CS511 – Endterm – April 27, 2023 – Part II – Topic B

Exercise 1 (*Promela*)

The following "solution" to the MEP problem (for two threads) was published in a journal[1]. It was later shown to be flawed (mutex fails). The Groovy code is presented below:

```
turn=0
flag = [false, false]

2.times {
    int me=it
    Thread.start{
      while (true) {
        flag[me] = true
        while (turn!=me) {
               while (flag[1-me]) {}
               turn = me
        }
        println (me+" went in\n");
        flag[me] = false
        println (me+" went out\n");
    }
    }
}
```

Write it in Promela.

Deliverable: `fmep.pml`. Your code must compile without errors (type `spin fmep.pml` and make sure the simulation runs without problems).

[1] Harris Hyman, Comments on a problem in concurrent programming control, Communications of the ACM, v.9 n.1, p.45, Jan. 1966

Exercise 2 (*Model Checking with Assertions*)

Address the following properties of the solution to the MEP problem from the previous exercise:

1. It enjoys absence of livelock. Introduce appropriate progress labels.
2. It does not enjoy mutex. Introduce the appropriate assertions and any necessary variables.

1. Deliverable 1: `fmep_mc.pml`. Code with progress labels and assertions.
2. Deliverable 2: `trail.txt`. This is the output from `./pan -r` for failure of mutex.