CSP 740, CSE, IIT Jammu

Software Engineering at BTech III (6^{th} semester)
Lab Assignment No 5 - SPIN PROMELA #1
SPRING Semester 2018-19

Instructions:

1. *The date of submission:* ***will be specified two days before****. However, for every delayed submission after the deadline, 10 marks per day will be deducted from the maximum marks of the assignment, without any exception, whatsoever may be the scapegoat.*
2. *The viva for this assignment will be taken on a future date as specified.*
3. ***Maximum Points 200.***

1. Get familiarity with the PROMELA/SPIN environment by executing the following PROMELA programs in SPIN illustrated in class and observing the MSC, Automaton, and the SPIN Output. In all the programs be sure that you run the syntax checker before you attempt to invoke the simulator.:

   (a) What are the various windows displayed for in a SPIN run ? Find the purpose of each. Consider the following Promela program:

   ```
   byte x = 2, y = 3;
   proctype A(){x = x + 1}
   proctype B(){x = x - 1; y = y + x}
   init { atomic{run A(); run B()} }
   ```

   Run this program using different seed values. What effect do the different seed values have on the output displayed within the Data Values window?

   (b) Consider the following Promela programas discussed in the class:

   ```
   byte state = 1;
   proctype A() { (state == 1) -> state = state + 1 }
   proctype B() { (state == 1) -> state = state - 1 }
   init { run A(); run B() }
   ```

   Run this program using different seed values. What effect do the different seed values have on the output displayed within the Data Values window?

   (c) Now consider the following modified program :

   ```
   byte state = 1;
   proctype A() { (state == 1) -> state = state + 1 }
   proctype B() { (state == 1) -> state = state - 1 }
   init { atomic{run A(); run B()} }
   ```

   Run this program using different seed values. What effect do the different seed values have on the output displayed within the Data Values window?

   (d) Repeat the same experiment using the following modified Promela program:

   ```
   byte state = 1;
   proctype A()
   { atomic{ (state == 1) -> state = state + 1 } }
   proctype B()
   { atomic{ (state == 1) -> state = state - 1 } }
   init { run A(); run B() }
   ```

   Again what effect do the different seed values have on the output displayed within the Data Values window?

   (e) Implement and run the PROMELA model for a sample HelloWorld program with printing the PIDs of the init process and the user process.

   (f) Extend the Hello World program with a process type called *control*. Define control so that it ensures that the hello process always performs its print statement before the print statement associated with the world process. You should use two channels to achieve the desired behaviour.

(g) Implement and run the PROMELA models for the Mutual Exclusion Algorithms 1, 2 and 3 as well as the Dekkers algorithm.

(h) Implement and run the for finding the quotient and the remainders - i.e. both the versions of the programs with proctype *quo_rem*() to find the quotient and remainder as a result of division as illustrated.

(i) Implement and run the PROMELA models for division as well as factorial programs.

(j) Implement and run the PROMELA model for the Producer Consumer problem with bounded buffer and using the semaphore P and V operation code. Assume the buffer size of 10 elements.

2. Design a PROMELA model for the Reader's Writer's problem with multiple readers and one writer with protection aginst anomolous updates. Your model must ensure the synchornization and the interprocess communication required in this case. Assume the buffer size of 10 elements.

3. Design a PROMELA model for the Classical Dining Philosopher's problem. Assume a fixed 5 number of philosophers for this problem.

4. Model the following system in PROMELA: A simple railway network involving three defective signals and two block sections of track. Model the defective signals as processes and the block sections of track by channels. Model the movement of trains by message passing, where a train is denoted by a 1 (bit). Although trains only travel in one direction, the network is unsafe because the defective signals allow multiple trains to enter the same block section of track at the same time, i.e. within the model, a crash (unsafe state), corresponds to two trains occupying the same block section of track at the same time.

5. Consider a vending machine that contains chocolate bars, both milk and plain. To select a milk chocolate bar one inserts Rs 5 coin while in the case of plain chocolate the cost is Rs 10 coin. Dene a process type called vender that models the vending machine. Moreover, dene a process called customer that models a chocoholic , i.e. someone who continuously consume chocolate bars. Assume that the vender has a limitless supply of chocolate while the customer has a limitless appetite for chocolate, both plain and milk. Again you should use message channels to model the various lines of communication between the customer and vender, i.e. coins and chocolate.

   (a) Next, rene your model of the vending machine by limiting the number of chocolate bars that are initially held, i.e. 10 milk and 5 plain chocolate bars. In addition, include a coin box that models the amount of money held within the vending machine . Assume that the coin box is initially empty and that once all the chocolate bars are sold that no money is accepted.

   (b) Next, add a local assertion to your vending machine model that expresses the fact that there exists an invariant relationship between the amount of money and chocolate it holds. Use the simulator to check your invariant.

