Lec 21: Boolean Algebra I

Prof. Adam J. Aviv

GW

CSCI 1311 Discrete Structures I
Spring 2020

Boolean Algebra

There are a lot of similarities between logical equivalences and set equivalences.

Logical Equivalences	Set Properties
For all statement variables p, q, and r:	For all sets A, B, and C:
a. $p \vee q \equiv q \vee p$ b. $p \wedge q \equiv q \wedge p$	a. $A \cup B = B \cup A$ b. $A \cap B = B \cap A$
a. $p \wedge (q \wedge r) \equiv p \wedge (q \wedge r)$ b. $p \vee (q \vee r) \equiv p \vee (q \vee r)$	a. $A \cup (B \cup C) \equiv A \cup (B \cup C)$ b. $A \cap (B \cap C) \equiv A \cap (B \cap C)$
a. $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ b. $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	a. $A \cap (B \cup C) \equiv (A \cap B) \cup (A \cap C)$ b. $A \cup (B \cap C) \equiv (A \cup B) \cap (A \cup C)$
a. $p \vee \mathbf{c} \equiv p$ b. $p \wedge \mathbf{t} \equiv p$	a. $A \cup \emptyset = A$ b. $A \cap U = A$
a. $p \vee {\sim}p \equiv \mathbf{t}$ b. $p \wedge {\sim}p \equiv \mathbf{c}$	a. $A \cup A^c = U$ b. $A \cap A^c = \emptyset$
${\sim}({\sim}p) \equiv p$	$(A^c)^c = A$
a. $p \vee p \equiv p$ b. $p \wedge p \equiv p$	a. $A \cup A = A$ b. $A \cap A = A$
a. $p \vee \mathbf{t} \equiv \mathbf{t}$ b. $p \wedge \mathbf{c} \equiv \mathbf{c}$	a. $A \cup U = U$ b. $A \cap \emptyset = \emptyset$
a. ${\sim}(p \vee q) \equiv {\sim}p \wedge {\sim}q$ b. ${\sim}(p \wedge q) \equiv {\sim}p \vee {\sim}q$	a. $(A \cup B)^c = A^c \cap B^c$ b. $(A \cap B)^c = A^c \cup B^c$
a. $p \vee (p \wedge q) \equiv p$ b. $p \wedge (p \vee q) \equiv p$	a. $A \cup (A \cap B) \equiv A$ b. $A \cap (A \cup B) \equiv A$
a. ${\sim}\mathbf{t} \equiv \mathbf{c}$ b. ${\sim}\mathbf{c} \equiv \mathbf{t}$	a. $U^c = \emptyset$ b. $\emptyset^c = U$

Table 6.4.1

Boolean Algebra

Definition

A Boolean Algebra is a mathematical construct $(B, +, \cdot, ', 0, 1)$ where B is a non-empty set of elements, $+$ and $\cdot$ are binary operators over elements of B, $'$ is a unary operator over B, and 0 and 1 are special *identities* in B.

Properties of a Boolean Algebra

- + and · are commutative
  - $\forall x, y \in B, \quad x + y = y + x$
  - $\forall x, y, \in B, \quad x \cdot y = y \cdot x$
- + and · are associative
  - $\forall x, y, z \in B, \quad x + (y + z) = (x + y) + z$
  - $\forall x, y, z \in B, \quad (x \cdot y) \cdot z = x \cdot (y \cdot z)$
- + and · are distributive over one another
  - $\forall x, y, z \in B, \quad x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
  - $\forall x, y, z, \in B, \quad x + (y \cdot z) = (x + y) \cdot (x + z)$
- Identity Laws
  - $\forall x \in B, \quad x + 0 = x$
  - $\forall x \in B, \quad x \cdot 1 = x$
- Complements Laws
  - $\forall x \in B, \quad x + x' = 1$
  - $\forall x \in B, \quad x \cdot x' = 0$

Propositional Logic as a Boolean Algebra

$$\text{Propositional Logic} = (\{\text{set of all propositions}\}, \wedge, \vee, \neg, \mathbf{t}, \mathbf{c})$$

