Software Engineering Seminar (WiSe 2016/17)

State Space Sampling Strategies for Probabilistic Symbolic Execution

Description

Probabilistic symbolic execution (PSE) was introduced by Geldenhuys et al. [3] as an adaptation of the traditional symbolic execution [4] in order to calculate the probability of the specific target events in the code. Therefore, it collects constraints on the inputs that lead to the target events and analyzes them to quantify how likely it is for an input to satisfy the constraints. This initial work was limited to inputs that are uniformly distributed within their domains. The work by Filieri et al. [2] generalized this to arbitrary usage profiles and used this advantage to assess the reliability of software. So far, these techniques are limited to the constraints they can solve. In 2015, Borges et al. proposed their approach *Iterative Distribution-Aware Sampling for Probabilistic Symbolic Execution* [1] that can handle arbitrarily complex mathematical constraints and continuous input distributions.

The student is supposed to focus on the work by Borges et al. and investigate the state of the art.

Prerequisites

A basic knowledge of software verification techniques is preferable, as well as the mathematical basics in probability theory.

References

[1] Mateus Borges, Antonio Filieri, Marcelo d'Amorim, and Corina S. Păsăreanu. Iterative distribution-aware sampling for probabilistic symbolic execution. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2015, pages 866–877, New York, NY, USA, 2015. ACM.

[2] Antonio Filieri, Corina S. Pasareanu, and Willem Visser. Reliability analysis in symbolic pathfinder. In *35th International Conference on Software Engineering, ICSE '13, San Francisco, CA, USA, May 18-26, 2013*, pages 622–631, 2013.

[3] Jaco Geldenhuys, Matthew B. Dwyer, and Willem Visser. Probabilistic symbolic execution. In *Proceedings of the 2012 International Symposium on Software Testing and Analysis*, ISSTA 2012, pages 166–176, New York, NY, USA, 2012. ACM.

[4] James C. King. Symbolic execution and program testing. *Commun. ACM*, 19(7):385–394, July 1976.

Contacts

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