Lesson 10 Boolean Algebra and DeMorgans's Theorem

Yes, it's Algebra. The nightmare continues:

When manipulating complex arrangements of *boolean* variables, they are found to follow many of the rules of ordinary algebra. This is easily seen if we think of **AND-ing as multiplication** and **OR-ing as addition**. Consider, for example, the following *boolean* expression where *a*, *b*, and *c* are *boolean* variables:

a &&(b || c)

Let us agree to rewrite this with "&&" replaced with "*" and "||" replaced with "+". So, our expression becomes,

a * (b + c) = a*b + a*c,

where we have taken the liberty of "multiplying" *a* into the parenthesis just as we would in regular algebra. Converting this back in terms of the familiar && and || symbols we have,

a && b || a && c which can be rewritten as (a && b) || (a && c) for clarity. The reason we can use the parenthesis like this is because && has higher precedence than does ||.

In summary we can state the following rule: **In *boolean* expressions in which each AND is expressed as a * (multiplication) and each OR is expressed as a + (addition), most simplification is done exactly as would be done with ordinary algebra.**

*From this point on in this lesson, we will use * for **AND** and + for **OR** when writing and simplifying* Boolean *expressions. Also, we will take the liberty of writing things like ab. Just as in ordinary algebra where this means "a times b," here it will mean "a*b" which of course ultimately means a **AND** b.*

DeMorgan's Theorem:

Before we offer more examples of Boolean simplification, we need the services of the most important theorem in Boolean algebra, **DeMorgan's Theorem**. Following are its two forms:

!(a + b) = (!a) * (!b)   and   !(a * b) = (!a) + (!b)

Thus, we see a way to turn **OR**s into **AND**s and vice versa. This is a **very powerful** tool. We can even use it where there isn't a "not" (!) in the original expression:

(a + b) = !!(a + b) = !(!(a + b)) = !(!a * !b))

Obvious Theorems:

Some other more obvious but still very useful theorems are(*a* and *b* are *booleans*):

a + false = a	a * true = a	a * false = false
a + true = true !	!a = a	a + !a = true
a + a = a	a * !a = false	
a * a = a		

The three examples in the top row just above are easy to obtain if we substitute 0 for *false* and 1 for *true*. The rules of ordinary algebra are then followed to produce the answers.

This illustrates one of the reasons why we express && as a multiplication sign and || as a plus sign…because it lets us work in terms of something with which we are already familiar (hopefully), regular algebra.

A Subtle Theorem:

This is subtle and not very obvious; however, it can be easily confirmed with a truth table.

a + b = a + (!a)(b) …same as a || b = a || (!a) && (b)

Law of Absorption:

In these theorems, the value of *boolean b* **does not matter** (it could just take a hike).

a * (a + b) = a …same as a = a && (a || b)
a + (a * b) = a …same as a = a || (a && b)

Now we are ready to present some examples of *boolean* simplification:

1. Example:
a(!b) + ab
= a(!b + b) = a(true) = **a**

2. Example:
ab + !ab + !ba + !b(!a)
= b(a + !a) + !b(a + !a)
= b(true) + !b(true) = b + !b = **true** (Amazing! Always true, doesn't depend on a and b)

3. Example:
!(!a + b + c)
= (! !a)(!b)(!c) = **a(!b)(!c)** (notice ! ! a = a)

4. Example:
Express a || b || c using **AND**s instead of **OR**s.
a || b || c
= (a || b || c)
= ! !(a || b || c)
= ! (! (a || b || c))
= **! (!a && !b && !c)**

5. Example:
Illustrate the equivalence of !(a + b) and !a * ! b using truth tables.

a	b	a + b	!(a + b)
false	false	false	true
false	true	true	false
true	false	true	false
true	true	true	false

a	b	!a	!b	!a * !b
false	false	true	true	true
false	true	true	false	false
true	false	false	true	false
true	true	false	false	false

Notice the two gray sections are identical. Also, notice that the two black sections are the same.

6. Example:
Derive the Boolean expression that produces the following truth table. This table uses 1's and 0's. Just thinks of a 1 as a *true* and a 0 as a *false*:

A	B	C	D
(input)			(output)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Choose only the rows that produce a 1 in the output.

(A)(B)(!C) + (A)(!B)(!C) + (!A)(B)(!C) = D

Notice that we always **AND** the inputs together that produce an output of 1. Then **OR** all of these groups. When **AND**-ing, any input that is a 0 should be inverted. Thus, for example, the A, B, and C values of 1, 1, 0 produce (A)(B)(!C).

Now let’s simplify the above expression by factoring out (!C)(A) from the left two terms as follows:

!CA(B + !B) + !AB !C

The parenthesis evaluates to 1, so we have:

!CA + !AB !C

Now factor out !C and get:

!C (A + !AB)

Apply the “subtle theorem” on page 2 of this lesson and get:

!C (A + B) = D