Fall 2016 COT 3100 Section 1 Homework 1
Assigned: 8/22/2016
Due: 8/26/2016

1) Create a truth table from scratch to evaluate the Boolean expression below. Your table should have 8 rows. While the number of columns may vary between responses make sure to create enough intermediate columns (so a minimum of 6 columns).

$(p \barwedge q) \vee ((r \barvee p) \wedge q)$

p	q	r	$(p \wedge q)$	$\neg(p \wedge q)$	$r \vee p$	$\neg(r \vee p)$	$\neg(r \vee p) \wedge q$	$\neg(p \wedge q) \vee (\neg(r \vee p) \wedge q)$
F	F	F	F	T	F	T	F	T
F	F	T	F	T	T	F	F	T
F	T	F	F	T	F	T	T	T
F	T	T	F	T	T	F	F	T
T	F	F	F	T	T	F	F	T
T	F	T	F	T	T	F	F	T
T	T	F	T	F	T	F	F	F
T	T	T	T	F	T	F	F	F

2) Perform the following bitwise operations on 8-bit unsigned integers. Please show your work (so that we know you understand how to do this instead of just plugging it into a compiler.)

a) `47 ^ 98 (xor)`
b) `13 & 211 (and)`
c) `99 | 131 (or)`
d) `13 << 3 (left shift)`
e) `198 >> 4 (right shift)`

a) 47 = **0**101**11**1
98 = **1**100**0**1**0**
xor: **1**00**11**0**1**
1001101 = **77**

b) 13 = 00001101
   211 = 11010011
   and: 00000001
   1 = **1**

c) 99 = 01100011
   131 = 10000011
   or: 11100011
   11100011 = **227**

d) 13 = 1101
   leftshift(3): 1101000
   1101000 = **104**

   alternatively:
   13 * 2^3 = 104

e) 198 = 11000110
   rightshift(4) = 1100
   1100 = **12**

   alternatively: 198 / 2^4 = 12

3) Show that the two following expressions are logically equivalent using the laws of logic:

a) $p \vee ((p \wedge (r \vee (\acute{r} \vee p))) \wedge q)$

b) p

Make sure to show every step and clearly label each step. If you feel it's obvious you may combine two steps in one, listing both rules applied.

$p \vee ((p \wedge (r \vee (\neg r \vee p))) \wedge q) - Given$

$p \vee ((p \wedge (r \vee \neg r \vee p)) \wedge q) - Associative$

$p \vee ((p \wedge (T \vee p)) \wedge q) - Inverse\ Laws (r \vee \neg r \Leftrightarrow T)$

$p \vee ((p \wedge T) \wedge q) - Domination\ Law (T \vee p \Leftrightarrow T)$

$p \vee (p \wedge T \wedge q) - Associative\ Law$

$p \vee (p \wedge q) - Identity\ Law (p \wedge T \Leftrightarrow p)$

$p - Absorbtion\ Law ((p \vee (p \wedge q)) \Leftrightarrow p)$

4) Create your own logic equivalence problem by finding two logical expressions that appear different but are logically equivalent. After showing your two expressions, explain ***how*** you created the problem. (Note: Most of the grade on this problem is based upon your explanation and not the two expressions.)

Example Response:

If you don't do your homework and you don't pay attention in class, then you will not get a good grade.

You can either do your homework, pay attention in class, or get a bad grade.

Three boolean variables can be created:
p: doing your homework
q: paying attention in class
r: getting a bad grade in the class
The initial expression can be shown as $\neg p \wedge \neg q \Rightarrow \neg r$

Using the definition of implications, the former can be rewritten as: $\neg r \vee \neg(\neg p \wedge \neg q)$

Which, using Demorgan's law, can be expressed as $\neg r \vee \neg\neg p \vee \neg\neg q$

Which can be further simplified as $\neg r \vee p \vee q$.

This can be distributed in a different order into $p \vee q \vee \neg r$, which creates the second expression.

5) Recount a paragraph or so about the contributions of Ada Lovelace, related to computer science. (You may do your research from anywhere, but please do not plagiarize. Write things in your own words.)

Ada Lovelace was a mathematician who is credited with being a first “computer programmer”. Along with Charles Babbage, the two worked on one of the earliest models of computers, the Analytical Machine. Her notes describes a program to compute Bernoulli numbers on it, which can be considered one of the computer programs written.

6) The recommended course textbook uses Boolean logic to describe a valid solution to a Sudoku puzzle. Imagine a tic-tac-toe board filled with trues and falses where the trues stand for a square with an X in it and the falses stand for a square with an O in it. Assign names to appropriate Boolean variables and create a Boolean expression that is true if and only if the team with X has won. (Note that since the contents of squares are limited to two things, we can only evaluate tic-tac-toe boards that have an X or O in every square.)

Assume the board looks like such:

X	X	O
X	X	O
O	O	X

You can create variables for each square on the board, a through i. This board can be represented as:

a	b	c
d	e	f
g	h	i

a: True
b: True
c: False
d: True
e: True
f: False
g: False
h: False
i: True

It is considered that 'X wins' if X gets three in a row, either horizontally, vertically, or diagonally (any corner to any opposite corner).

Thus, one can create a boolean expression using variables a through i as follows:

$$(a \wedge b \wedge c) \vee (d \wedge e \wedge f) \vee (g \wedge h \wedge i) \vee (a \wedge d \wedge g) \vee (b \wedge e \wedge h) \vee (c \wedge f \wedge i) \vee (a \wedge e \wedge i) \vee (c \wedge e \wedge g)$$

where each clause of ANDs corresponds to a “winning” line. If the expression is true, at least one of the clauses would be true, and thus, X would have a winning line, and would win. Likewise, if X wins, X would have a row, column, or diagonal corresponding to one of the clauses, and the expression would be true. Thus, the boolean clause is true if, and only if, X wins.

6 alternate solution) An alternate solution to this problem utilizes the symbols used for the Sudoku problem, where we can succinctly express several terms being and'ed together or or'ed together.

Using notation more similar to this, let the variable p(x, y) be set to true if and only if the entry in row x, column y is set to the character 'X'. If it's false, then that entry is set to 'O'. (These are the only possible settings for each square for this problem.)

We want to create a Boolean expression that is true if and only if there exists a row, column or diagonal that is completely set to true. To check to see if multiple items are all set to true, we must use an and. To see if at least one of many options is true, we use an or.

Thus, to check to see if team X has won a row, we use the following Boolean expression:

$$\bigvee_{x=1}^{3}(\bigwedge_{y=1}^{3}p(x,y))$$

It follows that to see if team X has won a column, we change the order of indexes and use the following Boolean expression:

$$\bigvee_{y=1}^{3}(\bigwedge_{x=1}^{3}p(x,y))$$

Finally, we have to create two separate checks for each diagonal as follows (the left portion of the expression is for the forward diagonal, the right portion for the backward diagonal):

$$[\bigwedge_{x=1}^{3}p(x,x)]\vee[\bigwedge_{x=1}^{3}p(x,4-x)]$$

Putting it all together we get:

$$[\bigvee_{x=1}^{3}\left(\bigwedge_{y=1}^{3}p(x,y\right))]\vee[\bigvee_{y=1}^{3}\left(\bigwedge_{x=1}^{3}p(x,y\right))]\vee[\bigwedge_{x=1}^{3}p(x,x)]\vee[\bigwedge_{x=1}^{3}p(x,4-x)]$$

Note: Although this expression is short enough to "brute force", for learning purposes, it's better to practice using the short-hand notation so that in new situations where "brute force" is too onerous, one can be comfortable using this more succinct notation.