IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Demystifying Defects: Test Case Purification for Augmented Fault Localization in Data-Centric Software Regression

[1]BINOOP KUMAR C A, [2]SUDHA P

[1]Assistant Professor, [2]Assistant Professor,
[1]Department of computer science,
[1]Thunchathezhuthachan College Elavanchery

Abstract:

Regression testing is a crucial process in software development, aimed at validating modified or added functionality and detecting faults introduced to previously tested code. However, the execution cost of regression testing can be substantial, particularly for large software systems with extensive test suites. This paper presents a novel approach to improve fault localization in regression testing for database applications by employing test case purification techniques. The proposed method involves clustering test cases based on their access to database entities, followed by the application of spectrum-based fault localization algorithms on purified test cases. Test case purification is a process that generates refined test cases containing a single assertion, thereby enhancing the discrimination power of fault localization techniques. The research evaluates the effectiveness of this approach using two database-driven software products, "Estafeta" and "Silabo," and compares the performance of three spectrum-based fault localization methods: Tarantula, Ochiai, and Jaccard. The results demonstrate that test case purification can significantly improve fault localization capabilities, with the Tarantula method outperforming the others in terms of fault detection rates. The proposed approach offers a promising solution for efficient and cost-effective regression testing in database applications, contributing to improved software quality and reduced development costs.

Introduction:

Software regression testing is an indispensable practice in the software development life cycle, particularly when modifications or enhancements are introduced to existing code bases. Its primary objective is to validate the modified software and ensure that no new faults have been introduced to previously tested functionality. However, the execution of regression test suites can be time-consuming and resource-intensive, especially for large and complex software systems with extensive test coverage.

One of the key challenges in regression testing is fault localization, which involves identifying the root cause of failures in test cases. Effective fault localization is crucial for efficient debugging and timely resolution of software defects. Traditional fault localization techniques often rely on the coverage information of test cases to rank the potential fault locations in the code.
In the context of database-driven applications, where the software interacts with and manipulates data stored in databases, regression testing poses additional complexities. The interactions between the application code and database entities, such as tables, views, stored procedures, and triggers, must be thoroughly tested to ensure data integrity and correctness.

This paper introduces a novel approach to enhance fault localization in regression testing for database applications by leveraging test case purification techniques. The proposed method involves clustering test cases based on their access to database entities and applying spectrum-based fault localization algorithms on purified test cases. Test case purification is a process that generates refined test cases containing a single assertion, thereby increasing the discrimination power of fault localization techniques.The research evaluates the effectiveness of this approach using two database-driven software products, "Estafeta" and "Silabo," developed for an Ecuadorian university. The performance of three widely-used spectrum-based fault localization methods, namely Tarantula, Ochiai, and Jaccard, is compared to assess the impact of test case purification on fault detection rates.

Background and Related Work:

Regression testing is a well-established practice in software engineering, and numerous techniques have been proposed to optimize its efficiency and effectiveness. Test case selection, prioritization, and minimization are commonly employed strategies to reduce the execution cost of regression testing while maintaining adequate fault detection capabilities.Fault localization is a crucial aspect of the debugging process, enabling developers to pinpoint the root cause of software failures quickly and accurately.

Various fault localization techniques have been explored, including spectrum-based fault localization, which utilizes the coverage information of test cases to rank potential fault locations in the code. In the context of database-driven applications, regression testing poses unique challenges due to the complex interactions between the application code and database entities. Prior research has explored various approaches to address these challenges, such as safe regression testing techniques that incorporate white-box and graph-based analysis of database interactions (Willmor & Embury, 2005), as well as black-box testing methodologies based on test case similarity and database state deviations (Vangala et al., 2009; Rogstad & Briand, 2010).Test case purification is a relatively recent technique that aims to enhance the effectiveness of fault localization by generating refined test cases containing a single assertion. By isolating individual assertions, test case purification increases the discrimination power of fault localization algorithms, enabling more accurate identification of fault locations (Zhang et al., 2013). However, the application of test case purification in the context of database-driven applications has not been extensively explored.

Proposed Approach:

The proposed approach combines test case clustering, test case purification, and spectrum-based fault localization techniques to improve fault localization in regression testing for database applications. The methodology consists of the following steps:

1. Test Case Selection: Identify the relevant test cases associated with the modified or added functionality, including those that access the database entities affected by the changes.

2. Test Case Similarity Matrix Construction: Create a binary matrix representing the similarity between test cases based on their access to database entities, such as tables, views, stored procedures, and triggers. A value of 1 indicates that the test case accesses the corresponding database entity.

