Experiment III c

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

Algorithm

1. Start
2. Read the name of the program to be analysed as input.
3. Check each pattern `p` against regular expressions in lex.
   1. If `p` “main() “if”, “else”, “while”, “int”, “char” or “float” return the corresponding token.
   2. If `p` is an identifier, return `VAR` token.
   3. If `p`p is a number return `NUM` token.
   4. If `p` is either ‘<’, ‘>’, ‘<=’, ‘>=’, ‘==‘ return `RELOP` token.
   5. If `p` is a new line character increment lineCounter.
4. Check whether the expression satisfies the context free grammar logic.

PROGRAM -> MAIN BLOCK

BLOCK-> { CODE }

CODE -> BLOCK STATEMENT CODE STATEMENT

STATEMENT -> DECLARE; ASSIGNMENT;CONDITION; LOOP;

DECLARE -> TYPE VAR;

ASSIGNMENT -> VAR = EXPR;

EXPR -> EXPR + EXPR

| EXPR - EXPR

| EXPR \* EXPR

| EXPR / EXPR

| VAR

| NUM

CONDITION -> IF ( STATEMENT) BLOCK

LOOP -> WHILE ( CONDITION) BLOCK

1. At each step build the abstract syntax tree and store it in an array.
2. Display the array in the form of a table.
3. If the expression does not match the context free grammar, print “Invalid expression”
4. Stop

Output

Enter the file name of the program: program.c

-----------------------------------------------

Pos Operator Arg1 Arg2 Result

-----------------------------------------------

0 = 1 a

1 = 3 b

2 = 4 c

3 + a b t0

4 = t0 a

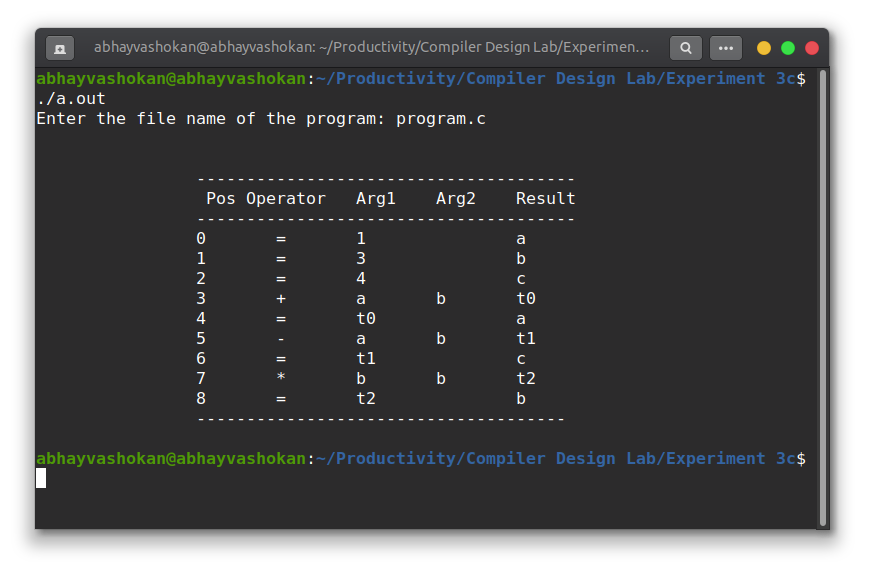
5 - a b t1

6 = t1 c

7 \* b b t2

8 = t2 b

----------------------------------------------

Screenshot

Readme

1. Compile the lex program using the command

**lex 2Abhay-P3c.l**

2. Compile the yacc program using the command

**lex 2Abhay-P3c.y -d**

3. Now compile and run the **lex.yy.c** and **y.tab.c** files generated using the command

**gcc lex.yy.c y.tab.c -ll && ./a.out**

4. Input the path of the program file.

5. The abstract syntax tree of the program would be obtained as output.

**Result**: Successfully implemented a program to generate an abstract syntax tree for a given program.