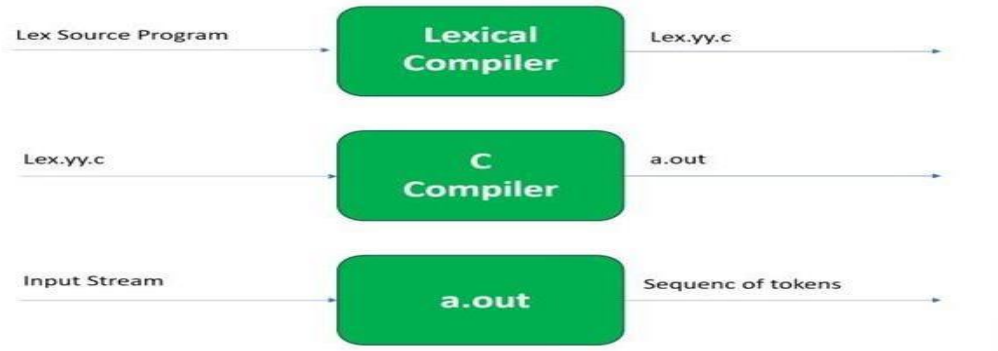
Practical 4

Title: : Prepare report for Lex and install Lex on Linux/Windows

Introduction:

Lex is a tool or a computer program that generates Lexical Analyzers (converts the stream of characters into tokens), also caled tokenizers. The Lex tool itself is a compiler. The Lex compiler takes the input and transforms that input into input patterns. It is commonly used with YACC(Yet Another Compiler Compiler).



Create a Lex file:
• A file with the .l extension is created.
• This file contains lexical rules that define tokens for the input stream.
• Let's assume the file is named Test.l.

Lex compilation:
• The Test.l file is passed through the Lex compiler.
• The Lex compiler generates a C program named Test.yy.c.
• This C program contains the code for recognizing the tokens defined in Test.l.

C compilation:
• The Test.yy.c file is compiled using a C compiler.
• This generates an object file named Test.out.
• The object file contains the machine code for the token recognition logic.

Tokenization:
• The Test.out object program is executed with an input stream.
• The input stream is processed, and the tokens defined in Test.l are identified.
• The output is a sequence of tokens.

Lex file format:

A Lex program consists of three parts and is separated by %% delimiters:-
Declaration
%%
Translation rules
%%
Auxiliary procedures
• Declarations: The declarations include declarations of variables.
• Transition rules: These rules consist of Pattern and Action.
• Auxiliary procedures: The Auxilary section holds auxiliary functions used in the actions.

Lex predefined functions and variables:

1. yyin :- the input stream pointer (i.e it points to an input file which is to be scanned or tokenised), however the default input of default main() is stdin .

2. yylex() :- implies the main entry point for lex, reads the input stream generates tokens, returns zero at the end of input stream .

3. yytext :- a buffer that holds the input characters that actually match the pattern (i.e lexeme) or say a pointer to the matched string .

4. yyleng :- the length of the lexeme .

5. yylval :- contains the token value .

6. yyval :- a local variable .*

7. yyout :- the output stream pointer (i.e it points to a file where it has to keep the output), however the default output of default main() is stdout .

8. yywrap() :- it is called by lex when input is exhausted (or at EOF). default yywrap always return 1.

9. yymore() :- returns the next token .

10. yyless(k) :- returns the first k characters in yytext .

11. yyparse() :- it parses (i.e builds the parse tree) of lexeme *sss

Lex installed screenshot:

```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>lex
```