International Journal of Open Source Software and Processes
Volume 7 • Issue 4 • October-December 2016

A Novel Method for Test Path Prioritization using Centrality Measures

Amita Jain, Ambedkar Institute of Advanced Communication Technology and Research, Delhi, India
Devendra Kumar Tayal, Indira Gandhi Delhi Technical University for Women, Delhi, India
Manju Khari, Ambedkar Institute of Advanced Communication Technologies and Research, Delhi, India
Sonakshi Vij, Indira Gandhi Delhi Technical University for Women, Delhi, India

ABSTRACT

Software testing is an essential stage of the software development life cycle which helps in producing bug free software. This paper introduces a strategy to create test data consequently from the beginning of test information which is tested against the Program under test (PUT) for ampleness criteria. Initially this process produces test information set arbitrarily where a unique approach for the test path prioritization process is presented that uses the concept of centrality measures. The proposed algorithm computes the importance of each node in the test paths by using various centrality measures and thus identifies the significance of each path. Furthermore, the proposed methodology shows the maximum number of potential nodes that are covered using a less number of prioritized paths. This paper tests the created test information against the product to check its sufficiency.

KEYWORDS

Betweenness, Centrality Measures, Closeness, Degree, Path Prioritization, Software Testing

DOI: 10.4018/IJOSSP.2016100102

INTRODUCTION

A major problem that the software industry encounters is to develop bug free software. As per latest IBM reports "*31% of the projects get cancelled before they are completed, 53% over-run their cost estimates by an average of 189% and for every 100 projects, there are 94 restarts*" (Aggarwal & Singh, 2006). Therefore, there is an urgent need to adopt better software engineering concepts and strategies in order to improve software quality. Software testing is of utmost importance amongst the phases of project development since it aims at demonstrating that errors are not present in the program under consideration. The purpose of testing is to show that a program performs its intended functions correctly and establish confidence that it does what it is supposed to do. The two major types of testing include structural and functional testing (Paul et al., 2011). In functional testing, various functions are tested for quality assurance but the internal structure of the program is rarely considered. It usually describes what the system does. On the other hand, structural testing aims at testing the internal structure of the software. The technique that has been discussed in this paper is Path Testing which is a type of structural testing (Tutorials Point, 2017). It is the name given to a group of testing techniques that are based on judiciously selecting a set of test paths through the program. A path in a graph is a finite or infinite sequence of edges which connect a sequence of vertices which, by most definitions, are all distinct from one another. If the set of paths is properly chosen then it means that one has achieved some measure of test thoroughness. For example, picking enough paths to assure that every source statement is executed at least once. Hence it becomes mandatory to have the complete knowledge of the program's structure. This type of testing involves generating a set of test paths that will cover every branch in the program as well as finding a set of test cases that will execute every path in this set of program paths.

An important aspect of path testing is path optimization which does the task of decreasing the number of paths to be tested by prioritizing the independent paths. Till date very little work is done in optimization of path testing. However, Cyclomatic Complexity is one of the approaches that were introduced to directly measure the number of linearly independent paths through a program's source code. This minimized the number of paths to be tested in comparison to normal path testing where all dependent and independent paths are tested. Researchers identified that the effectiveness of path testing rapidly deteriorates as the size of software under test increases (Goldberg & Harrelson, 2005; Malhotra & Manju, 2013). As the size of software increases it becomes difficult to test every possible path in the program. Hence there should be a technique for path prioritization so that number of paths to be tested can be minimized according to their prioritized value. Since the introduction of Cyclomatic Complexity, very little work has been done in the field of path prioritization.

This paper uses a graph based approach for path prioritizing that aims at path testing optimization. It relies on the concept that the paths which have more number of important nodes are considered to be more important from testing point of view. In case of reducing the testing efforts, one can neglect least important paths and consider