Notes on Truth Tables

Cynthia Bolton
Craig Carley

Arizona State University - Summer 2005

1 Introduction to Truth Tables

1.1 Introduction to Propositional Symbolic Logic

In propositional symbolic logic we use uppercase letters A…Z to represent indicative statements. For example, the statement, "It is raining." might be represented by the letter P. Given this convention, propositional logic is sometimes referred to as sentential logic.

Indicative (or, declarative) statements unlike questions, commands (imperatives), and expletives are always said to have a truth value. For instance, the statement "It is raining" is either true or false regardless of whether anyone knows anything about the current meterological conditions. But, questions such as, "Will it rain this afternoon?", or commands such as, "Close the door" are not the sorts of statements which can be true or false.

1.1.1 A Note on Ambiguity

Often students react to the idea that all indicative statements are either true or false. They accuse the instructor of narrow-mindedness, black-and-white thinking, or not being aware of how nuanced reality actually is. They point out the ambiguity which is inherent in many indicative statements and argue that the whole enterprise of logical analysis is doomed because of this fact. In the case the statement, "It is raining", some will raise the question as to the point at which a heavy fog or mist becomes rain. But, this objection misses the point, since the truth or falsity of the statement does not depend on our being able to determine its truth value. For our purposes, the important thing to note is that once the ambiguity of an indicative statement has been removed, it is impossible that both the statement itself and its negation should be simultaneously true. In other words, the statement "It is raining" and the statement "It is not raining" cannot be both true at the same time and place.

1.2 The Language of Propositional Logic

The objects in the language of propositional logic are: statement variables, logical operators (or, connectives), and parentheses. As mentioned above, statement letters (or, variables) are uppercase Roman letters A…Z that represent simple statements. These simple statements may be combined by means of the logical connectives to form more complex statements. For example, if we represent the statement, "It is raining" with the letter P, and the statement "There are clouds in the sky" with the letter Q, then we can form several more complex statements with the help of logical connectives. As in, “It is raining *and* there are clouds in the sky”, which we represent symbolically as: P & Q.

The truth value of these complex statements depends on the truth value of the simple (or, atomic) statements which comprise them as well as how these simple statements are combined. As the complexity of our statements increase, we must employ the use of parentheses so that the final product has but one possible reading. You can think of the parentheses as the punctuation marks in the language of propositional logic. (Imagine how confusing English would be if we had no periods, commas, or question marks.) In propositional logic, the parentheses designate the order of operation (or, precedence and scope) similar to that of mathematics.

In propositional logic, there are several logical operators which we use to connect simple statements in order to form more complex statements. These are summarized in the following table.

Operator	Name	Logical Function	Translation
˜ $\neg$	tilde logical not	negation	not, it is not the case that
& $\wedge$ $\cdot$	ampersand logical and dot	conjunction	and, also, moreover, but
$\vee$	wedge logical or	disjunction	or, unless, nor
$\supset$ $\rightarrow$	horseshoe arrow	material conditional (implication)	if ... then ..., only if
$\equiv$ $\leftrightarrow$	triple bar double arrow	material bi-conditional (equivalence)	if and only if, just in case

1.3 What is a Truth Table?

As noted above, in propositional logic every statement, simple or compound, has a truth value. The simple statement, P, for example, has a truth value. It is either true or false. The compound statement, (A & $\neg$B) $\vee$ $\neg$($\neg$A $\supset$ B), for example, also has a truth value. But the truth value for (A & $\neg$B) $\vee$ $\neg$($\neg$A $\supset$ B) depends upon the truth values for A and B. We often describe compound statements in propositional logic as truth-functional compound statements. This is because their truth value depends upon the truth value of the simple statements contained within the compound statement.

We can list the truth values for a statement, simple or compound, by means of a truth table. A truth table gives every possible truth combination (or every possible interpretation) for a set of simple statements.

1.3.1 Initial Setup

In general, if we have n simple statements, we have 2^n possible truth combinations. If we have one statement, then we have two truth combinations. If we have two statements, then we have four truth combinations. If we have three statements, then we have eight truth combinations.

We can represent these possible truth combinations by means of a table:

One Statement	Two Statements		Three Statements		
P	P	Q	P	Q	R
T	T	T	T	T	T
F	T	F	T	T	F
	F	T	T	F	T
	F	F	T	F	F
			F	T	T
			F	T	F
			F	F	T
			F	F	F

A compound statement not only includes simple statements but connectives. Thus, we need to know the truth values for connectives as well.

