Logicola "Truth Evaluation" Exercises

The Logicola exercises for Ch. 6.3 concern "truth evaluations," and in 6.4 this complicated to include "unknown" evaluations. I wanted to say a couple of things for those of you who might be having problems with these exercises.

<u>Truth Evaluations</u>

The first and most important thing to say is that these exercises all require you to be very familiar with the truth tables for the five basic truth functional connectives, i.e., the truth tables for *negations*, *conjunctions*, *disjunctions*, *conditionals*, and *biconditionals*. As I mentioned in class, you might want to have these written down in front of you as you do these exercises. But the point is that you need to know (or be able to see by looking at the tables), for example, that conjunctions are only true when both conjuncts are true; that disjunctions are only false when both disjuncts are false; that biconditionals are only true when both components have the same truth value, etc. If you want, you can find a printout of them online at:

http://homepages.wmich.edu/~baldner/ttconnectives.pdf

Still, even if you know these things, or at least understand how to look at the truth tables to see such things, you still have to understand the "technique" that is involved in doing the exercises. We know that the truth value of a compound statement (a *truth functionally* compound statement, but I will stop repeating that) is determined by the truth values of its component parts. So, these exercises give you the truth value assignments for the components and ask you to "compute" the truth value of a compound statement that has these components. The "box" on the bottom of the Logicola screen gives you a "workspace" to work out the answer. In the simpler exercises, you might not need a workspace: you'll just see the answer. But as the compounds get more complicated, you need to have a "technique" or process for calculating the truth value of the compound.

In logic, it's all about the process.

So, suppose we have the following compound statement:

(~(P v Q) ⊃ (R · ~Q))

and you are given the truth values of P=1, Q=1, and R=0.

Where do you begin? And what do you do next?

Here is the rule: **Compute the truth value for the *smallest* (shortest) well formed formulas (wffs) first**.

Then **continue, always computing for the smallest wffs remaining**, until you have the truth value for the compound statement you were given.

What are the “smallest” wffs? Well, the smallest wffs here are just the capital letters themselves. So replace all the capital letters with the truth values provided.

In the workspace box, simply hit the return key, and the current line will be repeated on the next line. So, with this example, the statement above would be in the box. Click inside the box and hit the return key, and the line will be repeated, giving you:

(~(P v Q) ⊃ (R · ~Q))
(~(P v Q) ⊃ (R · ~Q))

Now replace each capital letter in the second line with the truth value provided, giving you:

(~(P v Q) ⊃ (R · ~Q))
(~(1 v 1) ⊃ (0 · ~1))

Then, hit the return key again, and this last line will be repeated, giving you:

(~(P v Q) ⊃ (R · ~Q))
(~(1 v 1) ⊃ (0 · ~1))
(~(1 v 1) ⊃ (0 · ~1))

What's next? Evaluate the truth value for the next smallest wff. What is the next smallest wff remaining? ~1. To make sure you are replacing a *well formed formula* with its "calculated" truth value (and not some group of symbols that isn't actually a wff), what I suggest you do is to "highlight" (or "select") the wff (i.e., use your computer's mouse or keyboard commands to "select" the appropriate text on the screen), and "replace" that text by typing in the appropriate truth value. So, I would highlight the next smallest wff in the last line, "~1," which will look like this:

(~(1 v 1) ⊃ (0 · ~1))

And then I would replace the highlighted text by typing in the appropriate truth value. The truth table for negation tells us that "~" reverses the truth value, so we type in a "0." So, type in "0" to replace the highlighted text in the last line, giving us:

(~(1 v 1) ⊃ (0 · 0))

The point, again, of "highlight" and the "replacing" is that it forces you to pay attention to what the wff is that you are evaluating.

So, once again, *highlight the next smallest wffs*. What are the next smallest wffs? There are two that are of the same length, "(1 v 1)" and "(0 · 0)." Just because I'm compulsive, I like to do them one at a time, but what matters is that you do smaller wffs before doing the longer wffs that they are components of.

