w04-Lec

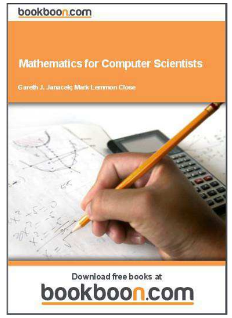
Mathematics and Computer Science: Boolean Algebra

Assembled for 204111

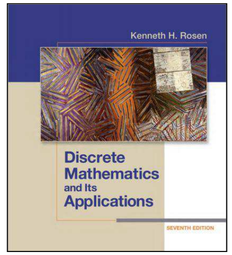
by Areerat Trongratsameethong

Boolean Algebra

- Mathematics for Computer Scientists – Janacek and Close
  - Introduction to Logical Value
  - Logical Operators



- Discrete Mathematics and Its Applications – K.H. Rosen
  - Introduction to Boolean Algebra
  - Rules of Precedence for Boolean Operators
  - Boolean Properties
  - Boolean Expression Simplification



Introduction to Logical Value

- A statement has its logical value, either true (T) or false (F)
  - "it is raining": may be true or false
  - "it is sunny": may be true or false
- Many statements can be combined with *and* and *or* to be a compound statement.
  - "it is raining and it is sunny"
  - "it is raining or it is sunny"
  - The logical values of the compound statements depend on the logical value of each combined statement and what connective ("and" or "or") is used.

Logical Operators

- Three logical operators are used in Boolean Algebra
  - Negation (not)
  - Conjunction (and)
  - Disjunction (or)

Logical Operators [2]

- Symbolic Notation: is used to make things shorter
  - Negation denoted by $\neg$
  - "and" denoted by $\wedge$
  - "or" denoted by $\vee$
- A symbolic can be also used for a statement
  - *p* can be used for "it is raining"

Logical Operators [3]

- Negation $\neg$
  - The negation of a statement is false when the statement is true.
  - The negation of a statement is true when the statement is false.
- Example
  - Let *p* = "It is raining",
    then $\neg p$ is "it is not raining"
  - If the logical value of *p* is F
    then logical value of $\neg p$ is T

Logical Operators [4]

- The logical values can be represented in form of truth table as an example below.

Example

Let *p* = "All computer scientists are men"

Two possible logical values of *p* are T and F

p	$\neg p$
T	F
F	T

Table 2.1: Truth table for negation ($\neg$)

Logical Operators [5]

- Conjunction $\wedge$
- If *p* and *q* are statements, then
  $p \wedge q$ is read as "*p* and *q*".
  - Let
    - *p* = "It is green",
    - *q* = "It is an apple" then

    $p \wedge q$ = "It is green and It is an apple"
  - The logical value of $p \wedge q$ depends on each logical value of *p* and *q* as shown in Table 2.2.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Table 2.2: Truth table for $\wedge$

Logical Operators [6]

- **Disjunction ∨**
- **If *p* and *q* are statements, then**

  $p \vee q$ is read as "*p* or *q*".
  - **Let**
    - **p = "It is green",**
    - **q = "It is an apple" then**

  $p \vee q$ = "It is green or It is an apple"
  - **The logical value of $p \vee q$ depends on each logical value of *p* and *q* as shown in Table 2.3.**

p	*q*	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Table 2.3: Truth table for ∨

Introduction to Boolean Algebra

- **A Symbolic form of Aristotle's system of logic sought by George Boole (1815-1864) - The English mathematician**
- **Mathematical language dealing with the questions of logic**
- **An Investigation of the Laws of Thought (Boole 1854),**
  - **Theories of Logic and Probabilities**
  - **Mathematical Relationship Quantities Rule**
    - **true or false**
    - **1 or 0**

Introduction to Boolean Algebra [2]

- **The mathematical system written by Boole became known as Boolean algebra.**
- **All Boolean quantities have two possible outcomes: 1 or 0.**
- **There is no such thing as "2" or "-1" or "1/2" in the Boolean world.**

Introduction to Boolean Algebra [3]

- **Boolean algebra as on-and-off circuits Control**
- **All signals are characterized as either "high" (1) or "low" (0).**
- **A Symbolic Analysis of Relay and Switching Circuits – MIT Thesis (Shannon 1938)**
  - **Mathematical tool for designing and analyzing digital circuits.**
  - **Defined the circuits in all electronic devices as 1 or 0 referring 'on' or in 'off' position.**

Introduction to Boolean Algebra [4]

- **Boolean algebra provides the operations and the rules for working with the set {0,1}.**
- **Operation for a circuit is called Boolean Function.**
- **Boolean Function produce output for each set of inputs.**
- **This function is built using Boolean expressions and operations.**

Rules of Operator Precedence

high

- **Order of Boolean Operators**
  1. **Complement (or Negation $\neg$) denoted by ~**
  2. **Boolean Product denoted by $\wedge$**
  3. **Boolean Sum denoted by $\vee$**

low

Rules of Operator Precedence [2]

