Shreeya Thigale
TE-TX
33168

PAGE No
DATE / /

ASSIGNMENT-12

TITLE: LEX & YACC for Intermediate Code generation.

PROBLEM STATEMENT: Write a program for intermediate code generation using LEX and YACC for Control flow Statement. (Either while loop or switch case)

OBJECTIVES:
- To understand fourth pase of compiler: intermediate code generation
- To learn and use compiler writing tools.
- To learn how to write address code for given statement.

THEORY:

In the analysis-synthesis model of compiler, the front end analyzes a source program and creates an intermediate representation, from which the back end generates target code. Ideally details of source language are confined to the front end, and details of target machine to back end. The front end translates a source program into an intermediate representation from which the back end generates target code

Scanned by CamScanner

Shreya Thigale
TE-IX
33168

PAGE No.
DATE / /

Intermediate Languages

Three ways of intermediate representation:

- Syntax tree
- Postfix notation
- Three address code

Steps to execute code:

```
1.  $ lex filename.l
2.  $ yacc -d filename.y
3.  $ cc lex.yy.c y.tab.c -ll -ly -lm
4.  $ ./a.out
5.  eg. comp.l
6.  eg. comp.y
```

ALGORITHM:

1. Declaration of header files specially y.tab.h which contains declaration for letter, digit, expr.
2. End declaration section by %%
3. Match regular expression
4. If match found then convert it into char & store it in yylval.p where p is pointer declared in YACC
5. Return token
6. If input contains new line character (\n) then return 0.

Scanned by CamScanner

Shreeya Thigale
TE-IX
33168

PAGE No.
DATE / /

7. If input contains "." then return yytext[0].
8. End rule-action section by %%
9. Declare main function
10. Open file given at command line
11. If any error occurs then print error & exit
12. Assign file pointer fp to yyin
13. Call function yylex until file ends.
14. End

YACC program

1. Declaration of header files.
2. Declare structure for three address code representation having fields of argument1, argument 2, operator, result.
3. Declare pointer of char type in union
4. Declare token expr of type pointer p.
5. Give precedence to "*", "/"
6. Give precedence to "+", "-"
7. End of declaration section by %%
8. If final expression of ~~the form~~ evaluates
8. then add it to the table of three address code
9. If input type is expression of form
10. Exp "+" exp then add to table the argument1, argument 2, operator
11. exp "-" exp then sub to table the argument1, argument2, operator
12. Exp "*" exp then add to table the argument1, argument 2, operator.

Scanned by CamScanner

Shreeya Thigale
TE-IX
33168.

PAGE No.
DATE / /

13. Exp"/" exp then add to table the argument1, argument2, operator
14. "(" exp ")" then assign $2 to $$
15. Digit OR letter then assign $1 to $$
16. End the section by %%.
17. Declare file "yyin externally
18. Declare main function and called yyparse function until yyin ends
19. Declare yyerror for if any error occurs
20. Declare char pointer s to print error
21. Print error messages
22. End of program.

CONCLUSION:
Thus we have successfully implemented program for intermediate code generation using LEX & YACC for control flow.

Scanned by CamScanner