Switching Circuits and Boolean Algebra

Ioan Despi

despi@turing.une.edu.au

University of New England

August 30, 2012

Outline

Let's Start!

- Switching circuits are a way of describing pictorially the symbolic logic that you met earlier.

Let's Start!

- Switching circuits are a way of describing pictorially the symbolic logic that you met earlier.
- Boolean algebras are abstract mathematical constructions that unify the apparently different concepts of sets, symbolic logic and switching systems.

Switches

- A **switch** is a device which is attached to a point in an electrical circuit.

Switches

- A **switch** is a device which is attached to a point in an electrical circuit.
- The switch can be in either of two states, **open** or **closed**:

Switches

- A **switch** is a device which is attached to a point in an electrical circuit.
- The switch can be in either of two states, **open** or **closed**:
    - in the `open` state the switch does not allow current to flow through the point.

Switches

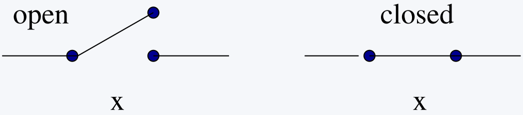
- A **switch** is a device which is attached to a point in an electrical circuit.
- The switch can be in either of two states, **open** or **closed**:
  - in the `open` state the switch does not allow current to flow through the point.
  - in the `closed` state the switch does allow current to flow through the point.

Switches

- A **switch** is a device which is attached to a point in an electrical circuit.
- The switch can be in either of two states, **open** or **closed**:
  - in the `open` state the switch does not allow current to flow through the point.
  - in the `closed` state the switch does allow current to flow through the point.
- We shall indicate a switch by means of the symbols

Switches

- A **switch** is a device which is attached to a point in an electrical circuit.
- The switch can be in either of two states, **open** or **closed**:
  - in the `open` state the switch does not allow current to flow through the point.
  - in the `closed` state the switch does allow current to flow through the point.
- We shall indicate a switch by means of the symbols



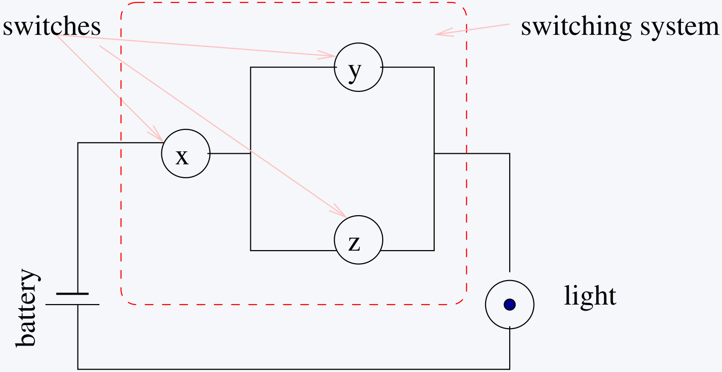
- In principle, "**x**" indicates a sentence such that the associated switch is closed when x is true and it is open when x is false. Another notation is $-x-$.

Switching Circuits

- Two points (available to the outside) are connected by a **switching circuit** if and only if they are connected by wires on which a finite collection of switches are located.

Switching Circuits

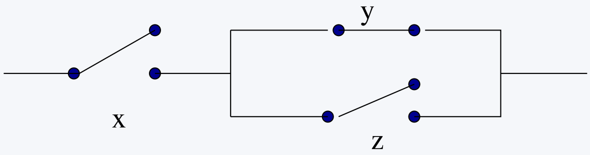
- Two points (available to the outside) are connected by a **switching circuit** if and only if they are connected by wires on which a finite collection of switches are located.
- For example, the following



is a **switching circuit**, making use of an energy source (battery) an output (light) as well as a switching system.

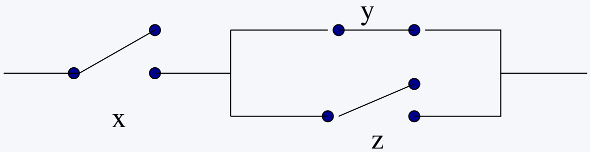
Switching Circuits

- If switches x and z are open while switch y is closed, then the state of the switching system may be represented by



Switching Circuits

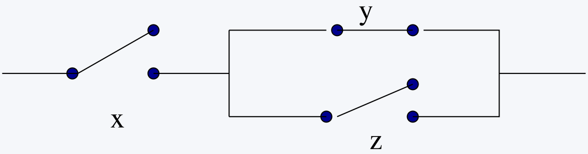
- If switches x and z are open while switch y is closed, then the state of the switching system may be represented by



- In order to describe switching systems formally and mathematically, we denote open and closed states by 0 and 1, respectively.

