4TB3 Final Project - Executive Summary

Justin Staples, Justin Licari

March 1, 2018

1 Project Description

As we have studied in class, a compiler is a tool that generally translates computer code written in a higher level language to a lower level target language. Ultimately, for an executable program to run on a computer, high level code needs to be translated to a machine (assembly) language that can be understood by the computer hardware.

Each type of computer processor (CPU) uses a different Instruction Set Architecture (ISA). An ISA is an agreement about how the software will communicate with the processor. The ISA describes all of the different instructions that the CPU is capable of understanding and processing. Therefore, the target language for the compiler must be an assembly langauge of instructions that are understood by that processor.

Currently, one of the most common types of processors used for mobile devices is the ARM (Advanced RISC Machine) processor, which uses a reduced ISA. As well, one of the most popular types of processors used for personal computers is the Intel processor (x86 architecture), which uses a complex ISA.

The aim of our project is to create a tool that will be able to generate assembly code for a particular hardware platform (ARM, Intel, etc.), given source code in a high level language. More specifically, our group is aiming to transform Java code into ARM assembly code.

2 Implementation Details

The main feature of our implementation will be a program that accepts a potential Java program as input. This Java code will be sent through lexical and syntactic analysis. This will be done by first validating the source code against a grammar that describes the Java programming language (or at least an appreciable subset of it). Our current objective is to write our own recursive descent parser for such a language. Once a valid Java program is detected, our program will begin a code generation phase where the Java code is mapped to the appropriate ARM instructions. The resulting ARM instructions will then be written to an output file. In theory, this ARM code could then be run on a computer that has an ARM processor or that has access to an ARM virtual environment.

3 Motivation and Relevance

Most Java programmers will know that Java code is not typically compiled into an assembly language like ARM. The Java compiler actually translates the Java code into Java Bytecode, which is the instruction set understood by the Java Virtual Machine (JVM). Therefore, to run a Java program, you must have the JVM installed on your computer. The motivation behind our project is to offer a different way of compiling and running a Java program (using ARM assembly instead of Bytecode). Because Java is such a ubiquitous language, we think that this could be a useful application.

Of course, the topics that I have described above have significant relevance to the course material as well as the field of computing. This project involves creating a compiler that will perform lexical and syntactic analysis, which have been studied vastly in the lectures. As well, the project also focuses on generating code that is compatiable with ARM hardware. Considering that ARM processors represent such a large share of the mobile processor market, we consider this a very relevant and interesting endeavour.