Let p, q, r be propositional statements, then

- $\wedge$ and $\vee$ is commutative:
  $p \vee q \equiv q \vee p$ and $p \wedge q \equiv q \wedge p$
- $\wedge$ and $\vee$ is associative:
  $(p \vee q) \vee r \equiv p \vee (q \vee r)$ and $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
- $\wedge$ and $\vee$ is distributive:
  $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ and $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
- $\wedge$ and $\vee$ is conservative under identities:
  $p \vee \mathbf{c} \equiv p$ and $p \wedge \mathbf{t} \equiv p$
- $\wedge$ and $\vee$ produce identities under complements:
  $p \vee \neg p \equiv \mathrm{t}$ and $p \wedge \neg p \equiv \mathbf{c}$

Set Operations as a Boolean Algebra

$$\text{Set Operators} = (2^U, \cup, \cap, ^c, \varnothing, U)$$

Let $A, B, r \in B$ be sets in the powerset of the universe U (i.e., 2^U), then

- $\cap$ and $\cup$ is commutative:
  $A \cup B \equiv B \cup A$ and $A \cap B \equiv B \cap A$
- $\cap$ and $\cup$ is associative:
  $(A \cup B) \cup C \equiv A \cup (B \cup C)$ and $(A \cap B) \cap C \equiv A \cap (B \cap C)$
- $\cap$ and $\cup$ is distributive:
  $A \cup (B \cap r) \equiv (A \cup B) \cap (A \cup C)$ and $A \cap (B \cup C) \equiv (A \cap B) \cup (A \cap C)$
- $\cap$ and $\cup$ is conservative under identities:
  $A \cup \varnothing \equiv A$ and $A \cap U \equiv A$
- $\cap$ and $\cup$ Aroduce identities under comAlements:
  $A \cup A^c \equiv U$ and $A \cap A^c \equiv \varnothing$

Uniqueness of Compliments

Theorem

For all elements $x \in B$, there is a unique complement of x. Formally

$$(\forall x, a_1, a_2 \in B)(x' = a_1 \wedge x' = a_2 \implies a_1 = a_2)$$

From the hypothesis of the implication, we have that $x' = a_1$ and $x' = a_2$, and by the complement law $x \cdot a_1 = 0$ and $x + a_1 = 1$ and $x \cdot a_2 = 0$ and $x + a_2 = 1$ because x and a_1, a_2 are complements.

$$\begin{aligned}
a_1 &= a_1 & \\
&= a_1 \cdot (x + a_2) & \text{by identity} \\
&= (a_1 \cdot x) + (a_1 \cdot a_2) & \text{by distributive} \\
&= 0 + (a_1 \cdot a_2) & \text{by hypothesis} \\
&= (x \cdot a_2) + (a_1 \cdot a_2) & \text{by hypothesis} \\
&= a_2 \cdot (x + a_1) & \text{by distributive} \\
&= a_2 \cdot 1 & \text{by hypothesis} \\
&= a_2 & \text{by identity}
\end{aligned}$$

Exercise

Prove the following is true for any Boolean Algebra where $x, y \in B$

- $x + x = x$ and $x \cdot x = x$
- $x + 1 = 1$ and $x \cdot 0 = 0$
- $x + (x \cdot y) = x$ and $x \cdot (x + y) = x$

DeMorgan's Law

DeMorgan's law also applies to all Boolean algebras

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

Proving DeMorgan's Law

Consider that the inverse of $x + y$ is $(x + y)' = x' \cdot y'$ by DeMorgan's Law. If that is so, then $(x + y) + (x' \cdot y) = 1$ and $(x + y) \cdot (x' \cdot y') = 0$ by the law of complements.

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

$$\begin{aligned}
0 &= (x + y) \cdot (x' \cdot y') \\
&= (x \cdot (x' \cdot y)) + (y \cdot (x' \cdot y')) \\
&= ((x \cdot x') \cdot y) + ((y \cdot y') \cdot x') \\
&= (0 \cdot y) + (0 \cdot x') \\
&= 0 + 0 \\
&= 0
\end{aligned}$$

Boolean Expressions

Binary Boolean Algebra of Digital Logic

To simplify our discussion going forward, we will focus on the Boolean algebra of Digital Logic.

- $B = \{0, 1\}$
- $+$: logical OR
- $\cdot$: logical AND
- $'$: logical NOT (or negation)
- 1 : 1 (logical true)
- 0 : 0 (logical false)

And, Or, Not – again!

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

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

x	x'
0	1
1	0

Definition

A boolean variable, like x or y, is a variable (placeholder) where the value is an element from the set B of the Boolean algebra.

Boolean Expressions and Variables

Definition

A boolean expression is any string that can be derived from the following rules and no other rules:

- 0 and 1 are Boolean expressions
- Any boolean variable is a Boolean expression
- If e and f are Boolean expressions, then e, $e + f$, $e \cdot f$, e' are all Boolean expressions.