Switching Circuits

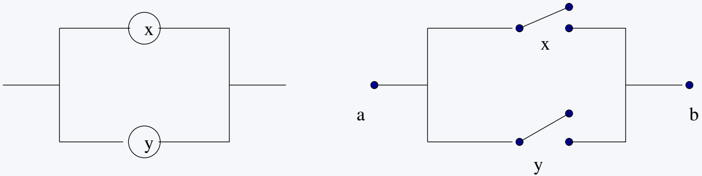
- If switches x and z are open while switch y is closed, then the state of the switching system may be represented by



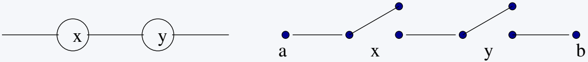
- In order to describe switching systems formally and mathematically, we denote open and closed states by 0 and 1, respectively.
- It's obvious that the state space S for any switch or switching system is composed of two states: 0 (open) and 1 (closed), i.e., $S = \{0, 1\}$.

Switching Circuits

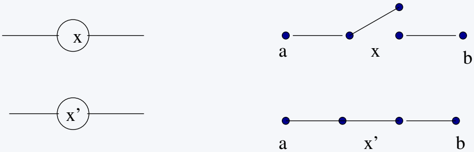
- In a switching system, switches may be connected with one another
  - ▶ in **parallel**: current flows between points a and b iff $x \vee y$ is true



  - ▶ in **series**: current flows between points a and b iff $x \wedge y$ is true



  - ▶ through the use of **complementary switches**: for any given switch x, the corresponding complementary switch, denoted by x', is always in the opposite state to that of x.



Switching Circuits

- Easy generalisation to the case of a finite number of switches $x_1, x_2, \ldots, x_n$:

Switching Circuits

- Easy generalisation to the case of a finite number of switches $x_1, x_2, \ldots, x_n$:
  - connected in parallel:
    current flows through the circuit iff $x_1 \vee x_2 \vee \ldots \vee x_n$ is true.

Switching Circuits

- Easy generalisation to the case of a finite number of switches $x_1, x_2, \ldots, x_n$:
  - connected in parallel:
    current flows through the circuit iff $x_1 \vee x_2 \vee \ldots \vee x_n$ is true.
  - connected in series:
    current flows through the circuit iff $x_1 \wedge x_2 \wedge \ldots \wedge x_n$ is true.

Switching Circuits

- Easy generalisation to the case of a finite number of switches $x_1, x_2, \ldots, x_n$:
  - connected in parallel:
    current flows through the circuit iff $x_1 \vee x_2 \vee \ldots \vee x_n$ is true.
  - connected in series:
    current flows through the circuit iff $x_1 \wedge x_2 \wedge \ldots \wedge x_n$ is true.
- For any two switches x and y, we use $x + y$ and $x \cdot y$ to denote their parallel and series connections respectively.

Switching Circuits

- Easy generalisation to the case of a finite number of switches $x_1, x_2, \ldots, x_n$:
  - connected in parallel:
    current flows through the circuit iff $x_1 \vee x_2 \vee \ldots \vee x_n$ is true.
  - connected in series:
    current flows through the circuit iff $x_1 \wedge x_2 \wedge \ldots \wedge x_n$ is true.
- For any two switches x and y, we use $x + y$ and $x \cdot y$ to denote their parallel and series connections respectively.
- The symbols "." and "+" here shouldn't be confused with those in the arithmetic, although there exist some similarities.

The state of compound switches

- With the above introduced notations, we can represent the state of, or the effect of switches in parallel, in series and so on by the following tables

Parallel

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

Series

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Complement

x	x'
0	1
1	0

switch y is open

switch x is closed

switching system with x and y in parallel is closed

The state of compound switches

- With the above introduced notations, we can represent the state of, or the effect of switches in parallel, in series and so on by the following tables

Parallel

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

Series

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Complement

x	x'
0	1
1	0

switch y is open

switch x is closed

switching system with x and y in parallel is closed

- As the special case in the above tables we have in particular $1+1=1$.

The state of compound switches

- With the above introduced notations, we can represent the state of, or the effect of switches in parallel, in series and so on by the following tables

Parallel

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

Series

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Complement

x	x'
0	1
1	0

switch y is open

switch x is closed

switching system with x and y in parallel is closed

- As the special case in the above tables we have in particular $1 + 1 = 1$.
- Notice a similarity between these tables and truth tables: here 0 can be considered as a F(alse) and 1 as a T(rue).

The state of compound switches

- With the above introduced notations, we can represent the state of, or the effect of switches in parallel, in series and so on by the following tables

Parallel

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

Series

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Complement

x	x'
0	1
1	0

switch y is open

switch x is closed

switching system with x and y in parallel is closed

- As the special case in the above tables we have in particular $1+1=1$.
- Notice a similarity between these tables and truth tables: here 0 can be considered as a F(alse) and 1 as a T(rue).
- The first two columns are constructed in a similar way to truth tables (with F's first instead of T's) to capture all possible combinations of 0's and 1's.

Boolean Algebra

- A `Boolean algebra` is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic and the foundation of electronic design of computers.

Boolean Algebra

- A `Boolean algebra` is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic and the foundation of electronic design of computers.
- It is a branch of mathematics developed around 1850 by George Boole as rules of *algebra* for logical thinking.

