Project phase3: Semantic checker for C- language

Due Date for Report and Code: November 14 , 2023

The objective of this assignment is to perform semantic analysis such as type and scope analysis and declaration processing, and integrate such analyses with the parser. Add semantic actions to your parser to produce abstract syntax for the C language.

Notes on how to conduct the experiments:

1.For declaration processing, do the following,

- Decide on the attribute information (i e type, dimension etc).Accordingly design the symbol table entry formats for storing information regarding variable identifiers, label identifiers, and function identifiers.
- Write semantic rules for processing of all declarations of data in a single block of source program and recording relevant information in the symbol table.

2. For type analysis, write semantic rules to do the following,

- For a use of a symbol, check for the declaration that the symbol is bound to using C scope rules. If the symbol is not bound to any declaration , declare an error. The symbol table should be rich enough to implement the scope rules.
- The symbol table entry for the declaration will provide all the attributes of the symbol.
- Use this information to check that use of a symbol is type compatible with its context. C language type compatibility should be followed .
- Perform type conversion if necessary.

3. Detects and reports semantic errors. You should detect as many semantic errors as possible in a single compilation of the program. Error messages should include the line number where they occurred.

Your project report should include:

- An overview explaining your code including info about what doesn't work(if anything).
- Listings of the code for your semantic checker.
- A description of the checks performed.
- A description of the symbol table
- A discussion of any assumptions you made beyond what is in the basic language description.
- A report on the test cases you ran along with the results.