The reason I'm suggesting that you highlight (or "select") the wff and then hit the "1" or "0" key to "replace" that text (as opposed, say, to simply backspacing or deleting text) is that I want you to be clear about the fact that you are replacing wffs with their truth values. That is, it is, in this example, "(1 v 1)" and "(0 · 0)" that needs to replaced with a truth value. You need to include the parentheses, or you will make an error that may give you the wrong answer in the end. Highlighting (selecting) the wff before replacing it forces you to pay attention to which group of symbols actually constitute the wffs that are components of the larger compound statement. So, I would hit Return, to repeat the line above, and then highlight the first of these two wffs, which would look like:

(~(1 v 1) ⊃ (0 · 0))

and then replace the highlighted text. Of course, to do this, I have know that the truth value of the highlighted wff. (Look at those truth tables!) I can then "replace" this highlighted text by typing in "1" (the truth value of "(1 v 1)"). This gives us:

(~1 ⊃ (0 · 0))

After doing this, my shortest wff is no longer "(0 · 0)," but "~1." So, I repeat the process for the shortest wff I now have, "~1." So I highlight "~1," and then replace it with its truth value, giving me:

(0 ⊃ (0 · 0))

I then hit the Return key, to repeat this line. I can now highlight the next smallest wff, "(0 · 0)," and replace it by typing in its truth value. Thus:

(0 ⊃ 0)

And the truth value of this is 0, or "false," which is the answer I would enter on the top part of the screen.

So, by the time I have finished, what I will have inside the workspace box would look like this:

(~(P v Q) ⊃ (R · ~Q))

(~(1 v 1) ⊃ (0 · ~1))

(~(1 v 1) ⊃ (0 · 0))

(~1 ⊃ (0 · 0))

(0 ⊃ (0 · 0))

(0 ⊃ 0)

0

This is *a lot* of detail describing a process that you may have figured out by just playing around. But the point is to know what the process actually is. Since we know that truth tables work, the only reason for making mistakes in exercises like these is that we have made some kind of mistake in the process. So, we focus on stating *exactly* what that process is, so that we can be sure to follow it *exactly*. (Again, this is why computer's can do logic: they don't make mistakes so long as we can "tell" them the process for "swapping" certain symbols with other symbols. They don't know anything about truth tables. They just follow "mechanical" rules. And that is what we are doing here.)

Unknown Evaluations

I won't repeat the whole description of the process here. The process is the same as above, except that now we will evaluate compounds where we don't know the truth value of all the its components.

Intuitively, this isn't hard to understand. A disjunction is true if even one of its disjuncts is true. So, if I know that one of the is true, I know the truth value of the disjunction (i.e., that its true) even without knowing the truth value of the other disjunct. Likewise, a conjunction is false if even one of its disjuncts is false. So if I know that P=0 but I don't know the truth value of Q (which we represent as "Q=?"), I still know that the truth value of "(P · Q)" is 0. Given that P=0 and Q=?, I also know that truth value of "(P ⊃ Q)" is 1. (Look at the truth table for conditionals:

if the antecedent of a conditional is false, the conditional is true, no matter what the truth value of the consequent.) On the other hand if I am told that P=? and Q=0, then what can I infer about the truth value of "(P ⊃ Q)?" I can't determine it. A true antecedent (with a false consequent) would give me a false conditional, but a false antecedent (with a false consequent) would give me a true conditional. So, if P=? and Q=0, then the truth value of the conditional is unknown,
i.e., if P=? and Q=0, then (P ⊃ Q)=?

For the exercises in 6.4, use the exact same process as in 6.3 (i.e., always compute the shortest wffs first), but just do it when using "?" as standing for an unknown truth value. This forces you, again, to become extremely familiar with the truth tables for the 5 truth functional connectives. And you will need this for the sections to come.

I will leave you to figure out the strategies for doing the exercises in 6.5 an 6.6. There is already on line line to suggestions for doing the exercises in 6.7. You can find this at:

http://homepages.wmich.edu/~baldner/s67.pdf

I hope this helps!