**Model-checking and Debugging Readers-Writer Lock Algorithm (due May 19th 11:59 PM)**

**Overview**

A readers-writer lock is an extension of binary lock (i.e., mutex) that allows multiple reading threads (calling it *readers*) to execute the critical section concurrently while guaranteeing mutually exclusive execution of a writing thread (calling it *writer*).

Your task is to debug the readers-writer algorithm in Slides 29—32 at Lecture note “CS492B Analysis of Concurrent Programs: Pthread and OpenMP” using the SPIN model checker. You have to find a bug in the buggy readers-writer lock algorithm, fix the algorithm, and confirm the correctness of your fix using the model checker SPIN. For the purpose, you have to

1. Write a Promela model corresponding to the algorithm and related posix pthread library functions. Your Promela model should contain 2 readers and 2 writers, which is a minimal model to check the requirements.
   1. For making a Promel model simple, assume that
      1. Mutex type is PTHREAD\_MUTEX\_DEFAULT (see the man page of pthread library)
      2. pthread\_mutex\_lock() is not invoked recursively.
      3. pthread\_mutex\_unlock() is invoked only when it was locked by the calling thread.
2. Specify the correctness requirements in assertion and LTL, and
3. Interpret model checking results and fix the bug using the results, and
4. Confirm that your fix is correct by checking if the fixed model satisfies all of the requirements

**Requirements for the Readers-writer lock algorithm**

The readers-writer lock algorithm should satisfy the following 4 requirements:

(1) Exclusive writing

* A reader cannot enter the critical section while a writer is in the critical section, and
* At most one writer can be in the critical section.

(2) Read concurrency

* It should be possible that multiple readers can enter critical section concurrently/at the same time

(3) High priority of writer

* A waiting writer should enter the critical section earlier than all waiting readers.

(4) Deadlock freedom

* The algorithm should not cause a deadlock

**Assignment**

You have to submit a report including the following items (note that your explanation will take a large portion of the HW grade):

* Your Promela model based on the provided rwlock\_template.pml corresponding to the original readers-writer lock algorithm and your explanation of how you have built your model from the algorithm systematically.
* Requirements specified as assertions or LTL properties
* Script to run the Spin model-checker for the Promela models (i.e., spin parameters) w.r.t. the requirement (TA should be able to replay your verification using the script). Don’t forget to enforce weak fairness constraint in a necessary case to get realistic verification result.
* Verification results (i.e., violation of the requirements) on the buggy algorithm model (i.e., spin output) and your own explanation for the bugs (i.e., interpretation of the violation and error trails)
* Fixed promela model and your explanation for the fix
* Verification results (i.e., no violation) on the fixed algorithm model (i.e., spin output)

**Submission**

Submit your report and result via TA’s e-mail [hongshin@gmail.com](mailto:hongshin@gmail.com) by the due date. Also, submit the hardcopy of your report to the HW box. TA will acknowledge your homework in 12 hours.