Boolean Algebra

- A `Boolean algebra` is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic and the foundation of electronic design of computers.
- It is a branch of mathematics developed around 1850 by George Boole as rules of *algebra* for logical thinking.
- Two preliminary definitions:

Boolean Algebra

- A `Boolean algebra` is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic and the foundation of electronic design of computers.
- It is a branch of mathematics developed around 1850 by George Boole as rules of *algebra* for logical thinking.
- Two preliminary definitions:
  - A **binary operation** on a set A is a mapping $f\colon A \times A \to A$, i.e., $f(a, b) \in A$ for any pair (a, b) with a and b both in A.

Boolean Algebra

- A `Boolean algebra` is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic and the foundation of electronic design of computers.
- It is a branch of mathematics developed around 1850 by George Boole as rules of *algebra* for logical thinking.
- Two preliminary definitions:
  - A **binary operation** on a set A is a mapping $f\colon A \times A \to A$, i.e., $f(a, b) \in A$ for any pair (a, b) with a and b both in A.
    - So a binary operation takes two elements of a set and produces a third.

Boolean Algebra

- A `Boolean algebra` is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic and the foundation of electronic design of computers.
- It is a branch of mathematics developed around 1850 by George Boole as rules of *algebra* for logical thinking.
- Two preliminary definitions:
  - A **binary operation** on a set A is a mapping $f\colon A \times A \to A$, i.e., $f(a, b) \in A$ for any pair (a, b) with a and b both in A.
    - So a binary operation takes two elements of a set and produces a third.
  - A **unary operation** on a set A is a mapping $f\colon A \to A$, i.e., $f(a) \in A$ for any a in A.

Boolean Algebra

- A `Boolean algebra` is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic and the foundation of electronic design of computers.
- It is a branch of mathematics developed around 1850 by George Boole as rules of *algebra* for logical thinking.
- Two preliminary definitions:
  - A **binary operation** on a set A is a mapping $f\colon A \times A \to A$, i.e., $f(a,b) \in A$ for any pair (a,b) with a and b both in A.
    - So a binary operation takes two elements of a set and produces a third.
  - A **unary operation** on a set A is a mapping $f\colon A \to A$, i.e., $f(a) \in A$ for any a in A.
    - So a unary operation takes one element of a set and produces another.

Boolean Algebra

- A `Boolean algebra` is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic and the foundation of electronic design of computers.
- It is a branch of mathematics developed around 1850 by George Boole as rules of *algebra* for logical thinking.
- Two preliminary definitions:
  - A **binary operation** on a set A is a mapping $f\colon A \times A \to A$, i.e., $f(a, b) \in A$ for any pair (a, b) with a and b both in A.
    - So a binary operation takes two elements of a set and produces a third.
  - A **unary operation** on a set A is a mapping $f\colon A \to A$, i.e., $f(a) \in A$ for any a in A.
    - So a unary operation takes one element of a set and produces another.

Boolean Algebra

- A `Boolean algebra` is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic and the foundation of electronic design of computers.
- It is a branch of mathematics developed around 1850 by George Boole as rules of *algebra* for logical thinking.
- Two preliminary definitions:
  - A **binary operation** on a set A is a mapping $f\colon A \times A \to A$, i.e., $f(a, b) \in A$ for any pair (a, b) with a and b both in A.
    - ⋆ So a binary operation takes two elements of a set and produces a third.
  - A **unary operation** on a set A is a mapping $f\colon A \to A$, i.e., $f(a) \in A$ for any a in A.
    - ⋆ So a unary operation takes one element of a set and produces another.

Example

For switching systems with state space $S=\{0, 1\}$, the "+" and "$\cdot$" operation are binary and the "$'$" operation is unary.

Boolean Algebra

- A `Boolean algebra` is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic and the foundation of electronic design of computers.
- It is a branch of mathematics developed around 1850 by George Boole as rules of *algebra* for logical thinking.
- Two preliminary definitions:
  - A **binary operation** on a set A is a mapping $f\colon A \times A \to A$, i.e., $f(a, b) \in A$ for any pair (a, b) with a and b both in A.
    - ⋆ So a binary operation takes two elements of a set and produces a third.
  - A **unary operation** on a set A is a mapping $f\colon A \to A$, i.e., $f(a) \in A$ for any a in A.
    - ⋆ So a unary operation takes one element of a set and produces another.

Example

For switching systems with state space $S=\{0, 1\}$, the "+" and "$\cdot$" operation are binary and the "$'$" operation is unary.

Solution. This is because for any switching systems x and y, we have that $x + y$, $x \cdot y$ and x' are all still switching systems with the same state space S.

Note. Binary operator or operation has nothing to do with binary numbers.