   (c) Next, make one nal renement to your vending machine model so that the customer only has Rs 45 to spend. Furthermore dene a system invariant which states that the amount of money in the overall system is always Rs 45. Check your solution using the simulator.

6. Consider a simple water storage system that involves, sensors, a user and inlet and outlet devices. The sensors measure the water level within a storage device. The outlet device provides water for the user. The demand for water by the user is not constant, i.e. at each moment in time the user decides randomly whether or not to request water. Whenever the water level reaches 20 units the sensors close the outlet and open the inlet. This causes the water level to rise . Once the water level reaches 30 units the inlet is closed and the outlet is opened again. Model the water storage system using distinct es to model the sensors, user, inlet and outlet. Include an assertion to ensure that the water level is always within the range 20 to 30 units. Evaluate your model using the simulator.

7. Consider the Traffic signals at intersection of a highway and a farm road. S1 is the signal for highway and S2 for farm road. There is a sensor on the farm road. Initially S1 will be Green and S2 Red. The sensor becomes On when a car passes on the farm road. When the sensor is On, the S1 goes to Orange and then to Red. Simultaneously, S2 goes to Green. S2 stays Green as long as sensor is On, after which it goes to Orange and then to Red. Now S1 becomes Green again.

   Design a PROMELA model to model the above scenario using the skeleton provided below:

```
        #define On 1
        #define Off 0
        mtype = { Red, Orange, Green};
        byte s1_status;
        byte s2_status;
        bit sensor_status;
        chan chan_s1=[0] of {byte};
         /*for msgs from HighwaySignal to FarmRoadSignal */
        chan chan_s2=[0] of {byte};
        /*for msgs from FarmRoadSignal  to HighwaySignal */
        chan chan_sensor=[0] of {bit};

        proctype HighwaySignal(mtype status)    {
                bit sense = Off;
                int count;
                s1_status=status;
                byte temp;
/* to model the case: if  sensor is ON then S1 goes from orange
 to red after some time.  Also send the status of the current signal to
 FarmRoadSignal process. Also, if able to read from chan_s2,
 then go green (why ?) */
        }

        proctype FarmRoadSignal(mtype status)   {
/* to model the case that it has to get the  message from HighwaySignal
and go Green.  If one is able to read from chan_s1  then it implies
(what ??) and so  S2 goes red and sends status to HighwaySignal */

                proctype Sensor()       {
                        do
                        ::          if
                        :: sensor_status = On
                        :: sensor_status = Off
                        fi;
                        chan_sensor!sensor_status
                        od
                }

                init {
                        atomic  {
                                run HighwaySignal(Green);
                                run FarmRoadSignal(Red);
                                run Sensor()
                        }
                }
```

8. Consider a model shown in Figure 1 for alternating bit protocol with no message loss. There are two processes - sender and receiver. Sender sends message 0 and receiver receives it and sends the acknowledgement 0. On receiving ack 0 from receiver sender sends message 1, receiver receives it and sends the acknowledgement 1. This process is repeated infinitely. Design a PROMELA model for the same.

9. Design a PROMELA model to model the fan controller shown in Figure 2.

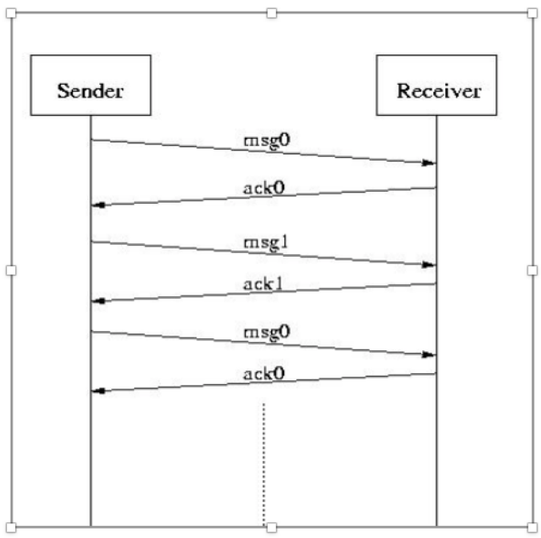

Figure 1: Alternating Bit Protocol)

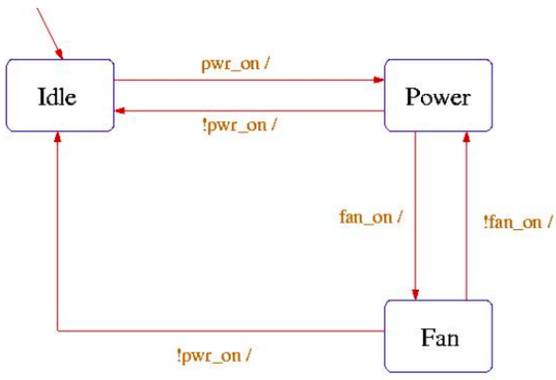

Figure 2: Fan Controller)