CSE 311: Foundations of Computing I

Homework 2 (due Wednesday, Oct 12 at 11:59 PM)

Directions: *Write up carefully argued solutions to the following problems. The first task is to be complete and correct. The more subtle task is to keep it simple and succinct. 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 any results proven in lecture without proof. Anything else must be argued rigorously. Unless otherwise specified, all answers are expected to be given in closed form.*

1. Logical Equivalences (20 points)

Prove the following assertions using equivalances. You can use commutativity and associativity an arbitrary number of times in a single line of the proof.

(a) [8 Points] $p \to (q \to r) \equiv q \to (p \to r)$.

(b) [12 Points] $((p \to q) \wedge (r \to \neg q)) \to (r \to \neg p) \equiv T$.

2. Boolean Algebra (10 points)

Prove that $X' + ((X + Y) \bullet X)' = X'$ using the axioms and theorems of boolean algebra.

3. $X \cdot Y$ (20 points)

In this question, you will construct a circuit that takes a pair of two-bit integers $(x_1x_0)_2$ and $(y_1y_0)_2$ and computes the four output bits for their integer product.

For example, if $(x_1x_0)_2 = (10)_2$ and $(y_1y_0)_2 = (11)_2$ then we're doing the following computation:

$$\begin{array}{cccc} & & 1 & 0 \\ \times & & 1 & 1 \\ \hline & & 1 & 0 \\ + & 1 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 \end{array}$$

(a) [8 Points] Give sum-of-products forms for the two output bits of the product, $(a_1a_0)_2$, of $(x_1x_0)_2$ and $(y_0)_2$. Do the same for $(x_1x_0)_2$ and $(y_1)_2$ yielding $(b_1b_0)_2$. These are the bits produced as part of applying the usual elementary school method for multiplying numbers.

Drawn out, this looks like:

$$\begin{array}{ccc} & x_1 & x_0 \\ \times & y_1 & y_0 \\ \hline & a_1 & a_0 \\ b_1 & b_0 & 0 \end{array}$$

(b) [12 Points] Use the minimized sum-of-products forms for one-bit adders given in class, together with the results of the above two products, to produce sum-of-products forms for the output bits z_3, z_2, z_1, z_0. Some of the inputs you give to the one-bit adders may be constants. Use Boolean algebra to minimize the resulting sum-of-products form as a sum-of-products using only x_1, x_0, y_1, y_0.

Drawn out, this looks like:

$$\begin{array}{cccccc} & & & a_1 & a_0 \\ + & & b_1 & b_0 & 0 \\ \hline & z_3 & z_2 & z_1 & z_0 \end{array}$$

4. Counting Courses with Combinational Logic (20 points)

In lecture 4, we considered a combinational logic example about days of class.

(a) [5 Points] Write c_1 in the product of sum form.

(b) [6 Points] Simplify the sum of product forms of c_0, c_2 using boolean algebra axioms and theorems. Make sure to cite which axioms and theorems you are using when simplifying.

(c) [9 Points] Simplify the sum of product forms of c_1 using boolean algebra axioms and theorems.

5. Hats, 311, and Bunnies (20 points)

It is often useful to assert that something exists and is unique. For instance,

> There exists a unique course at UW with the course number CSE 311.

(a) [4 Points] Let the domain of discourse be all vases. Let B be a constant representing "the painted vase in Shayan's office." Translate "If there is a vase that is painted, then it is the painted vase in Shayan's office." into first-order logic. Don't forget to define predicates.
Hint: Remember that a "constant" is a particular object in the domain. For example, "0" and "1" are constants in the domain of natural numbers.

(b) [4 Points] Let the domain of discourse be all university courses. Now, use an idea similar to part (a) to translate the sentence about the CSE 311 course number into first-order logic. Don't forget to define predicates.

For the remaining parts, we define the following predicates:

- Let $R(x)$ be "x is a rabbit."
- Let $H(x)$ be "x hops."

For these parts, let the domain be mammals. Translate each of the following into English.

(c) [3 Points] $\exists x(R(x) \wedge H(x))$

(d) [3 Points] $\forall x(R(x) \rightarrow H(x))$

(e) [3 Points] $\forall x(R(x) \wedge H(x))$

(f) [3 Points] $\exists x(R(x) \rightarrow H(x))$

6. Domain of discourse (10 points)

(a) [5 Points] Give examples of predicates P and Q and a domain of discourse so that the two statements

$$\forall x(P(x) \vee Q(x)) \text{ and } (\forall x P(x)) \vee (\forall x Q(x))$$

are not equivalent.

(b) [5 Points] Give an example where they have the same meaning.

(c) [0 Points]**Mini extra credit:** Say that a logical connective $p \otimes q$ is *non-trivial* if it sometimes evaluates to false and sometimes evaluates to true. Is there any such connective $p \otimes q$ such that $\exists x(P(x) \otimes Q(x))$ and $\exists x P(x) \otimes \exists x Q(x)$ are logically equivalent for every domain and choice of predicates? Explain.

7. Extra credit: Comparison circuit (-NoValue- points)

In this problem, you will design a circuit with a minimal number of gates that takes a pair of four-bit integers $(x_3x_2x_1x_0)_2$ and $(y_3y_2y_1y_0)_2$ and returns a single bit indicating whether $x_3x_2x_1x_0 < y_3y_2y_1y_0$. See the following table for some examples.

$x_3x_2x_1x_0$	$y_3y_2y_1y_0$	$x_3x_2x_1x_0 < y_3y_2y_1y_0$
0101	1011	1
1100	0111	0
1101	1101	0

Design such a circuit using at most 10 AND, OR, and XOR gates. You can use an arbitrary number of NOT gates, and a single gate can have multiple inputs. (Extra credit points start at 10 gates, but if you can use fewer, you will get even more points.)