

CS 569 Proposal

Instructor: Professor Alex Groce

Student Name: Yipeng Wang

ID: 931903609

Date: April 19, 2016

Feedback-directed Random Test Generation in TSTL

Background and Random test generation algorithm

In current period, the utilization and development of computer has been more and more common. We could utilize computer in our daily lives which could save us bunch of time and provide much more accuracy. However, a serious problem appeared, sometimes when we utilize the computer, we will meet some error which cannot return the expect result which we want, those bugs are annoying. Therefore, solve the bugs has become a very important issue for all of the software programmers. This term, we learned a new method to detect bugs from certain libraries, which called TSTL, template scripting testing language tool. The pre-installed tester of TSTL is random tester, which could generate data randomly to detect whether it has bugs and it is very useful and convenient technique to help the software programmers. However, there are some problems appearing, how to find the data which will have high radio coverage or even which part of data we should test to get the most efficient method. Usually, for random tester, we should simply generate some random data value in target program, but it will lead to a new problem that we cannot avoid the redundant for random test generation. To solve this problem, usually, people will choose to check the data after test, then after new test finish, we need to check the difference between this test and the previous one. For the feedback-directed random test generation, it will generate the incremental data sequences. Then it will take four types of input: classes, contracts, filters, and time limit. In time condition part, it will randomly choose the sequences to append with the data sequences which we test previous. After appending, classify the result, for each sequence, it has a Boolean flag to determine the set of contracts and filters: if it is redundant or illegal, we will put it into error sequence and ignore it; if it has bugs, we will record the bugs in logs; if it is useful, we can use it again in future test appending [1].

Describe your project plan and what is your idea for test generation?

For my project plan, first of all, since the feedback-directed random test method will use java to implement, and I am not very familiar with java, so I think I need to learn java more before start my project. In addition, I need to find more paper about feedback-directed random test, since only one paper is not enough. After that, I will focus on how to implement the feedback-directed random in TSTL. For the time condition part, it will classify into four different types, how to implement it should be one of the biggest problem I need to solve in this project. At last, since feedback-directed random will classify the result, which will take more time than just implement random tester, so I need to find some method to help me save time as much as possible.

Reference:

[1] C, Pacheco, S, K. Lahiri, M, D. Ernst, and T, Ball. Feedback-directed random test generation. In Proceedings of the 29th International Conference on Software Engineering, 2007.