A Relational Static Semantics for Call Graph Construction

Xilong Zhuo, Chenyi Zhang

Abstract

The problem of resolving virtual method and interface calls in object-oriented languages has been a long standing challenge to the program analysis community. The complexities are due to various reasons, such as increased levels of class inheritance and polymorphism in large programs. In this paper, we propose a new approach called type flow analysis that represent propagation of type information between program variables by a group of relations without the help of a heap abstraction. We prove that regarding the precision on reachability of class information to a variable, our method produces results equivalent to that one can derive from a points-to analysis. Moreover, in practice, our method consumes lower time and space usage, as supported by the experimental results.

Keywords: type analysis, static analysis, method resolving

1. Introduction

For object-oriented programming languages, virtual methods (or functions) are those declared in a base class but are meant to be overridden in different child classes. Statically determine a set of methods that may be invoked at a call site is important to program optimization...

☆describe that this paper had been published on ICFEM
*Corresponding author
Email address: chenyi_zhang@jnu.edu.cn (Chenyi Zhang)

2. Type Flow Analysis

We define a core calculus consisting of most of the key object-oriented language features...

2.1. $x \sqsubseteq y$

2.2. $x \xrightarrow{f} y$

3. Implementation

The analysis algorithm is written in Java, and is implemented in the Soot framework...

3.1. Static Analysis Tool

3.2. Dynamic Profiler

4. Evaluation

We evaluate our approach by measuring its performance on 13 benchmark programs...

4.1. Efficiency

We executed each benchmark program 10 times with the CHA, PTA and TFA algorithms. We calculated the average time consumption...

4.2. Precision

4.2.1. Reflection Call

4.2.2. JNI Call

4.2.3. Library

4.2.4. Array Approximation

5. Related Work

There are not many works focusing on general purpose call graph construction algorithms, and we give a brief review of these works first.

6. Conclusion

In this paper we have proposed Type Flow Analysis (TFA), an algorithm that constructs call graph edges for Object-Oriented programming languages.

References