Example Boolean Expressions

- $x + y \quad x' + y$, $x \cdot (y + z')$ and $x \cdot y + z'$
- $xyz + x'yz' + xyz' + (x + y)(x' + z)$

The following are *not* Boolean expressions

- x/y
- x^y

Boolean Functions

Definition

A Boolean function of n variables is a function

$$f : B^n \to B$$

where $f(x_1, x_2, \ldots, x_n)$ is a boolean expression in terms of $x_1, x_2, \ldots, x_n$

Examples:

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

$$g(x, y, z, w) = (x + 6 + z')(x' + y' + w) + xyw'$$

Equivalences of Boolean Expressions

Like in prepositional logic, there are two ways to show that two boolean expressions (or functions) are equivalent

1. Equivalent Statements: You can apply successive rules, theorems or axioms to show that they are the same equatoin.

2. Truth tables: Show that for every value of the variables of the expression/function, it evaluates the same

Truth tables of Boolean Functions

We can use truth tables to show the possible inputs and outcomes of evaluating a Boolean function/expression

x	y	$x \cdot y + x' \cdot y$
0	0	1
0	1	0
1	0	0
1	1	1

But what if we only had a truth table, can we find a boolean expression?

Minterms

Definition

A minterm is any product of n literals/Boolean-variables where each of the n variables appears once in the product.

For example, if an expression/function contains 3 variabls then

$$xy'z \quad x'yz \quad xyz$$

are all minterms of a 3 variable expression/function.

How many minters exist for a n-variable expression?

Finding a Boolean Expression from a Truth Table

Suppose you had the following truth table where f is the output of a function for inputs x, y, and z.

x	y	z	f	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$x' \cdot y \cdot z$
1	0	0	0	
1	0	1	1	$x \cdot y' \cdot z$
1	1	0	1	$x \cdot y \cdot z'$
1	1	1	1	$x \cdot y \cdot z$

Every row of the truth table, if f evaluates to 1, has a unique minterm that would evaluate to 1 in that arrangement of x, y, and z. Thus, to capture all the possible minterms, we can sum them up. Resulting int the following formula:

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

Disjunctive Normal Form (DNF)

Definition

Disjunctive normal form (or minterm canonical form) is a function/expression of n Boolean variables that is the sum of minterms.

The function we found before is in DNF

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

Every Boolean function can be expressed in DNF

Every boolean expression or boolean function of n variables can *always* be expressed in DNF. Why?

Every boolean function can be expressed as a truth table where each row maps to a unique minterm that would evaluate to 1 if that setting of variables in the row of the truth table evaluated to 1.

The sum of all the minterms, using logical or, requires only one minterm to evaluate to 1 for the entire expression to evaluate to 1. Thus the sum of all the minterms will be 1 whenever the input, as matched in the truth table, evaluates to 1.

Maxterms

definition

A maxterm for a Boolean expression or function of n terms is the sum where all n variables/literals appear.

For example

$$x + y + z' \qquad x' + y + z \qquad x + y' + z'$$

are all maxterms.

Conjunctive Normal Form (CNF)

Definition

Conjunctive normal form (or maxterm canonical form) is a function/expression of n Boolean variabls that is the product of maxterms.

The following function of 3 boolean variabls is in CNF

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

Finding CNF from a truth table

You can always derive CNF of a Boolean expression from a truth table, much in the same way you can for DNF, but instead you consider the cases where the output is 0, or when f' is 1.

x	y	z	f	f'		
0	0	0	0	1	$x' \cdot y' \cdot z'$	$= (x+y+z)'$
0	0	1	0	1	$x' \cdot y' \cdot z$	$= (x+y+z')'$
0	1	0	0	1	$x' \cdot y \cdot z'$	$= (x+y'+z)'$
0	1	1	1	0		
1	0	0	0	1	$x \cdot y' \cdot z'$	$= (x'+y+z)'$
1	0	1	1	0		
1	1	0	1	0		
1	1	1	1	0		

The complement of the function, when it's 0, provides a set of *minterms*, which can be written in DNF. Taking the complement of the complement, reverts the expression to its original evaluation, and by DeMorgan's law, leaving a product of maxterms.

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

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

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

Exercise

Find the DNF and CNF form of the function $f(x,y,z)$ from the truth table

x	y	z	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1