1.3.2 Basic Truth Table for the Five Connectives

Basic Setup Setup		Negation		Conjunction	Disjunction	Material Conditional	Material Bi-conditional
P	Q	$\neg$P	$\neg$Q	P & Q	P $\vee$ Q	P $\supset$ Q	P $\equiv$ Q
T	T	F	F	T	T	T	T
T	F	F	T	F	T	F	F
F	T	T	F	F	T	T	F
F	F	T	T	F	F	T	T

In general, we should remember that:

1. $\neg$P takes the opposite value of P.
2. P & Q is only true when P and Q are both true.
3. P $\vee$ Q is only false when P and Q are both false.
4. P $\supset$ Q is only false when P is true and Q is false.
5. P $\equiv$ Q is true when P and Q have the same truth value—either they are both true or they are both false.

1.3.3 How to Construct a Truth Table

A truth table is a two-dimensional representation (or matrix) of all possible truth values for any statement (either atomic or complex). The best method for learning how to construct a truth table by doing, so let's walk through two examples—one simple and one a bit more complex.

Consider the following statement: P $\supset$ (Q $\vee$ R).

1. In order to represent a statement in a truth table, the first step is to collect all the variables in the statement and list them in the first row of a table on the far left side. So far, our table should look like this:

P	Q	R

2. Next we must make truth value assignments which takes into account ALL possible values for this given set of variables. Since there are only two possible values (TRUE or FALSE), we can systematically capture all possibilities by a sort of binary counting—e.g., 000, 001, 010, 011, 100, 101, 110, 111 or in our case TTT, TTF, TFT, TFF, FTT, FTF, FFT, FFF. So, now we indicate this in the table as follows:

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

   All truth tables will have this sort of an index on the left-hand side either explicitly or implicitly.

3. The next step is to copy the formula in the first row to the right of the index area, as follows:

P	Q	R		P	$\supset$	(	Q	$\vee$	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								

4. Next we will work from the inmost parentheses outward. In this example, we will start with the $\vee$. The $\vee$ is a logical or and we know from the table above that the only occassion when a disjunction is false is when both of the disjuncts are false. So, the first row (other than the header row) would look like this:

P	Q	R		P	$\supset$	(	Q	$\vee$	R	)
T	T	T					T	T	T	

- NOTE: As an intermediate step you might wish to copy the values of the variables from the index part of the table to the computational part of the table as I have done (using a smaller typeset) above.
- Now lets complete the rest of the rows of the truth table under the $\vee$ operator.

P	Q	R		P	$\supset$	(	Q	$\vee$	R	)
T	T	T					T	T	T	
T	T	F					T	T	F	
T	F	T					F	T	T	
T	F	F					F	F	F	
F	T	T					T	T	T	
F	T	F					T	T	F	
F	F	T					F	T	T	
F	F	F					F	F	F	

5. Next we move look at the formula and see if there is another subformula which is at the same level of precedence (or scope) and perform the operations described in the preceeding step. In this case since there is no such subformula, we pop up to the next level—namely the subformula containing the $\supset$ operator. In this formula, the $\supset$ is the main connective and after we finish this step our truth table will be complete.

P	Q	R		P	$\supset$	(	Q	$\vee$	R	)
T	T	T		T	T		T	T	T	
T	T	F		T	T		T	T	F	
T	F	T		T	T		F	T	T	
T	F	F		T	F		F	F	F	
F	T	T		F	T		T	T	T	
F	T	F		F	T		T	T	F	
F	F	T		F	T		F	T	T	
F	F	F		F	T		F	F	F	

- Recall that a material conditional is only false when the antecedent (the first variable) is true and the consequent (the second variable) is false. Also note, that in this example the antecedent is the statement variable P, however, the consequent is the truth value of the subformula Q $\vee$ R.

1.3.4 A More Challenging Example

Consider the following formula:

$$\neg((A \,\&\, \neg B) \equiv \neg C) \supset (\neg A \vee B)$$

Before reading further, see if you can identify the main connective and then try to construct your own truth table for the above formula.

First we identify and collect the variables which are present in the formula and place them in the header row of the index section of our table.

A	B	C

Next we enter all possible truth values for this set of variables.

A	B	C
T	T	T
T	T	F
T	F	T
T	F	F
F	T	T
F	T	F
F	F	T
F	F	F

Next, we create a header row for the formula itself.

A	B	C		$\neg$	(	(	A	&	$\neg$	B	)	$\equiv$	$\neg$	C	)	$\supset$	(	$\neg$	A	$\vee$	B	)
T	T	T																				
T	T	F																				
T	F	T																				
T	F	F																				
F	T	T																				
F	T	F																				
F	F	T																				
F	F	F																				

Working from the inmost set of parentheses we begin to fill in the truth table. The subformula, A & $\neg$B, is first in the order of precedence. However, before we can determine the truth value of the conjunction, we should take care of the negation operator.

