


ECE 322 Lab Report #6

 Benjamin Kong
1573684

December 8, 2021

Part 1

Q1.1

Mutation testing is a form of white box testing that allows testers to check the effectiveness of unit tests. Mutation testing changes specific components—for example, equality operators and mathematical operators—of the code being tested in order to check if unit tests are able to detect the changes. Changes from mutation testing are meant to cause errors in the program. If a test that typically passes fails as a result of the code being mutated, then we say the mutant was killed.

Q1.2

Mutation testing tests how well unit tests are able to detect errors in the program being tested.

Q1.3

There are three types of mutations:

- Value mutations (boolean values can be swapped, variables can be altered),
- Decision mutations (swapping equality operators, swapping mathematical operators), and
- Statement mutations (deleting a statement, duplicating a statement).

Q1.4

In my opinion, mutation testing is effective. 100% code coverage means that all code statements have been run through but doesn't mean the code has been tested effectively. Mutation

testing allows testers to gauge how effective their unit tests are. Furthermore, mutation testing is typically not too difficult to set up. As a result, I think it is an effective tool for helping test a production type of system.

Q1.5

Writing more robust test cases that are able to catch small changes to the code can help increase the chance of killing mutants. For example, writing tests that can detect if the correct equality sign is being used (for example, “less than” versus “less than or equal to”) is a step in the right direction.

Part 2

Q1.1

Regression testing is a type of testing that helps verify that a code change does not impact the existing functionality of the system. In its simplest form, regression testing involves re-running all test cases to ensure a change has not broken any test cases. For large systems, however, it may be infeasible to re-run all tests due to the high cost of running tests. In these cases, only test cases affected by changed code are re-run.

Q1.2

Regression testing is used to make sure errors do not appear in any parts of the system when a change is made. If a change is made to a system that unknowingly causes an error, regression testing should catch it.

Q1.3

Regression testing is important because it makes sure bugs and errors are not introduced into a system. It is also beneficial for saving resources as only tests related to the changed

code are re-run. It helps ensure software quality stays high throughout development.

Q1.4

Regression testing is very effective because it creates a low-cost way of ensuring bugs are not introduced into a system. It verifies that a small change doesn't break or change the functionality of the system in an unintended way.