Software Engineering Seminar

Differential Program Analysis

Description

Differential program analysis can mean to detect paths in a program with different properties or to detect the semantic differences between multiple program versions / variants. For example *regression testing* is one instance of differential program analysis. The work by Winstead and Evans [4] describes some basic problems regarding differential program analysis. *Differential Symbolic Execution* by Person et al. [2] leverages symbolic execution and represents an important milestone in this research area. Recent work like *NEZHA* by Petsios et al. [3] and *Relational Symbolic Execution* by Farina et al. [1] show that the problems are still not solved and need further research.

The goal of this seminar topic is to collect the current research directions in differential program analysis. Therefore, it is is necessary to perform an initial literature analysis based on the provided publications. The student should examine and discuss the approaches given in the papers and compare them to each other and to similar existing techniques. Additionally, the student is asked to provide a critical discussion of the current research directions, which should also include an outlook for possible future work.

References

[1] Gian Pietro Farina, Stephen Chong, and Marco Gaboardi. Relational symbolic execution. *arXiv preprint arXiv:1711.08349*, 2018.

[2] Suzette Person, Matthew B. Dwyer, Sebastian Elbaum, and Corina S. Păsăreanu. Differential symbolic execution. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, SIGSOFT '08/FSE-16, pages 226–237, New York, NY, USA, 2008. ACM.

[3] T. Petsios, A. Tang, S. Stolfo, A. D. Keromytis, and S. Jana. Nezha: Efficient domain-independent differential testing. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 615–632, May 2017.

[4] Joel Winstead and David Evans. Towards differential program analysis. In *Workshop on Dynamic Analysis*. Citeseer, 2003.

Contacts

Yannic Noller (`noller@informatik.hu-berlin.de`)
Software Engineering Group
Institut für Informatik
Humboldt-Universität zu Berlin