Assignment 5

Computing Lab 2 (CS69012) 31/01/2018

Problem Statement

Lex is a lexical analysis tool that can be used to identify specific text strings in a structured way from source text. Yacc is a grammar parser; it reads text and can be used to turn a sequence of words into a structured format for processing. In this assignment, you have to develop a desk calculator application using both lex and yacc. The calculator should support BODMAS rules and operators. Standard precedence and associativity rules should be taken care of. Numbers can be integers or floats, positive or negative, so you have to take care of unary operators as well. For simplification consider that exponents will be only integer and positive.

Problem 1

Submit calc1.l file which contains pattern matching rules. lex will generate lex.yy.c file which contains yylex() function which is called by yyparse() to get the token. In this part instead of returning the token you will print the identified token on console. Following is list of tokens: INTEGER, MINUS_OP, PLUS_OP, LEFT_BRACKET, RIGHT_BRACKET, DIV_OP, MULT_OP, POWER_OP. Below is sample output.

Listing 1: Sample Output of lex

```
( 2 - 3 ) ^ 4 * 5 / 6 + 1 - 9
    Found token : LEFT_BRACKET ( lexeme : ( )
    Found token : INTEGER  ( lexeme : 2 )
    Found token : MINUS_OP ( lexeme : - )
    Found token : INTEGER  ( lexeme : 3 )
    Found token : RIGHT_BRACKET ( lexeme : ) )
    Found token : POWER_OP ( lexeme : ^ )
    Found token : INTEGER  ( lexeme : 4 )
    Found token : MULT_OP ( lexeme : * )
    Found token : INTEGER  ( lexeme : 5 )
    Found token : DIV_OP ( lexeme : / )
    Found token : INTEGER  ( lexeme : 6 )
    Found token : PLUS_OP ( lexeme : + )
    Found token : INTEGER  ( lexeme : 1 )
    Found token : MINUS_OP ( lexeme : - )
    Found token : INTEGER  ( lexeme : 9 )
hi2+3
    invalid string : hi
    Found token : INTEGER  ( lexeme : 2 )
```

```
20      Found token : PLUS_OP ( lexeme : + )
21      Found token : INTEGER  ( lexeme : 3 )
```

Problem 2

Submit both calc.l and calc.y files. This time yylex() will return tokens to yyparse() instead of printing on console. calc.y will contain grammar rules to parse the input, which you have to formulate with respect to this problem. For any input that does not conform to the grammar rules, the program should print 'syntax error'. Otherwise, it should print 'correct syntax'. Following is the sample input-output example:

Listing 2: Sample Output of lex for 2

```
1 user > ( 2 - 3 ) ^ 4 * 5 / 6 + 12 - 9
2 calc > correct syntax
3 user > 3++4
4 calc > syntax error
```

Useful Links

1. Lex Yacc Tutorial

Marking Scheme

TAs will check your understanding of how lex and yacc works. Go through tutorial link given above before you start implementation.

- Lex 30 Marks
- Yacc 30 Marks
- Viva 40 Marks

Reference Books

Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. 1986. Compilers: Principles, Techniques, and Tools. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.