

PROJECT PROPOSAL

Name: Xuan He

Project

It is widely recognized that software testing is an essential component of any successful software development process. A software test consists of an input that executes the program and a definition of the expected outcome. There are many testing methods to find the bug of the programs.

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application.

In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method.

Unit testing can find problems early in the development cycle. This includes both bugs in the programmer's implementation and flaws or missing parts of the specification for the unit. The process of writing a thorough set of tests forces the author to think through inputs, outputs, and error conditions, and thus more crisply define the unit's desired behavior. The cost of finding a bug before coding begins or when the code is first written is considerably lower than the cost of detecting, identifying, and correcting the bug later; bugs may also cause problems for the end-users of the software.

Software testing is a difficult and expensive task. Therefore, many automatic test generation tools aimed at lowering the cost of writing tests by enabling users are developed in order to derive tests automatically. EvoSuite is a tool that automatically generates JUnit test suites for Java classes, targeting code coverage criteria such as branch coverage. It uses an evolutionary approach based on a genetic algorithm to derive test suites. To improve readability, the generated unit tests are minimized, and regression assertions that capture the current behavior of the tested classes are added to the tests.

Therefore, after doing a lot of research, I decide to write a unit testing generator, which can do unit testing in TSTL, based on the design of EvoSuite in my project.

EvoSuite is a very popular tools for unit testing in Java and has been used on more than an hundred open-source software and several industrial systems, finding thousands of potential bugs, so I think the unit test generation algorithm that I develop based on EvoSuite is also an effective testing tools.

Plan

From April 20 to April 27, I will do more research about unit testing and EvoSuite to find out some information that could help me design the unit test generator.

From April 28 to May 3, I will try to do some basic implementations of my unit test generation algorithm using the TSTL API.

From May 3 to May 18, I will improve my unit test generation algorithm to make it more efficient and complete.

From May 18 to June 6, I will work on the final report and the final version of my unit test generation algorithm.

Reference

1. Gordon Fraser and Andrea Arcuri. 2011. EvoSuite: automatic test suite generation for object-oriented software. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering (ESEC/FSE '11)*. ACM, New York, NY, USA, 416-419.
<http://www.evosuite.org/wp-content/papercite-data/pdf/esecfse11.pdf>
2. Gordon Fraser and Andrea Arcuri. 2013. EvoSuite: On the Challenges of Test Case Generation in the Real World. In *Proceedings of the 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation (ICST '13)*. IEEE Computer Society, Washington, DC, USA, 362-369.
http://www.evosuite.org/wp-content/papercite-data/pdf/icst13_tool.pdf
3. Gordon Fraser and Andrea Arcuri. 2013. Whole Test Suite Generation. *IEEE Trans. Softw. Eng.* 39, 2 (February 2013), 276-291.
<https://pdfs.semanticscholar.org/e6a4/8b6943d578cd8532f510541f9e96cdf9ff28.pdf>