Mirror Adaptive Random Testing for TSTL

CS 569 – Project Proposal

Swathi Sri Vishnu Priya Rayala (932-696-872)

Background

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. Thus, RT is used to assess software reliability as well as to detect software failures by simply selecting test cases in a random manner from the whole input domain. However, there is a lot of scope in improving the Random Testing to detect the software failures depending on failure-causing inputs. Generally, there are three types [1] of failure-causing inputs – Strip Pattern, Block Pattern and Point Pattern. For most of the faulty programs, Block Pattern and Strip Pattern types occurs more frequently.

Here comes the enhancement for RT, the Adaptive Random Testing (ART) which uses fewer test cases to detect the first software failure. ART has been widely applied in different scenarios, such as numerical programs, some object-oriented programs, and mobile applications [2]. The intuition for ART is that the fault causing pattern is fixed but unknown. So, for such non-point failure patterns, an even spread of random test cases will enhance the fault detection capabilities. However, ART requires extra computations to ensure an even spread of test cases, which may render ART to be less cost-effective than RT.

Proposal

In this proposal, I want to implement the innovative approach, Mirror Adaptive Random Testing (MART) that reduced the computational costs significantly in TSTL. If the number of test cases to be tested are reduced, there is a great chance of finding the fault causing inputs in less time. Basically, in this [1] paper, the input domain of the program to be tested is divided into m disjoint subdomains. One subdomain is designated as the original subdomain, and the MART algorithm is only applied to this original subdomain. If the selected test case in the original subdomain is executed and failure is not detected, this test case is mirrored successively into other subdomains, known as mirror subdomains. This is achieved by means of mirror functions, which generate (m-1) distinct test cases, one within each of the mirror subdomains. If none of these (m-1) test cases reveals a failure, resume the process of ART in the original subdomain. When compared to the other methods, MART showed 75% reduction in distance computations and performed the best when fault casing inputs are block-patterned.

Plan

So my plan for implementing this approach in TSTL would be initially survey the papers related to ART and MART. Next, I try to find out the methods in TSTL that supports MART functions. I will the existing AVL tree example in order to test the performance of the test cases generated with MART. Later, compare code coverage and execution time with the existing random tester of TSTL.

References

[1] “Mirror Adaptive Random Testing”, T. Y. Chen ; Sch. of Inf. Technol., Swinburne Univ. of Technol., Hawthorn, Vic., Australia ; F. C. Kuo ; R. G. Merkel ; S. P. Ng

[2] “Adaptive Random Testing”, T.Y. Chen1, H. Leung, and I.K. Mak