Project Description: Static Type Analysis of Dynamically Typed Programming Language

Štěpán Šindelář

March 12, 2014

Introduction

There are several studies showing poor success rate of software projects, for example [1]. Not only are software projects delivered late or they fail completely before they can be even released, but often released software contains errors and security vulnerabilities. One way we can improve this is providing developers with tools that are capable of automatic analysis of the software they are developing for potential errors.

Project Aim

Three out of top ten programming languages in TIOBE index ([2]) fall into a category called dynamic languages. When using statically typed languages, as opposed to dynamic languages, the programmer must choose the data type for each variable and then stick to that type. So it is not possible to assign a floating point number into a variable that was chosen to be of integral numeric type. Dynamic programming languages allow dynamic type of variables, meaning that the type of the variable can change depending upon the type of the value we assign to it. This makes dynamic languages easier to use, but at the same time can lead to error prone code, which then leads to problems in later phases of the development and maintenance. The aim of the project is to provide automatic analysis of variables types and thus discover possible type related errors in the source code.

Research Problem

Because of their dynamic nature, dynamic languages are more difficult to analyse than statically typed languages, especially if we want the analysis to be reasonably fast so that it can be used in everyday development. There is an ongoing research of the static analysis methods for many different families of programming languages, including dynamic languages. The problem this project is trying to address is to adapt and apply those methods on a real world and widely used programming language PHP. Especially to automatically infer the type information from the source code in order to find possible type mismatch errors and inconsistencies with the explicit type documentation in the comments embedded in the source code.

This kind of analysis can provide a feedback to the developers and can help them to improve their source code quality as well as the correspondence between the source code and documentation comments.

Methodology

- Research into possible methods of static analysis of source code.
- Analysis of applicability of those methods to the PHP programming language.
- Adaptation and implementation of the selected method(s).
- Evaluation on real world open source projects.
- Final report including the research and analysis as well as documentation of the implementation and evaluation results.

Outcomes

Expected outcome of this project is a console application that can process PHP source code files (several at once), analyse them and print out possible errors it discovered. This application will form a basis for a plugin for Visual Studio IDE.

References

[1] K. Ellis, “The impact of business requirements on the success of technology projects,” *Benchmark, IAG Consulting*, 2008.

[2] “Tiobe index for march 2014.” `http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html`. Accessed: 2014-09-03.