

Proposal of Final Project

CS569

Wei Liu

ID: 932305582

April 19 2016

Section 1: Introduction

Based on many research papers and reports about different test generation methods, automatic test generation is discussed in these recent papers. In my opinion, to implement the principle of test generation complex algorithm is the best way. So, I prefer to use static test and dynamic test driven generation code automatic test generation. In this project, I will use it to realize the algorithm.

Random tester has been proved to be a simple and effective way to perform software testing tasks. However, the performance of random tester is really bad based on repeat and useless test cases. To compare with it, SUT has its better performance to reduce overhead, save time and so on. Based on these information, I prefer to find a way or implement a test harness on random tester. Specifically, Adaptive Random Testing is a good way to implement adaptive test. For more information, that is to say, the adaptive random testing is an intelligent algorithm, can choose a relatively good test case, in order to reduce unnecessary testing input. Commonly used random testing study, the fault caused by input rate used in the determination of the validity. This paper also shows that ART provide higher efficiency and more confidence in the reliability of the SUT, even without a failure is detected. As a large number of flexible API, TSTL is absolutely feasible to implement ART algorithm. There are many different standard to describe the uniform. In the article, the author mainly consider using in the execution group and the Euclidean distance between the elements in the candidate group. Generally, the more evenly distributed with less test cases have a better chance of test cases to a failure mode. Also, ART, which is considered the distribution of the input space of test cases, and attempts to generate test cases more evenly. In addition, based on article Code Coverage of Adaptive Random Testing, ART can have the same number of test cases,

and this is an interesting fact random testing to a higher code coverage. We picked up the next test case, making the minimum distance should be the largest minimum distance. According to this standard, we actually choose more appropriate test cases the next iteration. After taking the information of test cases into consideration, the tester will be more reasonable and experiment significantly better.

Section 2: PLAN

Before my project begins, I prefer to read the article carefully and try to search more information on about it. The algorithm is a little hard for me to understand, but I will discuss it with other student. More than this, this is my first time meet with random tester and I will search for more codes and sources and implement them. Before this term, I was joined CS562 and learned TSTL in that term. I think TSTL is a really good tool to test codes and search for bugs.

Section 3: References

[1] Chen, T. Y., Kuo, F., Liu, H., & Wong, W. E. (2013). Code Coverage of Adaptive Random Testing. IEEE Transactions on Reliability IEEE Trans. Rel., 62(1), 226-237. doi:10.1109/tr.2013.2240898

[2] I. M. T.Y. Chen, H. Leung. Adaptive random testing, 2004.