

CS 569 - Part 1 – Project proposal
Prof. Alex Groce
Eman Almadhoun – 932909951

Nowadays, computers are very important part in our life which needs many programs to run on them. But for ensuring the reliability and quality of these programs, we need to test them to ensure that there are no faults. Software testing plays a very important role in the software development life cycle. We need Software testing to ensure the quality and reliability of the software and caught any bugs in the system and analyze them. Doing that by hand is much harder, detecting all the bugs are impossible and takes so long time. So, we need an automated tool for generating unit test to find bugs in very short time with the lowest cost. Test generation is a test that generates set of test cases, each of them has a sequence of statements which depend on criterion specified [1][2].

In my project, I will use seeding strategies for solving the testing problems and finding bugs from previous knowledge in very short time. Some of the seed strategies which I am going to look at in my project, seeding of constants string which consequent dynamically and statically from the source code, seeding of values detected during test execution, seeding of information type and seeding from previous tests. Software Under Tests (SUT) needs to pass parameters via its functions call. By scan the source code, we can generate these values instead of generating them randomly. So, this method will improve the performance of the test generation by maximized code coverage [2].

These seeding techniques are used in an automated testing tool called EVOSUITE. This tool is based on a genetic algorithm(GA). Genetic Algorithm is based on the population of individuals which need to be selected iteratively and applying crossover and mutation operations on them [2]. EVOSUITE tool maximizes the coverage by using fitness function [4] which is the feature of finding a good solution [3]. Investigation on EVOSUITE tool can help understanding the seeding technique.

This technique allows us to generate more test cases which increase the chance of finding bugs. So, it will increase the reliability of test generation. Why seeding technique is import because it is very cheap. When the test generation tests target, for example, braches coverage, it searched for it one at a time. If we have many targets inside the same branch, so when we chose one of the targets and generate test data, the result of this test can be reused as a start point to test other targets, instead of restart the test from the start [2]. Scanning the source code for Software Under Test and asserting the right inputs for the SUT could be a very challenge as the string domain, for example, is so large. Also, test generation using seeding maximizes code coverage according to [2].

Finally, I will investigate more about seeding technique by reading some papers and I will focus on “Seeding strategies in search-based unit test generation” paper which explain this technique using Java Language. But I am focusing on my project to implement the test generation using Python Language as required to test sut.py code which generated by TSTL complier. The seeding technique reduces the testing time compare to the random testing and it is not an expensive technique. I will assert that my test generator can test the source code of AVL tree and other codes.

References:

1. Sao N., Patel N., “A Survey on Automated Test Data Generations by Using Hybrid Approach”, International Journal of Innovative Science Engineering & Technology (IJSET), 2015, Vol. 2 Issue 4.
2. S Rojas1 J., Fraser G., Arcuri A. Seeding strategies in search-based unit test generation. Software Testing, Verification and Reliability (STVR) 2016;
3. Harman M, Jones BF. Search-based software engineering. Journal of Information & Software Technology 2001; 43(14):833–839.
4. Fraser G, Arcuri A. Whole test suite generation. IEEE Transactions on Software Engineering 2013; 39(2):276–291.