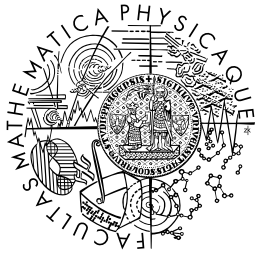

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Tomáš Husák

Improving Type Inference in the C# Language

Department of Distributed and Dependable Systems

Supervisor of the master thesis: Mgr. Pavel Ježek, Ph.D.
Study programme: Computer Science
Study branch: Software Systems

Prague 2024

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

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

In ..... date ....

Author's signature

TODO: Dedication.

Title: Improving Type Inference in the C# Language

Author: Tomáš Husák

Department: Department of Distributed and Dependable Systems

Supervisor: Mgr. Pavel Ježek, Ph.D., Department of Distributed and Dependable Systems

Abstract:

TODO: Abstract.

Keywords: Type Inference C# Roslyn

Contents

Introduction

Note: Describe what is type inference.

Statically typed languages have many advantages like revealing bugs in compilation time or performance. To achieve these benefits, the languages demand type annotations from a programmer. These type annotations define an actual type of variable during runtime protecting to make operations on incompatible data. Because code usually contains a lot of variables whose type has to be known during compilation time, type inference was introduced to eliminate type annotations that can be deduced from a context. Type inference tries to deduce a type of a variable using a context, where the variable is used. That's used operations and interactions with other parts of the code.

TODO: Describe type inference in C#.

TODO: Compare it with type inference in Rust or Haskell as an example of Hindel-Millner type inference.

TODO: Describe Roslyn.

TODO: Mention CSharplang repo, community, and describe a process of accepting lang changes.

TODO: Give an overview of chapters.

1. Related work

TODO: Describe type inference in C#.

TODO: Describe Hindel-Millner type inference.

TODO: Describe type inference in Rust or Haskell (Mention related papers about type inference.

TODO: Describe Roslyn(Focused on the part where the type inference is done).

TODO: Mention related Github issues and csharplang repo.

2. Problem analysis

TODO: Describe outputs of this work(Proposal and prototype). Why these outputs are necessary.

TODO: Describe the set of related issues.

TODO: Describe the selection and scope of this work based on the issues and other factors.

TODO: Describe problems of C# lang architecture which prohibits some advanced aspects of type inference.

TODO: Describe goals of the work and explain benefits of proposed changes.

3. Solution

TODO: Describe process of making proposal and the prototype.

TODO: Describe partial method type inference.

TODO: Describe constructor type inference.

TODO: Describe generic adjusted algorithm for type inference.

TODO: Describe decisions of proposed change design.

TODO: Describe changed parts of C# standard.

4. Evaluation

TODO: Describe achieved type inference. Mention interesting capabilities.

TODO: Note about the performance.

TODO: Links to csharplang discussions.

5. Future improvements

TODO: Mention next steps which can be done.

TODO: Discuss which steps would not be the right way(used observed difficulties).

Conclusion

TODO: Describe issue selection.

TODO: Describe proposed changes in the lang.

TODO: Describe the prototype and proposal.

TODO: Mention csharplang discussions.

TODO: Mention observed future improvements.

Bibliography

List of Figures

List of Tables

List of Abbreviations

A. Attachments