



Software Engineering Seminar (SoSe 2016)

Program Slicing

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 of importance for 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, ...) were proposed so far. The aim of this topic is to review the current state of the literature on program slicing (starting for example from the referenced articles given below) and to compare the different existing approaches while highlighting their differences and commonalities. Another main goal is to evaluate the usefulness of the different approaches, especially considering "real-life" applications.

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 and Systems*, 12(1):26–60, 1990.
- [3] A Edwin Robert. Program Slicing Techniques And Its Applications. 2(3):50–64, 2011.

Contacts

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