A	B	C		$\neg$	(	(	A	&	$\neg$	B	)	$\equiv$	$\neg$	C	)	$\supset$	(	$\neg$	A	$\vee$	B	)
T	T	T							F	T												
T	T	F							F	T												
T	F	T							T	F												
T	F	F							T	F												
F	T	T							F	T												
F	T	F							F	T												
F	F	T							T	F												
F	F	F							T	F												

NOTE: If the character following a negation operator is a statement variable the truth value of the negation may be assigned at anytime. However, if the negation operator is followed by a left parenthesis, then the truth value of that particular subformula must be determined prior to assigning a truth value to the negation operator.

Now we can determine the truth value of the conjunction we are working on.

A	B	C		¬	(	(	A	&	¬	B	)	≡	¬	C	)	⊃	(	¬	A	∨	B	)
T	T	T					T	F	F	T												
T	T	F					T	F	F	T												
T	F	T					T	T	T	F												
T	F	F					T	T	T	F												
F	T	T					F	F	F	T												
F	T	F					F	F	F	T												
F	F	T					F	F	T	F												
F	F	F					F	F	T	F												

Since there are no other subformulas that are at the current depth (that is, no other subformula is nested as deep as the one we have just been considering), we can pop up a level. When we do so, we find that there are two—one which has a material biconditional as a connective and the other is a disjunction. It doesn't matter which one we tackle first, so let's do the biconditional first.

A	B	C		¬	(	(	A	&	¬	B	)	≡	¬	C	)	⊃	(	¬	A	∨	B	)
T	T	T					T	F	F	T		T	F	T								
T	T	F					T	F	F	T		F	T	F								
T	F	T					T	T	T	F		F	F	T								
T	F	F					T	T	T	F		T	T	F								
F	T	T					F	F	F	T		T	F	T								
F	T	F					F	F	F	T		F	T	F								
F	F	T					F	F	T	F		T	F	T								
F	F	F					F	F	T	F		F	T	F								

You will notice that we took care of the negation in front of the C first, and then applied the rule of the material biconditional to the truth value obtained from the conjunction in the previous step and the truth value of the negated C. Remember, a material biconditional is true if the truth values of its antecendent and its consequent agree, otherwise it is false.

Now let's determine the truth values for the subformula containing the disjunction. But first, we must take care of the negation preceeding the A.

A	B	C		¬	(	(	A	&	¬	B	)	≡	¬	C	)	⊃	(	¬	A	∨	B	)
T	T	T					T	F	F	T		T	F	T				F	T	T	T	
T	T	F					T	F	F	T		F	T	F				F	T	T	T	
T	F	T					T	T	T	F		F	F	T				F	T	F	F	
T	F	F					T	T	T	F		T	T	F				F	T	F	F	
F	T	T					F	F	F	T		T	F	T				T	F	T	T	
F	T	F					F	F	F	T		F	T	F				T	F	T	T	
F	F	T					F	F	T	F		T	F	T				T	F	T	F	
F	F	F					F	F	T	F		F	T	F				T	F	T	F	

Now we have only two operators to solve for in order to complete our truth table—namely, the negation and the material conditional. Two questions must be answered at this point: "Which one should we do first?" and "Does it matter?"

To answer the second question first, it definitely does make a difference which connective we treat first. If we mistake which operator is the main connective, we will draw wrong conclusions about the formula under consideration. Regarding the first question, we always want to do the main connective last. So, which connective is the main connective? Consider the following formulas:

1. ¬(((A & ¬B) ≡ ¬C) ⊃ (¬A ∨ B))
2. (¬((A & ¬B) ≡ ¬C)) ⊃ (¬A ∨ B)

The question is which one of the above is equivalent to the original formula:

- ¬((A & ¬B) ≡ ¬C) ⊃ (¬A ∨ B)

In formula (1), the main connective is the negation, while in formula (2) the main connective is the material conditional (the horseshoe). It is formula (2) that is equivalent to the original rendering. Although the original formula does not make the parentheses explicit, they are nevertheless implicitly understood. Consequently, we should first deal with the negation and save the horseshoe for the final operation. When you have finished the truth table, it should resemble the following:

A	B	C		¬	(	(	A	&	¬	B	)	≡	¬	C	)	⊃	(	¬	A	∨	B	)
T	T	T		F			T	F	F	T		T	F	T		**T**		F	T	T	T	
T	T	F		T			T	F	F	T		F	T	F		**T**		F	T	T	T	
T	F	T		T			T	T	T	F		F	F	T		**F**		F	T	F	F	
T	F	F		F			T	T	T	F		T	T	F		**T**		F	T	F	F	
F	T	T		F			F	F	F	T		T	F	T		**T**		T	F	T	T	
F	T	F		T			F	F	F	T		F	T	F		**T**		T	F	T	T	
F	F	T		F			F	F	T	F		T	F	T		**T**		T	F	T	F	
F	F	F		T			F	F	T	F		F	T	F		**T**		T	F	T	F	