ECE 322 Lab Assignment 6

Regression and Mutation Testing

Overview

The objective of this laboratory assignment is to become familiar with regression and mutation testing techniques. This lab makes use of a mutation testing tool called **mutmut**.

Introduction

The following section provides a brief overview of mutation testing and regression testing, serving as an introduction to these topics for the purposes of this lab assignment.

Mutation testing:
Mutation testing uses the concept of automatic code mutation to determine how well unit tests are able to detect errors in the application. This type of testing changes the existing source code by changing equality operators, mathematical operators, etc. The existing unit tests are then run against the mutated source to see how many mutations are caught. If a previously passing test fails as the result of source mutation we say that mutant was killed. For this lab session we will be using the **mutmut** mutation tool which is available at https://mutmut.readthedocs.io/en/latest/. For additional information on the use of mutmut refer to the resource document available on the course website

Regression Testing:
Regression testing is concerned with re-testing a system after updates (such as bug fixes, or new features) have been added to the application. The intention of regression testing is to show that the changes to the code have not introduced any new bugs in the system or impacted the system negatively. The simplest form of regression testing involves simply running all existing test cases after the modifications have been introduced. In some cases, however, this can be excessive and take a long time when the system is very large. An alternative approach is to selectively re-run test cases based on where changes to the system were made. A final essential aspect of regression testing is to update and create test cases for new functionality in the system, and to change existing test cases where the requirements of the application have changed.

Lab Experiment (Tutorial, watch the online tutorial - No code is required for this section)

Tutorial Task 1

In this part of the lab tutorial, we will be applying the concept of mutation testing.

The lab tutorial for this lab is in two parts:

First:

- Create test cases fulfilling the requirements of statement coverage
- Create test cases fulfilling the requirements of branch coverage

- As with all unit testing the test cases must make meaningful assertions regarding the functionality of the array library
- Ensure that the test cases cover (and test!) the basic functionality of the library

Second:

- Create mutations using the mutmut library and run the tests against them (via mutmut interface).
- Record the results of the testing (number of mutants killed and total mutants etc.). If there are surviving mutants, improve the test cases so that all, or almost all mutants are killed.
- Save the mutation testing report

Tutorial Task 2:

In this part of the lab tutorial, we will be applying the concept of regression testing to a simple maths library which you are in charge of testing before the release of a new version. This package contains a number of simple mathematical functions which you will need to test: the tutorial for this task is divided in two parts:

First:

- We Prepare and implement test cases for the existing functions in the math package and run the tests against the code. As always these must be meaningful, comprehensive unit tests.
- Report any errors found in the source code with the test cases (We Do not fix them at this point).
- Record the first set of next cases in a test case table indicating failed test cases. Make note of what caused these failures.

Next:

- Replace the math library class file in your project with the contents of the file labelled commit which can also be found in the code. This simulates the next addition to the library from the development team in which they claim to have fixed the errors you found previously, and they have also added some additional functionality. 3
- Some functions have been added to the library. Adjust your test suite to cover the added functionality following the same coverage requirements as part 1.
- Test the new version of the software
- Resolve and errors found in the new version

Lab 6 Exercise (optional):

The submission of the Exercise is optional. Your submission will help improve your overall grade for this lab.