Bachelor/Master Thesis Topic

Improving SBFL with Machine-Learning II (Algorithm Tuning)

Motivation and Background

Given a faulty program, the localization of occurring faults can be very difficult and may take a long time, which, in turn, leads to higher development costs and often to frustration for the developer in charge. Automated fault localization techniques as, e.g., Spectrum Based Fault Localization [1] (SBFL), have been developed to aide developers with this task by pointing them to program elements with a supposedly high fault probability (suspiciousness). 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. To rank the program elements, SBFL techniques only require the execution of a test suite to generate a program spectrum, which is a matrix in which each cell contains simple execution information (i.e., whether the program element was/was not executed by a specific test). This spectrum is then used to generate a ranking of program elements, often based on a similarity coefficient as, e.g., the Jaccard index.

Goals

The student uses a special machine-learning technique to improve the results of SBFL rankings, which, among others, needs parameter tuning to be successfully applicable. The goal of this project is to evaluate the approach (based on a benchmark) using different parameters and, desirably, develop own approaches/modifications and implement/evaluate these.

Description of the Task

A detailed description of 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/master of software engineering or bachelor/master of computer science program, and has completed the required course modules to start a bachelor/master thesis.

Skills required

Programming skills in (preferably) Java and C/C++, 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

Lars Grunske/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: “SBFL-ML-II” to se-career@informatik.hu-berlin.de