Boolean Algebra

Definition

A **Boolean algebra** is a set S on which are defined two binary operations $+$ and $\cdot$ and one unary operation $'$ and in which there are at least two distinct elements 0 and 1 such that the following properties hold for all $a, b, c \in S$

$$
\begin{array}{llcll}
\text{B1.} & a+b & = & b+a & \multirow{2}{*}{} \\
 & a \cdot b & = & b \cdot a & \\
\text{B2.} & (a+b)+c & = & a+(b+c) & \\
 & (a \cdot b) \cdot c & = & a \cdot (b \cdot c) & \\
\text{B3.} & a+(b \cdot c) & = & (a+b) \cdot (a+c) & \\
 & a \cdot (b+c) & = & (a \cdot b)+(a \cdot c) & \\
\text{B4.} & a+0 & = & a & \\
 & a \cdot 1 & = & a & \\
\text{B5.} & a+a' & = & 1 & \\
 & a \cdot a' & = & 0 &
\end{array}
$$

- B1: $a+b = b+a$, $a \cdot b = b \cdot a$ — commutativity
- B2: $(a+b)+c = a+(b+c)$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ — associativity
- B3: $a+(b \cdot c) = (a+b) \cdot (a+c)$, $a \cdot (b+c) = (a \cdot b)+(a \cdot c)$ — distributivity
- B4: $a+0 = a$, $a \cdot 1 = a$ — identity relations
- B5: $a+a' = 1$, $a \cdot a' = 0$ — complementation

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.
- They are *symbols* rather than normal numerical values.

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.
- They are *symbols* rather than normal numerical values.
- Likewise the operators "+", "$\cdot$" and "$'$" are also symbols, each representing the *designated* special roles.

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.
- They are *symbols* rather than normal numerical values.
- Likewise the operators "+", "·" and "′" are also symbols, each representing the *designated* special roles.
- A Boolean algebra is thus often represented by a *tuple* $(S, +, \cdot, ', 0, 1)$ which carries all the relevant components.

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.
- They are *symbols* rather than normal numerical values.
- Likewise the operators "+", "·" and "′" are also symbols, each representing the *designated* special roles.
- A Boolean algebra is thus often represented by a *tuple* $(S, +, \cdot, ', 0, 1)$ which carries all the relevant components.
- A variable $x \in S$ is called a **Boolean variable**.

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.
- They are *symbols* rather than normal numerical values.
- Likewise the operators "+", "$\cdot$" and "$'$" are also symbols, each representing the *designated* special roles.
- A Boolean algebra is thus often represented by a *tuple* $(S, +, \cdot, ', 0, 1)$ which carries all the relevant components.
- A variable $x \in S$ is called a **Boolean variable**.
- A combination of some elements of S (variables) via the connectives $+$ and $\cdot$ and the complement $'$ is a **Boolean expression**.

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.
- They are *symbols* rather than normal numerical values.
- Likewise the operators "+", "$\cdot$" and "$'$" are also symbols, each representing the *designated* special roles.
- A Boolean algebra is thus often represented by a *tuple* $(S, +, \cdot, ', 0, 1)$ which carries all the relevant components.
- A variable $x \in S$ is called a **Boolean variable**.
- A combination of some elements of S (variables) via the connectives $+$ and $\cdot$ and the complement $'$ is a **Boolean expression**.
- Among the 3 operations "$'$", "$\cdot$" and "+" in a Boolean expression, the "$'$" operation has the highest *precedence*, then comes the "$\cdot$" operation, and then the "+" operation.

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.
- They are *symbols* rather than normal numerical values.
- Likewise the operators "+", "$\cdot$" and "$'$" are also symbols, each representing the *designated* special roles.
- A Boolean algebra is thus often represented by a *tuple* $(S, +, \cdot, ', 0, 1)$ which carries all the relevant components.
- A variable $x \in S$ is called a **Boolean variable**.
- A combination of some elements of S (variables) via the connectives $+$ and $\cdot$ and the complement $'$ is a **Boolean expression**.
- Among the 3 operations "$'$", "$\cdot$" and "+" in a Boolean expression, the "$'$" operation has the highest *precedence*, then comes the "$\cdot$" operation, and then the "+" operation.
  - For example, $a + b' \cdot c$ in fact means $a + ((b') \cdot c)$.

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.
- They are *symbols* rather than normal numerical values.
- Likewise the operators "+", "·" and "′" are also symbols, each representing the *designated* special roles.
- A Boolean algebra is thus often represented by a *tuple* $(S, +, \cdot, ', 0, 1)$ which carries all the relevant components.
- A variable $x \in S$ is called a **Boolean variable**.
- A combination of some elements of S (variables) via the connectives $+$ and $\cdot$ and the complement ′ is a **Boolean expression**.
- Among the 3 operations "′", "·" and "+" in a Boolean expression, the "′" operation has the highest *precedence*, then comes the "·" operation, and then the "+" operation.
  - For example, $a + b' \cdot c$ in fact means $a + ((b') \cdot c)$.
- Each property in the definition of a Boolean algebra has its **dual** as part of the definition.

