

Bachelor Thesis Topic

Abstract Syntax Tree (AST) Based Fault Localization

Motivation and Background

Spectrum Based Fault Localization [1] (SBFL) techniques have been developed to aide developers with locating faulty program elements by computing a ranking of all (executable) program elements, in which elements with higher fault suspiciousness are ranked accordingly. In recent times, SBFL techniques have also been used in various automated program repair tools [2], as they provide reasonable results at a negligible cost. Depending on the abstraction level, usual program elements are statements, methods or whole classes. These abstraction levels may not be optimal, though. Especially for automated repair tools - but also for human developers, it may make sense to use (a viable subset of) the nodes of the AST, instead.

Goals

The goal of this project is to develop, implement and evaluate (based on a benchmark) different SBFL techniques that work on AST-level. A main goal is the development of a hierarchical approach that combines the suspiciousness scores of a parent and its child nodes in a suitable way.

Description of the Task

The student has to either modify coverage-tools (e.g. Cobertura/JaCoCo) to work on AST-level or find ways to map statement-level coverage data to the AST nodes (preferably the first one, possibly both). A general framework (Java) to generate SBFL rankings with access to a popular benchmark (Defects4J) is available and should be used by the student. The framework also includes a basic implementation of hierarchical spectra elements, which may be utilized, if applicable.

Further information on the task and the underlying techniques will be given personally on interest.

Research Type

Theoretical Aspects: *****

Industrial Relevance: *****

Implementation: *****

Prerequisite

The student should be enrolled in the bachelor of software engineering or bachelor of computer science program, and has completed the required course modules to start a bachelor thesis.

Skills required

Programming skills in Java, Understanding of, or willingness to learn, the architectural and statistical foundations needed for the project.

References

[1] J. A. Jones, M. J. Harrold, and J. Stasko, "Visualization of test information to assist fault localization," in Proceedings of the 24th International Conference on Software Engineering, ser. ICSE '02. ACM, 2002, pp. 467–477.

[2] T. Durieux, M. Martinez, M. Monperrus, R. Sommerard, and J. Xuan, "Automatic repair of real bugs: An experience report on the defects4j dataset," CoRR, vol. abs/1505.07002, 2015.

Contacts

Simon Heiden, Humboldt-Universität zu Berlin, Institut für Informatik, Lehrstuhl Software Engineering, Unter den Linden 6, 10099 Berlin, Germany

Application

Please contact during office hours or write an email with the title: "AST-SBFL" to se-career@informatik.hu-berlin.de