

Lexical Analyser

(GeoC) by Team 11

J. Sriram - CS21BTECH11025

Manoj - ES21BTECH11020

K. Jhanavi - CS21BTECH11032

P. Bindu Sree - CS21BTECH11048

Project objective:

**To Build a lexer for GeoC(A DSL
for implementing geometry
operations with C**

Overview

- Lexical Analysis is the first stage of building a Compiler.
- It generates the token stream from the code you gave to compiler
 - Scans the user code from top to end of file
 - Generates tokens from the code.
 - Resolves conflicts by generating the longest match token.
 - When a token cannot be generated from the code shuts the process and sends corresponding error message to the terminal
 - Ignores white spaces, tabs and comments

Implementation

- The input file will be given in .geoc format
- we used lex tool to build lexical analyser so further explanation will be of lexer.l file rather than lexer.yy.c which is autogenerated.
- In the rules section we defined the rules for identifying each particular sequence of characters that are possible in geoc.
 - In definitions section we defined some particular sequence of characters for easy usage
- Once a particular sequence of characters is matched with a rule it then takes the corresponding action that is associated with the rule.

Implementation

- When multiple rules are matched to a sequence of characters, the rule which is matched with the longest sequence is considered.
- If both have same length, the the rule that is defined first will be considered.
- For example all rules corresponding to reserved keywords should be defined before the identifier rule.
- In this stage we are printing the sequence of character that is matched and the type of token in tokens.txt file as action

Implementation

- First run the `lex lexer.l` file in terminal. It generates the `lexer.yy.c` file
- Compile the `c` file with any C compiler that is preinstalled in your device.
- If GCC, then `gcc lexer.yy.c`. It generates an executable `a.out`
- run the `a.out` program with the `geoc` file as an argument
 - The `geoc` files are in `testCases` folder. Bring them to the working directory.
- The tokens will be generated in `tokens.txt`

Proposed deliverables

Deliverable 1

- `lexer.l`
- The lex file corresponding to lexical analysis

Deliverable 2

- PPT and videos(Folder)
- The description,demo and design overview videos

Deliverable 3

- `testCases(Folder)`
- contains the testcases