Using Adaptive Random Testing for Test Generation

CS569 Proposal

Xuhao Zhou

Introduction

The white box and the black box approach are mainly two approaches towards the selection of test cases. Random selection of test cases is a simple and intuitively attractive technique among the black box techniques. Test cases are chosen randomly based on a uniform distribution in random testing. A study [1] shows that in addition to depending on the failure rate, the performance of a partition testing strategy depends on the geometric pattern of the failure-causing inputs. A study conducted by Chen et al. [2] shows that a new type of random testing namely adaptive random testing obviously improved the effectiveness of random testing without increasing significant overheads. Chan et al. [1] have classified the patterns of failure-causing inputs into three categories: point, strip and block patterns. An inspection shows that for non-point patterns, the failure detection capability can be greatly improved by modifying the normal random testing technique, because more evenly spread test cases have a better chance of hitting the failure patterns. One thing we can do is to make sure that the new test case should not be too close to any of the previously generated ones. In order to achieve this, we can generate some random test cases and then choose the most suitable one among them and try to distribute the selected test cased as spaced out as possible. According to Chen et al. [2], "adaptive random testing makes 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 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." An experiment-based research was conducted by Chen et al. [2] to compare the performance between adaptive random testing and ordinary random testing. This research shows an encouraging and promising result. That is adaptive random testing outperforms random testing greatly for most of the cases because an even spread of test cases is more likely to find out failures using fewer test cases than random testing. Therefore, I want to use adaptive random testing method for test generation.

Plans

Apr 20 - May 3: Finish a basic implementation of a novel test generation algorithm using the TSTL API.

May 3 - May 18: Vary the parameters and SUTs.

May 19 - June 6: 1. Refine the implementation with a more-flexible version of algorithm. The implementation takes more parameters, or has a nicer configuration interface, like the TSTL random tester and other generators.
2. Finish a project document that includes graphs to evaluate the algorithms and describe related work.

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, I. K. Mak: Advances in Computer Science - ASIAN 2004. Higher-Level Decision Making Volume 3321 of the series Lecture Notes in Computer Science pp 320-329
http://link.springer.com/chapter/10.1007%2F978-3-540-30502-6_23#page-1