

Generating Test Cases

Dependable Systems - Dirk Nowotka

Project Description

Safety and correctness of computer systems is of vital importance in a society depending heavily on computer systems. No matter how well we design our systems, errors will sneak into the final product. Errors that may have catastrophic outcomes.

The usual way of validating software systems behave correctly is using testing, where the system is exposed to an input and the systems output compared to an expected output. Since we cannot explore all possible inputs, the input space is partitioned into subsets that are expected to behave equivalently, and a single case from each is used for testing. In this project, a student can investigate techniques for finding these behavioural equivalent classes. Among the techniques worth investigating is

- Symbolic Execution,
- Fuzzing

Classic testing works very well on pure input-output systems but much harder to integrate into full reactive systems, that are expected to run constantly and react to stimuli from an environment. A natural extension of the above investigation is considering how to generalise the investigated techniques to reactive systems.

Applicable For

Bachelorstudents

Masterstudents

Keywords

Test Case Generation

Fuzzing

Symbolic Execution

Contact

Danny Bøgsted Poulsen

@ dbp@informatik.uni-kiel.de