Hao Liu

Applying ART by using APIs in TSTL

Milestome#2

5/18/2016

Introduction:

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:

The input is almost the same as tester1. 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. And faults are just a switch for the tester. Coverage gives the coverage report and running is just the switch for new branch discovered report. Last time I failed to put internal report in my tester and I add it this time. With internal report we can easily find the new action and which action has been tested the most time.
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.
The program will randomly test action until hitting a bug. And any new actions will be stored as new statement in order to count the depth of the test. By considering the coverage and depth we can figure out whether my tester is efficient or not.
Another limitation for the whole program is width, which means how many states have been visited during the testing. And for the same level of state the tester will decide if a new action appears or it is an enabled action.

Future work

The ART Algorithm describes how to choose set and test case efficiently. The process is

repeated with various randomly selected inputs as the first test case. And in the class mutation is another method of picking up the test sample, which is from genetic algorithm. I think use genetic algorithm in ART is kind of interesting and in genetic algorithm, one is identifying the most critical path clusters in a program. The SUT is converted into a CFG. Weights are assigned to the edges of the CFG by applying 80-20 rule. 80 percentage of weight of incoming credit is given to loops and branches and the remaining 20 percentage of incoming credit is given to the edges in sequential path. How to covert the SUT states into CFG and find an algorithm to search might be the future work

Reference

[1] 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.

[2] Praveen Ranjan Srivastava and Tai-hoon Kim,“Application of genetic algorithm in software testing” International Journal of software Engineering and its Applications, 3(4), 2009, pp.87 – 96