CSC034U4E: Software Engineering
IIT Jammu, Autumn Semester, 2022-23
Lab Assignment No 3: Modeling Distributed Applications using PROMELA

<u>Date uploaded/emailed :18$^{\text{th}}$ August 2022</u>

<u>Instructions</u>:

I. *Date of Final submission: Will be specified two days before the due. This would then be a hard deadline.*

II. ***Maximum Points 200.***

III. *Submission Guidelines:*

(a) *You may have to show/share your screen of the program that you are asked to show by the examiner.*

(b) *You have to write a .tex file. Copy the source of each of your programs therein, along with brief analysis/inference of executing that program. The name of the file must containing the full extension used for writing the program file, when executing.*

(c) *You will have to write a brief README.txt file - explaining how to execute the programs.*

(d) *Compile all of the above in a .zip or .tar file.*

(e) *Upload our .tar or .zip file as your submission on the Classroom.*

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 Dekker's 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.

2. (a) Implement and run the PROMELA model for the Producer Consumer problem with bounded buffer and using the Mutual Exclusion Algorithm#1 illustrated in the class. Assume the buffer size of 10 elements. Simulate your code with different seed values and analyze whether the simulation shows the limitation of this algorithm. Write your observations clearly.

   (b) Implement and run the PROMELA model for the Producer Consumer problem with bounded buffer and using the Mutual Exclusion Algorithm#2 illustrated in the class. Assume the buffer size of 10 elements. Simulate your code with different seed values and analyze whether the simulation shows the limitation of this algorithm. Write your observations clearly.

   (c) Implement and run the PROMELA model for the Producer Consumer problem with bounded buffer and using the Mutual Exclusion Algorithm#3 illustrated in the class. Assume the buffer size of 10 elements. Simulate your code with different seed values and analyze whether the simulation shows the limitation of this algorithm. Write your observations clearly.

   (d) Implement and run the PROMELA model for the Producer Consumer problem with bounded buffer and using the Dekker's Algorithm illustrated in the class. Assume the buffer size of 10 elements. Simulate your code with different seed values and analyze whether the simulation shows the limitation of this algorithm. Write your observations clearly.

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

   (f) IS the semaphore P and V operation code illustrated in the class simulating a binary semaphore or a counting semahore. Specify the limitations of a binary semaphore. Now modify the semaphore P and V operation code to simulate a counting semaphore (let's assume counting upto 5). Implement and run the PROMELA model for the Producer Consumer problem with the modified semaphore definition. Assume the buffer size of 10 elements.

3. Design a pertinet for a single reader and a single writer in a typical RW problem. Your problem should support mutual exclusion between the reader and writer.

4. Design a PROMELA model for the Reader's Writer's problem with three readers and one writer with protection aginst anomolous updates. Your model must ensure the synchronization and the interprocess communication required in this case. Assume the buffer size of 10 elements. The corresponding logic is shown using the Petrinet as in Figure .

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

6. 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.

7. Model the following behaviour in PROMELA: A system with two lamps and a button. When the lights are OFF, pushing the button causes the first lamp to go on. Pushing the button again causes the second lamp to go ON and the first to go OFF. Pushing the button yet again causes both the lamps to go ON and pushing it once more causes both to go OFF.
8. Model the following system in PROMELA: Consider a network of sensors could be used in an industrial setting to monitor groundwater contamination. Power consumption is a major issue for sensors, so a monitoring system should be designed to conserve power as much as possible. The system will be initialized upon power up If a normal reading is received, no action will occur. If an abnormal reading is received, the system will send an alert. Additional abnormal readings will not be reported (to conserve power). When a normal reading is received, the system will again send an alert.
9. Design a PROMELA program to model the following behavior: In a Chemical plant, if there is a High-pressure signal OR there is a High-temperature signal then the plant is shut OFF. Here, temperature & pressure have two different signals – one indicating a slight deviation from the acceptable & the other a dangerous deviation from the acceptable value. In the latter case, system must shut off immediately.
10. Modify the chemical plant simulation model above, to cope also with simultaneous signals and design a PROMELA model for the same.