CS251: Introduction to Language Processing

LEX Tool

Abhay Deep Seth

Teaching Assistant

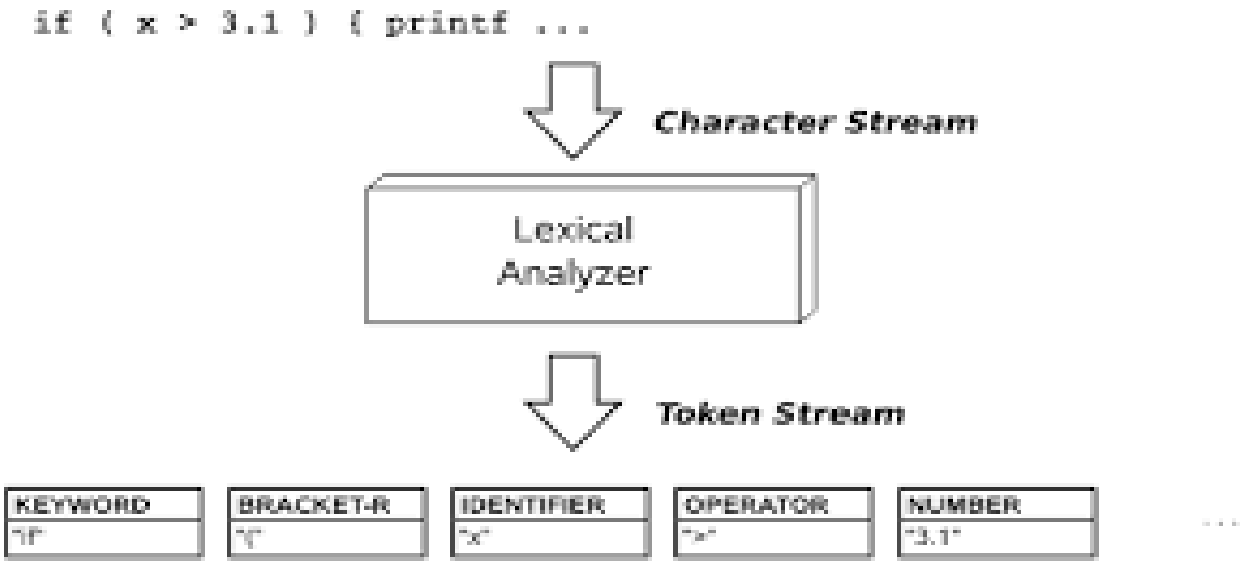
Indian Institute of Technology, Bhilai

abhays@iitbhilai.ac.in

2020-21 W

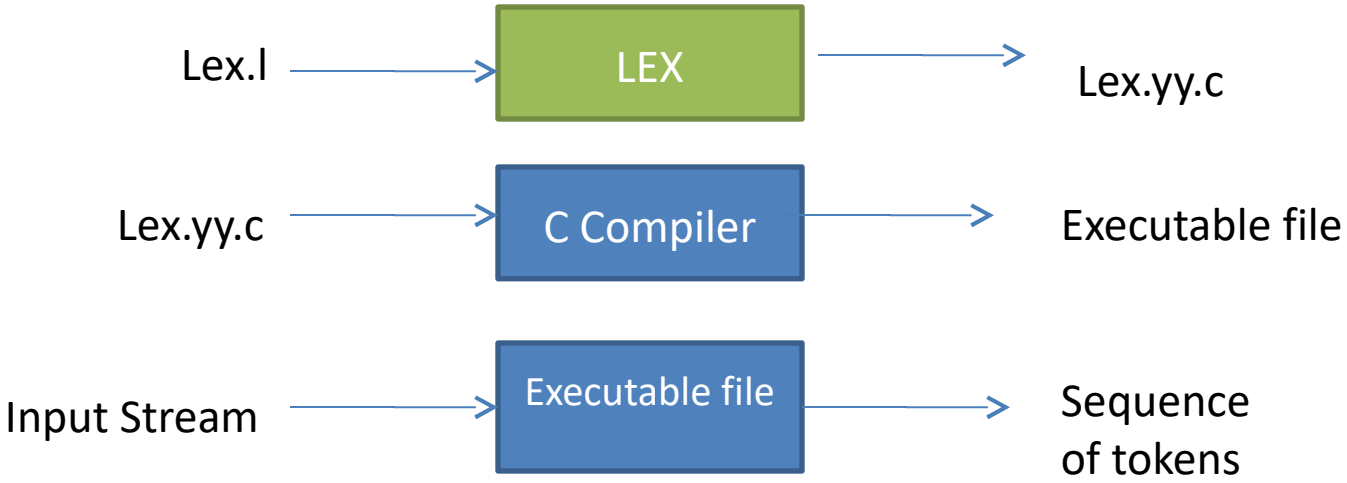
Lexical Analyzer Generator

- Lex is a tool that generates lexical Analyzer.
- Lexical Analyzer first phase of compiler.
- Lexical Analyzer takes input as source code and generate output as tokens



Contd...

- The code is written in lex language. The extension .l is used for the lex program e.g filename.l
- The Lex compiler after compiling the lex source file i.e., filename.l always generates output as lex.yy.c



Structure of LEX Program

- The LEX program has following structure:

```
%{
      C header files \* if required *\
      Definition section
%}

%%

Translation Rules

%%

Auxilliary functions
```

Contd...

