CS335A-Project

Milestone3

We have used python version 3.8.18 grammar for the implementation

1. **Our compiler includes the following python features:**
   - (a) Native data types like bool,int,string
   - (b) Assignment statements and expressions
   - (c) If else statements
   - (d) Functions
   - (e) Support for print statements

2. **Folder content:**
   milestone3 directory contains the following folders and files:
   - (a) milestone3/src folder contains all the required source files. Let the files be "lexer.l","parser.y","symbol_table.cpp","3ac.cpp" and "x86.cpp". The bash script file "compile.sh" should be in this folder.
   - (b) milestone3/doc/doc.pdf which contains the instructions

3. **Compilation instructions:**
   To install flex, Bison, Dot and Graphviz use the following commands in linux:

```
sudo apt-get update
sudo apt-get install flex
sudo apt-get install bison
sudo apt-get install graphviz
```

   To compile and execute the files, open the milestone3 directory in terminal and execute the following instruction.
   To compile:

   - Execute the following command to give execute permission to bash script file. (The bash script is in "compile.sh" file which is used to compile the program.)

```
$ chmod +x src/compile.sh
```

   - Execute the following command to compile the source files.

```
$./src/compile.sh
```

   The following files will be created after compilation in src folder:

```
parser.tab.h
parser.tab.c
parser.tab.output
lex.yy.c
parser
convert
```

To execute:

```
$./src/parser tests/testcase.py
$g++ -o convert src/x86.cpp
$cd src
$./convert --input=3ac.txt
$gcc -o 3ac.s -o 3ac
$./3ac
```

- To execute various test cases, change "testcase.py" to the respective ".py" test case file
- The "symboltable.csv" files will be created in the src folder
- The "3ac.s" file will be created in src folder
- The executable file for x86 will also be created in src folder

4. **Modifications for 3AC from milestone2:**
   (a) We have removed instruction numbers
   (b) We have labeled if and for loops as .L index
   (c) We have removed indices for goto instructions and replaced them with end.L
5. No manual changes to be done to the generated assembly file.
6. We did not implement 3ac for classes We did not implement x86 for classes , arrays and strings
7. **Contribution:**
   Chitte Charan Kumar:33%
   Deeven Kumar Ogirala:33%
   Hemasai Narni:33%
8. To do:
   Highlight the required and optional (if any) language features supported by your implementation,
   • Provide command lines and test cases to demo end-to-end execution of the test cases,
   • Modifications (if any) to the 3AC from Milestone 2 for implementing Milestone 3,

• Manual changes (if any) to the generated assembly file to run it successfully with as or gcc,
• List any required features that you could not support and the associated challenges,
• Effort sheet.