - 3

Summary of common Boolean algebra properties

The following table summarizes commonly-used basic properties of Boolean algebra.

Table 11.8.3: Commonly-used basic properties of Boolean algebra.

Property	Name	Description
a(b + c) = ab + ac a + (bc) = (a + b)(a + c)	Distributive (AND) Distributive (OR)	(AND) Same as multiplication in regular algebra (OR) Not at all like regular algebra
ab = ba a + b = b + a	Commutative	Variable order does not matter. Good practice is to sort variables alphabetically.
(ab)c = a(bc) (a + b) + c = a + (b + c)	Associative	Same as regular algebra
aa' = 0 a + a' = 1	Complement (AND) Complement (OR)	(AND) Clearly one of a, a' must be 0 1 · 0 = 0 · 1 = 0 (OR) Clearly one of a, a' must be 1 1 + 0 = 0 + 1 = 1
a · 1 = a a + 0 = a	Identity (AND) Identity (OR)	(AND) Result of a · 1 is always a's value *0 · 1 = 0 1 · 1 = 1* (OR) Result of a + 0 is always a's value *0 + 0 = 0 1 + 0 = 1*
a · 0 = 0 a + 1 = 1	Null elements	Result doesn't depend on the value of a.
a · a = a a + a = a	Idempotent	Duplicate values can be removed.
(a')' = a	Involution	(0')' = (1)' = 0 (1')' = (0)' = 1
(ab)' = a' + b' (a + b)' = a'b'	DeMorgan's Law	*Discussed in another section*

PARTICIPATION ACTIVITY 11.8.6: Basic properties of Boolean algebra.

1) Which property allows one to change zxy into xyz?
   - Associative
   - Commutative
   - Identity

2) Which property allows one to change a + a into just a?
   - Identity
   - Idempotent
   - Complement

3) Which property allows transforming xy + xy' into x(y + y')
   - Complement

○ Distributive

4) Which property allows transforming x(y + y') into x(1)?

○ Complement

○ Identity

5) Which property allows transforming x(1) into x?

○ Complement

○ Identity

CHALLENGE ACTIVITY 11.8.2: Simplify the equation using Boolean algebra properties.

Start

Simplify z'w+z'w' to z'

Properties

Distributive			Identit
ab+ac	≡	a(b+c)	a·1
(a+b)(a+c)	≡	a+bc	a+(
Commutative			**Null el**
ab	≡	ba	a·0
a+b	≡	b+a	a+1
Complement			**Idemp**
aa'	≡	0	aa
a+a'	≡	1	a+a

1	2	3

Check Next

Provide feedback on this section