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.
- They are *symbols* rather than normal numerical values.
- Likewise the operators "+", "·" and "′" are also symbols, each representing the *designated* special roles.
- A Boolean algebra is thus often represented by a *tuple* $(S, +, \cdot, ', 0, 1)$ which carries all the relevant components.
- A variable $x \in S$ is called a **Boolean variable**.
- A combination of some elements of S (variables) via the connectives $+$ and $\cdot$ and the complement $'$ is a **Boolean expression**.
- Among the 3 operations "$'$", "$\cdot$" and "$+$" in a Boolean expression, the "$'$" operation has the highest *precedence*, then comes the "$\cdot$" operation, and then the "$+$" operation.
  - For example, $a + b' \cdot c$ in fact means $a + ((b') \cdot c)$.
- Each property in the definition of a Boolean algebra has its **dual** as part of the definition.
  - The dual is obtained by interchanging $+$ with $\cdot$ and 0 with 1.

Boolean Algebra

- The "0" and "1" in the above are just a notation, two special elements of S.
- They are *symbols* rather than normal numerical values.
- Likewise the operators "+", "$\cdot$" and "$'$" are also symbols, each representing the *designated* special roles.
- A Boolean algebra is thus often represented by a *tuple* $(S, +, \cdot, ', 0, 1)$ which carries all the relevant components.
- A variable $x \in S$ is called a **Boolean variable**.
- A combination of some elements of S (variables) via the connectives $+$ and $\cdot$ and the complement $'$ is a **Boolean expression**.
- Among the 3 operations "$'$", "$\cdot$" and "+" in a Boolean expression, the "$'$" operation has the highest *precedence*, then comes the "$\cdot$" operation, and then the "+" operation.
  - For example, $a + b' \cdot c$ in fact means $a + ((b') \cdot c)$.
- Each property in the definition of a Boolean algebra has its **dual** as part of the definition.
  - The dual is obtained by interchanging $+$ with $\cdot$ and 0 with 1.
  - Unary operation $'$ remains unchanged.

Boolean Algebra

Theorem

Let $(S, +, \cdot, ', 0, 1)$ *be a Boolean algebra. Then the following properties hold for all* $a, b, c \in S$

P1.	$a + a = a\,; \quad a \cdot a = a\,.$	*(idempotent laws)*
P2.	$a + 1 = 1\,; \quad a \cdot 0 = 0\,.$	*(dominance laws)*
P3.	$(a')' = a\,.$	*(double complement)*
P4.	$a + a \cdot b = a\,.$	*(absorption law)*
P5.	*If* $a + c = 1,\ a \cdot c = 0,$ *then* $c = a'.$	*(uniqueness of inver*
P6.	*If* $a \cdot c = b \cdot c,\ a \cdot c' = b \cdot c',$ *then* $a = b.$	*(cancellation law)*
P7.	*If* $a + c = b + c,\ a + c' = b + c',$ *then* $a = b.$	*(cancellation law)*

Proof of the Theorem

Proof.

For P1, the first half is derived from B3–B5 by

$$a + a \overset{B4}{=} (a + a) \cdot 1 \overset{B5}{=} (a + a) \cdot (a + a') \overset{B3}{=} a + (a \cdot a') \overset{B5}{=} a + 0 \overset{B4}{=} a\ ,$$

where the names on the equality sign indicate the property being used, while the second half is derived by

$$a \overset{B4}{=} a \cdot 1 \overset{B5}{=} a \cdot (a + a') \overset{B3}{=} a \cdot a + a \cdot a' \overset{B5}{=} a \cdot a + 0 \overset{B4}{=} a \cdot a\ .$$

For P2, we need to observe

$$a + 1 \overset{B5}{=} a + (a + a') \overset{B2}{=} (a + a) + a' \overset{P1}{=} a + a' \overset{B5}{=} 1$$
$$a \cdot 0 \overset{B5}{=} a \cdot (a \cdot a') \overset{B2}{=} (a \cdot a) \cdot a' \overset{P1}{=} a \cdot a' \overset{B5}{=} 0\ .$$

□

Proof of the Theorem

Proof.

The proof of P3 is

$$
\begin{aligned}
a'' &\overset{B4}{=} a'' \cdot 1 \overset{B5}{=} a'' \cdot (a + a') \overset{B3}{=} a'' \cdot a + a'' \cdot a' \overset{B5}{=} a'' \cdot a + 0 \overset{B5}{=} \\
&\quad a'' \cdot a + a' \cdot a \overset{B3}{=} (a'' + a') \cdot a \overset{B5}{=} 1 \cdot a \overset{B4}{=} a \,.
\end{aligned}
$$

The proof of P4 is

$$
a + a \cdot b \overset{B4}{=} a \cdot 1 + a \cdot b \overset{B3}{=} a \cdot (1 + b) \overset{B1,P2}{=} a \cdot 1 \overset{B4}{=} a \,.
$$

The other properties, P5–P7, can be derived similarly. □

Note. An immediate corollary of P5 in the above theorem is that $0' = 1$ and $1' = 0$ hold on any Boolean algebra $(S, +, \cdot, ', 0, 1)$.

Example 2

Example

Switching system $(S, +, \cdot, ', 0, 1)$ with $S = \{0, 1\}$ is a Boolean algebra.

