Fundamentals of Logic

Statements/Propositions **– Sentences that are true or false but not both. (Just like a simple boolean expression or conditional expression in a programming language.)**

For our purposes, statements will simply be denoted by lowercase letters of the alphabet, typically p, q, and r. I will refer to these as boolean variables as well.

Given simple statements, we can construct more complex statements using logical connectives. Here are 4 logical connectives we will use:

1) **Conjunction: This is denoted by the '□' symbol. The statement p □ q is read as "p and q." Only if both the values of p and q are true does this expression evaluate to true. Otherwise it is false.**

2) **Disjunction: This is denoted by the '□' symbol. The p □ q is read as "p or q." As long as at least one of the values of p or q is true, the entire expression is true**

3) **Implication: This is denoted by the '□' symbol. The statement p □ q is read as "p implies q". Essentially, in a programming language, this logic is captured in an if then statement. If p is true, the q must be true. However, if p is not true, there is no guarantee of the truth of q. An important observation to note: when statements are combined with an implication, there is no need for there to be a causal relationship between the two for the implication to be true.**

Consider the following implications:

If my bread is green, then I will not eat it.

Here, if the bread is not green, that does not guarantee that I will eat it. Perhaps it is wheat bread and I hate wheat bread. All I know is that if my bread is green, I will definitely NOT eat it.

If Pluto is the largest planet in our solar system, then pigs will fly out of my butt.

Wayne would actually be making a correct implication here, (assuming that we currently have accurate knowledge about our solar system...) Since the first part of our implication is false, the entire implication is automatically true.

4) Biconditional: This is denoted by the '□' symbol, and the statement p □ q is read "p if and only if q." The phrase "if and only if" is often abbreviated as "iff". Breaking this down into pieces, this means that p □ q AND q □ p. Hence exactly when p is true, q is true. (And in all cases where p is false, q must be false as well.)

Here is an example of a biconditional:

If and only if my alarm clock rings in the morning, then I will attend my morning classes.

From this statement we CAN deduce that if the alarm clock does NOT ring, then I will NOT go to my morning classes.

Finally, it will be important to have a symbol to denote the negation of a statement/proposition. We will use the '□' symbol.

The statement □p will be read "not p." This statement will have the opposite value of p.

Truth Tables

Now, given a particular compound statement, we can use a truth table to determine which values of the boolean variables result in the statement being true.

The idea here is to simply make a table, listing all the possible combinations of values for each of the boolean variables in a statement. Then, plug these values into the statement and see if it is true or not with these values. This is probably easiest seen with an example.

Consider the statement: p □ (q □ □r). Here is a truth table:

p	q	r	q □ □r	p□(q□□r)
0	0	0	1	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

Thus there are three possible combinations of values for p, q, and r that make p □ (q □ □r) true.
To see an example, let p, q and r be the following statements:

p: I have taken out the trash.
q: I have finished cleaning the dishes.
r: I have watched 5 hours of TV.