

Course: CS569
Name: Yanshen Wang
Date: 04/19/16
Proposal of Project

Feedback – directed Random Test Generation in TSTL

1 Project Description

For a long time, there exists a controversial topic about the efficiency between the systemic testing and the random testing. As we know, the random testing is a useful and convenient technique to help programmers generate some independent inputs and verify the output testing is whether pass or fail. However, some theoretical experts believe that in the practical works the random testing is less effective than the systemic testing since a couple of reasons [1, 2]. The first reason is that they think some interesting and specific tests cannot be generated randomly, so the random testing will waste lots of time for those testing work [2]. Furthermore, the random test generation will get a lower level of code coverage than the systemic testing, in case of the modeling checking, symbolic execution, chaining and so on [4].

However, the Feedback – directed random test is a good technique to improve the random generation because this method can provide some feedbacks for the created inputs to check out whether or not they are suitable for the latter function calls [3]. First of all, it will select the inputs randomly from the previous method calls. Then according to a set of constraints and filters, it will mark and divide those generated inputs into 2 kind of list sequence, which called Error Sequence and No Error Sequence. After execute this method, it can generate more useful inputs for the random test. In the article about Feedback – directed random test generation, authors executed this technique by Sun' JDK 1.5, so in this project I want to try to understand this method and implement it by TSTL.

2 Random test generation algorithm

Actually, the author has already introduce some general idea about this Feedback – directed Random Test Generation [3]. In this algorithm, the generate sequence takes 4 inputs including classes, contracts, filters, and time limit. Under the time condition, it create a new sequence, which contains the inputs after the random public method execute the classes. In fact, each sequence associate with a Boolean flag according the set of contracts and filters: if it is a redundant illegal input, this new sequence belongs to the Error Sequence, otherwise it will belong to None Error Sequence. After the algorithm create the new sequence, there is another helper function to check this sequence in each method calls and return the value to indicate it is satisfied or violated.

3 Project Plan

I will implement this project in the following steps:

- 1) I will continue to research more information about the method of Feedback – directed random test generation, and try to understand and find more details about this technique.
- 2) I need to continue to study more knowledge about the python language, and find ideas about how to implement this algorithm by python.
- 3) I will try to write the first edition of the randomtest.py, which can import the SUT file given by the professor.
- 4) According to the requirement, I will try to improve the previous edition of the code and complete my final paper.

References

- [1] D. Hamlet and R. Taylor. Partition testing does not inspire confidence. IEEE TSE, 16(12):1402–1411, Dec. 1990.
- [2] J. W. Duran and S. C. Ntafos. An evaluation of random testing. IEEE TSE, 10(4):438–444, July 1984.
- [3] Pacheco, Carlos, et al. "Feedback-directed random test generation." Software Engineering, 2007. ICSE 2007. 29th International Conference on. IEEE, 2007.
- [4] R. Ferguson and B. Korel. The chaining approach for software test data generation. ACM TOSEM, 5(1):63–86, Jan. 1996.