

DD2488 - COMPILER CONSTRUCTION - PROJECT REPORT

SAMUEL WEJEUS EMIL ARFVIDSSON

1. INTRODUCTION

A compiler is a complicated piece of machinery, a computer program, designed to transform high level source code written in some programming language into some lower level language understandable, and runnable on various target architectures i.e. different types of processors.

This report presents the construction, and inner workings, of a compiler for the high level programming language *MiniJava* (which is a subset of the Java programming language) created as a project in the course *DD2488 - Compiler Construction*[1].

The report will present the various tools used to facilitate the construction of the compiler and explain how the different stages of compilation have been implemented. The intended reader is a person who has rudimentary understanding of the workings of a compiler and our goal is to give the reader insight and a basic understanding of the inner workings and design choices we made for our implementation of MiniJava.

2. OVERVIEW

Topics: stages of compilation, introduction to visitors

3. LEXING AND PARSING

Topics: JFlex, JavaCUP

4. SCOPE AND TYPE CHECKING

Topics: scope and type

5. TRANSLATION TO INTERMEDIATE CODE

Topics: frames, IR tree, canonicalization

6. INSTRUCTION SELECTION

Topics: maximal munch, architecture specifics of frames

7. REGISTER ALLOCATION

Topics: liveness analysis, graph coloring

8. CONCLUSIONS

REFERENCES

- [1] Andrew W. Appel, *Modern Compiler Implementation in Java, 2nd edition*.
- [2] G.J.O. Jameson, *Finding Carmichael numbers*. www.maths.lancs.ac.uk/~jameson/carfind.pdf