

Detecting Roles of Variables in Intermediate Programming Languages: Enhancing the Approach and Evaluating its Performance in Comparison with Other Role Detectors

Background

The concept of roles of variables [1, 2] refers to the manner in which variables are used in computer programs. Each variable plays a particular role that is specific to the way it is used and its value is updated. For example, a variable that is used for storing a value for a short period of time has a *temporary* role, whereas variable that holds the most desirable value that is found so far plays a *most wanted holder* role.

Since roles of variables were introduced, there have been several attempts to detect them automatically, see for example [3] and [4]. In the former work, a set of different characteristics of variables of Java programs is extracted using dataflow analysis. Based on these, logical sentences are formed to determine the role of the variables. The latter work—which is only able to analyze Pascal programs—also computes some flow characteristics of variables using dataflow analysis, but applies machine learning techniques to detect roles. In a recently completed master's thesis [5], roles of variables are detected from JVM bytecode. This makes role detection language independent, and enables detection of roles of variables from all programs that compile to JVM bytecode. The main contribution of this work is that it extracts the JVM bytecode characteristics to detect roles of variables using the same methods previously applied in [3] and [4] for Java.

The main goals of the master's thesis

In addition to the theoretical part of the thesis, where a literature survey on the topic is carried out, the following tasks should be completed in the constructive part.

1. There are few characteristics that the prototype developed in [5] cannot compute. The prototype should be further developed to cover all the appropriate characteristics. It is also desirable if some key characteristics other than those used in [3] and [4] can be identified from bytecode to be used in detection of roles. Furthermore, detecting roles based on the computed characteristics is not fully implemented in the prototype. The prototype should be improved in this regard as well.
2. Building an annotated corpus of different roles of variables and assessing the performance of the method by carrying out comprehensive tests against the corpus.
3. Conducting an experiment using benchmarking to evaluate the performance of [3], [4] and [5] and compare their accuracy.

Successful completion of the thesis requires good knowledge of dataflow analysis as well as JVM bytecode.

The thesis will be supervised by Docent Ari Korhonen and instructed by M.Sc. Ahmad Taherkhani.

References

- [1] J. Sajaniemi. An empirical analysis of roles of variables in novice-level procedural programs. In Proceedings of IEEE 2002 Symposia on Human Centric Computing Languages and Environments, pages 37–39. IEEE Computer Society, 2002.
- [2] J. Sajaniemi. The Roles of Variables Home Page. http://cs.joensuu.fi/~saja/var_roles/, 2003
- [3] C. Bishop and C. G. Johnson. Assessing roles of variables by program analysis. In Proceedings of 5th Baltic Sea Conference on Computing Education Research, Koli Calling 2005, 2005.
- [4] P. Gerdt (2006) A System for the Automatic Detection of Variable Roles. Licentiate's Thesis, Department of Computer Science, University of Joensuu, Finland.
- [5] S. Lampinen (2010). Detecting Roles of Variables in an Intermediate Language. Master's Thesis, Faculty of Information and Natural Sciences, School of Science and Technology, Aalto University. *Copies available on request.*