CSEP590 – Problem Set 2
Summer 2003
Due: Monday, July 14 by midnight.
Directions: Answer each of the following questions. Email your solutions in doc, pdf, or html format to evan@cs.washington.edu

1. Recall our discussion of Automata. This problem will draw on most of what you learned in lecture. Here's a little make-believe story to provide some motivation. Let's suppose that a certain, devious computer hacker has discovered a way to compromise the security of the Microsoft Outlook binary by inserting a section of malicious code into the binary. Researchers at CERT have discovered though that the hacker wasn't entirely perfect. In fact, each inserted section begins with the binary sequence of the form 1110^*111 (for purposes of review, 0^* means "zero or more copies of 0." So for example, 111111, 1110111, 11100111, 111000111… are examples of 1110^*111). CERT would like design an Automaton M that recognizes this sequence. If M detects the sequence 1110^*111 in the binary, M should alert the user that a suspicious sequence has been detected by entering an "alert" state and cycling back to this alert state for any remaining input. For example, if the entire binary was 1010100011100111100101, M should detect this as suspicious because the binary contains the sequence 1110^*111 (specifically, 11100111).

a. Design an Automaton M that correctly models this behavior. Give both a graphical schematic of M (showing its transitions and states like we saw in class), and a formal definition of M of the form $M = (Q,E,T,q_0,l)$, where Q, E, T, q_0, and l are defined from class.
b. Draw the first 4 levels of the complete execution tree of M. Is the execution tree finite or infinite?
c. Name one possible length-12 partial execution of M.
d. Suppose we introduce a variable "ctr" (initialized to 0) that keeps track of the number of 0's seen between the first occurrence of 111 in the sequence and the following occurrence of 111 (that is, ctr = 2 for the sequence …11100111…). However, ctr is reset to 0 if the 1110^*111 sequence doesn't correctly complete (that is, if you have …11100110…, then ctr was equal to 2, but should now be reset to 0 to await the next possible 1110^*111 sequence). Which transition(s) should update/change the value of ctr? Point these transitions out in your schematic of M and label the transitions to show the nature of the change.
e. Suppose that if the value of ctr is ever at least 5, then M can't proceed as usual and instead is only able to transition to a new, deadend state. This can be accomplished by adding guards of the form "if ctr < 5" to some of your transitions. Thus, M can only transition as normal as long as ctr < 5. Then, you can add the new deadend state with incoming transition(s) with the guard "if ctr ≥ 5", thereby ensuring that transition to the deadend state is only possible if ctr ≥ 5. On your schematic of M, add the deadend state, new transitions, and mark those transitions that need to be guarded with either of the two guards, showing which guard is needed for which transition.

f. Finally, if we wished to unfold this version of M with guards and variable ctr, how many global states would there be in the unfolded Automaton? You don't have to draw the unfolded version of M, just make an argument supporting your number.

2. This question will give you practice with formally modeling a system as a Kripke Structure, in particular, a digital, synchronous circuit. Consider the following description of a synchronous, digital circuit D. D maintains 3 bits of state, called b_1,b_2,b_3. Initially, $b_1,\ldots,b_3$ are set to 110 or 010. The 3 bits of state are updated according to the following logic:

$$b_1' = (b_1 \vee b_3) \wedge b_2$$
$$b_2' = \neg b_2 \wedge b_3$$
$$b_3' = b_1 \wedge (\neg b_2 \vee b_3)$$

But, there's one problem…you only have NAND gates to build D from. Recall from problem set 1 that the NAND gate comprises a functionally complete set. Therefore, we can luckily translate the equations for the bit updates using just NAND gates! Your task in this problem is as follows. You need to formally model D as a Kripke Structure along the lines of what we saw in class.

a) Translate the update equations for bits $b_1,\ldots,b_3$ so that the equations use only NAND gates. Call the resulting circuit that uses NAND gates only, circuit D'.
b) Now, let's model D' as a Kripke Structure. First, we need to determine the 2 first order formulae that represent the initial states of D' and the transitions of D'. What are the values of V = the set of system variables for D', the domain of variables in V, $S_0(V)$ = the formula representing the initial states of the system, and R(V,V') = the formula representing the transitions of the system?
c) Now that you have defined the first order formulae that represent the circuit D' and the set of variables V (and implicitly V'), your final task is to use these formulae to define the Kripke Structure K modeling D'. What is the Kripke Structure $K = (S,S_0,R,L)$ as defined in class for circuit D'?

3. In the final problem, you will create a formal model as a Kripke Structure of a hypothetical system. Consider the following description of an elevator. The elevator spans 5 floors, numbered 1 through 5. The elevator has only 2 buttons, an "up" button and a "down" button. Initially, the elevator starts at floor 1, the building's lobby, and the up and down buttons are unlit (not pressed). When the up button is pressed, the elevator goes up one floor, stops, and awaits for another button to be pressed. Similar behavior occurs for the down button. Pressing the up button at floor 5 does nothing, as does pressing the down button at floor 1. If both the up and down button are pressed at the same time, the elevator does nothing. When the elevator is on floor 1, a bell rings in the elevator indicating that the elevator has reached the lobby. Your task is to model this elevator as a Kripke Structure $K = (S,S_0,R,L)$ as defined in class. As in problem 2, you will first have to carefully define your set of system variables V and the domains that your variables range over (recall that valuations of V correspond to states in the model). Then, you will have to define the 2 formulae $S_0(V)$ and R(V,V') for your system. With

those in hand, you can define the Kripke Structure K in a straightforward manner. Be sure to account for the ringing bell! Explain and justify all abstractions/assumptions you use in constructing your model. A formal definition of K without any English descriptions and justifications will receive little credit.