Compiler Design

Milestone 2

March 15, 2019

Group Members:

1. Nidanshu Arora (160443) - nidanshu@iitk.ac.in
2. Niket Agrawal (160446) - niketagl@iitk.ac.in
3. Siddharth Chinmay (160685) - schinmay@iitk.ac.in

Language :

1. The Source Language **S** for our compiler is **C**+.
2. The Implementation Language **I** is **C**++ and the Target Language is **x86-64**.

Project Desciption :

In this milestone, we have extended the project to perform semantic analysis. The goal was to convert the program into an Intermediate Representation to be used by later stages (Final code generation, optimization, etc.).We have used actions in the lexer/parser to achieve the desired outcome.

1. We have referred to this lexer - ANSI C grammar, Lex specification
2. We have modified this grammar - ANSI C Yacc grammar
3. We have added actions corresponding to each rule in our grammar for type checking.
4. We have created the following two-level symbol table structure.

   (a) A global symbol table (GST) that maps function names to their local symbol tables.

   (b) A local symbol table for structures that contains the relevant information for it.

   Our symbol table is extensible since we might discover the need to store new information as the project progresses.
5. The output for good programs consists of:

   (a) A dump of the symbol table as a CSV file, and

   (b) A dump of the 3AC of the functions in text format.

   For bad programs, the output also mentions the error that caused the program to be rejected with the line number.

Steps to build and run the Project:

Steps to build and run:

```
$ cd <project-top>
$ make
$ ./compile <input file path>
```

Example :

```
./compile tests/input/input1.c
```