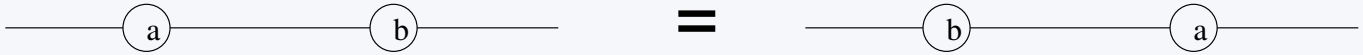
Example 2

Example

Switching system $(S, +, \cdot, ', 0, 1)$ with $S = \{0, 1\}$ is a Boolean algebra.

Solution. We need to show B1 – B5, by letting "+", "$\cdot$" and "$'$" specifically denote switches in parallel, in series and in complementation respectively. The identities in B1 are valid because

i.e. a + b = b + a



i.e. a. b = b . a

- The proof of B2 – B5 is elementary.

Example 2

- Hence we'll simply show only the first half of B3, i.e.
  $a + (b \cdot c) = (a + b) \cdot (a + c)$.

a	b	c	$b \cdot c$	$a + b$	$a + c$	$a + (b \cdot c)$	$(a + b) \cdot (a + c)$
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

all corresponding values exactly same

Example 2

- Hence we'll simply show only the first half of B3, i.e. $a + (b \cdot c) = (a + b) \cdot (a + c)$.
- We'll show the identity by the use of the evaluation table below

a	b	c	$b \cdot c$	$a + b$	$a + c$	$a + (b \cdot c)$	$(a + b) \cdot (a + c)$
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

all corresponding values exactly same

Example 3

Example

Let $\mathbb{R}$ be the set of real numbers, $+, \times$ be the normal addition and multiplication, $0, 1 \in \mathbb{R}$ be the normal numbers. If we define $'$ by $x' = 1 - x$ for any $x \in \mathbb{R}$, then is $(\mathbb{R}, +, \times, ', 0, 1)$ a Boolean algebra?

Example 3

Example

Let $\mathbb{R}$ be the set of real numbers, $+, \times$ be the normal addition and multiplication, $0, 1 \in \mathbb{R}$ be the normal numbers. If we define $'$ by $x' = 1 - x$ for any $x \in \mathbb{R}$, then is $(\mathbb{R}, +, \times, ', 0, 1)$ a Boolean algebra?

Solution. No. Because neither the first half of B3 nor the second half of B5 is satisfied. For example,

$$1 + (2 \times 3) \neq (1 + 2) \times (1 + 3).$$

Example 4

Example

Suppose $S = \{1, 2, 3, 5, 6, 10, 15, 30\}$. Let $a + b$ denote the least common multiple of a and b, $a \cdot b$ denote the greatest common divisor of a and b, and $a' = \dfrac{30}{a}$. Prove $(S, +, \cdot, ', 1, 30)$ is a Boolean algebra.

Algebraic Equivalence

- Switching systems and symbolic logic are essentially the same.

Algebraic Equivalence

- Switching systems and symbolic logic are essentially the same.
- Furthermore they both form a Boolean algebra $(S, +, \bullet, ', \mathbf{0}, \mathbf{1})$ on a set $S = \{\mathbf{0}, \mathbf{1}\}$, where

Algebraic Equivalence

- Switching systems and symbolic logic are essentially the same.
- Furthermore they both form a Boolean algebra $(S, +, \bullet, ', \mathbf{0}, \mathbf{1})$ on a set $S = \{\mathbf{0}, \mathbf{1}\}$, where
  - $\mathbf{0}$ is `open` in the switching systems and is F in the symbolic logic, while

Algebraic Equivalence

- Switching systems and symbolic logic are essentially the same.
- Furthermore they both form a Boolean algebra $(S, +, \bullet, ', \mathbf{0}, \mathbf{1})$ on a set $S = \{\mathbf{0}, \mathbf{1}\}$, where
  - $\mathbf{0}$ is `open` in the switching systems and is F in the symbolic logic, while
  - $\mathbf{1}$ is `closed` in the switching systems and is T in the symbolic logic.

Algebraic Equivalence

- Switching systems and symbolic logic are essentially the same.
- Furthermore they both form a Boolean algebra $(S, +, \bullet, ', \mathbf{0}, \mathbf{1})$ on a set $S = \{\mathbf{0}, \mathbf{1}\}$, where
  - $\mathbf{0}$ is `open` in the switching systems and is F in the symbolic logic, while
  - $\mathbf{1}$ is `closed` in the switching systems and is T in the symbolic logic.
- The correspondence can be seen in the following table

Algebraic Equivalence

- Switching systems and symbolic logic are essentially the same.
- Furthermore they both form a Boolean algebra $(S, +, \bullet, ', \mathbf{0}, \mathbf{1})$ on a set $S = \{\mathbf{0}, \mathbf{1}\}$, where
  - $\mathbf{0}$ is `open` in the switching systems and is F in the symbolic logic, while
  - $\mathbf{1}$ is `closed` in the switching systems and is T in the symbolic logic.
- The correspondence can be seen in the following table

Algebraic Equivalence

- Switching systems and symbolic logic are essentially the same.
- Furthermore they both form a Boolean algebra $(S, +, \bullet, ', \mathbf{0}, \mathbf{1})$ on a set $S = \{\mathbf{0}, \mathbf{1}\}$, where
  - **0** is `open` in the switching systems and is F in the symbolic logic, while
  - **1** is `closed` in the switching systems and is T in the symbolic logic.
- The correspondence can be seen in the following table

