CPS3233 – Verification Techniques Assessment

Assignment instructions (read carefully and thoroughly):

- This is an individual assignment and carries 100% of the final CPS3233 grade.

- This assignment consists of multiple tasks. While it is strongly recommended that you start working on the tasks as soon as the related material is covered in class, the firm submission deadline for all tasks is **Monday 3rd May 2021**.

- The submission of the full assignment, including a soft-copy of the report and all relevant files, should be uploaded to the VLE by the deadline. All files must be archived into a single .zip file.

- You are to allocate 100 hours to complete this assignment, including 70 hours for independent study.

- The first page in your assignment report must be the title of your assignment clearly showing your name, surname and study unit code.

- Reports that are difficult to follow due to low quality in the writing-style/organization/ presentation will be penalized.

Specify the elevator system (25%)

The first part of the assignment consists of specifying the elevator system (see the attached document providing more information) using different formal notations which are covered in class:

1. Finite state automata
2. Regular expressions
3. Duration calculus
4. Timed automata

Deliverable: In each case, you are expected to give informal explanations of how the formal notation corresponds to the intuitive meaning.

Runtime verification (25%)

Once we have specified the elevator system, we can attempt to monitor the elevator behaviour according to the formal properties. In this section, you are expected to use a runtime verification tool (Larva will be covered in class) to show how the elevator system can be monitored.

Deliverable: A report explaining:

1. The script used for monitoring (using Larva script, based on timed automata).
2. How you ensured that the monitor is working correctly (try to insert bugs in the specification and/or the system).
3. What could have been done better.

Model-based testing (25%)

Following runtime verification, we turn our attention to test the formal properties by generating inputs to simulate the elevator behaviour (using structures inspired from finite state automata). In this section, you are expected to use a model-based testing tool (ModelJUnit will be covered in class) to test the properties, generating test sequences intelligently.

Deliverable: A report explaining:

1. What has been tested and how (explaining the model used).
2. Why you think the testing carried out is adequate (which test coverage measures did you consider, why? What do the results obtained mean? etc).
3. How it was ensured that testing worked correctly (did your approach detect bugs?).

Runtime verification and testing (25%)

As a last step, you are expected to show how runtime verification and testing can be combined. Use a runtime verification tool to detect violations during an execution of model-based testing.

Deliverable: A report explaining:

1. The architecture you have set up for this experiment.
2. The violations you were able to detect (as above, you can artificially insert bugs to check detection)
3. The advantages/disadvantages of such a set up.