CS431CompilerDesignLab

List of Experiments

1. Design and implement a lexical analyzer for given language using C. The lexical analyzer should ignore redundant spaces, tabs and newlines.

Sample input(in file):

int a,b,c;

C = a+b;

output(in file or console): kwd id,id,id; id oper-equ id oper-plus id ;

Date of completion & upload: before 23/8/2020 3.30 pm

Continuous Evaluation: 25/8/2020

2. Implementation of Lexical Analyzer using Lex Tool.

3. Generate YACC specification for a few syntactic categories.

a. Program to recognize a valid arithmetic expression that uses operator +, –, * and /.

b. Implementation of Calculator using LEX and YACC.

c. Convert the BNF rules into YACC form and write code to generate abstract syntax tree..

4. Write program to find ε – closure of all states of any given NFA

with ε transition

5. Write program to convert NFA with ε transition to NFA without ε transition.

6. Write program to convert NFA to DFA.

7. Write program to minimize any given DFA.

8. Develop an operator precedence parser for a given language.

9. Write program to Simulate First and Follow of any given grammar.

10. Construct a recursive descent parser for an expression.

11. Construct a Shift Reduce Parser for a given language.

12. Write a program to perform loop unrolling.

13. Write a program to perform constant propagation.

14. Implement Intermediate code generation for simple expressions.

15. Implement the back end of the compiler which takes the three address code and produces the 8086 assembly language instructions that can be assembled and run using an 8086 assembler. The target assembly instructions can be simple move, add, sub, jump etc.