# Truth Functions

Here are some questions we will answer in this lecture:

1. What is a **truth function**? A **statement variable**?
2. What is a **truth table**? How can truth tables be used to provide *definitions* for the logical operators?
3. How can the truth values of compound propositions be computed, supposing one knows the truth values of the simple propositions that compose them?
4. How do the logical operators (NOT, AND, OR, IF-THEN, EQUALS) differ from the ordinary English words “not”, “and”, “or,” “if-then”, and “is equal to”?

## What is a Truth Function?

A **truth-function** is a compound statement whose truth depends *only* on the truth values of the simple statements that compose. The following are examples of truth functions:

1. “The Giants are a football team and the Packers are a football team.”
   * This is composed of two simple statements G and P.
   * The compound statement is true if and only if G and B are *both* true.
2. “It is not the case that Buffy is a vampire.”
   * This is composed of one simple statement B.
   * The compound statement ~B is true if and only if B is false.
3. “If Frodo is wearing the ring, then Frodo is invisible.”
   * This is composed of two simple statements R and I.
   * The compound statement is true if (a) R is false or (b) I is true. It is false *only* if R is true and I is false (i.e., Frodo is wearing the ring but is still visible).

The following compound statements are NOT truth functions:

1. “Callie believes that Jason is a jerk.”
   1. This is composed of a simple statement J (“Jason is a Jerk”).
   2. Even ifwe *know* whether Jason is a jerk (i.e., we know that J is true), we still wouldn’t be able to determine what Callie believes. Figuring out whether this statement is true or false requires inductive reasoning (and is beyond the scope of propositional logic, which concerns deductive arguments).
2. “If it were the case that the Nazis had won, then everyone in the U.S. would speak German.”
   1. This is composed of two simple statements N and G.
   2. Even though we know the truth value of N (false) and W (false), we can’t use this to deduce the truth value of the whole thing. Figuring out whether this statement is true or false requires inductive reasoning.

## Defining the Logical Operators Using Truth Tables

A **truth table** shows how the truth of compound statements is determined by the truth of its component propositions *for every possible combination of truth values.* In truth tables, lower-case letters like *p, q, r,* etc. are called **statement variables.** These can stand for any statement whatsoever. A truth table will contain enough lines to show *every possible combination* of the values of the statement variable. So, if there is ONE statement variable, there will be TWO rows (since the statement has two possible truth values: TRUE or FALSE). By contrast, if there are TWO variables, you will need to use FOUR rows (TT, TF, FT, and FF—every possible combination). Three variables will require eight rows, four variables will require 16 rows, and so on.

**Defining the Logical Operators.** Truth tables can be used to give *definitions* of the logical operators. This is because a truth table can *show* us exactly how the truth value of a compound statement changes with the truth value of the component statements. For example, here is the truth table that defines NOT (or “~”). Notice that, since there is only one statement variable (p), there are only *two* rows.

|  |  |
| --- | --- |
|  |  |
| T | F |
| F | T |

By itself, this might not mean much at first. So, here’s the same table, with a little explanation of what is going on:

|  |  |  |
| --- | --- | --- |
| **Explanation** |  |  |
| This line of the truth table considers the possibility p is TRUE. In this case, ~p is false. | T | F |
| This line considers the possibility p is FALSE. In this case, ~p is TRUE | F | T |

**Truth Tables for Conjunction, Disjunction, Implication, and Material Equivalence.** Unlike NOT (which works on individual statements), the other logical operators work on pairs of statements. Because of this, we will need to use a two-variable truth table to define them. In order to capture every possible combination of truth values for these two variables, this truth table will have four rows. Again, here it is, in its simplest form:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **p** | **q** | **p** ∙ **q** | **p** ∨ **q** | **p** ⊃ **q** | **p** ≡ **q** |
| T | T | T | T | T | T |
| T | F | F | T | F | F |
| F | T | F | T | T | F |
| F | F | F | F | T | T |

And here’s an expanded version of the table, with a little more explanation of what is going on:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Here, we consider EVERY POSSIBLE combination of truth values for p and q.** | | | **AND is only true when both are true** | **OR is only false when both are false** | **IF-THEN is only false when p is true, q false.** | **EQUALS is true when both true or both false.** |
|  |  |  |  |  |  |  |
| T | T | Both are true | T | T | T | T |
| T | F | Only p is true | F | T | F | F |
| F | T | Only q is true | F | T | T | F |
| F | F | Both are false | F | F | T | T |

A few things worth noting:

* is true if is true, is true, or both are true. This is a bit different than the English word “or,” which often means “not both.” For example it is *true* (at least in logic!)that “Either apples are a type of fruit or oranges are.”
* is false *only* if is true and is false. This is a bit different than the English “if p then q,” which is sometimes false, even if *p* is true—e.g. “if I am one inch taller than I actually I am, then I am Batman” is true in logic.

