Software Engineering Seminar

Delta Debugging

Description

In *delta debugging*, one tries to isolate a minimal root cause of an occurring error by removing all elements that are not relevant to the failure. For example, JINSI [2] is a tool to capture and replay interactions between Java components and their environment. It is also able to apply delta debugging to automatically isolate a subset of interactions that is relevant for the failure. In a multi-threaded environment, DEJAVU [1] allows to record a thread schedule and replay it in a deterministic way. Then, comparing successful and unsuccessful schedules, it uses delta debugging to narrow down the exact location of the occurred error.

The goal of this topic is to examine the merits of delta debugging in general and the mechanics of the given tools in particular.

References

[1] Jong-Deok Choi and Andreas Zeller. Isolating failure-inducing thread schedules. In *Proceedings of the 2002 ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA '02, pages 210–220, New York, NY, USA, 2002. ACM.

[2] Alessandro Orso, Shrinivas Joshi, Martin Burger, and Andreas Zeller. Isolating relevant component interactions with jinsi. In *WODA '06: Proceedings of the 2006 international workshop on Dynamic systems analysis*, pages 3–10, New York, NY, USA, February 2006. ACM.

[3] A. Zeller and R. Hildebrandt. Simplifying and isolating failure-inducing input. *IEEE Transactions on Software Engineering*, 28(2):183–200, Feb 2002.

Contacts

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