ECE 5790/6790 Design Assignment #2
Principles of Cyber-Physical Systems
Fall 2022
Z. Zhang

Asynchronous Model

The goal of this assignment is to formally model and verify properties for a mutual exclusion protocol using the IVy verification tool.

Consider an arbitrary, but finite, number of identical processes[1], that execute in parallel. Each process consists of a noncritical part and a critical part, usually called the critical section. In this exercise we are concerned with the verification of a mutual exclusion protocol, that is, a protocol that should ensure that at any moment of time at most one process (among the N processes in our configuration) is in its critical section. There are many different mutual exclusion protocols developed in the literature. In this exercise we are concerned with Szymanski's protocol [2]. Assume there are N processes for some fixed $N > 0$. There is a global variable, referred to as flag, which is an array of length N, such that $flag[i]$ is a value between 0 and 4 (for $0 \leq i < N$). The idea is that $flag[i]$ is the status of process i. The protocol executed by process i looks as follows:

```
0:  loop
1:      Noncritical section
2:      flag[i] := 1;
3:      wait until (flag[0] < 3 and flag[1] < 3 and .... and flag[N-1] < 3)
4:      flag[i] := 3;
5:      if (flag[0] = 1 or flag[1] = 1 or ... or flag[N-1] = 1) then
6:          flag[i] := 2;
7:          wait until (flag[0] = 4 or flag[1] = 4 or ... or flag[N-1] = 4);
8:      flag[i] := 4;
9:      wait until (flag[0] < 2 and flag[1] < 2 and ... and flag[i-1] < 2)
10:     Critical section
11:     wait until (flag[i+1] ∈ 0, 1, 4) and ... and (flag[N-1] ∈ 0, 1, 4)
12:     flag[i] := 0;
13: end loop
```

Before doing any of the exercises listed below, try first to informally understand what the protocol is doing and why it could be correct in the sense that mutual exclusion is ensured. If you are convinced of the fact that the correctness of this protocol is not easy to see, then start with the following questions.

1. Model Szymanski's protocol in IVy. Assume that all tests on the global variable *flag* (such as the one on line 3) are atomic. Look carefully at the indices of the variable flag used in the tests. Make the protocol description modular such that the number of processes can be changed easily.

2. Check for several values of N ($N \geq 2$) that the protocol indeed ensures mutual exclusion. Report your results for N equal to 4.

3. The code that a process has to go through before reaching the critical section can be divided into several segments. We refer to statement on line 4 as the *doorway*, to segments on lines 5, 6, and 7 as the *waiting room* and to segments on lines 8 to 12 (which contains the critical section) as the *inner sanctum*. Give for each case the changes to your original IVy

[1] Only the identity of a process is unique.

[2] B.K. Szymanski. A simple solution to Lamports concurrent programming problem with linear wait. In International Conference on Supercomputing Systems, pages 621626, 1988.

specification for Szymanskis protocol and present the verification results. In case of negative results, simulate the counterexample by means of guided simulation.

(a) Whenever some process is in the inner sanctum, the doorway is locked, that is, no process is at location line 4.

(b) If a process i is at line 10, 11 or 12, then it has the least index of all the processes in the waiting room and the inner sanctum.

(c) If some process is at line 12, then all processes in the waiting room and in the inner sanctum must have flag value 4.

Submission Requirements

- Submit your work in a **single zip** file (in the format **designAssign2-{your last name}.zip**) on Canvas. Your submission should include the following:
  - Project files including well-commented IVy model(s) to the stated task above. Comments should be succinct and clear.
  - A textfile containing counterexample(s) generated by IVy for each property violation, if any.
  - A **single PDF file containing your name on the first page**. It should be in the format **designAssign2-{your last name}.pdf** and it should contain answers and/or figures to the above problems. You should give a brief description to each IVy model you created, answer (Pass/Fail) to each property you are asked to verify. For each failing property, describe in words the action sequence of the counterexample generated by IVy.
- All writings and figures must be **clear and readable**. Otherwise, substantial loss of points may be incurred.