Compilers Project Proposals
Dr. D.M. Akbar Hussain

Important: These proposals can serve just as a guide line text, it gives you a clear idea about what sort of work you will be doing in your projects. If things are still not clear please come and have a conversation with me or your supervisor.

Compiler for Java to C

Today Java has become a standard for programming high level embedded systems like cellular phones and PDA's. Therefore a lot of software has been developed for the Java platform. This has some clear advantages since the Java platform is independent of the system architecture and can therefore run on many different targets without significant changes to the code.

One disadvantages of this is the fact that the Java Virtual Machine (JVM) must be ported to each system. Programs written in Java are only interpreted by the JVM which decreases the speed and efficiency of the program. Most low level embedded systems like microprocessors and digital signal processors are programmed using assembly or C code. This is preferable since the success of many applications of these systems requires a runtime specified by some real-time demands. Since many programs have been developed for the Java it would be nice if these programs could be ported to the lower level embedded systems. This will cause the number of available programs written for the lower level embedded systems to be raised. Furthermore Java programmers are able to program these systems without learning a new programming language. If an efficient compiler for this purpose is developed it would also mean greater speed and efficiency on runtime for the programs.

Compiler for Java to C++

The goal of this project is to apply the theory from the subjects of "Language and Compilers" and "Programming Paradigms". The students need to show practical approach to construct a compiler or an interpreter and to demonstrate that they understand the theory of the two subjects. Furthermore, this project is an opportunity for the group to gain practical experience as opposed to only possessing theoretical knowledge with the theory involved.

As a way of achieving this goal create a compiler which is able to translate a source code written in a subset of Java to C++. The compiler itself will be written in Java.

The problem of this project is depicted in the following T-diagram:

```
Java ---------------------------- C++
                |
                |
                |
                |
              Java
```

Compiler for subset of language: micro Java, micro C, micro Pascal

The aim of this project is to build a compiler that handles a subset of java language, C language or Pascal Language. The compiler will translate micro Java into an assembly code for Java Virtual machine. The compiler will translate micro C or micro Pascal into assembly code that could be assembled by a classic micro-assembler into an executable file.

Compiler for Haskell lambda language

Haskell is a general purpose, purely functional programming language. Haskell compilers are freely available for almost any computer. Haskell is a computer programming language. In particular, it is a poly-morphicy typed, lazy, purely functional language, quite different from most other programming languages. The language is named for Haskell Brooks Curry, whose work in mathematical logic serves as a foundation for functional languages. Haskell is based on lambda calculus. Writing large software systems is not only difficult but expensive as well. Maintaining those systems is even more difficult and expensive. Functional programming languages, such as Haskell, can make it easier and cheaper.

A compiler or interpreter written in C or may be even in Haskell will be a useful development tool. As practice application, a merge-sort algorithm (or another classic algorithm) is suggested.

A Java to C translator (C to Java)

A translator that translate Java class file into C source code (or C++) is suggested. This allows the construction of directly executable programs that avoid the overhead of interpretation. The translator deals with stand-alone applications, not applets.

The translator reads Java class files, the object code files produced by the Java compiler for an abstract "Java virtual machine" (JVM). The generated code is designed not for human readability (although that's not impossible) but rather for easy optimization by a reasonably clever C compiler.

Translation of a class file yields a .c file and a .h file. Each .c file references its own .h file and also the .h files of other classes referenced by the class file.

An alternative to this project is a translator that does not convert Java class files, but instead processes Java source code directly. It converts whole programs at a time, which can cause the conversion process to be quite lengthy, but has the advantage that a number of optimizations can be performed that would be very difficult to achieve with other techniques.

Environment and compiler for GCL - guarded command language

The programming Language GCL is a Pascal-like language with guarded commands and parallel assignment. The GCL was developed from a language originally designed by E. Dijkstra. Given an EBNF grammar for GCL, build a visual environment compiler with small debugging facilities.

Fuzzy Parser Java (or C++)

A Fuzzy parser is a form of syntax analyzer that performs analysis on selected portions of its inputs rather than performing a detailed analysis for all the source code. Fuzzy parsers are components of software tools and have a notable presence in development environments that extract information from source text. A tool that implements the fuzzy parser in object-oriented language is suggested.

Schematic tool for context-free grammars verification

A schematic tool for the analyses of the correctness of a grammar is suggested. Given a grammar in a predefined text form, the tool must analyse the production rules in order to identify the grammar, verify the ambiguity and a small suggestion is generated to eliminate the ambiguity made by an "intelligent" module of the tool.

A PHP - hypertext pre-processor

PHP is what is known as a server-side scripting language. When a user looks at a WWW page that is a PHP page, the server gets a PHP interpreter to examine the page. The PHP interpreter will generate some HTML, which is then shipped by the WWW server and is interpreted by the user's WWW browser. PHP's manual says that PHP's syntax is borrowed primarily from C. Java and Perl have also influenced the syntax. One can develop a PHP hypertext pre-processor.

WebSQL compiler for a subset of Data Mining SQL

A mini grammar for Data Mining (using a similar semantic as in SQL) is suggested. The compiler will translate the Data Mining SQL queries into SQL syntax language usable for relational databases (even a direct implementation is possible). WEBminer language is also taken into account and could be anther interesting project

Java Subset Compiler

The aim of this project is to build a compiler that handles a subset of the programming language Java. The compiler will translate the Java subset into an assembly code for the Java Virtual Machine (JVM). The development of the compiler can be done either from scratch by writing everything or using generators. The project objective is to use and achieve knowledge of the theory of compiler construction in developing a compiler, which translates a subset of the Java programming language.