CS569 – Project Proposal

Xu Zheng

Onid: zhengxu

Student ID: 932-509-227

April 19, 2016

1. Introduction

As we know, Random Testing is a Black-box testing method. Instead of analyzing the source code, generating random inputs to test the software is the purpose of random testing. The disadvantage of random testing is that there will be lots of useless test cases generated. So that it is very important to have efficient algorithms to generate its test cases. To this end, there are many researchers developed enhanced random testing such as "Feedback-Directed Random Test Generation" [1], "Feedback-Controlled Random Test Generation" [2] and "Adaptive Random Testing" [3]. There common idea, which is also my idea for this project, is selecting the test cases to improve the effectiveness of testing.

According to the previous study, "In random testing, test cases may be randomly chosen based on an uniform distribution or according to the operational profile." The authors of adaptive random testing [3] points out the performance of a partition testing strategy depend not only on the failure rate, but also on the geometric pattern of the failure-causing inputs. The basic algorithm of adaptive random testing is making use of two sets of test cases, namely the executed set and the candidate set which are disjoint. The executed set is the set of distinct test cases that have been executed without causing any failure. And the candidate set is a set of test cases that are randomly selected. At the beginning, the executed set is empty and the first test case is randomly chosen from the input domain. The executed set will keep updating with the selected element from the candidate set until a failure is caused. From the candidate set, an element that is farthest away from all executed test cases, is selected as the next test case.

Feedback-directed random test generation is a technique to automatically generate software tests. According the authors [1], "the technique can generate random method sequences using public methods from the classes of a system-under-test (SUT)". And the main difference between feedback-directed random testing is that it makes the use of the feedback from the previously generated inputs to guide the generation. It is convincing me that feedback-directed random testing is effectiveness with the respect to the fact it has been used widely in industries filed.

Basing on the feedback-directed random testing, some researchers [2] proposed an enhanced version called "Feedback-controlled random testing". The basic idea is that to use multiple pools concurrently to generate the inputs, instead of using a single pool. To be more specific, each pool has different contents, so that different pools will guide the generation toward different direction. By using different pools concurrently, the algorithm can dynamic determine in which direction to continue generate inputs by analyzing the feedback information. According to their experiments result, feedback-controlled random test generation is highly effective.

2. Project Plan

As I mentioned above, my main idea for this project is selecting the test cases generated to improve the effectiveness of testing. And I want to combine the algorithms from Feedback-directed random testing and Adaptive random testing. In order to implement this idea in TSTL, I will continue reading the relevant papers. After I implement the first version of tester, I will use it to test buggy avltree to check if my implement is in the right track. Afterward I will improve my tester with the feedback of the testing results.

3. Checkpoints

	Competitive Milestone 1	Competitive Milestone 2	Final Submission
May 3	■		
May 18		■	
June 6			■

References

[1] C. Pacheco, S. K. Lahiri, M. D. Ernst, and T. Ball. Feedback-directed random test generation. In Processing of the 29th international conference on Software Engineering, ICSE'07, pages 75-87, 2007

[2] K. Yatoh, K. Sakamoto, F. Ishikawa, and S. Honiden. Feedback-controlled random test generation. In Processing of the 2015 international symposium on Software Testing and Analysis, ISSTA'2015, pages 316-326, 2015

[3] T. Y. Chen, F. C. Kuo, R. G. Merkel, and T. H. Tse. Adaptive random testing. In Processing of the 9th Asian Computing Science Conference, pages 320-329, 2004