Software Engineering Seminar

Program Slicing for Fault Localization

Description

In large software systems, keeping track of control and data flow is a very hard task. To this end, *program slicing* allows the developer to concentrate on the important parts of the program under consideration by essentially *hiding* other parts of the program from the developer that are not relevant to the problem at hand. That means: parts of a program that are not influenced by a chosen set of variables, or parts that do not, themselves, influence a specific variable or a set of variables are simply not shown to the user.

Various variants of program slicing (e.g. static, dynamic, ...) have been proposed so far. The aim of this topic is to shortly review the current state of the literature on program slicing, as well as to examine and discuss its usage, advantages and disadvantages in combination with debugging and fault localization.

References

[1] Mark Harman and Robert M. Hierons. An Overview of Program Slicing. *Software Focus*, 2(3):85–92, 2001.

[2] Susan Horwitz, Thomas Reps, and David Binkley. Interprocedural slicing using dependence graphs. *ACM Transactions on Programming Languages ans Systems*, 12(1):26–60, 1990.

[3] A Edwin Robert. Program Slicing Techniques And Its Applications. 2(3):50–64, 2011.

[4] Jifeng Xuan and Martin Monperrus. Test case purification for improving fault localization. *CoRR*, abs/1409.3176, 2014.

Contacts

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