**CSEP590 – Problem Set 4**
**Summer 2003**
**Due: Thursday, July 31 by midnight.**
**Directions: Answer each of the following questions. Email your solutions in doc, pdf, or html format to evan@cs.washington.edu**

**Note:** you should plan on getting an early start on this problem set, especially if you haven't read over the SMV paper yet…

1. This question will deal with Binary Decision Diagrams. Consider the Boolean formula $f(x,y,z) = (!x+y+!z)\bullet(x+!y+z)\bullet(x+y)$. For the variable orderings below, compute the unique reduced OBDD $B_f$ of f with respect to that ordering.

   a. [x,y,z]
   b. [y,x,z]
   c. [z,x,y]
   d. Find an ordering of variables for which the resulting reduced OBDD $B_f$ has a minimal number of edges (ie, there is no ordering for which the corresponding $B_f$ has fewer edges).

2. This problem will be a warm-up for problem 3 to get you comfortable with SMV. First, download SMV from the course website and read the SMV paper. Feel free to play around with SMV in order to get familiar with it. The SMV paper describes a mutual exclusion example using a semaphore on page 6. Code up this example in the SMV modeling language (just like in the paper). Now, we wish to specify that this system satisfies the property: "if proc1 wants to enter its critical region, it eventually does." Add this specification in CTL to your SMV model and run SMV on your model. Turn in a copy of the SMV output along with your model's code. Describe the behavior of the system that caused SMV to return a counter-example to your specification.

2. The following problem will have you model a simplified elevator model in SMV and check some properties of the model.

The simplified elevator that we will model works as follows. The elevator has 2 state variables: *cabin* that records the position of the elevator cabin, and *dir* that indicates the cabin's next direction of movement from its current state. The position is a floor number between 0 and 3. The global state of the system also indicates, for each $0 \leq i \leq 3$, whether there is a pending request for the elevator at floor i. In this simplified model, the cabin goes up and then down and then up etc. according to the infinite sequence 0,1,2,3,2,1,0,1,2,3,2,…. Our simplified elevator model allows a request for some floor to appear at any time *except if the cabin is actually on this floor.* Then a request cannot disappear until the cabin does reach that floor, in which case the request immediately disappears (that is, the request becomes satisfied). You will of course have to model the evolution of *cabin*, *dir* and the requests according to this behavior. Note that this elevator system is non-deterministic in the sense that requests for floors MUST occur non-deterministically. This allows us to model a freely evolving environment. Initially, the elevator is on floor 0 and there are no pending requests.

a. Model this simplified elevator in SMV. Be sure to comment your code so we understand what you did. When you turn in your code, also provide a brief outline describing your model and the choices you made.
b. Now that you have your SMV model, we'll check a few properties of the elevator system. For each of the following properties, translate the property into CTL (be sure to include the CTL formula in your solution so we know what you did), add it as a specification to your SMV model, run SMV, and turn in the output from SMV. Does your model satisfy the specification? If so, why? If not, what counter-example did SMV return and by analyzing the counter-example, what can you conclude about the behavior that caused your model to fail the specification?
    1) "The elevator has no deadlock."
    2) "All floor requests are eventually satisfied."
    3) "Floor requests are eventually all satisfied simultaneously (that is, eventually there is some moment when there are no pending floor requests)."