Name: Hengrui Guo

Student ID: 932504737

Date: April 19, 2016

Project Proposal

1. Background

Random test is a kind of test technique where the program is tested by the random and independent input. This test technique is popular in software testing. However, most random testing studies consider only the rate of failure-causing inputs is used to evaluate the measurement of effectiveness. The study by Chan et al. [1], figures out that the performance of a partition testing strategy depends not only on the failure rate, but also on the geometric pattern of the failure-causing inputs. As a new random testing technique is introduced, called adaptive random testing, in the paper “Adaptive Random Testing”. In this paper, the authors introduce an algorithm, which focus more on evenly distributing test cases within the input space. In this new method, the evenly distributed test cases are more likely to find out the failure using fewer test cases than ordinary random testing. The key point of this algorithm is to distribute the test cases evenly. The authors divide the test cases into two parts: the candidates set and the executed set. The executed set stores the test cases executed without revealing any failure. The candidate set is a set of test cases that are randomly selected without replacement. The test case are picked from the candidate set to test is the farthest away from all executed test cases. The author defined this farthest distance as the Euclidean distance. In other words, the distance from the selected test case in candidate set to every test cases in executed set is larger than distance from any other test cases in candidate set to every test cases in executed set. In this algorithm, the adaptive technique can reveal the failure quickly than the ordinary random test method.

2. Plan to Implement

In this term, I am going to implement an adaptive random test algorithm using TSTL. First of all, I need to read and study carefully the paper, “Adaptive Random Test”, especially its pseudo code to implement the algorithm. Secondly, I should study the some functions in sut.py considering how to use this function in TSTL. Meanwhile, I will carefully check the source code of the TSTL. Though it’s a lot I should find the source code related the random tester. And then I will try to figure out how to use my algorithm in the random tester in TSTL to improve the effective of the TSTL random tester. What is more, I will find some instance for the adaptive random tester, such as some data structure like AVL tree, Heap or some algorithm with bug to check the effective of the adaptive random tester comparing to the ordinary random tester. Finally, I will write a graphic report about comparing the effective between the adaptive random tester and ordinary random tester in detecting different kinds of program or algorithm.

3. References

[1] Chan, F.T., Chen, T.Y., Mak, I.K., Yu, Y.T.: Proportional sampling strategy: guidelines for software testing practitioners. Information and Software Technology 38 (1996) 775–782
[2] T.Y. Chen, H. Leung, and I.K Mak. Adaptive Random Testing, 2004