Decaf Project Report

Compilers CSE 419
Hemanth Kumar Veeranki
201401145

Project Outline

The aim of this project is to build a compiler for a simple object-oriented programming language called Decaf. Decaf is a strongly-typed, object-oriented language with support for inheritance and encapsulation. By design, it has many similarities with C/C++/Java. The compiler was iteratively developed in 4 stages

- **Phase1**
    - Deals with writing a lexical analyser for the given language. Flex was used to give the specifications to automatically generate a lexer which generates tokens.
    - **Challenges Faced:**
        - Regular expressions knowledge and comping up with correct regular expressions
        - Learning the syntax for lex
    - This phase helped us to learn how the concepts of Finite automatons and regular languages were used in compiler construction

- **Phase 2**
    - Deals with writing a syntax analyser for the given grammar. Bison was used to specify the CFG and automatically generate the parser.
    - **Challenges Faced:**
        - Coming up with given cfg in such a manner that they don't raise Shift-reduce conflict errors.
        - Understanding the syntax for specification in bison

- **Phase 3**
    - Deals with developing an IR from the given tokens. We used the Abstract syntax tree approach to build an AST Parse tree with hierarchy of various nodes.
    - **Challenges Faced:**
        - Hierarchy of classes and inheritance patterns need to be decided which is tougher.
        - Coming up with proper OOP techniques to do that is also one more challenge.
        - OOP coding in C++ is a little tougher than doing it in Java Python. So that is one more challenge faced.
        - Also the pattern in which we have to display AST have to be displayed.

- **Phase 4**
    - This phase required converting AST IR to LLVM IR which is a form of three address instruction or more specifically Static Single Assignment.
    - **Challenges Faced:**
        - Learning the api for LLVM was a very difficult task. The tutorial in the llvm website has many new things in it and learning all of them was a big task.
        - Some features were not covered in the tutorial present there. For instance the concept of arrays and accessing the elements in array dealing with statements like break, continue are not there. Learning them is yet another difficult task by looking at the Documentation