CSE 311: Foundations of Computing I

Homework 1 (due Wednesday, April 4 at 11:59 PM)

Directions: *Write up carefully argued solutions to the following problems. Your solution should be clear enough that it should explain to someone who does not already understand the answer why it works. You may use results from lecture, the theorems handout, and previous homeworks without proof.*

1. Translation into Logic (24 points)

Translate these English statements into logical language, decomposing each sentence as much as possible into atomic propositions.

(a) [10 Points] Use the same atomic propositions to turn all of these sentences into logic.

i) If the stack is empty, you can push but not pop,

ii) If the stack is full, you can pop but not push.

iii) If the stack is neither full nor empty, you can both push and pop.

(b) [6 Points] If we can develop cars that run on solar power or hydrogen power then we will not have a 3 degree rise in global temperatures this century and sea levels will rise less than 10 feet.

(c) [8 Points] Use the same atomic propositions to turn both of these sentences into logic.

i) You can either take the highway or surface streets to go to work.

ii) If you take the highway and there is no accident you will arrive in time but if there is an accident you will arrive late unless you take surface streets.

2. Nonequivalent Logical Statements (16 points)

Use truth assignments to show that the two propositions in each part are not logically equivalent:

(a) [4 Points] $p \vee q$ vs. $\neg(p \wedge q)$.

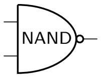
(b) [4 Points] $(p \oplus q) \vee (p \oplus r)$ vs. $p \vee q \vee r$.

(c) [4 Points] $(p \rightarrow q) \rightarrow (q \rightarrow p)$ vs. $(q \rightarrow p) \rightarrow (p \rightarrow q)$.

(d) [4 Points] $(((p \rightarrow q) \rightarrow r) \rightarrow s) \rightarrow p$ vs. $p \rightarrow (q \rightarrow (r \rightarrow (s \rightarrow p)))$.

3. All You Need is Nand, Nand, Nand... (20 points)

The **NAND** connective takes two propositions and evaluates to False when both propositions are True and evaluates to True otherwise. In circuit diagrams, the gate for **NAND** is denoted by



The **NAND** of p and q is written as $p \mid q$. Demonstrate that we can construct *all the other connectives* by just using **NAND** by writing propositional formulae for each of the following while *only using* ***NAND*** *connectives and no constants like T or F*:
Hint: It's okay to use a single input/output more than once.

(a) [5 Points] $\neg p$

(b) [5 Points] $p \lor q$

(c) [5 Points] $p \land q$

(d) [5 Points] $p \leftrightarrow q$

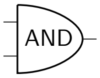
4. The Majority Wins! (10 points)

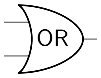
Find a compound proposition involving the propositional variables p, q, and r that is true precisely when a majority of p, q, and r are true. Explain why your answer works.

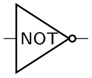
5. $MMMM$ Good! (10 points)

Using only...

AND Gates, **OR** Gates, and Inverters (**NOT** Gates),







draw the diagram of a circuit with **three** inputs that computes the function $M(p, q, r)$, where the following define M:

$$M(1, q, r) \coloneqq q$$
$$M(0, q, r) \coloneqq r$$

6. The Curious Case of The Lying TAs (10 points)

A new UW CSE student wandered around the Paul Allen building on their first day in the major. They found (as many do) that there is a secret room in its basements. On the door of this secret room is a sign that says:

> `All ye who enter, beware! Every inhabitant of this room is either a TA who always lies or a student who always tells the truth!`

(a) [5 Points] The CSE student walked into the room, and two inhabitants walked up to the student. One of them said "at least one of us is a TA." Determine (with justification) all the possibilities for each of the two inhabitants.

(b) [5 Points] Three inhabitants walk up to the CSE student and surround the UW CSE student. One of them says "every TA in this circle has a TA to their immediate right." Determine (with justification) all the possibilities for each of the three inhabitants.

7. EXTRA CREDIT: XNORing (0 points)

For two bits a and b, we define $\texttt{XNOR}(a, b) = \neg(a \oplus b)$. You are given two memory registers, each with the same number of bits. You have an operation, $\texttt{XNOR}(R_1, R_2)$, which takes two registers, R_1 and R_2, performs bitwise XNOR between them, and stores the result in R_1.

Show how you can swap the contents of the two registers using a sequence of XNORs without temporary memory registers.