Ch. VII

BOOLEAN ALGEBRA

Boolean Algebra is a powerful mathematical tool for logical circuits design. It takes its name from its founder George Bool in the 19th century

7.1 Boolean Expressions:

A Boolean Expression in n variables $x_1, x_2, x_3, \dots, x_n$ is any finite string of symbols formed by applying the following rules: -1- $x_1, x_2, \dots, x_n$ are Boolean expressions.

-2- If $\mathscr{X}$ and $\mathscr{Y}$ are Boolean expressions, so are $(\mathscr{X}+\mathscr{Y})$, $(\mathscr{X}\cdot\mathscr{Y})$ and $\mathscr{X}'$

In our context we build a Boolean Algebra on a set $B=\{0,1\}$ all together with the operations of disjunction, conjunction, and negation

"$\vee$" or "+"	0	1
0	0	1
1	1	1

"$\wedge$" "and" "$\cdot$"	0	1
0	0	0
1	0	1

x	x'
0	1
1	0

Also remember that always:

$x+0=x$, $x+1=1$

$x\cdot 0=0$, $x\cdot 1=x$

$x+x=x$, $x\cdot x'=0$

$x\cdot x=x$, $x+x'=1$

Some complex Boolean expressions:

$$x\cdot\left[x\cdot y+y'(x+y)'\right]$$

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

can apply truth table. And 2 expressions are said to be equivalent if they have the same truth table.

Theorem 7.1: Let x, y, and z be Boolean Variables; the following identities always hold:

-1- a) $x\cdot y=y\cdot x$ Commutative law b) $x+y=y+x$

-2- a) $x\cdot(y\cdot z)=(x\cdot y)\cdot z$ Associative law b) $x+(y+z)=(x+y)+z$

-3- a) $x\cdot(y+z)=x\cdot y+x\cdot z$ Distributive law b) $x+(y\cdot z)=(x+y)\cdot(x+z)$

-4-

Now using theorem 7.1. to show that

$$\left(x\cdot(y+z')\right)+(x\cdot y) = x\cdot(y+z')$$

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

<u>De Morgan's Laws</u>:

<u>Th 7.2</u> For all Boolean Expressions x and y:

-1- $(x + y)' = x'\cdot y'$

-2- $(x\cdot y)' = x' + y'$

<u>Th. 7.3</u> : If $x+y=1$ and $x\cdot y = 0 \Rightarrow x = y'$

proof: $x = x\cdot 1 = x\cdot(y+y')$

$$\Rightarrow x = x\cdot y + x\cdot y' = 0 + x\cdot y'$$
$$= y\cdot y' + x\cdot y' = y'(x+y) = y'$$
$$\Rightarrow \boxed{x = y'}$$

<u>Exercises</u>: 7-1. page 259. #5.

-a- $(p+q')'\cdot r' = p'\cdot q\cdot r'$

Use De Morgan's Law; $p'\cdot q\cdot r'$

-c- $\left((p+q')\cdot(r\cdot(p+q'))\right)' = p'\cdot q$

De Morgan Law:

$$(p+q')' + \left(r\cdot(p+q')\right)' = p'\cdot q + r' + (p+q')'$$
$$= p'\cdot q + r' + p'\cdot q = r' + p'\cdot q$$

<u>or</u>:

$$\left((p+q')\cdot(rp+rq')\right)' = (p+q')' + (rp+rq')' = p'\cdot q + (rp)'(q'\cdot r)'$$
$$= p'\cdot q + (r'+p')(q+r') = p'q + r' + p'\cdot q + r'(q+p')$$

-d- $x\cdot y\cdot x'\cdot y + x'\cdot y' + x\cdot y' = y'$

?

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

-e- $\underbrace{x\cdot y + x'\cdot y}_{1\cdot y} + \underbrace{x'\cdot y' + x\cdot y'}_{1\cdot y'} \stackrel{?}{=} 1$

$$= 1$$

7.2 Representation of Expressions:

What is the best way to write a Boolean Expression. We consider first by considering the different fcts of Boolean function -

$\forall n, \quad n \geq 1 \quad$, Let $F^n = \{ f \,/\, f : B^n \to B \}$, where $B = \{0, 1\}$.

Exple: List all the elements of F^1 and F^2

F^1: $\quad x \in B^1 = B = \{0, 1\}$.

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

for F^2:

x_1	x_2	f_0	f_1	f_2	f_3	f_4	.....	f_{14}	f_{15}
0	0	0	0	0	0	0		1	1
0	1	0	0	0	0	1		1	1
1	0	0	0	1	1	0		1	1
1	1	0	1	0	1	0		0	1

what is $f_1 + f_2$? $\qquad f_1 + f_2 = f_3$,

what is $f_1 + f_{14}$? $\qquad f_1 + f_{14} = f_{15}$,

what is $f_3 \cdot f_4$? $\qquad f_3 \cdot f_4 = f_0$

what is f_3'? $\qquad f_3' = f_{12}$

The # of fcts in F^n is 2^{2^n}

Minterms: since a large number of possible fct for few Boolean Variables, we look for a method of representing them in a canonical form, called set of minterms: a minterm is a Boolean fct that has exactly one 1 in its truth table.

