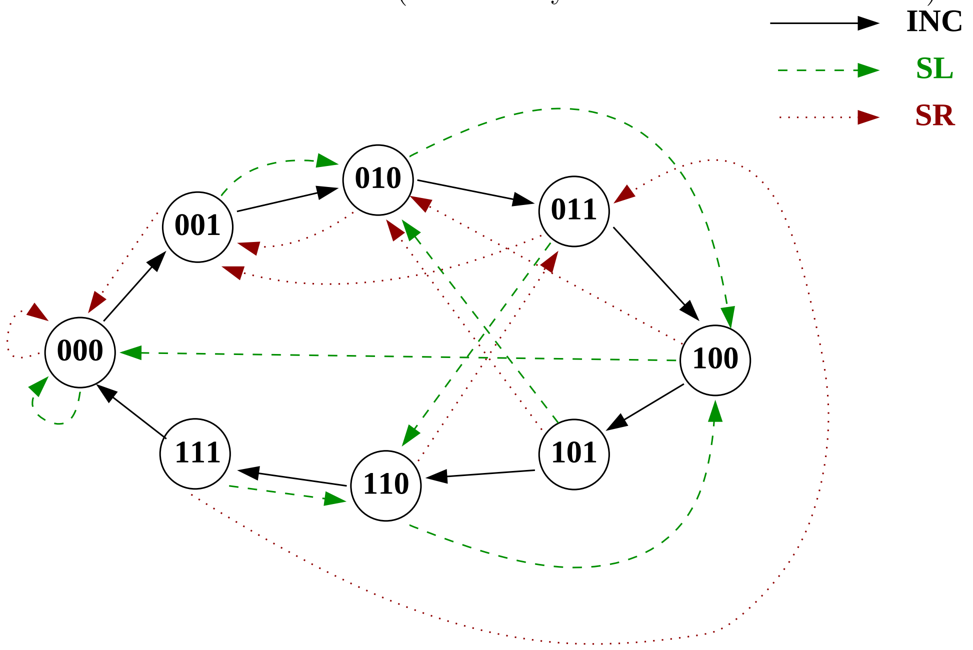
Embedded Real-Time Systems (AME 3623)
Homework 4 Solutions

May 6, 2010

Question 1

Consider a FSM whose state is described by a 3-bit binary number (i.e., there are 2^3 states). This finite state machine has three possible events: shift left (SL), shift right (SR) and increment (INC).

(20 pts) Draw the FSM diagram that describes the behavior of this device when the different events occur (don't worry about actions in this case)



Question 2

We are designing a control subsystem for a Botball-related task. In this scenario, the robot has a set of blue and green balls scattered around it. Using its camera, the robot can identify whether either type of ball is in front of it or if there are no balls.

- In the case of no balls (NB) or a green ball (GB), the robot should continue to search to its right by first making a small right turn (RT)
- If the ball is blue (BB), then the robot should approach it and attempt a grasp (both of these steps are handled by a GRASP action).
- Should the grasp be unsuccessful (NoSUCCESS), then the robot should move backwards for a short distance (BACKUP) and search again
- Should the grasp be successful (SUCCESS), then the robot should place the ball into the basket on its back (PLACE)
- Once the place is complete (COMPLETE), then the robot should turn right (RT) and continue its search
- If the robot's basket is full (FULL), then robot should stop moving

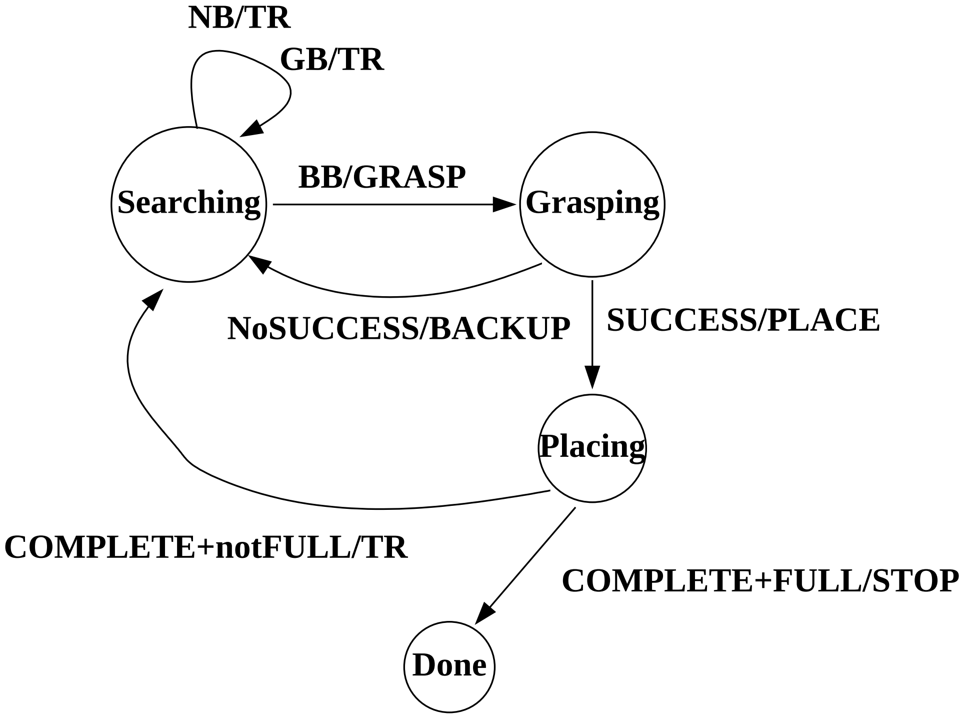
1. (5 pts) What are the events?

