RUHR
UNIVERSITÄT
BOCHUM RUB

Bachelor Topic

Benchmark Curation for APR in Software Testing Education

Motivation and Background:
Software testing is usually part of the Software Engineering (SE) curriculum of Computer Science (CS) study programs. However, software testing education suffers from issues similar to programming education: the rising number of students and the need to guide them properly. This thesis project aims at curating a benchmark for Automated Program Repair (APR) [1] in software testing education. Such benchmark will support the development of automated techniques, in particular APR, in the context of intelligent tutoring. For example, LLM-based techniques for test code improvement [2-4] could support students in building test suites.

Student Task and Responsibilities:

- Make yourself familiar with the state-of-the-art in automated test code improvement, i.e., conduct a literature review on these topics.
- Systematically explore available education repositories for software testing courses and their exercises.
- Conduct mining campaigns on open-source projects to collect improvements of test code.
- Based on your findings, design and develop a benchmark framework, e.g., inspired by the Defects4J (https://github.com/rjust/defects4j).
- Apply an existing test improvement technique on your benchmark to showcase its effectiveness.
- Analyze the results and document your findings.

Deliverables:

- Benchmark for APR in Software Testing education
- Evaluation artifacts (dataset, tools, etc.)
- Documented findings of the conducted experiments

Pre-Requisites: (Programming Languages, OS, Skills, Papers, etc)
Strong knowledge in Java and unit testing with the JUnit is helpful for this project.

[1] C. Le Goues, M. Pradel, A. Roychoudhury and S. Chandra, "Automatic Program Repair," in *IEEE Software*, vol. 38, no. 4, pp. 22-27, July-Aug. 2021. https://doi.org/10.1109/MS.2021.3072577

[2] S. Fatima, H. Hemmati, and L. Briand, "FlakyFix: Using Large Language Models for Predicting Flaky Test Fix Categories and Test Code Repair," Jan. 29, 2024, http://arxiv.org/abs/2307.00012

[3] S. Gu, C. Fang, Q. Zhang, F. Tian, and Z. Chen, "TestART: Improving LLM-based Unit Test via Co-evolution of Automated Generation and Repair Iteration," Aug. 07, 2024, http://arxiv.org/abs/2408.03095

[4] N. Alshahwan *et al.*, "Automated Unit Test Improvement using Large Language Models at Meta," Feb. 14, 2024, http://arxiv.org/abs/2402.09171.

Contacts
Prof. Dr. Yannic Noller (`sq-office@rub.de`)
Software Quality group, Faculty of Computer Science, Ruhr University of Bochum