## Making Your Own Truth Table

To make your *own* truth table, you’ll need to begin by filling out the columns beneath the statement variables (we’ll talk more about this in future classes). Here’s how you do this:

1. Your table will need to have 2^(number of statement variables) rows. So, two variables = rows. Three variables would require rows, and so on.
2. Divide the total number of rows in half (in the case of the truth table above, 4/2 = 2).
3. Go to the first statement variable on the left (in this case, p), and fill up the rows beneath it by writing this many (two) Ts followed by the same number of Fs: TTFF.
4. Now, divide *this* number in half (2/1 = 1), and move on to the next statement variable (in this case, q). Fill up the rows beneath it by writing this many Ts, followed by the same number Fs, and repeating: TFTF.
5. Continue doing this until you’ve completed every statement variable. We’ll talk later about filling out the rest of the truth table.

So, for example, let’s suppose we want to make a truth table for P ⊃ ~Q. This has TWO simple statements, and we would need FOUR lines. We would end up with something like the following:

|  |  |  |
| --- | --- | --- |
| P | Q | P ⊃ ~ Q |
| T | T |  |
| T | F |  |
| F | T |  |
| F | F |  |

By contrast, if we started with a statement such as (P ⊃ ~Q) ∙ ~~R, we would have TREE simple statements, and would need EIGHT LINES. We would now want to start with something more like this:

|  |  |  |  |
| --- | --- | --- | --- |
| P | Q | R | (P ⊃ ~Q) ∙ ~~R |
| T | T | T |  |
| T | T | F |  |
| T | F | T |  |
| T | F | F |  |
| F | T | T |  |
| F | T | F |  |
| F | F | T |  |
| F | F | F |  |

If we considered a statement with four variables, we would need a table with 16 rows; one with five variables would need 32 rows, etc. In future lessons, we’ll learn how to actually complete these sorts of truth tables.

## Computing the Value of Compound Statements

Now that we know how each of the logical operators “work,” we can start using them to determine the value of compound propositions. To compute the truth value of compound propositions, you should (1) write the truth values of the simple propositions directly underneath them, (2) use these to figure out the truth values of propositions in parentheses, working outward until (3) you reach the main operator. The truth value of the main operator determines the truth value of the statement as a whole (that is, if the main operator ends up being true, the whole statement is true. If it is false, the whole statement is false.)

**Example:** Suppose A, B, and C are true; and D and E are false. What is the truth value of (A ∨ E) ⊃ E?

In this case, the main operator is the horseshoe that we filled in at the very end. Since this is false, the whole expression is false!

## Solved Problem: Determining Truth Values

Suppose A and B are true, and C and D are false. For the following problems (1) identify the main operator, and (2) determine the truth value of the statement as whole.

|  |  |  |
| --- | --- | --- |
| Statement | Main Operator? | Truth of whole statement? |
| A | None. | True |
| C | None. | False |
| ~A | Negation. | False (because A is true) |
| ~~A | Negation (the far left one) | True (because ~A is false) |
| ~A ∙ C | Conjunction | False (both conjuncts are false) |
| A ∙ (C v D) | Conjunction | False |
| (A ∙ C) v D | Disjunction | False |
| B ≡ ~D | Equiv | True (both parts true) |
| ~(A ∙ C) | Negation | True (since the part in parentheses is false) |
| D ⊃ C | Conditional | True (since antecedent D is false) |
| A ⊃ (B ≡ C) | Conditional | False (antecedent A is true, consequent B ≡ C is false) |
| [A ⊃ (B ≡ C)] v B | Disjunction | True (since B is true, whole thing is true) |
| {[A ⊃ (B ≡ C)] v B} ≡ A | Equiv (the far right one) | True (A is true, as is the stuff to which it is being compared) |
| ~{{[A ⊃ (B ≡ C)] v B} ≡ A} | Negation | False (because what it negates is true). |

## Review Questions

1. Suppose A, B, and C are true, and D and E are false. Determine the truth value of the following propositions:
   1. E ⊃ A
   2. (A ∙ B) ∙ D
   3. ~(B ⊃ C)
   4. ~(~E ∙ ~A)
2. Suppose that C = “The moon is made of cheese” and O = “The moon orbits the earth.” What is the truth value of the following?