Boolean algebra	Switching systems	Symbolic logic
$S = \{\mathbf{0}, \mathbf{1}\}$	{open, closed}	$\{F, T\}$
$+$	$x + y$ (in parallel)	$p \vee q$ ("or")
$\bullet$	$x \cdot y$ (in series)	$p \wedge q$ ("and")
$'$	x' (complement)	$\sim p$ ("not")
$\mathbf{0}$	circuit open	contradiction
$\mathbf{1}$	circuit closed	tautology

Algebraic Equivalence

- Hence properties B1–B5 and P1–P7 will also hold in symbolic logic, when "$+$", "$\cdot$", "$'$", "0" and "1" are replaced by "$\vee$", "$\wedge$", "$\sim$", *contradiction* and *tautology* respectively.

Algebraic Equivalence

- Hence properties B1–B5 and P1–P7 will also hold in symbolic logic, when "+", "·", "′", "0" and "1" are replaced by "$\vee$", "$\wedge$", "$\sim$", *contradiction* and *tautology* respectively.
  - Hence, for example,

Algebraic Equivalence

- Hence properties B1–B5 and P1–P7 will also hold in symbolic logic, when "$+$", "$\cdot$", "$\prime$", "0" and "1" are replaced by "$\vee$", "$\wedge$", "$\sim$", *contradiction* and *tautology* respectively.
  - Hence, for example,

Algebraic Equivalence

- Hence properties B1–B5 and P1–P7 will also hold in symbolic logic, when "+", "·", "′", "0" and "1" are replaced by "$\vee$", "$\wedge$", "$\sim$", *contradiction* and *tautology* respectively.
  - Hence, for example,

$$a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c), \quad a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c),$$

Algebraic Equivalence

- Hence properties B1–B5 and P1–P7 will also hold in symbolic logic, when "$+$", "$\cdot$", "$'$", "0" and "1" are replaced by "$\vee$", "$\wedge$", "$\sim$", *contradiction* and *tautology* respectively.
  - Hence, for example,

$$a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c), \quad a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c),$$

$$a \vee a \equiv a, \quad a \wedge a \equiv a, \quad \sim(\sim a) \equiv a, \quad a \vee \perp \equiv a, \quad a \wedge \top \equiv a,$$

Algebraic Equivalence

- Hence properties B1–B5 and P1–P7 will also hold in symbolic logic, when "+", "·", "′", "0" and "1" are replaced by "$\vee$", "$\wedge$", "$\sim$", *contradiction* and *tautology* respectively.
  - Hence, for example,

$$a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c), \quad a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c),$$

$$a \vee a \equiv a, \quad a \wedge a \equiv a, \quad \sim(\sim a) \equiv a, \quad a \vee \bot \equiv a, \quad a \wedge \top \equiv a,$$

$$a \vee (a \wedge b) \equiv a, \quad a \wedge (\sim a) \equiv \bot, \quad a \vee (\sim a) \equiv \top, \quad a \wedge \bot \equiv \bot, \quad a \vee \top \equiv \top$$

hold for any propositions a, b and c, where $\bot$ represents a contradiction and $\top$ represents a tautology.

Example 5

Example

Convert $(p \vee q) \to r$ into the corresponding Boolean expression.

Example 5

Example

Convert $(p \vee q) \to r$ into the corresponding Boolean expression.

Solution. Since $p \to q$ is equivalent to $(\sim p) \vee q$,
we see that $(p \vee q) \to r$ is equivalent to $(\sim(p \vee q)) \vee r$
which is thus converted to $(p+q)' + r$.

Example 6 (De Morgan's Laws)

Example

Let $(S, +, \cdot, ', 0, 1)$ be a Boolean algebra, then for any $x, y \in S$

$$(x + y)' = x' \cdot y',$$

$$(x \cdot y)' = x' + y'.$$

Example 6 (De Morgan's Laws)

Example

Let $(S, +, \cdot, ', 0, 1)$ be a Boolean algebra, then for any $x, y \in S$

$$(x + y)' = x' \cdot y',$$

$$(x \cdot y)' = x' + y'.$$

Solution. Proof obvious from the theorem in the previous section.

Example 7

Example

Suppose $(\mathbb{T}, \clubsuit, \diamondsuit, \heartsuit, \sqcup, \sqcap)$ is a Boolean algebra. Show $(\sqcup^{\heartsuit})\clubsuit\sqcup = \sqcap$.

Example 7

Example

Suppose $(\mathbb{T}, \clubsuit, \diamondsuit, \heartsuit, \sqcup, \sqcap)$ is a Boolean algebra. Show $(\sqcup^{\heartsuit})\clubsuit\sqcup = \sqcap$.

