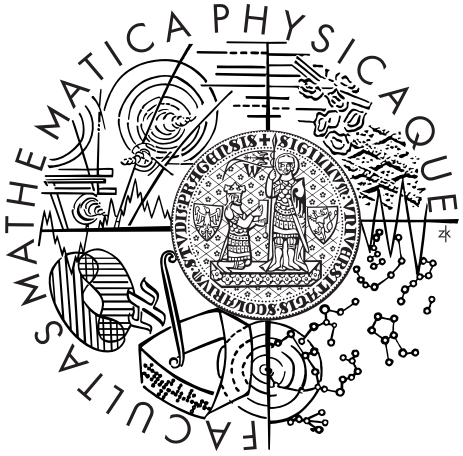
Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



First and last name of the author

Implementing control flow resolution in dynamic language

Department of Software Engineering of the Charles University in Prague

Supervisor of the master thesis: Filip Zavoral, Ph.D.

Study programme: programme

Specialization: specialization

Prague YEAR

Dedication.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In ..... date ..... signature of the author

Název práce: Název práce

Autor: Jméno a příjmení autora

Katedra: Název katedry či ústavu, kde byla práce oficiálně zadána

Vedoucí diplomové práce: Jméno a příjmení s tituly, pracoviště

Abstrakt:

Klíčová slova:

Title:

Author: Jméno a příjmení autora

Department: Název katedry či ústavu, kde byla práce oficiálně zadána

Supervisor: Jméno a příjmení s tituly, pracoviště

Abstract:

Keywords:

Contents

1. Introduction

1.1 Problem Description

Static type analysis of a dynamic language, benefits: compiler optimizations, IDE support - bug hunting analysis.

1.2 Thesis structure

2. Static Analysis

More detailed description of what static analysis is (as opposed to for example explicit model checking, verification, etc.).

2.1 Existing Approaches

List and short description of existing approaches to static analysis and related terminology (Data Flow, context-sensitive, path-sensitive, ...).

2.2 Data Flow Analysis

Detailed description of Data Flow analysis or just reference to the literature in the previous chapter and this chapter would be left out.

2.3 The PHP Programming Language

Note: maybe a new chapter?

What makes static analysis in PHP harder than of statically typed language

what are the caveats of PHP we had to deal with

PHPDoc annotations and their significance for type analysis (especially for the bug hunting)

3. Existing Software

There are lots of different incomplete solutions to slightly different but relevant problems: e.g. Weverca focuses on security vulnerabilities. How many other implementations should I mention and in what detail?

3.1 HipHop

3.2 Weverca

4. Implementation

specific constraints:

must use Phalanger front end and its AST data structures,

should be ready to be connected in between the Phalanger front end and back end as a middle end,

should be ready to be used continuously: interactively re-analyze updated code when used inside IDE,

overall description:

Control Flow graph construction

discussion of the choice of intermediate representation (or in fact, why we do not use any intermediate representation).

generic framework for Data Flow analyses

type analysis

5. Results

Analyses of well known OSS PHP projects.

6. Conslusion

6.1 Future Work

Branched Data Flow analysis, integer interval analysis, integration with the compiler back-end.

Bibliography

List of Tables

List of Abbreviations

Attachments