

CS 6110 – Formal Methods – Spring 2011

March 1, 2011

Assignment 5, Handed out: Feb 24rd, Due Mar 4rd Fri midnight

1. (10%) Describe the liveness violation (“non-progress cycle”) in the Promela model of distributed locking called `distributed_locking.prm` kept online. Walk through the Promela code path described by this violation, describing the Promela statements encountered and what they do (about a page of bulleted crisp and short explanations).
2. (5%) Describe the state initialization method in the model `distributed_locking.m` given to you in detail. Why is this method symmetry-preserving? (About half a page.)
3. (5%) Run the Murphi model with/without symmetry, and make a brief note about the state savings due to symmetry.
4. (16% per property) Express five interesting LTL properties about this Promela model. Describe each property in English, asserting why they are interesting for this model. Evaluate these LTL properties using SPIN and report the results.

Your properties must be selected as follows:

- Two properties that succeed
- Two that fail
- Three must be liveness

For example:

- two liveness properties that fail
- one liveness property that passes
- one safety property that passes
- one safety property that fails

NOTES: In the model you are given for the present assignment, i.e.

http://www.eng.utah.edu/~cs6110/week7/distributed_locking.prm there is a 'progress' label in the model. There is also a never automaton given under an `ifdef NEVER`. Now I'm giving you some options on how to do this assignment:

1. Well, since I gave you one safety property thru the above `NEVER`, you are allowed to write the contents of this never automaton as one of your LTL properties... of course turned into a proper LTL formula.

You are also allowed to completely remove (or not define the `NEVER` flag) thru the `-D` flag. See the “Extra Compile-Time Directives” in the `Advanced Verification Options` on how to turn on this `ifdef`.

2. Also since the the **progress** label may throw violations that you don't want to see, you can remove this **progress** label if you want. Then you can focus purely on the LTL formulae (and their violations) that you write. This way, you won't be "distracted" by the progress label caused violations.