Analysis of Algorithms
CS 375, Fall 2022
Project 3 Supplement

Project 3 Supplement: Truth Tables!

This document is referred to in Section 1 of the Project 3 assignment: This contains the *truth tables* that show the meaning (the *semantics*) of the operators for our set of propositional logic expressions (*PLEs*).

Definitions of Propositional Operations (Semantics)

Here are truth tables that define the four propositional operations from which we'll be building PLEs, along with examples.

The *not* operation The *not* operation (called "negation" by logicians) takes a single argument, and returns the negation of the value of its argument—that is, the *other* boolean value. Here's a truth table representing that:

P	(not P)
T	F
F	T

As with all propositional or boolean expressions, input argument P could only evaluate to one of two possible values: *True*, represented here by T; or *False*, represented by F. This truth table gives the value of (not P) in each of those cases: The top line says that if P evaluates to *True*, (not P) will evaluate to *False*; and if P evaluates to *False*, (not P) will evaluate to *True*. This gives a complete definition of how the *not* operation will evaluate on any valid input.

The *and* operation The *and* operation (called "conjunction" by logicians) takes two input arguments, and returns *True* exactly when *both* of the arguments evaluates to *True*. This is intended to capture what we *intuitively* mean by the word "and": The statement "Today is Saturday and it's sunny out" is only true in circumstance where it true that the day *is* Saturday and also it *is* sunny out; both have to be true for the "and" to be true. Here's a truth table representing that:

P	Q	(P and Q)
T	T	T
T	F	F
F	T	F
F	F	F

Note that there are four lines in this truth table: P could be either *True* or *False*, and for either of those options, Q could be either *True* or *False*, for the four possibilities in the table (one per line). (If it's not clear why there are four lines in the table, please ask me or a TA about it!) The values in the last column of the table show that (P and Q) evaluates to *True* exactly when both P and Q evaluate to *True*; (P and Q) evaluates to *False* in the other three cases.

The *or* operation The *or* operation (called "conjunction" by logicians) takes two input arguments, and returns *True* exactly when *one or both* of the arguments evaluates to *True*.

This is *close* to what we intuitively mean by the word "or," but not exactly. In English, sometimes the word "or" can have an *exclusive* meaning—like, when you're told you can get soup or salad with an entree at a restaurant, that doesn't mean you get both, you only get one, *excluding* the other option. The meaning for us when evaluating propositional logic expressions, however, is an *inclusive* meaning: For example, when I say that you can ask me *or* a TA about CS375 material, I mean that you are welcome to do both!

Here's a truth table representing that for our *or* operation:

P	Q	(P or Q)
T	T	T
T	F	T
F	T	T
F	F	F

There are four lines in this truth table, for the same reasons as for *and*, above. Here, the values in the last column of the table show that (P or Q) evaluates to *True* exactly when one or both of P and Q evaluate to *True*; (P and Q) evaluates to *False* only when both P and Q evaluate to *False*.

The *implies* operation The *implies* operation (called "implication" by logicians, and "if-then" by many people) takes two input arguments, and returns *True* exactly when *either* the first argument evaluates to *False or* the second argument evaluates to *True*, or both (this is an *inclusive* or).

This meaning of (P implies Q) isn't always a great match to what we intuitively mean by "If P then Q" in English. It *is*, however, very connected to what we discussed involving *vacuous truth*! This can be a bit subtle, so rather than put a lot into writing here, we'll just talk about it in class. Meanwhile, this truth table encodes the definition of *implies* that you should use for your algorithm for evaluating propositional logic expressions:

P	Q	(P implies Q)
T	T	T
T	F	F
F	T	T
F	F	T

There are four lines in this truth table, for the same reasons as for *and*, above. Note that the only case that evaluates to *False* is when the first argument evaluates to *True* and the second evaluates to *False*; in all other cases, the expression (P implies Q) evaluates to *True*.

Some example exercises In case you're looking to test your understanding, here are a few little exercises. *Do not submit these—they are not part of the project*—but I'll be happy to go over them with you, if you'd like! I've included an answer the first one, but the second and third are for you to try on your own.

1. What does PLE (p implies (q and (not (r or p)))) evaluate to when p is *True*, q is *True*, and r is *False*?

   Answer: By the truth table for *or* above, we know (r or p) evaluates to *True* when p is *True* and r is *False*. Then, by the truth table for *not*, we know (not (r or p)) evaluates to *False*. By the truth table for *and*, when q is *True* and (not (r or p)) evaluates to *False*, we know (q and (not (r or p))) evaluates to *False*. And finally, when p is *True* and (q and (not (r or p))) is *False*, the truth table for *implies* shows us that (p implies (q and (not (r or p)))) evaluates to *False*.

2. What does PLE (p implies (q and (not (r or p)))) evaluate to when p is *True*, q is *True*, and r is *True*?

3. What does PLE (p implies (q and (not (r or p)))) evaluate to when p is *False*, q is *True*, and r is *False*?