TUTORTUBE: INTRO TO BOOLEAN ALGEBRA

Hi my name is Jeff, I'm a Lead Tutor at the Learning Center, and today I'll be going over the basics of Boolean Algebra.

Boolean Algebra is a branch of regular algebra focused in logic, where everything is either true or false. Just like the algebra we're used to, we have a set of operations we can use as well as some definitions and identities.

Let's start with our operations. We have 3 basic operations: AND, OR, and NOT. Here I'm going to show truth tables for each of these operations. A truth table is just a table that shows the result from any set of inputs.

A good way to remember what each operation does is to think about the name itself.
- AND can be remembered by saying "If A *and* B are both 1, then I get a 1." Otherwise I get 0.
- OR can be remembered by saying "If A *or* B is 1, then I get a 1." Otherwise I get a 0.
- NOT is a lot like a negative sign. Whatever we would get normally, we would flip it: either from 0 to 1 or 1 to 0.

There are some other operations that are possible, but they can be reduced to a combination of any of these three. So we'll just focus on these three.

Now that we have our operations let's define some identies. Recall from regular algebra identies that are commonly used. Things like "Anything times 1 is itself", "Anything times 0 is 0", "Anything divided by 1 is itself", etc. Here is a table of Boolean Postulates and Theorems that's from your "Digital Logic Circuit Analysis and Design" textbook.

TABLE 2.2 BOOLEAN ALGEBRA POSTULATES AND THEOREMS

Expression	Dual
$P2(a): a + 0 = a$	$P2(b): a \cdot 1 = a$
$P3(a): a + b = b + a$	$P3(b): ab = ba$
$P4(a): a + (b + c) = (a + b) + c$	$P4(b): a(bc) = (ab)c$
$P5(a): a + bc = (a + b)(a + c)$	$P5(b): a(b + c) = ab + ac$
$P6(a): a + \bar{a} = 1$	$P6(b): a \cdot \bar{a} = 0$
$T1(a): a + a = a$	$T1(b): a \cdot a = a$
$T2(a): a + 1 = 1$	$T2(b): a \cdot 0 = 0$
$T3: \quad \bar{\bar{a}} = a$	
$T4(a): a + ab = a$	$T4(b): a(a + b) = a$
$T5(a): a + \bar{a}b = a + b$	$T5(b): a(\bar{a} + b) = ab$
$T6(a): ab + a\bar{b} = a$	$T6(b): (a + b)(a + \bar{b}) = a$
$T7(a): ab + a\bar{b}c = ab + ac$	$T7(b): (a + b)(a + \bar{b} + c) = (a + b)(a + c)$
$T8(a): \overline{a + b} = \bar{a}\bar{b}$	$T8(b): \overline{ab} = \bar{a} + \bar{b}$
$T9(a): ab + \bar{a}c + bc = ab + \bar{a}c$	$T9(b): (a + b)(\bar{a} + c)(b + c) = (a + b)(\bar{a} + c)$
$T10(a): f(x_1, x_2, \ldots, x_n) = x_1 f(1, x_2, \ldots, x_n) + \bar{x}_1 f(0, x_2, \ldots, x_n)$	
$T10(b): f(x_1, x_2, \ldots, x_n) = [x_1 + f(0, x_2, \ldots, x_n)][\bar{x}_1 + f(1, x_2, \ldots, x_n)]$	

Table 1 - Table 2.2 Boolean Postulates and Theorems[1]

All the expressions labeled with P are postulates, which is a fancy word for identites. All the expressions are sorted into the left and right for their AND and OR forms. Some of them, like the commutative property on postulate 3, and the associative property on postulate 4, appear the same way as they do in regular algebra, so we don't need to spend too much time on them. The most important identites are Postulate 2 and Theorems 1 and 2. However, you don't need to memorize these as long as you've memorized the truth tables for AND and OR that we had earlier. Take theroem 1a for example. It says that if you OR two of the same variable together, you get that same variable back. Since A can only be either 0 or 1, look at the rows of the truth table where both A and B are either a 1 or 0. Notice how we get the same number as a result when we OR two of the same number together. You can do the same thing with some of the other simpler theorems, so I encourage you to memorize the tables rather than those smaller theorems and postulates so they don't take up space in your head.

So then the question is what should you memorize? Theorems 4, 5, and 8 are the most important in my opinion. Even if you have this table available to you to use when simplyfing equations, you'll want to be able to recognize those 3 theroems off the top of your head. But the best way to memorize these theorems and get better with them is to practice, so let's look at an example.

Let's say we have the boolean function:

(x+z)(~x+y)(y+z)

Our goal is to simplify the equation. So just by looking it seems like in order to get all these parentheses out of the way, we need to distribute multiple times. But that's a pain, so let's look back at our table of thereoems to see if anything can help us.

Now we need to look for a theorem or postulate that matches part or all of our equation. Take a moment to pause and look through the table to decide what we should use. If you said Theorem 9b, good job! That theorem matches pretty well with our equation. The letters arn't the same but that's okay: they're just variables. So applying our theorem, our function is reduced to:

(x+z)(~x+y).

