Bachelor/Master Thesis Topic

Generator-based Fuzzing with Input Features

Motivation and Background

Fuzzing for fuzz testing [1] is an established technique that aims to discover unexpected program behavior (e.g., bugs, security vulnerabilities, or crashes) by feeding automatically generated data into a program under test. Generator-based fuzzing tools like Zest [2] rely on domain-specific input generators to produce syntactically valid inputs (e.g., XML or JavaScript). These input generators produce test inputs based on a sequence of different *random choices* that determine the syntactic structure and semantic elements of the input. Currently, there is no straightforward way to guide these generators towards producing inputs with particular *input features*, e.g., JavaScript programs containing nested loops or strings with a specific length. Such guidance would enable different applications, such as targeted test generation or explainable debugging based on input features [3].

Goals

The goal of this thesis is to extend Zest to enable input generation based on (predetermined or user-defined) input features and evaluate the implemented approach.

Description of the Task

The specific tasks are:

- Understand the overall approach of Zest and become familiar with the implementation
- Design a set of input features based on the existing input generators of Zest
- Extend Zest in a way that the generators produce test inputs with the given features
- Evaluate the efficiency and effectivity of the implemented approach against baseline Zest

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.

Skills required

Programming skills in Java, understanding of, or willingness to learn, the software engineering methods (like fuzz testing) and tools (e.g., Zest) needed for the project.

Contacts

Hoang Lam Nguyen (nguyehoa@informatik.hu-berlin.de)
Software Engineering Group, Institut für Informatik, Humboldt-Universität zu Berlin

References

[1] Manès, Valentin Jean Marie, et al. "The art, science, and engineering of fuzzing: A survey." IEEE Transactions on Software Engineering (2019).
[2] Padhye, Rohan, et al. "Semantic fuzzing with zest." *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 2019.
[3] Kampmann, Alexander, et al. "When does my program do this? learning circumstances of software behavior." *Proceedings of the 28th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*. 2020.