Exple

x	y	z	$m(x,y,z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	

$\Rightarrow \; m(x,y,z) = 1$ only if $x = 0, y = 1, z = 1$

$\Rightarrow \; m(x,y,z) = x' \cdot y \cdot z$

This is the product representation of the minterm.

only conjunction and negation

Normal form: Any minterm can be written or expressed only in terms of "and" and "negation". The next step is to show that any Boolean fct can be written as a disjunction of minterms, "Sum of Products".

Theorem: Every Boolean fct can be expressed as a disjunction of minterms.

Exple:

x	y	z	$f(x,y,z)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$\Rightarrow f(x,y,z) = x'y'z' + x'yz + x.y'.z' + x.y.z$$

Complete Set of Operators: Any Boolean Algebra can be represented in terms of "+", ".", "'". All sets of operators that are sufficient to represent all Boolean ~~fct~~ expressions is called **Complete set of operators**

$\{+, \cdot, '\}$ is complete set of operators

$\{\cdot, '\}$ is also a complete set of operators.

NAND defined as not and is also a Complete set of operations

7.3 **Minimization of Boolean Expression:** Now, not only we are interested in finding a form of a set of product, ~~of~~ but we want a simpler expression that is equivalent to the original one. This can be done with the use of Karnaugh map.

~~Exple:~~ Karnaugh map w/ 3 variables

	xy	xy'	$x'y'$	$x'y$
z				
z'				

w/ 4 variable

	xy	$x'y$	$x'y'$	xy'
zw				
$z'w$				
$z'w'$				
zw'				

Exple: ① $f(x,y,z) = x'y'z' + x'yz + xy'z' + xyz' + xyz$

	xy	xy'	$x'y'$	$x'y$
z	1			1
z'		1	1	

$f(x,y,z) = yz + y'z'$

②

	xy	$x'y$	$x'y'$	xy'
z	1			1
z'	1			1

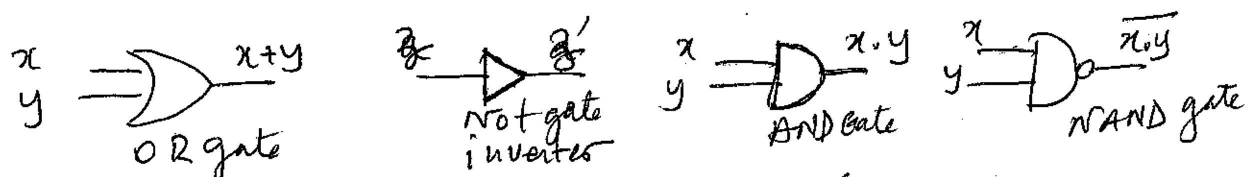
$f(x,y,z) = x$

	xy	xy'	$x'y'$	$x'y$
wz	1			1
wz'		1	1	
$w'z'$		1	1	
$w'z$	1			1

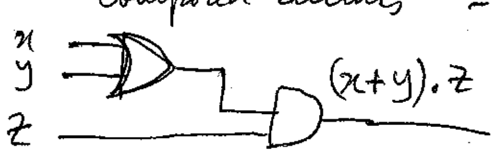
$f(x,y,z,w) = z'y' + yz$

7.4 Switching Theory: Application to binary devices. Combinational devices or networks.

Basic combinational circuit - Gates;



Compound circuits - logical network



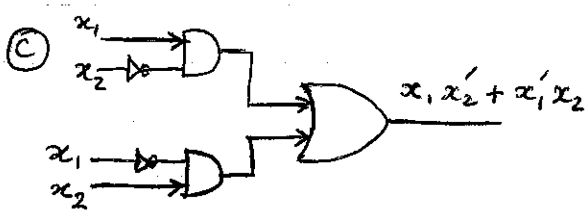
Exple: A Hall light is controlled by two light switches one at each end. Find (a) a truth set. (b) a Boolean Expression. (c) a logic network that allows the light to be switched on or off by either side.

(a)

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

(b) $x_1 x_2' + x_1' x_2$

(c)



Exple: A mail order cosmetics firm, an automatic control device is used to supervise the packaging of orders. The firm sells lipstick, perfume, makeup, and nail polish. As a bonus item, shampoo is included with any order that includes perfume or any order that includes lipstick, makeup, and nail polish. How can we design the logic network that controls whether shampoo is packaged with an order?

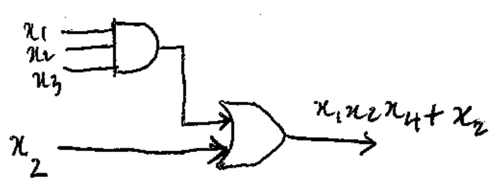
Input to the network → 4 items.

x_1 : Lipstick
x_2 : perfume
x_3 : make up
x_4 : nail polish

$x_i = 1 \Rightarrow$ item i is ordered.

$f(x_1, x_2, x_3, x_4) = 1 \Rightarrow$ shampoo to be included.

x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



HW #6: Pb # 2, 3, 5, 6, 7, 13, 14, 15, 16, 18, 19, 20, 21, 23, 25, 26
Chap7 30, ~~35~~, ~~36~~, 40 → optional
Review Pb page 292