Bachelor Thesis Topic

Hypothesis-Based Debugging with LLMs

Motivation and Background

Avicenna [1] has proven the effectiveness of a hypothesis-driven approach for diagnosing failure-inducing inputs. By iteratively refining hypotheses about the conditions leading to a bug, Avicenna systematically pinpoints the input properties responsible for program failures. Concretely, it employs a learner and generator that work in tandem to refine or refute hypotheses, ultimately yielding both general and precise explanations of *why* a program fails. Building on these ideas, our new **DBG framework** generalizes the hypothesis-driven paradigm by decoupling the learner from the input generator, thereby enabling flexible and reusable debugging workflows. Meanwhile, large language models (LLMs) offer the potential to automatically synthesize, interpret, and refine hypotheses based on diverse information sources (e.g., input specifications or structured inputs). Integrating LLMs as learners within DBG can thus further automate the diagnosis process, uncover more intricate failure patterns, and reduce the need for manual intervention.

Goals

The goal of this thesis is to:

1. **LLM-Based Learner Integration**. Implement a plug-and-play Learner component that leverages an LLM to generate or refine debugging hypotheses.
2. **Automatic Diagnosis Generation.** Use the LLM to propose explanatory constraints or diagnoses for the failure-inducing inputs.
3. **Iterative Hypothesis Improvement.** Employ the DBG feedback loop so that newly generated tests and LLM suggestions inform subsequent refinements of the learner's model.
4. **Evaluation of Accuracy.** Assess how well the LLM-based diagnoses align with ground-truth bugs in real or synthetic systems.

Research Type

Theoretical Aspects:	*****
Industrial Relevance:	*****
Implementation	*******

Prerequisite

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

Skills required

We are seeking a candidate with **strong Python programming skills** and practical knowledge of large language models (LLMs), including hands-on experience. Completed courses in software engineering-II, compiler construction or machine learning research is a plus, as is the ability to communicate results effectively.

Contacts

Martin Eberlein
martin.eberlein@hu-berlin.de
Software Engineering, Institut für Informatik, Humboldt-Universität zu Berlin

References

[1]: Martin Eberlein, Marius Smytzek, Dominic Steinhöfel, Lars Grunske, and Andreas Zeller. 2023. Semantic Debugging. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2023).