Name - Onkar Mendhapurkar
PRN - 2207012213 5, CSE B2, 22-26

CCL Experiment - 1

Title - Write details about LEX, YACC, file format, & related functions. Solve post lab questions too

- LEX - It stands for lexical analyzer. LEX is a UNIX utility which generates the lexical analyzer. LEX is a tool for generating scanners. scanners are programs that recognize lexical pattern in text. These lexical patterns are defined in particular syntax. A matched regular expression may also have an associated actions. This action may also include returning a token. when LEX receive input in the form of a file or text, it attempts to match the text with the regular expression. It takes input one character at a time & continue until a pattern is matched.

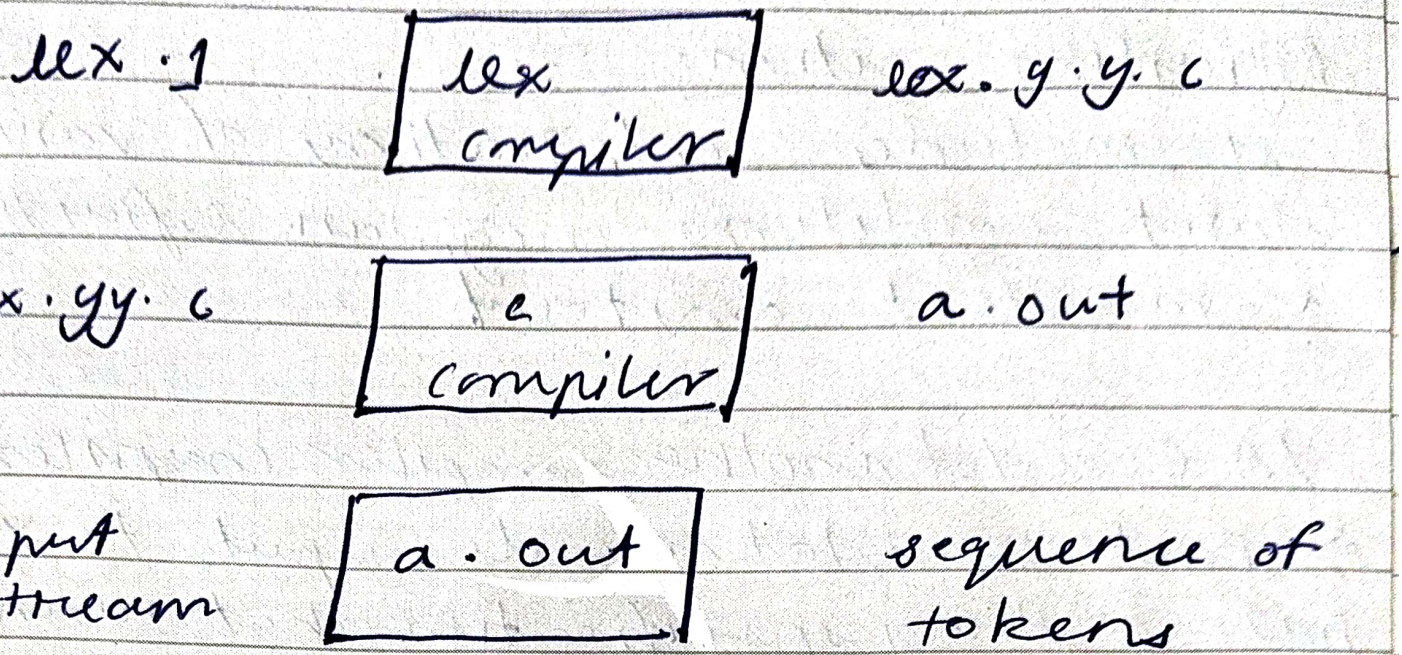

Fig 1.1: Creating a lexical analyzer with LEX

To generate a lexical analyzer two important things are needed. Firstly it will need a precise specification of the tokens of the language. Secondly it will need a specification of the actions to be performed on identifying each token.

⇒ LEX specification (Structure

```
%{
    Definition section
%}
%%
    Rules section
%%
    Auxiliary functions
```

Definition section:
It includes declarations of variables, start conditions regular definitions & manifest constant

YACC (Yet another compiler compiler)
⇒ Each translation rule input to YACC has a string specification that resembles a production of grammer - It has a non-terminal on the LHS & a few alternatives on the RHS. For simplicity, we will refer to a string specification as a production. YACC generates an LALR(1) parser would operate as follows:

- For a shift action, it would invoke the scanner to obtain the next token & continue the parse by using that token. While performing a reduced action in accordance with production, it would perform the semantic action associated with that production.
- The semantic actions associated with productions achieve the building of an intermediate representation or target-code as follows:

the semantic action associated with a production can access attributes of nonterminal symbols used in that production symbols "$n" in the semantic action where n is an integer, designated the attributes of nonterminal symbols in RHS of the production & the symbols "$$" designated the attributes of LHS nonterminal.

Input file: YACC input file is divided into three parts.

```
/* definitions */
  ....
%%
/* rules */
  ....
%%
/* auxilary routines */
  ....
```

Definition part:

```
%token number
%token ID
%token Number 621
```

Q1) What are specifications for Regular expression?

⇒ . → any character
* → zero or more occurance
+ → one or more occurance
[] → any one character in the set
^ → beginning of line
$ → end of line
| → OR operator

Q 2) Explain with an example syntax LEX and YACC program.

⇒ %. %.
[0-9] + {printf("Number: %s \n", yytext);}
[a-z A-Z] + {printf("word: %s \n", yytext);}

YACC example

expr: expr '+' expr {printf("Add \n");}
{expr '*' expr {printf("multiply \n");};}
{Number};
%. %.

Q3) With an example explain tokens

⇒ Tokens are the smallest units recognized

by a lexical analyzer

example in LEX:

"if" { return IF; }

[0-9]+ { return Number; }

[a-z A-Z]+ { return ID; }

Q4) Lists commands for LEX & YACC execution

⇒ lex file.l

yacc -d file

cc lex.yy.c y.tab.c -o output

./output