CS 335 Milestone 3 - Group 22

April 19, 2024

Tools Used in the Project

- **Flex** for constructing the lexer.
- **Bison** for developing the parser.
- **Git** for version control and collaborative work.
- **Bash** for writing and running scripts.

Compilation Instruction

- We used lexer.l file for lexer in flex
- The command we use is flex lexer.l to compile the lexer.
- We used parser.y file for parser in Bison
- The command we use is bison -d parser.y to compile the lexer.
- We used g++ -o ir lex.yy.c parser.tab.c -lfl to compile and combine both commands
- We used ./ir -input $1 -output out.txt to output out.txt file for symbol table and 3AC code
- As now symbol tables and not necessary we used command rm -f *.csv to remove all csv files that is symbol tables.
- We used loop.cpp to modify and remove some unnecessary 3AC code and we will run loop.cpp by following command g++ loop.cpp -o loop and ./loop out.txt ac.txt
- We used string.cpp file to generate assembly code i.e asm.s file from ac.txt by using commands g++ -o string string.cpp and ./string ac.txt asm.s
- We used all this commands in run.sh file to execute the whole program.
- We used ./run.sh testinput.py to generate the assembly code and 3AC code and output of the assembly code generated for testinput.py input file.

Execution Details

- We should run the run.sh script file with input file path as argument.
- The generated assembly code will be stored in asm.s file and generated 3AC code will be stored in ac.txt file and output of the generated assembly will be printed to terminal.

Language Features and Limitations

- Our compiler mainly supports basic arithmetic operations, relational operations and string comparison.
- It also supports declaration and type error.
- Can also implicit convert int to bool and vice versa but can't support float.
- Supports basic functions including recursion but not support function overloading.
- Also supports basic features of class including single parent inheritance but all class variables should be declared in their methods(function). We can only intialize class through __init__ function
- We also supports list but you have to intialize the list at the time of declaration but can't resize the list.
- Doesn't support string concatenation
- Supports range, print, power and len function
- Also supports Loops, break and continue

Modifications

- We did some changes in parser for classes to support inheritance and allocate space to classes
- Added support for space allocation in list
- Add loop.cpp file to remove unnecessary 3AC line