Parser

(GeoC) by Team 11

J. Sriram - CS21BTECH11025

Manoj - ES21BTECH11020

K. Jhanavi - CS21BTECH11032

P. Bindu Sree - CS21BTECH11048

Project objective:

To Build a parser and lexical analyser for GeoC(A DSL for implementing geometry operations with C)

Overview

- We know that regular expressions suffices to check for a particular sequence in characters, but will it be enough to check the syntax of a language ???
- Well, the answer is No and we need Context Free Grammar(CFG).
- Parser uses CFG to define the syntax of a language and checks whether the input code follows that syntax or not.
- It takes stream of tokens as input and verifies them against the grammar rules
- We use yacc/bison to build the parser. So further explanation is done with parse_source.y as reference

Lexical Analyser

- While building lexical analyser just for the Lexer stage(lexer.l), we printed the token type and the sequence of characters that is matched in tokens.txt file.
- But that needs to be modified a little for making it work with parser(lex_source.l)
- we remove the main function from the lexer.l file and when a sequence of characters is matched instead of printing, we return a token
- we include parse_source.tab.h in the lex_source.l

Accepted tokens & parse_source.tab.h

The tokens that the parser can accept is are defined before the grammar section.

When we run bison -d -t parse_source.y, a “.tab.c” file and “.tab.h” files are generated

parse_source.tab.h file contains the tokens that parser can accept.

Implementation

- The lexer is called by the parser iteratively till the lexer reaches end of input file. Each call of lexer generates only one token.
- The parser is LALR(1).
- We wrote the grammar rules for verifying the syntax of almost all components in C, like declaration, expression, predicate, if-else statements, for loops, while loops, print statements, function calls, structs, operations, break-continue, scopes, extra brackets() to define precedence, e.t.c.
- We added the point, triangle as new datatypes and the ic, cc, ir, cr, oc, centroid, slope, area inbuilt functions.

Execution

- Run the following commands in terminal after entering the Parser directory
  - lex lex_source.l
  - bison -d -t parse_source.y
  - cc lex_source.yy.c parse_source.tab.c
  - ./a.out testcase1.geoc
- In output.txt, you will get the type of statement corresponding to each line.

Test Case Example

```
1  int main(){
2      int x;
3      x = 2;
4      if(x==2){
5          print("The value of x is " + x  );
6      }
7      return 0;
8  }
```

```
1  Method
2  Decleration stmt
3  Expression stmt
4  If stmt
5  Print stmt
6  end of If stmt
7  Return stmt
8  end of Method
```

Proposed deliverables

Deliverable 1	• lex_source.l • parse_source.y
Deliverable 2	• PPT and videos(Folder) • The description,demo and design overview videos
Deliverable 3	• testCases(Folder) • contains the testcases

Thank You