CSE 311: Foundations of Computing (Spring, 2015)

Homework 2 Out: Friday, 10-Apr. Due: Friday, 17-Apr, **before class** on **Gradescope**

Additional directions: You should write down carefully argued solutions to the following problems. Your first goal is to be complete and correct. A secondary goal is to keep your answers simple and succinct. The idea is that your solution should be easy to understand for someone who has just seen the problem for the first time. (Re-read your answers with this standard in mind!)

1. Logical equivalence ($10 = 6 + 4$ points)

(a) Prove that $\big((p \to q) \land (q \to \neg r)\big) \to \big(\neg(p \land r)\big)$ is a tautology using logical equivalences. You should use only the equivalences in the "Properties of logical equivalences" handout on the web page (http://courses.cs.washington.edu/courses/cse311/15sp/documents/logical_properties.pdf). You don't need to label the equivalences by name, but you should use only one equivalence per step, except for associativity and commutativity of $\lor$ and $\land$ which you can use multiple times in the same line (for a set of ANDs or ORs at the same level, e.g. $a \land b \land c \land d \equiv (c \land b) \land (d \land a)$ can be one step).

(b) Show that $\big((p \to q) \to (q \to \neg r)\big) \to (\neg p \to \neg r)$ is **not** a tautology.

2. Zombies are people too, sort of (10 points)

Define the predicates $Z(x)$ = "x is a Zombie," $H(x) =$ "x is a humanoid", and $E(x) =$ "x eats brains." For the purposes of the problem, you can assume that every Zombie is a humanoid. The domain consists of all objects in the Milky Way galaxy.

Consider the question: "Does there exist a humanoid such that if he/she is a Zombie, then every humanoid that eats brains is a Zombie?" Translate it into logical notation, and then explain when it is true and when it is false based on the state of the galaxy. (Be careful about the situation when the galaxy contains no humanoids!)

3. Parenthetical asides (10 points)

(a) Give an example of predicates and domains such that the statements $\exists x \left(P(x) \oplus Q(x)\right)$ and $\exists x\, P(x) \oplus \exists x\, Q(x)$ are not equivalent.

(b) Given an example where they **are** equivalent.

(c) [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.

4. Try not to be so negative (12 points)

For each of the following, translate the English statement into first-order logic and then negate it. Your answer should put the negation symbols as far inside as possible. The domain of discourse is the set of all people in Washington and the set of all paper-based objects. The predicates are Student(x), Exam(x), Takes(x, y) (meaning that student x takes exam y), and Equal(x, y) (meaning that x and y are the same object).

(a) "All exams are taken by at least one student."

(b) "Every student takes exactly one exam."

(c) Translate the following statement into English (try to make the translation as simple as possible):

$$\forall x \forall y \left(\left(\mathrm{Student}(x) \land \mathrm{Student}(y) \land \neg \mathrm{Equal}(x, y) \right) \rightarrow \left(\exists z\, \mathrm{Exam}(z) \land \mathrm{Takes}(x, z) \land \neg \mathrm{Takes}(y, z) \right) \right)$$

5. Multiplication (15 points)

In this problem, you will design 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 of their integer product. You will simulate the elementary-school algorithm. For example to multiply $2 = (10)_2$ and $3 = (11)_2$:

```
   1 0
   1 1
------
   1 0
 1 0 0
------
 1 1 0
```

(a) Give sum-of-product 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 supposed to be the bits produced as part of applying the usual elementary school method for multiplying numbers.)

(b) Use the minimized sum-of-product forms for one-bit adders give in lecture, 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.

(c) Draw circuit diagrams for the results.

6. **Extra credit**: Get these MF Pokémon off this MF private jet.

Consider the predicates $\mathrm{Pokemon}(x), \mathrm{Celebrity}(x), \mathrm{Jet}(x), \mathrm{WearsPants}(x)$, $\mathrm{Flies}(x, y)$ which denote, respectively, that x is a Pokémon, celebrity, or jet, that x wears pants, and that person x flies on jet y. We also use $\mathrm{Alive}(x)$ to denote that x is alive. (For the purposes of this problem, celebrities are alive, even if they're a little dead inside. Pokémon are definitely alive.) $\mathcal{T}$ is Taylor's jet and is $\mathcal{B}$ Bieber's jet. The domain of discourse is the set of all alive things and all flying things.

Suppose the following are true:

(i) $\forall x \left(\mathrm{Flies}(x, \boldsymbol{\mathcal{T}}) \leftrightarrow \left(\mathrm{Pokemon}(x) \wedge \mathrm{WearsPants}(x)\right)\right)$

(ii) $\forall x \left(\mathrm{Flies}(x, \boldsymbol{\mathcal{B}}) \leftrightarrow \left(\mathrm{Celebrity}(x) \wedge \mathrm{WearsPants}(x)\right)\right)$

(iii) $\forall j \left(\mathrm{Jet}(j) \to \exists x \left(\mathrm{Alive}(x) \wedge \neg \mathrm{Flies}(x, j)\right)\right)$

(iv) $\forall j \left(\mathrm{Jet}(j) \to \exists h \left(\mathrm{Jet}(h) \wedge \forall x \left(\mathrm{Alive}(x) \to \left(\mathrm{Flies}(x, h) \leftrightarrow \neg \mathrm{Flies}(x, j)\right)\right)\right)\right)$

Explain precisely why there is at least one Pokémon who is not allowed to fly on Taylor's jet.

7. **Extra credit:** Aliens who like to run their computers backwards

Why does time run forward and what does it mean?

https://www.youtube.com/watch?v=zvFFNkg7Mvo

Suppose in one of our weekend trips through the galaxy, we encounter an Alien species who has yet to develop computers because they're superstitious. They only do anything if it can also be done backwards. Computers don't seem to go backwards very well: You might start with a circuit that takes two n-bit numbers, adds them together, and outputs an $(n+1)$-bit number. But you can't run that $(n+1)$-bit number through the computer backwards to recover the inputs.

Your goal is to help them out. First, design a three-input / three-output circuit that can implement $\mathrm{AND}(x, y)$, $\mathrm{OR}(x, y)$, and $\mathrm{NOT}(x)$. The third input is used to control what functionality the circuit computes. It should be the case that, from the three outputs, we can always recover the three inputs uniquely. When you've finished, you will have constructed a single gate that can be run *backwards*---an Alien gate.

Now show them how to use this to create a full-blown computer. Suppose you have a normal human circuit C that takes n-input bits $x_1, x_2, \dots, x_n$ and outputs a value $C(x_1, x_2, \dots, x_n)$. The circuit just uses normal AND, OR, and NOT gates. You should describe how to design an Alien circuit C' of the following form:

C' consists of layers of logic. At the beginning are the n input bits. Every layer has n wires coming in and n wires coming out. The first bit of the final layer should compute the value $C(x_1, x_2, \dots, x_n)$. Moreover, every layer should have the property that from its n output wires, you can uniquely recover the values on the n input wires. In other words, the whole computer can be run backwards. If the normal human circuit was built using s NOT, AND, and OR gates, how many layers does your C' use?