CSE 311: Foundations of Computing (Autumn, 2015)

Homework 1 Out: Friday, 2-Oct. Due: Friday, 9-Oct, **11:59PM** on **Gradescope**

1. English Translation [12 points]

Write each of these statements in the form "if p, then q" in English.

(a) To become a president of the Unites States, it is necessary to be a natural-born citizen.

(b) Whenever I hike for an 10 hours, I have to rest for a day.

(c) To travel to the moon, it is enough to have a billion dollars.

(d) TAs will grade the solutions to the homework when the deadline is passed.

2. Nonequivalent Logical Statements [12 points]

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

(a) $p \vee q$ vs. $\neg(p \wedge q)$

(b) $(p \oplus q) \vee (p \oplus r)$ vs. $p \vee q \vee r$

(c) $(p \rightarrow q) \rightarrow (q \rightarrow p)$ vs. $(q \rightarrow p) \rightarrow (p \rightarrow q)$

(d) $(((p \rightarrow q) \rightarrow r) \rightarrow s) \rightarrow p$ vs. $p \rightarrow (q \rightarrow (r \rightarrow (s \rightarrow p)))$

3. Equivalence of NAND [18 points]

Define the NAND logical connective: $p\#q \equiv \neg(p \land q)$. In this part, we will see that you can design a digital circuit made up of only the NAND connectives for any logical proposition. For each of the following propositions, write a logically equivalent proposition using only the connective # and possibly the literals **T** and **F**.

(a) $\neg p$

(b) $p \land q$

(c) $p \lor q$

(d) $p \oplus q$

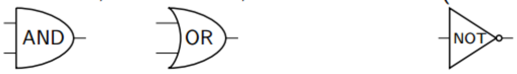
(e) $p \rightarrow q$

(f) $p \leftrightarrow q$

4. Simple circuit design [10 points]

Using only the following gates:

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



Design a circuit with **three inputs** that computes the function $M(p, q, r)$ where

$$M(T, q, r) = q \wedge \neg r$$

$$M(F, q, r) = \neg q \vee r$$

Extra credit: XNORing

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

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