CS 17L1 LANGUAGE PROCESSOR LABORATORY

Cycle of Experiments

Cycle 1 (C Programs)

1. Write a C program to implement a DFA for the regular expression (a/b)*abb using IF-ELSE.
2. Write a C program to implement a DFA for the regular expression 1*2*3* using transition table.
3. Write a C program to implement a DFA accepting binary strings ending with '00'.
4. Write a C program to implement a DFA accepting strings made of {a,b,c} having 'abc' as a substring.
5. Write a C program to implement a DFA accepting binary strings such that every '00' is immediately followed by 1.

Cycle 2 (LEX Programs)

1. Study of Lex
2. Write a LEX program to recognize the regular expressions
   i) aa(a+b)*bb
   ii) (0+1)*000(0+1)*
3. Write a program LEX program to count the number of characters,words,lines in a text file.
4. Write a LEX program which accepts a C program and display
   i) number of C statements
   ii) number of Identifiers
   iii) number of assignment operators
   iv) number of relational operators
   v) number of keywords
   vi) number of integers
5. Write a LEX program to copy a text file. The new file should contain only words(no numbers) which are separated by one blank space.

Cycle 3(YACC Programs)

1. Study of YACC.
2. Write a YACC program to recognize arithmetic expressions having identifiers and numbers using all basic operators such as +,-,*, / .
3. Write a program using LEX and YACC to recognize IF statements in C. ie if (condition) statements;
4. Write a program using LEX and YACC to implement a calculator.

Cycle 4 (Parsers)

1. Write a program to implement a Shift Reduce parser for arithmetic expressions with operators with all basic arithmetic operators .(+, - , * and /)

2. Write a program to implement a recursive descent parser for arithmetic expressions with basic arithmetic operators.

3. Write a program to implement a Operator Precedence parser.