Now we repeat the process. If we look again at the table, unfortunately we won't find any theorems that will help us as the equation currently stands. So we should try to rearrange the equation and see what happens. Oftentimes this looks like making the equation slightly more complicated so that a theorem will appear for us and make things simpler in the end. So let's distribute this equation. We can do this the same way we do normal algebra, using FOIL for instance. This will give us:

~xx+~xz+xy+zy

Now let's clean this up a little bit. Refering back to the table, x times x bar will give us 0, and anything OR'd with 0 is itself, so this first term can vanish and we're left with:

~xz+xy+zy

Again, let's look at the table and see if anything appeared for us. Thereom 9a works really well for us here. Applying it, I get:

~xz+xy

Now the equation is fully simplified. No more theorems will work here and we can't arrange, distribute, or manipulate this equation any further.

Let's look at another example:

(wx + y)(wx+~y)

Now it looks like Theorem 6 is perfect here, and that's true. But you might be wondering, "but the theorem has just an *a* where *wx* should be. Will it still work?" The answer is yes. We can rewrite wx as a single variable, let's say z.

$$(z + y)(z + \sim y)$$

Now we could apply theorem 6, but let's say we forgot it, or didn't recognize it. What then? Well let's distribute anyway and see what happens. We get:

$$zz+zy+yz+\sim yy$$

Like before, the last term vanishes, and this time we can reduce zz to z giving us:

$$z+zy+yz$$

Now this is why I said earlier memorizing theorem 4 was important, because using this theorem allows all of this stuff to reduce to z, which is what we would've gotten using Theorem 6. Congrats, we just derived Theorem 6. So as long as you remember Theorem 4, there's no stress to memorize Theorem 6. Now before we say we've completed the problem, let's turn z back into wx, giving us our final answer:

$$wx$$

Let's turn our attention to Theorem 8. This is called DeMorgan's Law and is probably the most important theorem on this page. This theorem defines how we handle the NOT operation across multiple variables. While you might guess that the NOT operation is similar to multiplying by negative one, that isn't quite how it works here. When we have a NOT operation over another expression, instead of just complementing each variable, we also have to complement the operation as well. AND and OR are complementary operations, so when we use DeMorgan's Law, we are also going to change the operation from AND to OR, or vice versa. I like to think about this like pushing or pulling the bar in or out of an expression. When we want to apply the bar onto an expression, we push it down, and when we want to factor out a NOT operation, we pull it back out of the expression. So let's see an example where we can use DeMorgan's Law.

We'll start with the equation:

$$\sim(AB)(\sim A+B)(\sim B+B)$$

I encourage you to try to pause and try to simplify this yourself, and then compare with my solution.

Using Postulate 6 and then 2, I can turn this last term into a 1, which then doesn't do anything, so it can go away and just leave us with:

~(AB)(~A+B)

Now here you might be tempted to distriubte this through, but if we try that, the NOT signs become a real annoyance in trying to simplify this expression. If you're curious you can try it that way and compare with my answer at the end, but for now, let's try something else. I can use DeMorgan's Law on this first term and push this NOT bar through. Now I have:

(~A+~B)(~A+B)

We can apply Theorem 6b here and this will all reduce down to ~A.

So let's recap the big things to know and the strategy for simplifying Boolean expressions.

- Memorize the tables for each of the three primary operations, AND, OR, and NOT. Knowing these tables will show you a good handful of the postulates found on the reference table.
- Speaking of the reference table, try to at least memorize Theorems 4, 5, and 8, which is DeMorgan's Law. These three are the most useful theorems, and as we saw earlier, knowing these three often allow you to derive some of the other theorems. Granted this is only if you don't have the table with you. If you can, always have this table close by as a reference.
- Don't be afraid to try things. It's hard to see immediately if a certain idea or solution will work or not. The only way to know for sure is to try and get lots of practice. So make sure to do these problems in pencil, just incase you need to back up a few steps.

With those tools in mind let's go over the strategy for simplifying Boolean expressions.

- Step one is to compare any part of the expression to the theorems on the reference table. If there is one that matches, go to step two, if not go to step three.
- Step two is to use the theorem that you found and apply it to the equation. Write down your new expression and return to step one.
- Step three is to rearrange or manipulate the expression to see if it appears like a theorem on the table. This includes, distributing, factoring out a term, or using DeMorgan's Law. Once you've done that, return to step one. Repeat this process until you can no longer rearrange the expression in any meaningful way and no theorems apply.

Here's the reference from the textbook that we used for the table:

References

[1] Nelson, V. P., Nagle, H. T., Carroll, B. D., & Irwin, J. D. (1995). *Digital Logic Circuit Analysis and Design.* Englewood Cliffs: Prentice Hall.

I hope this video helps get you started on how to understand and simplify Boolean expressions. If you have more questions, feel free to visit learningcenter.unt.edu to see what other resources we can offer you! See you next time!