CSCE 747 - Assignment 4:
Symbolic Execution and Finite State Verification

Due Date: Tuesday, April 11, 11:59 PM

There are 3 questions, worth a total of 100 points. You may discuss these problems in your teams and turn in a single submission for the team, in zip format, on Moodle. Answers must be original and not copied from online sources.

Problem 1 (7 Points)

Exercise 7.1 from the textbook.

Problem 2 (8 Points)

Exercise 7.5 from the textbook.

Problem 3 (60 Points)

For this exercise, you are required to create a finite-state model of a traffic-light controller and verify its properties using the NuSMV symbolic model-checker (download from http://nusmv.fbk.eu/ - **we will not provide technical support for this tool**)

- Assume that the controller manages traffic and pedestrian lights at the intersection of two roads, both with two-way traffic.
- Pedestrians can request access to cross the road by pressing a “walk button”.
- Assume that the system has traffic sensors for each direction to detect if vehicles are present and waiting to pass through, which allows the system to manage traffic flow efficiently by varying the amount of time the lights are green for each road/direction based on demand. Your model should capture and represent this notion of varying time in some manner (i.e., do not abstract away time).
- There are emergency vehicle sensors for each direction which lets the system provide priority access for emergency vehicles by switching lights appropriately.

You may state and make any other reasonable simplifying assumptions that you need.

In your submission, you must address the following:

1. Define the scope and the requirements for the system that you intend to model – a brief description of what you have modeled, any assumptions that you have made and the key requirements you expect the system to satisfy. (10 Points)

2. Build a finite state model of the system in the NuSMV language. Be sure to write sufficient comments. (Though not required, you may find drawing state diagrams helpful). (20 Points)
3. Write at least three safety properties in temporal logic (CTL or LTL) that must be satisfied by the system. Explain your properties informally and state which system requirements those properties are derived from. (10 Points)
4. Write at least three liveness properties in temporal logic (CTL or LTL) that must be satisfied by the system. Explain your properties informally and state which system requirements those properties are derived from. (10 Points)
5. Verify your properties on your system using the NuSMV symbolic model checker and provide a transcript of your NuSMV session. (5 Points)
6. Write at least one "anti-property" – i.e., a temporal property that does not hold for the system - and show the counter-example generated by NuSMV for the anti-property. Briefly explain both your anti-property and the counter-example informally. (10 Points)

Problem 4 (25 Points)

In Problem 3, you used a model-checker to verify properties of a finite-state model. When a property is not true for the model, the model-checker produces a counterexample that shows an execution trace leading to a violation. One may leverage this feature of the model checker for test generation.

Suppose our goal is to exercise a certain state (or transition) in a finite-state model. If that goal is expressed as some propositional (or temporal) Boolean formula, one can then formulate a temporal property that states "the goal cannot be reached for the model". Such a property is called a trap property – attempting to verify that property using the model-checker will result in a counterexample (if the goal is reachable). This counterexample can be treated as test case for an implementation whose behavior is expected to conform to the FSM.

Now, apply this approach for the traffic light FSM that you developed in Problem 3:

1. Formulate at least 5 different test goals (e.g., reaching a particular state, exercising a specific transition, reaching the same state twice, etc.). (10 Points)
2. Show the trap properties that you derived for each of the test goals. (10 Points)
3. Check those properties using NuSMV on your traffic lights FSM and show the test cases you obtained from the counter-examples produced by the tool. (5 Points)

You must submit your trap properties and the NuSMV output file which contains your test cases. You only need to redirect NuSMV output to a file and submit it, do not edit the output file.