Compilers Project VGM

Phase3

Phase3

- The phase-3 of this project is Semantic Analysis
- Semantic Analysis is about checking the semantics of the program
- We have static and dynamic semantic checks
- Static semantic checks include type checking, declaration of variable, number of arguments of a function, checking the return type of a function,etc..
- Dynamic checks include overflow while performing ‘+’ or ‘*’, using the index of an array upper or lower than the declared, stack overflow, etc..
- Static checks are done in the attributed grammar
- Dynamic checks are done at the run time

- The semantics which we have included are :
    - Expression type checking
    - duplicate variable name, function name, label name,class and struct name checking
    - Function should contain at least one return statement
    - Variables must be declared before use
    - Parameters type checking in functions
    - Function declared type and return type checking
    - Giving valid conditional arguments
    - Dimension checking for matrix, number of vertices for graphs and the maximum size of array

- Element-type checking in vectors and arrays
- Checking if the accessed attributes are present in structs and classes
- Maintaining scope
- Maintaining positive dimensions for matrix and graph
- Type checking for in-built functions, arithmetic operations and logical operations

SYMBOL TABLE

- We have maintained different symbol tables for variables, class names, structs and functions.
- The variables declared in functions get deleted whenever the parser reaches the end of the function. However, the attributes declared inside struct and those in classes outside methods have to be stored for accessing them in the program.
- All the declarations get stored along with their level number for scope-checking.
- Insertion, Search and deletion functions related to symbol table are written separately in symbol_table.c file.

- symbol_table.c file is included in parser.y yacc file.
- For execution, we can run the ‘bash run.sh’ command.
- The following commands are present in the “run.sh” file:
    - yacc -d parser.y
    - lex lexer.l
    - gcc lex.yy.c y.tab.c -w
    - ./a.out
- In case there is an error in the syntax of the given input program, an error message with the line number can be seen in the terminal