   NB, BB, GB, SUCCESS, NoSUCCESS, COMPLETE, FULL, Not-FULL (needed to add this last one)

2. (5 pts) What are the actions?

   RT, GRASP, BACKUP, PLACE, STOP (needed to add this last one)

3. (20 pts) Show the states and transitions. Label the transitions with events and actions.



Note: there are multiple solutions (in particular, having to do with how you handle the question of "full").

Question 3

Consider a keypad and two locked doors with the following properties:

1. The keypad has 4 keys, labeled: L, 1, 2 and 3
2. Entering the sequence: 3, 1, 2 results in unlocking the left door
3. Entering the sequence: 3, 3, 2 results in unlocking the right door
4. Entering *L* will cause both doors to be locked and the sequence to be reset
5. Only one door can be unlocked at a time (entering a code for a door when the other is unlocked will be ignored)
6. Notes: 1) extraneous button presses result in a resetting of the sequence, and 2) the first occurrence of the above sequences should result in the unlocking of the respective door (i.e., the sequence 3, 3, 3, 1, 2 will unlock the left door as soon as the 2 is pressed)

We will design a finite state machine that performs this task. Remember that every state must respond to each event in exactly one way.

1. (10 pts) What are the events?

   Pressing L, 1, 2 and 3

2. (10 pts) What are the actions?

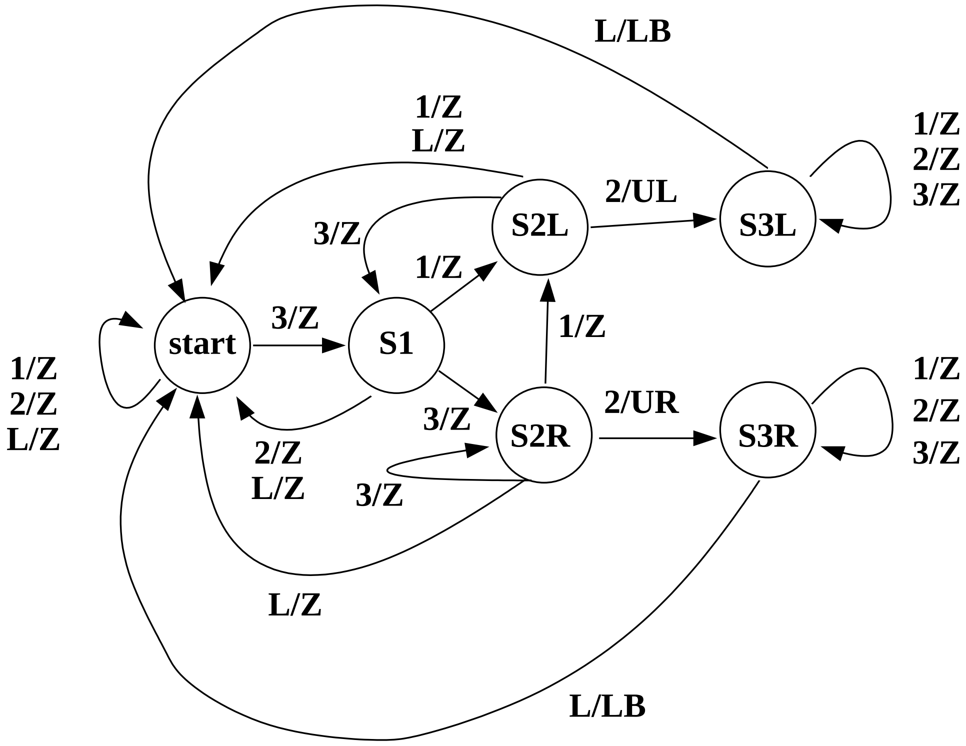
   Unlock left (UL)

   Unlock right (UR)

   Lock both (LB)

   Nothing (Z)

3. (20 pts) Show the states and transitions. Label the transitions with events and actions.



Note: several solutions are possible. In particular, how button presses are handled once a door is unlocked could vary dramatically.