Experiment IV

**Aim**: Construct a recursive descent parser for an expression.

Algorithm

1. Start
2. Input the expression.
3. Check whether the first term is alpha-numeric or ‘(‘. Else it is an invalid expression.
4. If the first term is ‘(‘ recursively check whether the second term is alpha-numeric till a ‘)’ is encountered. Else, it is an invalid expression.
5. Repeat this process till the whole expression is parsed.
6. After all the iterations, if the expression is not invalid, print “Valid expression”.
7. Stop

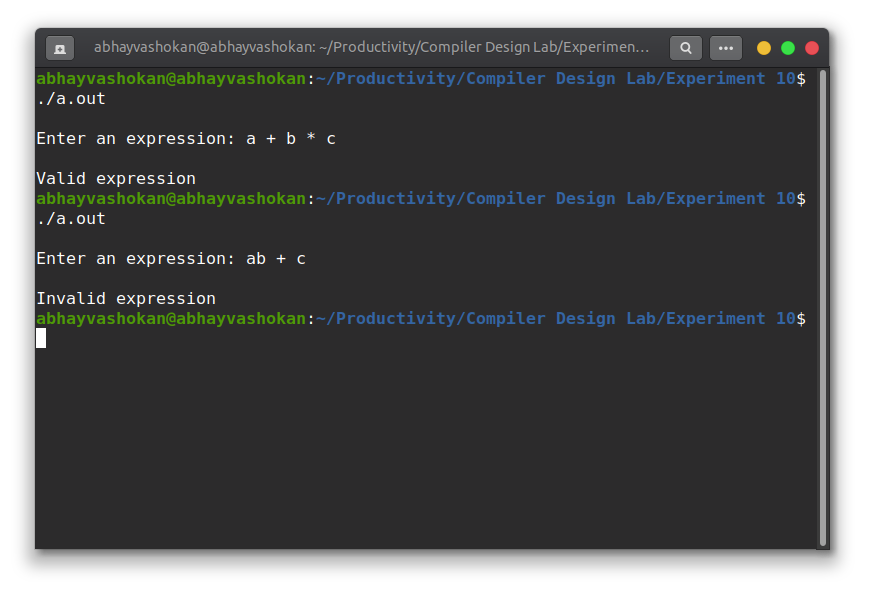
Output

Enter an expression: a + b \* c

Valid expression

Enter an expression: ab + c

Invalid expression

Screenshot

Readme

1. Compile and run the C program using the command

**gcc 2Abhay-P10.c && ./a.out**

2. Enter an expression

3. The program returns whether the expression is valid or not

**Result**: Successfully implemented a program to find the ε - closure of all the states in an NFA with ε transitions.