

Property-Based Testing

Description

Property-based testing (PBT), popularized by Quickcheck [1], is a semi-automatic, random testing technique that relies on two main components: 1) general *properties* that the system under test (SUT) is expected to satisfy, and 2) random *input generators* that produce well-distributed test inputs. Traditional black-box PBT continuously sampled random test cases using the provided input generators and reported a bug whenever execution resulted in a property violation. However, random sampling of test inputs is inefficient since the specified properties may only be violated by rare corner-cases. As a result, recent advances in this field proposed techniques to automatically guide the input generation process, e.g., by leveraging code coverage feedback [3], or search-based meta-heuristics [2]. In this seminar topic, the goal is to examine recent advancements in the field of property-based testing. The student should compare and discuss the different approaches and be able to give insights into possible further research directions.

References

- [1] Koen Claessen and John Hughes. Quickcheck: a lightweight tool for random testing of haskell programs. *Acm sigplan notices*, 46(4):53–64, 2011.
- [2] Andreas Löscher and Konstantinos Sagonas. Targeted property-based testing. In *ISSTA*, pages 46–56, 2017.
- [3] Rohan Padhye, Caroline Lemieux, Koushik Sen, Mike Papadakis, and Yves Le Traon. Semantic fuzzing with zest. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 329–340, 2019.

Contacts

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