Improving of Automatic Software Repair with Probabilistic Symbolic Execution

Motivation and Background
Automatic software repair aims to identify potential errors automatically and to propose code patches that fix these bugs. The state-of-the-art techniques for automatic software repair are Nopol [1] and GenProg [2], which show promising results. Nevertheless, according to the study by Qi et al. [3] these and other tools are not able to produce effective bug fixing solutions so far. Probabilistic symbolic execution (PSE) [4] is a static analysis technique aiming at quantifying the probability of a target event occurring during a program execution. Filieri et al. [5] proposed the characterization of code changes by calculating the probability of the repaired code to terminate the program successfully.

Goals
This project will investigate automatic software repair techniques and try to improve them by leveraging PSE.

Description of the Task
The specific tasks are:
- Survey the state of the art in automatic software repair and probabilistic symbolic execution
- Identify automatic software repair approaches which could be improved with PSE
- Implement the procedure for a selected automatic software repair approach
- 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 master of computer science program, and has completed the required course modules to start a master thesis.

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

Contacts
Yannic Noller
noller@informatik.hu-berlin.de
Humboldt-Universität zu Berlin, Institut für Informatik, Lehrstuhl Software Engineering, Unter den Linden 6, 10099 Berlin, Germany

References