

Software Engineering Seminar

Mutation-Based Fault Localization

Description

The search for an error that occurred in a program is often times more time-consuming than the actual repair. In automated fault localization, a tool attempts to locate the source of the bug automatically with or without the support of the developer. Different techniques were (and are still being) developed and steadily improved to accomplish this goal, including, for example, spectrum-based fault localization (SBFL) [1] and mutation-based fault localization (MUSE) [3]. In SBFL, program traces of failing and succeeding test cases are compared to identify program elements that are likely to be faulty, whereas in MUSE, the mutation of program statements allows for conclusions about the mutated statements being correct or faulty based on the results of executed test cases. Additionally, combinations of the two mentioned techniques are possible, as presented in [2].

The student should examine and discuss the current state of mutation-based fault localization techniques.

References

- [1] Rui Abreu, Peter Zoetewij, and Arjan J. C. van Gemund. On the accuracy of spectrum-based fault localization. In *Proceedings of the Testing: Academic and Industrial Conference Practice and Research Techniques - MUTATION*, TAICPART-MUTATION '07, pages 89–98, Washington, DC, USA, 2007. IEEE Computer Society.
- [2] Seokhyeon Moon, Yunho Kim, Moonzoo Kim, and Shin Yoo. Hybrid-muse: Mutating faulty programs for precise fault localization.
- [3] Seokhyeon Moon, Yunho Kim, Moonzoo Kim, and Shin Yoo. Ask the mutants: Mutating faulty programs for fault localization. In *Proceedings of the 2014 IEEE International Conference on Software Testing, Verification, and Validation*, ICST '14, pages 153–162, Washington, DC, USA, 2014. IEEE Computer Society.

Contacts

Simon Heiden (heiden@informatik.hu-berlin.de)
Software Engineering Group
Institut für Informatik
Humboldt-Universität zu Berlin