CS323 Documentation

Luc Dang

ID: 888858644

1. **Problem Statement**

   The assignment is to build a semantic analyzer and generate assembly code for each semantic action using a top down parser of our choice.

2. **How to use my program**

   From the terminal of a Unix operating system (preferably Ubuntu), run pre-compiled executable file "test" with following command:

   ./test.

   If there is any problem with "*test*" file, compile "*semantic_icg.cpp*" with following command, then execute again.

   clang++ -std=c++17 semantic_icg.cpp -o test

   The program will prompt for input file name for syntax analysis. You must include the file extension in the name in order for the file to be read because the program will not assume .txt for reading. The program will then prompt for the desired name of your output file which the analysis will be printed to. The program will then print instruction table and symbol table for the file.

3. **Design of my program**

   To add semantic analysis and intermediate code generation functionality, I continued with my previous table-driven predictive parser. Since it is more difficult to implement these functionalities with my parser than with a recursive parse, I made 2 separate stacks to keep track of the operations and operands. I followed the rules including the semantic operations given by Dr. Choi to implement the stacks properly and create an instruction table. For the symbol table, I just added the token whenever a Declaration statement was made.

4. **Any Limitations**

   My program isn't very memory efficient because of the wide use of data structures for each process.

5. **Any Shortcomings**

   - Program will sometimes get the operand (memory location) wrong for an unknown reason
   - Program does not fully implement if-while functionality