Proposal of Testing Project in CS569

Student: Kun Chen
Email: chenk4@oregonstate.edu

Introduction of ART and its efficiency:

Random Test (RT) algorithm is a widely used and known as a black-box testing method by using random technique. As RT can generate the input values randomly, the cost of RT in the test is not expensive. On the base of RT, Chen Y. T et al has proposed another version named Adaptive random testing (ART). Compactness of failures in both regular and irregular failure regions are studied and the obtained results reveal that the performance of ART will be better with the increment of compactness of failure regions [1]. In another reference [2], it shows that RT algorithm will create more duplicated testing cases than the ones in ART algorithm if given the same size of testing cases. In addition, ART algorithm will not be affected by the given input type and size. In addition, adaptive Random Testing (ART) can be affected by dimensions. Reference [3] studies the reason why ART algorithm may not become effective for higher failures. And then in the same reference the spatial distribution of cases in higher dimensions are well studied.

TSTL part:

Alex has proposed TSTL, the template scripting testing language, to help find bugs in libraries which is required to be written in Python [4]. By making use of TSTL, the Software Under Test (SUT) is needed to support the TSTL, and SUT is created by the functions in libraries for testing. Alex also have given application examples such as the one by using Random Test (RT) algorithm to test the AVL Tree library, the source code are available in the github[5].

Application of ART in TSTL:

As related studies in Reference [1][2][3] reveal that ART have better characterizes than RT algorithm, and RT have been proved to be available in TSTL, so the ART algorithm is expected to be effective in TSTL. We can use AVL tree library as a testing example and then compare the results of RT and ART. And during the test task, we can also try to reveal the disadvantage of ART algorithm in TSTL. There still several problems I may confront, e.g., one parameter called "distance" in ART is determined by users, and this parameter will affect the success of implementation of ART, so how the give a reasonable definition of this parameter is a challenge for me. In addition, how to well apply ART algorithm in high dimension with less failure is another challenge too.

In order to complete the present proposal well, the master of the knowledge in Python programming language, TSTL, SUT and ART are necessary. With the process of this project, small revisions of the proposal will be carried out for a better testing result.

Reference:

1) CHEN Tsong-Yueh, KUO Fei-Ching, SUN Chang-Ai. Impact of the Compactness of Failure Regions on the Performance of Adaptive Random Testing. Journal of Software, Vol.17, No.12, 2006, pp.2438−2449.

2) CHEN Tsong-Yueh, KUO Fei-Ching. Is Adaptive Random Testing Really Better Than Random Testing. Proceedings of the First International Workshop on Random Testing (RT'06), 2006, pp.64-69.

3) Johannes Mayer, Christoph Schneckenburger. Adaptive Random Testing with Enlarged Input Domain. Proceedings of the Sixth International Conference on Quality Software (QSIC'06). 2006

4) Alex Groce, Jervis Pinto, Pooria Azimi, Pranjal Mittal. TSTL: A Language and Tool for Testing (Demo). ISSTA'15 , July 12–17, 2015

5) https://github.com/agroce/tstl