3. Test Case Clustering: Apply an unsupervised clustering algorithm, such as the Expectation-Maximization (EM) algorithm, to group test cases based on their similarity matrix. This step helps identify clusters of related test cases that access similar database entities.

4. Cluster Selection: Select the clusters containing test cases associated with the modified or added functionality, as well as test cases that revealed faults during previous executions.

5. Test Case Purification: For each selected cluster, perform test case purification to generate refined test cases containing a single assertion. This process involves three sub-steps:
   a. Test Case Atomization: Replace each failed test case with multiple single-assertion test cases, where each assertion is isolated.
   b. Test Case Slicing: Remove unrelated statements from each single-assertion test case using dynamic slicing techniques.
   c. Rank Refinement: Re-rank the statements in the code based on the coverage information of the purified test cases, using a spectrum-based fault localization algorithm.

6. Fault Localization: Apply spectrum-based fault localization algorithms, such as Tarantula, Ochiai, and Jaccard, to the purified test cases and analyze their effectiveness in identifying fault locations.

Experimental Evaluation:
To evaluate the proposed approach, two database-driven software products were used: "Estafeta" and "Silabo." Estafeta is a system for organizing the weekly schedules of professors at , while Silabo is used for managing course materials taught by the professors.

The experimental setup involved the following steps:

1. Test Case Collection: Collect the test cases associated with each version of the software products, along with the number of faults identified in each version.

2. Clustering and Test Case Selection: Perform test case clustering based on database entity access and select the relevant clusters for regression testing.

3. Test Case Purification: Apply the test case purification process, including atomization, slicing, and rank refinement, to the selected test cases.

4. Fault Localization: Execute the Tarantula, Ochiai, and Jaccard spectrum-based fault localization algorithms on the purified test cases and record the fault detection rates.

5. Performance Evaluation: Analyze the effectiveness of the proposed approach by comparing the fault detection rates, test suite reduction percentages, precision, recall, and F-measure values across the different fault localization methods and software products.

Results and Discussion:

The experimental results demonstrated the effectiveness of the proposed approach in enhancing fault localization through test case purification. The key findings are as follows:

1. Fault Detection Rates: The Tarantula fault localization method outperformed the Ochiai and Jaccard methods in terms of fault detection rates for both the Estafeta and Silabo software products. Specifically, Tarantula identified 43.28% and 51% of faults in Estafeta and Silabo, respectively, compared to 30% and 28% for Ochiai, and 12% for both products using Jaccard.

2. Test Suite Reduction: The clustering and test case selection steps resulted in significant reductions in the test suite size, ranging from 40% to 60% across different versions of the software products. This reduction in test suite size can lead to substantial savings in execution time and computational resources during regression testing.

3. Precision, Recall, and F-measure: The proposed approach achieved favorable precision, recall, and F-measure values, indicating its ability to select relevant test cases and effectively detect faults. The precise values varied across different software versions and fault localization methods.

The improved fault localization capabilities demonstrated by the proposed approach can be attributed to the synergistic effects of test case clustering, purification, and spectrum-based fault localization techniques. By isolating individual assertions and removing unrelated statements, test case purification enhances the discrimination power of fault localization algorithms, enabling more accurate identification of fault locations.

REFERENCES

[1] . Garousi et al., “A Survey of Software Engineering Practices in Turkey,” J. Systems and Software, vol. 108, 2015, pp. 148–177.
[2] V. Garousi and J. Zhi, “A Survey of Software Testing Practices in Canada,” J. Systems and Software, vol. 86, no. 5, 2013, pp. 1354–1376.
[3] M. Grindal, J. Offutt, and J. Mellin, “On the Testing Maturity of Software Producing Organizations,” Proc. Testing: Academia & Industry Conf.—Practice and Research Techniques, 2006, pp. 171–180.
[4] V. Garousi, M. Felderer, and M.V. Mäntylä, “The Need for Multivocal Literature Reviews in Software Engineering: Complementing Systematic Literature Reviews with Gray Literature,” Proc. 2016 Int’l Conf. Evaluation and Assessment in Software Eng. (EASE 16), 2016, pp. 171–176
[5] T. Hall et al., “What Do We Know about Developer Motivation?,” IEEE Software, vol. 25, no. 4, 2008, pp. 92–94. 10. W. Afzal et al., “Software Test Process Improvement Approaches: A Systematic Literature Review and an Industrial Case Study,” J. Systems and Software, Jan. 2016, pp. 1–33.