CS4710: Model-Driven Software Development, Spring 2017
Computer Science Department
Michigan Technological University

SPIN Group Assignment #1: Parallel Swap

Deadline: April 14, 2017 at mid-night

Problem description: Consider a shared memory program with *N* processes that have access to an array A[] of integers of size *N*. The array A[] is initialized with *distinct* positive integer values. Each process can read and write to any array cell.

Specifications: Each process P_i should randomly pick a value *j* modulo *N* and then swap A[i] with A[j]. After the swap, P_i terminates. The safety and liveness requirements of the program are as follows:

- *Liveness*: All processes must eventually terminate.
- *Safety*: After the termination of all processes, the array A[] must contain a permutation of its initial values. Notice that, A[] must not contain duplicate, nor should there be a missing value in A[] after termination. In order to guarantee that, you should ensure data race-freedom. That is, no two processes should simultaneously access the same array cell where one of them performs a write operation. In this problem, data race-freedom would imply no duplicate/missing values.

Note 1: The part of the code that performs the swap operation must NOT be in an atomic block. You are allowed to introduce new variables if needed.

Note 2: You should NOT *serialize* the processes. That is, two pairs of processes that have no data dependencies (in terms of the cells of A[]) must be able to concurrently perform the swapping. In other words, a good design is the one that maximizes the degree of parallelism and does not unnecessarily impose a sequential order for ensuring data race freedom.

Deliverables: You must turn in the following material:

1. **Promela model**: You should design a model of the Parallel Swap in Promela. Your model should be parameterized in terms of the process ID (_pid) variable. That is, your solution must be in the SIMD (Single Instruction Multiple Data) model of parallelism. (40 points)
2. **Property specification in LTL and their verification**: You should specify the safety and liveness properties as LTL expressions, and verify them using SPIN. (30 points)
3. **Written report**: Your team must turn in a written report that explains your Promela model, your LTL expressions and your verification activities. (30 points)

Extra Credit: After you verify the safety and liveness properties, revise your Promela code such that each proctype performs the swapping several times. Under this scenario, you must ensure that each process performs the swapping infinitely often and does not get deprived from doing the swapping forever. (50 points)

- Are the safety and liveness properties still satisfied? Explain what you experience.
- Should the proctypes be synchronized after each round of swapping?