Test Case Selection Using Feature Extraction and Clustering

Angelin Gladston, College of Engineering, Anna University, Chennai, India
H. Khanna Nehemiah, College of Engineering, Anna University, Chennai, India
P. Narayanasamy, College of Engineering, Anna University, Chennai, India
A. Kannan, College of Engineering, Anna University, Chennai, India

ABSTRACT

This article explains the selection of important parameters from an execution pattern which brings out the details of the application of test cases. Hence, execution profiles are captured and a new execution profile-based clustering approach is chosen for test case selection, which uses three new features. These are Function frequency, Branches taken and Block percentage. The test cases are clustered using the extracted features. The experiments show that the proposed FBB selects smaller size of more relevant test cases which are more fault revealing compared to the existing Function Call Profile approach.

KEYWORDS

Clustering, Selection Heuristic Feature Extraction, Test Case Selection, Test Suite

INTRODUCTION

An efficient way to conduct regression testing is to find a minimal subset of test cases (Sumit, Rajesh, & Dhavleesh, 2016) so that the minimized test suite can be executed even with resource constraints (Mohamad, Joanna, & Philip, 2016), involving both time and cost. A suitable subset could be found during test case generation or after creating the test suite. Test suite size reduction results in quick conduct of testing. The process of finding a minimal subset is called test suite reduction, test suite minimization or test case selection, which results in a so called representative set and normally coverage based techniques are used to arrive at it.

Test suite contains test cases meant for varied program requirements. Out of evolution the test suite grows and few test cases may become redundant. Thus, not all test cases need to be used for conducting regression testing. Coverage based test suite reduction techniques help in keeping test cases based on the coverage requirement used and the redundant ones based on the coverage requirement are removed. But the test case removed because it is redundant with respect to a coverage requirement may not be redundant with respect to another coverage requirement. Further the coverage based test suite reduction technique does not address the dynamic behavior of software. There is no room to bring in the importance of the frequent usage of functionality, complex conditional block capable of revealing more faults and sequence of unusual branches taken, which shows the possibility of having fault. The execution pattern of software throws light on information like, what part of the software gets executed often, what are the functionalities used repeatedly, critical functions, coverage information, and detail about whether a branch is taken or not. Rather than reducing test suite based
on coverage information (Gregory, Matt, Michael, & Mats, 2015), if the execution behavior of the software can be profiled and analyzed, then more relevant test cases which are more fault revealing can be selected. Identifying and using vital information for deciding or selecting relevant test cases plays an important role (Johny, Mor, Pnina, & Yaron, 2013).

Execution profile of a software contain the details of functionalities, blocks, branches, statements, conditions and the complex sequences of them exercised in the executions. The profile information collected can be analyzed using clustering technique (Michail & Nikolaus, 2013). Since clustering is a technique which helps in grouping candidates with similar behavior, test cases can be clustered based on execution pattern which are captured from execution profiles. Hence test cases can be selected based on dynamic behavior of software, rather than by coverage alone, which brings in hope for selecting more fault revealing test cases. This kind of selection is made possible by exploring the execution profile and identifying the complex combination of executions and rare sequence of executions which are more fault revealing. Execution profile based clustering technique based on function call profile has been used by Zhang et al. (2010).

The persisting necessity to conduct effective regression testing is the main motivation behind this work. The sole focus is to improve existing test case selection approach so as to assist in regression testing. The review on existing works highlighted the importance of finding and using new decision making parameters. To incorporate all possible aspects, multiple parameters need to be identified and used (Arunasalam & Vidyasagar, 2017). Based on that, in this work, an execution profile based clustering technique, based on three new features namely, function frequency, branches taken and block percentage of the test suite is used. The test cases are executed and these features are extracted from their execution profile. The test cases are then clustered using K-means algorithm based on these extracted features. From the resulting test case clusters, required number of test cases is selected using the selection heuristic. First, one test case from each cluster is selected, and if there are more unsatisfied requirements, more test cases are selected from the clusters. Cluster holding fault revealing test case and cluster having unusual number of test cases are taken into consideration since they exhibit that behavior, those test cases are selected.

The remaining of this paper is organized as follows: Related work describes works related to test case selection and works related to clustering of execution profiles. Proposed profile based clustering describes the proposed profile based clustering approach for test case selection and the approach used for evaluation. Experimental design and results provides the implementation details along with discussion on result analysis and conclusion discusses the conclusion and future work.

RELATED WORK

Selecting test cases to assist in regression testing is usually carried out using coverage-based as well as profile-distribution based techniques. In coverage-based test case selection, test cases are selected based on coverage criterion. A minimal set of test cases satisfying the adequacy criteria are selected. These selected test cases form reduced test suite in test suite reduction. Even though solutions exist for test suite reduction, still redundant test cases remain in the reduced test suite. In order to solve this problem, two HGS based heuristic algorithms, namely, Non Redundant HGS and Enhanced HGS are introduced by Angelin, Nehemiah, Narayanasamy, and Kannan (2015). The former utilized the redundant strategy available with Greedy, Redundant, and Essential (GRE) to get rid of redundancy, whereas the latter selected a test case for higher cardinalities based on overall coverage of unmarked associated testing sets and thus arrived at reduced test suite without redundant test cases. The Enhanced HGS and Non Redundant HGS algorithm selected a minimal subset that was much smaller when compared to that of original HGS. These techniques improved upon the HGS heuristics by removing redundancy in the selected test cases. In the experiments, Enhanced HGS showed 4.65% improvement and Non Redundant HGS showed 2.81% improvement compared to existing HGS in suite size reduction.
Outsourcing in Knowledge-Based Service Firms
[www.igi-global.com/article/outsourcing-knowledge-based-service-firms/55600?camid=4v1a](www.igi-global.com/article/outsourcing-knowledge-based-service-firms/55600?camid=4v1a)

Strategies to Implement Edge Computing in a P2P Pervasive Grid
[www.igi-global.com/article/strategies-to-implement-edge-computing-in-a-p2p-pervasive-grid/193590?camid=4v1a](www.igi-global.com/article/strategies-to-implement-edge-computing-in-a-p2p-pervasive-grid/193590?camid=4v1a)