Homework 1 Logic

Instructions:

- The assignment has to be uploaded to Blackboard by the due date. *NO* assignment will be accepted after 11:59pm on that day.
- We expect that you will study with friends and often work out problem solutions together, but *you must write up your own solutions, in your own words*. **Cheating will not be tolerated**. Professors, TAs, and peer tutors will be available to answer questions but will not do your homework for you. One of our course goals is to teach you how to think on your own.
- You may turn in work to Blackboard that is either handwritten and scanned, written in a word processor such as Word, or typeset in LaTeX. In the case of handwritten work, we may deduct points if the scan is upside down or the work is illegible.
- *To get full credit*, **show INTERMEDIATE steps** leading to your answers, throughout.

Problem 1 Logical formulas

Write a logical formula that is true if and only if the described condition is true. For example, if the description is "A, B, and C are all true, but D is not," you would write $A \wedge B \wedge C \wedge \neg D$. Use $\wedge$, $\vee$, and $\neg$ for AND, OR, and NOT.

i. At least one of A, B, or C is true.

ii. A, B, and C, all have the same value (of true or false).

iii. At least two of A, B, and C are true.

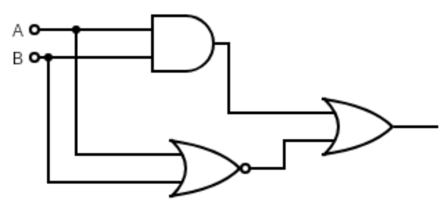
iv. At least one member of each group must be true: group 1 contains A, B, and C, group 2 contains D, E, and F, and group 3 contains A, F, and G. (Thus if A is true, it satisfies the requirement for both group 1 and group 3.)

v. Exactly one of A, B, or C is true.

vi. At least one of A, B, or C is false. Do not use $\vee$ (OR).

Problem 2 : Truth Tables, Statements, and Circuits

Each row of this table corresponds to an equivalent truth table, boolean statement, and circuit. (10 points) Fill out the following table. For this problem, you may hand write/draw the table and put that picture into your PDF.

<table>
<tr><th>Truth Table</th><th>Boolean Statement</th><th>Circuit</th></tr>
<tr><td><table><tr><th>A</th><th>B</th><th>Out</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td></td><td></td></tr>
<tr><td></td><td>$\neg(A \lor B) \land (\neg A \lor B)$</td><td></td></tr>
<tr><td></td><td></td><td>
</td></tr>
</table>

Problem 3 : English Statement Negations

Formalize each statement with formal logic and the following statements and $\land, \lor, \neg, \implies$. Then negate the formal logic you wrote. For full credit, fully distribute your negation.

- P = Alex wants to eat lunch at the park
- Q = The park is open
- R = Pigeons will eat lunch at the park
- S = Pigeons are at the park

i. (2 points) Alex wants to eat lunch at the park but the park is not open.

ii. (2 points) If the park is open and the pigeons are not at the park, then Alex wants to eat lunch at the park.

iii. (2 points) Either the pigeons are not at the park or the park is not open.

iv. (2 points) If the pigeons will eat lunch at the park, then if the pigeons are at the park and the park is open, then Alex does not want to eat lunch at the park.

v. (2 points) The park is open and either the pigeons will eat lunch at the park or Alex wants to each lunch at the park.

Problem 4 : NAND: The universal gate

In this problem we'll explore the fact that all logical circuits can be implemented using just NAND gates.

i. (5 points) Let's denote p NAND q as $p \barwedge q$. Write a logical expression for the three circuits corresponding to AND, OR, and NOT.

ii. (5 points) Validate your three logical expressions with three truth tables. For clarity and full credit, show each variable and distinct statement in a separate column, culminating in your final formula. For instance, if we wanted a table for the statement $(p \wedge q) \vee q$, we would need one column for p, one for q, one for $p \wedge q$ and one for the whole statement.

Problem 5 : Satisfiabilty Problem

A boolean formula is satisfiable if there exists some variable assignment that makes the formula evaluate to true. Namely, a boolean formula is satisfiable if there is some row of the truth table that comes out true. Determining whether an arbitrary boolean formula is satisfiable is called the *Satisfiability Problem*. There is no known efficient solution to this problem, in fact, an efficient solution would earn you a million dollar prize. While this is hard problem in computer science, not all instances of the problem are hard, in fact, determining satisfiability for some types of boolean formulae is easy.

i. First, let's consider why this would be hard. If you knew nothing about a given boolean formula other than that it had n variables, how large is the truth table you would need to construct? Please indicate the number of columns and rows as a function of n

ii. Now consider the following 100 variable formula.

$$x_1 \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge \ldots \wedge (\neg x_{99} \vee x_{100})$$

Without constructing a truth table, how many satisfying assignments does this formula have, explain your answer.

iii. Now consider an arbitrary 3-DNF formula with 100 variables and 200 clauses. 3-DNF means that the formula is in disjunctive normal form and each clause has three literals. (A literal is the instantiation of the variable in the formula, so for x, $\neg x$ or x.) An example might be something like:

$$(\neg x_1 \wedge x_3 \wedge x_{10}) \vee (\neg x_3 \wedge x_{15} \wedge \neg x_{84}) \vee (x_{17} \wedge \neg x_{37} \wedge x_{48}) \vee \ldots \vee (\neg x_{87} \wedge \neg x_{95} \wedge x_{100})$$

What is the largest size truth table needed to solve this problem. What is the maximum number of such truth tables needed to determine satisfiabilty.