Boolean Algebra

Algebra is the branch of mathematics that deals with variables. *Variables* represent unknown values and usually can stand for any real number. Because computers use only 2 numbers as we saw with Computer Number Systems, 0 or 1, *George Boole* developed a form of algebra that is used with variables A and B that have TRUE (1) or FALSE (0) values.

Using AND, OR, and NOT with Words or Operations

For example, we use the words AND, OR, and NOT in every day life such as “The sun is shining.” AND “It is raining.” If both are TRUE, then the entire statement is TRUE. If not, then the entire statement is FALSE. In arithmetic, we say that “2+3 > 5” OR “2+3 = 5” which is TRUE because they are both not FALSE. The statement NOT “2+3 > 5” is TRUE because “2+3 > 5” is FALSE while NOT “2+3 = 5” is a FALSE statement.

Just as in algebra, operations are used to write expressions. *Addition* (+) is used for the word OR and *multiplication* (*) is used for the word AND. In this Division, we will use the tilde (~) to represent NOT. For example, in words “NOT (A AND B)” = “~(A*B)” and “A OR NOT B” = “A+~B”. Parentheses are needed to override the order of operations. The order without parentheses from left to right is NOT (~), AND (*), and then OR (+). Fortunately, this follows the same rule that we use in all of mathematics that multiplication is always done before addition.

Truth Tables

Truth tables can be used to represent the outcomes of a Boolean Algebra expression. If there are 2 variables, then there are only 4 possible combinations of 0 or 1 values as follows

A	B	A AND B = A*B
1	1	1
1	0	0
0	1	0
0	0	0

A	B	A OR B = A+B
1	1	1
1	0	1
0	1	1
0	0	0

Therefore, a truth table can be used to evaluate the expression “~(A * B)” in column 4 as well as the expression “A + ~B” in column 6 as follows:

1	2	3	4	5	6

A	B	A*B	~(A * B)	~B	A + ~B
1	1	1	0	0	1
1	0	0	1	1	1
0	1	0	1	0	0
0	0	0	1	1	1

If you are asked "Which ordered pairs make the expression FALSE?" for the first expression, the answer would be (1, 1) by looking for the 0's in column 4. If you are asked "How many ordered pairs make the expression TRUE?" for the second expression, the answer would be 3 by counting the number of 1's in column 6.

Simplification and DeMorgan's Law

There are also some simple rules of simplification that are very similar to those used in regular algebra:

1. A + B = B + A   A * B = B * A   (*Commutative* Property)
2. (A+B)+C=A+(B+C)   (A*B)*C = A*(B*C)   (*Associative* Property)
3. A * (B + C) = A * B + A * C   (*Distributive* Property)

4. NOT (A OR B) = NOT A AND NOT B   (*DeMorgan's Law*)
   = ~A * ~B
5. NOT (A AND B) = NOT A OR NOT B   (*DeMorgan's Law)*
   = ~A + ~B

6. A OR 0 = A + 0 = A   A AND 0 = A * 0 = 0
7. A OR 1 = A + 1 = 1   A AND 1 = A * 1 = A
8. A OR NOT A = A + ~A = 1   A AND NOT A = A * ~A = 0
9. A OR A = A + A = A   A AND A = A * A = A
10. NOT (NOT A) = ~(~A) = A

The most useful one is *DeMorgan's Law* that changes the negation of an OR to the negation of each separately with an AND between them. The same law changes the negation of an AND to the negation of each separately with an OR between them. *Augustus DeMorgan* was a 19^{th} century mathematician who formulated it. It can be shown to be TRUE with the following truth table because columns 4 and 7 are the same as well as the values in columns 9 and 10:

1	2	3	4	5	6	7	8	9	10	

A	B	A*B	~(A*B)	~A	~B	~A + ~B	A+B	~(A+B)	~A * ~B	
1	1	1	0	0	0	0	1	0	0	Same
1	0	0	1	0	1	1	1	0	0	Same
0	1	0	1	1	0	1	1	0	0	Same
0	0	0	1	1	1	1	0	1	1	Same

Equivalent Expressions

This also indicates that two expressions are *equivalent* if they produce the same truth table values for all possible inputs. An example of using the above laws of simplification is as follows:

NOT (A AND B) OR B	~(A * B) + B
(NOT A OR NOT B) OR B	(~A + ~B) + B
NOT A OR (NOT B OR B)	~A + (~B + B)
NOT A OR 1	~A + 1
1	1

Therefore, the original expression simplifies to always being TRUE no matter what the values of A or B are. This is called a *tautology* in Boolean Algebra. The Boolean expressions A + 1 and A + ~A are also considered *tautologies*. If the question was "How many ordered pairs make the following expression TRUE?", you could solve it by using a truth table or by simplification. Either way the answer is 4 because there are 4 total possibilities for two variables that can only be TRUE (1) or FALSE (0):

(1, 1), (1, 0), (0, 1), and (0, 0).

