

---

# Bachelor/Master Thesis Topic

## Automated Test Case Generation with Symbolic Execution for Regression Testing

### Motivation and Background

Software changes, typically referred to as patches, are needed during the software maintenance in order to incrementally improve the software qualities and to satisfy the user's requirements. Ideally, the patches should be comprehensively tested. Palikareva et al. [1] introduced a technique based on dynamic symbolic execution that can generate test inputs that cover the new program behaviors introduced by a patch. The Java verification tool Java PathFinder (JPF) [2] can be used to automatically generate test cases.

### Goals

This project should implement the idea of Palikareva et al. for Java applications by extending the symbolic execution extension of JPF, namely Symbolic PathFinder (SPF) [3]. Additionally, the approach should be evaluated on open source Java applications.

### Description of the Task

The specific tasks are:

- Understand the approach by Palikareva et al. and get familiar with SPF.
- Design an extension for SPF and plan the implementation.
- Implement the procedure as an extension of SPF.
- Perform experimental evaluation and comparison of the implemented idea on set of established benchmarks.

### Research Type

Theoretical Aspects: \*\*\*\*\*

Industrial Relevance: \*\*\*\*\*

Implementation: \*\*\*\*\*

### Prerequisite

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

### Skills required

Programming skills in Java or C++, understanding of, or willingness to learn, the software engineering methods like symbolic execution and patch testing needed for the project.

### Contacts

Yannic Noller (noller@informatik.hu-berlin.de)

Software Engineering Group, Institut für Informatik, Humboldt-Universität zu Berlin

### References

[1] Palikareva, H.; Kuchta, T. & Cadar, C. Shadow of a Doubt: Testing for Divergences Between Software Versions. Proceedings of the 38th International Conference on Software Engineering, ACM, 2016, pages 1181-1192.

[2] Visser, W.; Pasareanu, C. S. & Khurshid, S. Test Input Generation with Java PathFinder. SIGSOFT Softw. Eng. Notes, ACM, 2004, 29, pages 97-107.

[3] Pasareanu, C. S. & Rungta N. Symbolic PathFinder: Symbolic Execution of Java Bytecode. Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, ACM, 2010, pages 179-180.