Part 1: Proposal

Xinran Peng

CS569s16

Prof Groce Alex

2016/4/19

Introduction

Random testing is a black-box software testing technique where programs are tested by generating random, independent inputs. Results of the output are compared against software specifications to verify that the test output is pass or fail. In case of absence of specifications the exceptions of the language are used which means if an exception arises during test execution then it means there is a fault in the program. A recent study has shown that failure patterns may be classified as point, strip or block failure patterns. Intuitively speaking, when the failure pattern is not a point pattern, more evenly spread test cases have a better chance of hitting the failure patterns. Based on this intuition, we propose a modified version of random testing called adaptive random testing. This method seeks to distribute test cases more evenly within the input space. It is based on the intuition that for non-point types of failure patterns, an even spread of test cases is more likely to detect failures using fewer test cases than ordinary random testing. An empirical analysis of 12 published programs has shown that adaptive random testing outperforms random testing significantly for most of the cases. The experimental results provided by the paper 'Adaptive random testing' have been very encouraging, providing evidences that our intuition of spreading test cases more evenly within the input space is potentially very useful. Nevertheless, there are a number of issues of adaptive random testing that need to be considered, such as various criteria of evenly spreading of test cases, ways of defining the candidate sets.

In this project, I think I will use the method provided by the paper 'Adaptive random testing'. The method implementation of Adaptive Random Testing is as follows: Adaptive random testing makes use of two sets of test cases, the executed set and the candidate set which are disjoint. The executed set is the set of distinct test cases that have been executed but without revealing any failure;

while the candidate set is a set of test cases that are randomly selected without replacement. The executed set is initially empty and the first test case is randomly chosen from the input domain. The executed set is then incrementally updated with the selected element from the candidate set until a failure is revealed. From the candidate set, an element that is farthest away from all executed test cases, is selected as the next test case.

Plan

The plan for this project is as follow.

April 20 to April 24: Read more papers related with adaptive random testing and random testing, and find some other way to realize adaptive random testing.

April 25 to May 16: Implement adaptive random testing with TSTL. This includes finding functions within the TSTL API that supports ART and the methods found in the papers.

May 17 to May 31: Improve the first version of adaptive random testing. Make this method more efficient.

June 1 to June 6: Finish the project: write up documentations, and final optimization.

Reference

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

[2] Richard Hamlet (1994). "Random Testing". In John J. Marciniak. Encyclopedia of Software Engineering (PDF) (1 ed.). John Wiley and Sons.