HW1: Hello There

Due: Sep 15, 2010

Include your *full name*, *CS login*, and the problem number(s) on each piece of paper you hand in, and please staple your pages together before handing in.

While collaboration is encouraged in this class, please remember not to take away notes from collaboration sessions other than your scheduled lab section.

In general, if you submit a complicated, messy FSM and include no explanation of how it works, it will probably not be graded.

Problem 1

Draw a circuit and a straight-line program for the Boolean function whose value is **True** exactly when x or y or both is **True** and z is **False**.

The following questions are lab problems. Please remember to prepare a solution for your assigned problem before going to your lab section.

Lab Problem 1

When it was first discovered that there are problems that cannot be solved by computers, it was a very surprising result. An equally unsettling discovery by Gödel was that some mathematical statements have no proofs. In this problem, we will do a high-level proof of this fact. Instead of using mathematical statements, we will simply use English sentences.

This is the sentence we are trying to prove:

There exists a true sentence for which there is no proof.

We will argue this by contradiction.[1] That is, we will assume the following statement, then obtain a contradiction:

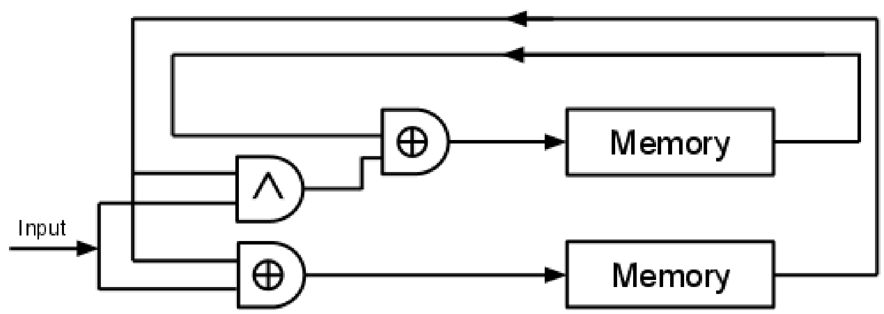
[1] We are being informal here, because we have not given a mathematical formalization of the statement we are trying to prove.

Assumption: All true sentences have proofs.

You may assume that any statement for which there exists a proof is true.

Lab Problem 2

Convert the following circuit diagram for an FSM into a state diagram (where each state is represented by a vertex and each transition is a directed edge).



In the above diagram, $\bigoplus$ is exclusive OR and $\bigwedge$ is AND.

Hint: Think of the memory bits as a single 2-bit binary integer.

Lab Problem 3

Vending Machines and FSMs: Finite state machines can be used for many purposes. Here you'll design a finite state machine to model your own vending machine. Your vending machine will take quarters and distribute two flavors of juice, Apple Juice and Orange Juice, each for \$.50.

In order to model the vending machine as an FSM, the input and action alphabets must be defined. Your vending machine has four possible inputs at each step, **Q, O, A**, and **C** which have the following meanings:

Q: A quarter (\$.25) has been inserted into the machine.

O: The *Orange Juice* button has been pressed.

A: The *Apple Juice* button has been pressed.

C: The *Return Change* button has been pressed.

Note: We assume that multiple buttons are not pushed at the same time and that a quarter cannot be inserted at the same time as a button pressed.

Your FSM has four possible actions, **DO, DA, N**, and **R**. These have the following meanings:

DO: Distribute an Orange Juice.

DA: Distribute an Apple Juice.

N: Do nothing.

R: Return one quarter.

You should think about your design carefully. Make sure that your machine behaves reasonably. An example of an unreasonable behavior would be to distribute an Orange Juice when someone presses the *Apple Juice* button. Also note that the *Return one quarter* action will have the effect of returning at most one quarter. That is, *Return one quarter* after $.50 has been inserted will result in returing a single quarter, not two quarters.

The FSM you submit should be given as a graph (circles and lines). Briefly discuss any significant design decisions that you make.