CS335 Project Milestone 3 Report

Group Members: Divyansh Kankariya (200352), Nishi Mehta (200645), Mohd. Umam (200595)

April 4, 2023

1 Tools Used

In this project, we used the following tools:

1.1 Flex

Flex is a tool for generating scanners (also known as lexers or tokenizers) that can recognize regular expressions in the text. We used Flex to generate a lexer for the Java 17 language.

1.2 Bison

Bison is a tool for generating parsers that can recognize context-free grammar. We used Bison to generate a parser for the Java 17 language.

2 Compilation and Execution Instructions

To compile and execute our lexer and parser, follow the steps below:

1. Unzip the file `cs335_project_200645_200595_200352`
2. Navigate to the project directory
3. Compile the lexer and parser using the makefile provided.

```
// Your code goes here
make
make st
```

On executing the 'make st' command, we will have to enter the filename of the test file. After the execution of the parser, all the symbol tables are stored in the folder 'outputSymTabs' with the filename as 'Scope_SCN.csv' where SCN is the scope number. For example, In our representation, the scope number is represented as '1_2a_3a' for three scopes.

The type-checking errors are saved into a new text file log.txt.

The 3AC code generated has been stored as output3AC/3AC.txt file.

3 Features we worked on during this milestone

We have updated following features in this milestone which were implemented in previous milestone:

- We have implemented support for recursive calls, previously it give an error messge saying "err: recursive call", while in this submission, it does typechecking of the return type and argument list of the recursive call and implements procedure calls required in this milestone.
- Multiple declarations in single line was not functional in previous submission which has been updated to work in this milestone submission.
- Array initialization gave some error in last submission (for eg. int a[3] = 1,2,3;) gave syntax error which has been debugged now.
- In last milestone we were supposed to generate a 3AC code for the java test file, which was incomplete in our submission, we have implemented 3AC code generation for every major functionalities like arithmetic statements, conditional statements, assignments, if-else statements, while, do-while, for statemets, function calls etc.

After updating previous shortcomings we implemented procedure/sub-routine calls for Java-17 in this milestone: `milestone 3`

- Each and every statement added to 3AC code for this milestone is a natural language prompt (according to the requirements of milestone 3) and not an actual x86 instruction, although at some places we have written commands that resemble x86 instructions.
- In this milestone we have implemented support for Data movement instructions (eg. mov, push, pop)
- We have also implemented support for Control flow instructions like call, ret.
- We have pushed parameters of argumentlist during methodInvocation, constructor invocation.
- Apart from this we ensured that the object of the size of the class is returned and memory is allocated in heap using allocmem when we use "new" keyword for memory allocation.
- We also have extended the fields of symbol table to store the size and offset of every local variable, field declaration, parameters of the function.
- This way we have made space for the locals and actual parameters.
- While saving the offset of formal parameters we left space of 8 bytes which denotes the space for base pointer and return value.
- When the declaration of a class is completed, we fill its size in its symbol table entry. This way, we know the size of the object of the class each time we create its object.
- We have ensured correctness in the movement of stack pointer at each and every place.

Note:

- Indexing of our 3AC code starts from 0.
- This list is not exhaustive and may be updated as we continue to develop our implementation.

In addition, we have created 11 test cases to ensure the correctness of our lexer plus parser. These test cases are located in the `tests` folder.