Example: Find the value of $1 \wedge 0 \vee \sim (0 \vee 1)$

$$1 \wedge 0 \vee \sim (\underline{0 \vee 1})$$

$$1 \wedge 0 \vee \underline{\sim 1}$$

$$\underline{1 \wedge 0} \vee 0$$

$$0 \vee 0$$

$$0$$

Boolean Properties

- **The simpler that we can make a Boolean function, the smaller the circuit that will result.**
- **Simpler Circuits**
  - **Cheaper to build**
  - **Consume less power**
  - **Run faster than the complex circuits**
- **With this in mind, we always want to reduce our Boolean functions to their simplest form.**
- **There are a number of <u>Boolean identities that help us to do this</u>.**

Rules of Boolean Algebra

- **Most Boolean identities have an AND (product) form as well as an OR (sum) form.**
- **We give our identities using both forms.**

Identity Name	AND	OR
Identity Law	$1 \wedge x = x$	$0 \vee x = x$
Null Law	$0 \wedge x = 0$	$1 \vee x = 1$
Idempotent Law	$x \wedge x = x$	$x \vee x = x$
Inverse Law	$x \wedge \sim x = 0$	$x \vee \sim x = 1$

x	1	$1 \wedge x$
0	1	0
1	1	1

Truth table of $1 \wedge x$

How about the others?

Note: These laws can be proved by truth table.

Rules of Boolean Algebra [2]

Identity Name	AND
Commutative Law	$x \wedge y = y \wedge x$
Associative Law	$(x \wedge y) \wedge z = x \wedge (y \wedge z)$
Distributive Law	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$

Identity Name	OR
Commutative Law	$x \vee y = y \vee x$
Associative Law	$(x \vee y) \vee z = x \vee (y \vee z)$
Distributive Law	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

Remark: The above identities can be translated to logical equivalences about propositions and to identities about sets.

Rules of Boolean Algebra [3]

Identity Name	AND	OR
Absorption Law	$x \wedge (x \vee y) = x$	$x \vee (x \wedge y) = x$
De Morgan's Law	$\sim(x \wedge y) = \sim x \vee \sim y$	$\sim(x \vee y) = \sim x \wedge \sim y$
Double Complement Law	$\sim(\sim x) = x$	

Proof – Absorption Law:

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

Rewrite: $\wedge \rightarrow \cdot$, $\vee \rightarrow +$

$$\begin{aligned} x \cdot (x + y) &= x \cdot x + (x \cdot y) \\ &= x + (x \cdot y) \\ &= (x \cdot 1) + (x \cdot y) \\ &= x \cdot (1 + y) \\ &= x \cdot 1 \\ &= x \end{aligned}$$

Proof – Absorption Law:

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

Rewrite: $\wedge \rightarrow \cdot$, $\vee \rightarrow +$

$$\begin{aligned} x + (x \cdot y) &= (x \cdot 1) + (x \cdot y) \\ &= x \cdot (1 + y) \\ &= x \cdot 1 \\ &= x \end{aligned}$$

Boolean Expression Simplification

Example 1: $(x \vee y) \wedge (x \vee \sim y) \wedge \sim(x \wedge \sim z)$

1	$(x \vee y) \wedge (x \vee \sim y) \wedge \sim(x \wedge \sim z)$	
2	$(x \vee y) \wedge (x \vee \sim y) \wedge (\sim x \vee z)$	De Morgan's Law
3	$(x \wedge x) \vee (x \wedge \sim y) \vee (y \wedge x) \vee (y \wedge \sim y) \wedge (\sim x \vee z)$	Distributive Law
4	$x \vee (x \wedge \sim y) \vee (y \wedge x) \vee 0 \wedge (\sim x \vee z)$	Idempotent and Inverse Laws
5	$x \vee (y \wedge x) \wedge (\sim x \vee z)$	Absorption and Identity Laws
6	$x \wedge (\sim x \vee z)$	Absorption Law
7	$(x \wedge \sim x) \vee (x \wedge z)$	Distributive Law
8	$0 \vee (x \wedge z)$	Inverse Law
9	$x \wedge z$	Identity Law

หมายเหตุ: ในขั้นตอนที่ 3 สามารถดำเนินการได้อีกวิธี อย่างไร?

Boolean Expression Simplification [2]

- **Example 2: Find the <u>complement</u> of the Boolean expression below. (De Morgan's Law)**

$$(x \wedge y) \vee (\sim x \wedge z) \vee (y \wedge \sim z)$$
$$\underline{\sim((x \wedge y) \vee (\sim x \wedge z) \vee (y \wedge \sim z))}$$
$$\underline{\sim(x \wedge y)} \wedge \underline{\sim (\sim x \wedge z)} \wedge \underline{\sim (y \wedge \sim z)}$$
$$(\sim x \vee \sim y) \wedge (x \vee \sim z) \wedge (\sim y \vee z)$$

Summary

- **Introduction to Logical Value**
- **Logical Operators**
- **Introduction to Boolean Algebra**
- **Rules of Precedence for Boolean Operators**
- **Boolean Properties**
- **Boolean Expression Simplification**

References

- **Mathematics for Computer Scientists, Janacek and Close**
- **Discrete Mathematics and Its Applications, K.H. Rosen**