Solution. Recall that when we say $(\mathbb{T}, \clubsuit, \diamondsuit, \heartsuit, \sqcup, \sqcap)$ is a Boolean algebra, the tuple *means* that
$\clubsuit$, $\diamondsuit$ and $\heartsuit$ correspond respectively to the
"$+$", "$\cdot$" and "$\prime$" operations entailed by a Boolean algebra, and that
$\sqcup$ and $\sqcap$ correspond respectively to the
"0" and "1" elements possessed by the Boolean algebra.
Hence $(\sqcup^{\heartsuit})\clubsuit\sqcup = \sqcap$ is the same as $(0') + 0 = 1$ and is thus obviously true.

Example 8

Example

Let $B = \{0, 1\}$ and $(B, +, \cdot, ', 0, 1)$ be a Boolean algebra.
Let B^n denote the set of all the tuples $(x_1, x_2, ..., x_n)$ with $x_1, ..., x_n$ being any elements of B, i.e.

$$B^n \stackrel{\text{def}}{=} \{(x_1, ..., x_n) \,|\, x_1 \in B, ..., x_n \in B\}$$

Then $(B^n$, $+$, $\bullet$, **'**, $\mathbf{0}$, $\mathbf{1})$ is also a Boolean algebra if

$$\begin{aligned}
\mathbf{0} &= \overbrace{(0, \cdots, 0)}^{n \;\; 0's}\,, \\
\mathbf{1} &= (1, \cdots, 1)\,, \\
\mathbf{x} + \mathbf{y} &= (x_1 + y_1, \cdots, x_n + y_n)\,, \\
\mathbf{x} \bullet \mathbf{y} &= (x_1 \cdot y_1, \cdots, x_n \cdot y_n)\,, \\
\mathbf{x'} &= (x_1', \cdots, x_n')\,.
\end{aligned}$$

Sets connection with Boolean Algebra

- If we make the correspondence between

$$\varnothing,\, U,\, \cup,\, \cap,\, ' \quad \text{and} \quad 0,\, 1,\, +,\, \cdot,\, '$$

  for sets and Boolean algebra respectively, we see that properties S1–S5 are exactly those B1–B5 for the definition of Boolean algebra.

Sets connection with Boolean Algebra

- If we make the correspondence between

$$\varnothing,\, U,\, \cup,\, \cap,\, ' \quad \text{and} \quad 0,\, 1,\, +,\, \cdot,\, '$$

  for sets and Boolean algebra respectively, we see that properties S1–S5 are exactly those B1–B5 for the definition of Boolean algebra.
- Hence for any nonempty set S, for instance, $(\mathcal{P}(S), \cup, \cap, ', \varnothing, S)$ is a Boolean algebra.

Sets connection with Boolean Algebra

- If we make the correspondence between

$$\varnothing,\ U,\ \cup,\ \cap,\ ' \quad \text{and} \quad 0,\ 1,\ +,\ \cdot,\ '$$

  for sets and Boolean algebra respectively, we see that properties S1–S5 are exactly those B1–B5 for the definition of Boolean algebra.
- Hence for any nonempty set S, for instance, $(\mathcal{P}(S), \cup, \cap, ', \varnothing, S)$ is a Boolean algebra.

Sets connection with Boolean Algebra

- If we make the correspondence between

$$\varnothing,\ U,\ \cup,\ \cap,\ ' \quad \text{and} \quad 0,\ 1,\ +,\ \cdot,\ '$$

  for sets and Boolean algebra respectively, we see that properties S1–S5 are exactly those B1–B5 for the definition of Boolean algebra.
- Hence for any nonempty set S, for instance, $(\mathcal{P}(S), \cup, \cap, ', \varnothing, S)$ is a Boolean algebra.

Theorem

If $\mathcal{B}$ is a Boolean algebra with exactly n elements then $n = 2^m$ for some m. Furthermore $\mathcal{B}$ and $(\mathcal{P}(\{1, 2, \cdots m\}), \cup, \cap, ', \varnothing, \{1, 2, \cdots, m\})$ essentially represent the same Boolean algebra.

Sets connection with Boolean Algebra

- If we make the correspondence between

$$\varnothing,\ U,\ \cup,\ \cap,\ ' \quad \text{and} \quad 0,\ 1,\ +,\ \cdot,\ '$$

  for sets and Boolean algebra respectively, we see that properties S1–S5 are exactly those B1–B5 for the definition of Boolean algebra.
- Hence for any nonempty set S, for instance, $(\mathcal{P}(S), \cup, \cap, ', \varnothing, S)$ is a Boolean algebra.

Theorem

If $\mathcal{B}$ is a Boolean algebra with exactly n elements then $n = 2^m$ for some m. Furthermore $\mathcal{B}$ and $(\mathcal{P}(\{1, 2, \cdots m\}), \cup, \cap, \prime, \varnothing, \{1, 2, \cdots, m\})$ essentially represent the same Boolean algebra.

Example

9. Let $U = \{2, 3\}$ and $S = \mathcal{P}(U)$. Then $(S, \cup, \cap, ', \varnothing, U)$ is a Boolean algebra. Notice that the set S in this case contains more than 2 elements.