Hao Liu
Applying ART by using APIs in TSTL
Milestome#1
5/5/2016

Introduction:

My proposal is about applying genetic algorithm in TSTL, but it is hard and I can’t get any help from the class. So I decide to change my subject to the common one. And all my works are based on the paper “Adaptive Random Testing” by T.Y. Chen, H. Leung and I.K. Mak.

The mainly differences between adaptive random testing and ordinary random testing are uniform distribution and without replacement. These features not only give adaptive random testing easier mathematical model for analyzing, but also provide testing result which is closer to the reality. Therefore, adaptive random testing is believed that providing a better random testing method. Because I don’t take the testing input as alone points in adaptive random testing, now we can judge the following test case is humble or not by observing the range or distance of testing inputs. In other word, in adaptive random testing, a pure random test is necessary, in order to get credible result and good performance.

Algorithm:

Input:
Since I’m new to python, I have used codes in coverTester.py. And the seven inputs are showed at the beginning of the code. TIME_BUDGET is the total time my tester will run on the sut.py. From my result, usually within 1 min the output is almost the same. And faults is just a switch for the tester. Coverage gives the coverage report and running is just the switch for new branch discovered report.
Process:
Within a given time we will keep doing random test until we hit a bug. The input will judge whether we should print out the new branch or not. And any new statement will be stored in the storedTest. While we are exploiting we will count the statement number as coverage.
Output: bug numbers, action numbers and total running time.

Future work

This is just an implementation of what we have learned in class. So far I have not applied any concept in ART in my tester. In the paper, authors give two algorithms for different purposes. Algorithm 1 describes how to choose set and test case efficiently. The process is repeated with various randomly selected inputs as the first test case. If we have enough statement as input, the result will be more credible. Algorithm 2 is known as FCFS which is *Fixed Size*

Candidate Set Version of the Adaptive Random Testing. This algorithm selects new sets and constant size of the input, which means each time we run algorithm 2 the input set is filtrated. I think if I apply algorithm 2 somewhere in my tester, it will be more efficient.
But first I'm considering finding some classmate to work with each other. I need to discuss some python problem with someone who has experience in python.

Reference

Chen T Y, Leung H, Mak I K. Adaptive random testing[M]//Advances in Computer Science-ASIAN 2004. Higher-Level Decision Making. Springer Berlin Heidelberg, 2004: 320-329.