|  |  |
| --- | --- |
| MONO | **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING****NATIONAL INSTITUTE OF TECHNOLOGY PATNA** Ashok Raj Path, PATNA 800 005 (Bihar), India |
|  Phone No.: 0612 – 2372715, 2370419, 2370843, 2371929, 2371930, 2371715 Fax – 0612- 2670631 Website: [www.nitp.ac.in](http://www.nitp.ac.in/) |

***CSL5404 Compiler Design Lab***

**L-T-P-Cr: 0-0-3-1**

**Pre-requisites:** Basics of Compiler Design

**Objectives/Overview:**

The Objectives of this lab is to make students capable of developing different phases of a compiler for a programming language

**Course Outcomes:**

At the end of the course, a student should:

|  |  |  |
| --- | --- | --- |
| Sl. No | Outcome | Mapping to PO |
|  1 | Development of lexical analyzers.  | PO2, PO3 |
|  2 | Development of top-down and bottom-up parsers and semantic analyzer. | PO2, PO3 |
|  3 | Development of intermediate code generation phase. | PO2, PO3 |
|  4 | Development of code optimization and code generation phase. | PO2, PO3 |

**List of Experiments:**

1. Consider the following regular expressions:

a) (0 + 1)\* + 0\*1\*

b) (ab\*c + (def)+ + a\*d+e)+

c) ((a + b)\*(c + d)\*)+ + ab\*c\*d

Write separate programs for each of the regular expressions mentioned above.WAP to shuffle a deck of cards.

1. Design a Lexical analyzer for identifying different types of token used in C language.

[Note that the reserved keywords such as if, else, class, struct etc must be reported as invalid identifiers. C allows identifier names to begin with underscore character too. Further, the real numbers can also be expressed in the form 1.75e-30 which is 1.75 x 10-30.]

1. Given a text-file which contains some regular expressions, with only one RE in each line of the file. Write a program which accepts a string from the user and reports which regular expression accepts that string. If no RE from the file accepts the string, then report that no RE is matched.
2. Write a program to design a minimized DFA for a given regular expression.
3. Write a program which accepts a regular expression from the user and generates a regular grammar which is equivalent to the R.E. entered by user. The grammar will be printed to a text file, with only one production rule in each line. Also, make sure that all production rules are displayed in compact forms e.g. the production rules:

S--> aB, S--> cd

S--> PQ

Should be written as

S--> aB | cd | PQ

And not as three different production rules. Also, there should not be any repetition of production rules.

1. Allocation of mini projects
2. a. Write a program to eliminate left recursion.

b. Write a program to find First and Follow sets of a grammar.

1. Consider the following grammar:

S --> ABC

A--> abA | ab

B--> b | BC

C--> c | cC

Following any suitable parsing technique(prefer top-down), design a parser which accepts a string and tells whether the string is accepted by above grammar or not.

[This grammar is not as simple as it appears to be. Try to check if the string abbcc is accepted or not. It is accepted, but improper logic of parsing can cause the program to get stuck or go into infinite loop !! The students are permitted to generate an equivalent grammar by removing ambiguities (if any) and immediate left-recursion from the given grammar.]

1. Design a parser which accepts a mathematical expression (containing integers only). If the expression is valid, then evaluate the expression else report that the expression is invalid.

[Note: Design first the Grammar and then implement using Shift-Reduce parsing technique. Your program should generate an output file clearly showing each step of parsing/evaluation of the intermediate sub-expressions. ]

[Extra credit will be given if the program can handle advanced operations such as exponentiation, square-root, cube-root, logarithm etc]

1. Write a program to find three address codes for the given set of input expressions.
2. Write a program to find quadraples for the given set of input expressions.
3. Write a program to optimize given three address codes.
4. Develop a program for Part of speech tagging (POS) of english words.

[One may consult wordnet for this]

1. Develop a program for Part of speech tagging (POS) of hindi words.

[One may consult hindiwordnet for this]