
Bachelor/Master Thesis Topic

Program Repair via “Naturalness” of Code

Motivation and Background

Bug-fixing generally is a difficult and time-consuming task. Therefore, tools are being developed which help the programmer to find and repair erroneous code or even have the purpose of *automatically* fixing detected bugs. A recent finding about a certain “naturalness” [1] of *correct* source code compared to erroneous code points to the examination of techniques that seize on this idea in order to improve current *automatic program repair* tools. *GenProg* [2], for example, is a tool for automatically repairing C programs, using a genetic algorithm to generate program patches through simple forms of mutations.

Goals

Examine and explore the possibilities of applying the concept of “naturalness” of code to the algorithms used in *GenProg* (and possibly other tools like, e.g., *PAR* [3]) and statistically evaluate the differences in performance.

Description of the Task

The specific tasks are:

- Understand *GenProg* (and resp. other tools) and find sensible ways for improvement based on the idea of “naturalness” of code depicted above.
- Create the respective implementation(s) based on your findings.
- Perform experimental evaluation and comparison of the original tools and your implementations on a suitable benchmark.

Research Type

Theoretical Aspects:

Industrial Relevance:

Implementation

Prerequisite

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

Skills required

Programming skills in C/C++ (and possibly Java). Understanding of, or willingness to learn, the software engineering methods and the statistical techniques needed for the project.

References

[1] Ray, B. et al. (2015). On the “Naturalness” of Buggy Code. arXiv Preprint.

[2] Le Goues, C., Nguyen, T., Forrest, S., & Weimer, W. (2012). *GenProg*: A generic method for automatic software repair. *IEEE Transactions on Software Engineering*, 38(1), 54–72.

[3] Kim, D., Nam, J., Song, J., & Kim, S. (2013). Automatic patch generation learned from human-written patches. *Proceedings - International Conference on Software Engineering*, 1(c), 802–811.

Contacts

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

Application

Please contact me during my office hours or send me an email with the title: “[ThesisProject]-PRvNoC” to se-career@informatik.hu-berlin.de