It is possible to have more than 2 variables in a Boolean expression. For 3 variables, there are 8 possible *ordered triples* such as (0, 1, 1) and for 4 variables, there are 16 possible *ordered quadruplets* such as (1, 0, 1, 1). We will only being using 2 variables in the Elementary Division.

References

http://tryengineering.org/lesson-plans/boolean-algebra-elementary
http://alex.state.al.us/lesson_view.php?id=11203
http://mathmaniacs.org/lessons/04-boolean/
https://www.tes.com/teaching-resource/boolean-algebra-1-6307568

Sample Problems

<table>
<tr>
<td>Determine if the following statement is TRUE or FALSE in mathematics:

3+4>6 AND 7-2>6</td>
<td>The statement 3+4>6 is TRUE while the statement 7-2>6 is FALSE. That makes the entire statement FALSE because both must be TRUE for the entire statement to be TRUE. The answer is FALSE.</td>
</tr>
<tr>
<td>Determine if the following statement is TRUE or FALSE in mathematics:

(NOT (3+4<6) AND (7-2>6)) OR NOT (5+1=6)</td>
<td>The statement 3+4<6 is FALSE, so NOT (3+4<6) is TRUE. The statement 7-2>6 is FALSE, so TRUE AND FALSE is FALSE. The statement 5+1=6 is TRUE, so NOT (5+1=6) is FALSE. Therefore, FALSE OR FALSE is FALSE because at least one must be TRUE for the entire statement to be TRUE. The answer is FALSE.</td>
</tr>
<tr>
<td>Find all ordered pairs (A, B) that make the following expression FALSE.

(NOT (NOT A OR B)) AND (NOT (A OR NOT B))</td>
<td>By using DeMorgan's Law with each NOT and then using simplification rule #8 above,
(NOT (NOT A OR B)) AND (NOT (A OR NOT B))
= ((NOT (NOT A)) AND (NOT B)) AND
(NOT A AND NOT (NOT B))
= (A AND NOT B) AND (NOT A AND B)
= (A AND NOT A) AND (B AND NOT B)
= 0 AND 0
= 0
This yields 0 solutions so the ordered pairs (1, 1), (1, 0), (0, 1), and (0, 0) make the expression FALSE.</td>
</tr>
<tr>
<td>Which of the following is a tautology?

a) A * ~A + B
b) A + ~A * B
c) (A + ~A) + B</td>
<td>Remember that multiplication (AND) has precedence over addition (OR). Letter a) simplifies to 0 + B = B. Letter b) doesn't simplify. Letter c) simplifies to 1 + B = 1 so the answer is (c).</td>
</tr>
<tr>
<td>How many ordered pairs make the following expression TRUE?</td>
<td><table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr>
<tr><td>A</td><td>B</td><td>1+2</td><td>1*3</td><td>~4</td><td>~1</td><td>2*5</td><td>5+7</td><td>~8</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table></td>
</tr>
</table>

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

1	0	1	1	0	0	0	0	1
0	1	1	0	1	1	1	1	0
0	0	0	0	1	1	0	1	0

Therefore, both (1,1) and (1,0) make the statement TRUE so the answer is 2.

Which of the following Boolean Algebra expressions are equivalent?

a) A * ~(B + ~A)
b) A * ~B
c) A * ~(~B + A)

1	2	3	4	5	6	7	8	9	10	11
A	B	~1	~2	2+3	4+1	~5	~6	1*7	1*4	1*8
1	1	0	0	1	1	0	0	0	0	0
1	0	0	1	0	1	1	0	1	1	0
0	1	1	0	1	0	0	1	0	0	0
0	0	1	1	1	1	0	0	0	0	0

Look at columns 9, 10, 11. (a) & (b) are equivalent.

Simplify:
NOT (((A OR NOT A) AND NOT B) AND (NOT A AND B))

Then find how many ordered pairs make the expression FALSE.

NOT (((A OR NOT A) AND NOT B) AND (NOT A AND B))
= NOT ((1 AND NOT B) AND (NOT A AND B))
= NOT (NOT B AND NOT A AND B)
= NOT (NOT B) OR NOT (NOT A) OR NOT B
= B OR A OR NOT B
= (B OR NOT B) OR A
= 1 OR A
= 1
This is a *tautology*, so 0 ordered pairs make it FALSE.

Simplify:
~(A + ~B) + ~A * B

Then find what ordered pairs make the expression TRUE.

~(A + ~B) + ~A * B
= ~A * ~(~B) + ~A * B
= ~A * B + ~A * B
= ~A * B by using the order of operations.

From the simplified expression, A must be FALSE and B must be TRUE so the only ordered pair is (0, 1).

Simplify: NOT (A AND NOT B) OR NOT (NOT A AND B) Then find what ordered pairs make the expression TRUE.	NOT (A AND NOT B) OR NOT (NOT A AND B) = NOT A OR NOT (NOT B) OR NOT (NOT A) OR NOT B = NOT A OR B OR A OR NOT B = (NOT A OR A) OR (B OR NOT B) = 1 All 4 ordered pairs make the expression TRUE which are (1, 1), (1, 0), (0, 1), and (0, 0).