- The definition section defines macros and imports header files written in C.
- The rules section associates regular expression patterns with C statements.
- The C code section contains C statements and functions and contain code defined by the rules in the rules section

Patterns

Table 1: Special Characters	
Pattern	Matches
.	any character except newline
\.	literal .
\n	newline
\t	tab
^	beginning of line
$	end of line

Contd...

Table 2: Operators	
Pattern	Matches
?	zero or one copy of the preceding expression
*	zero or more copies of the preceding expression
+	one or more copies of the preceding expression
a|b	a or b (alternating)
(ab)+	one or more copies of ab (grouping)
abc	abc
abc*	ab abc abcc abccc ...
"abc*"	literal abc*
abc+	abc abcc abccc abcccc ...
a(bc)+	abc abcbc abcbcbc ...

Important Functions

- yywrap called by lex when i/p is exhausted or finished (returns 1)
- yylex() : The main point for lex. It reads the i/p stream and generates tokens also return 0 at the end of i/p stream. It is called to invoke the lexer.
- yyleng: It defines the length of matched string

```
%{
#include<stdio.h>
int vowel=0;
int cons=0;
%}
%%
"a"|"e"|"i"|"o"|"u"|"A"|"E"|"I"|"O"|"U" {vowel++;}
[a-zA-z] {printf(cons++;}
%%
int yywrap()
{return 1;}
main()
{
printf("Enter String\n");
yylex();
printf("vowel=%d and Consonant=%d",vowel,cons);
}
```

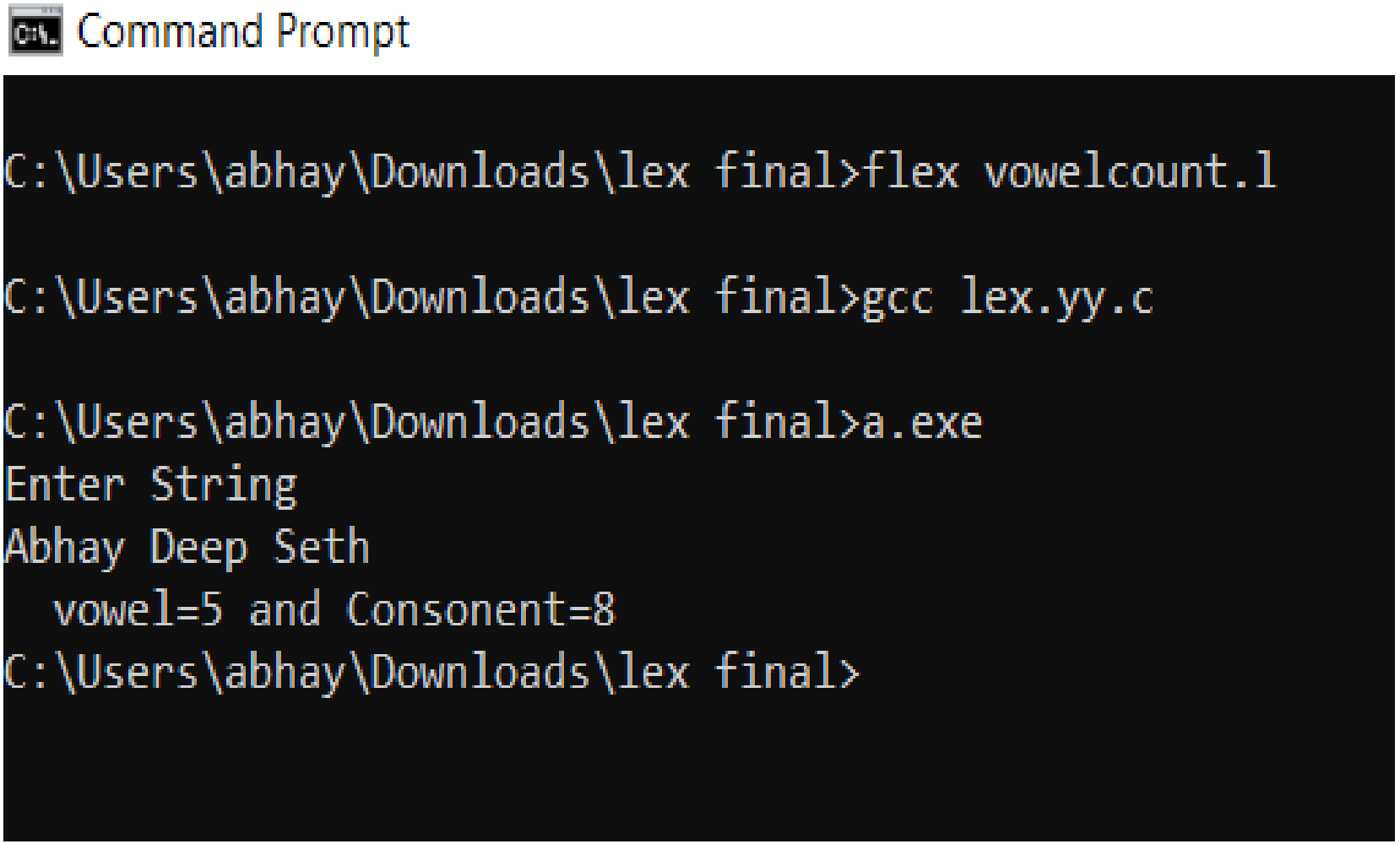
DEFINITION

RULES

Auxilliary functions

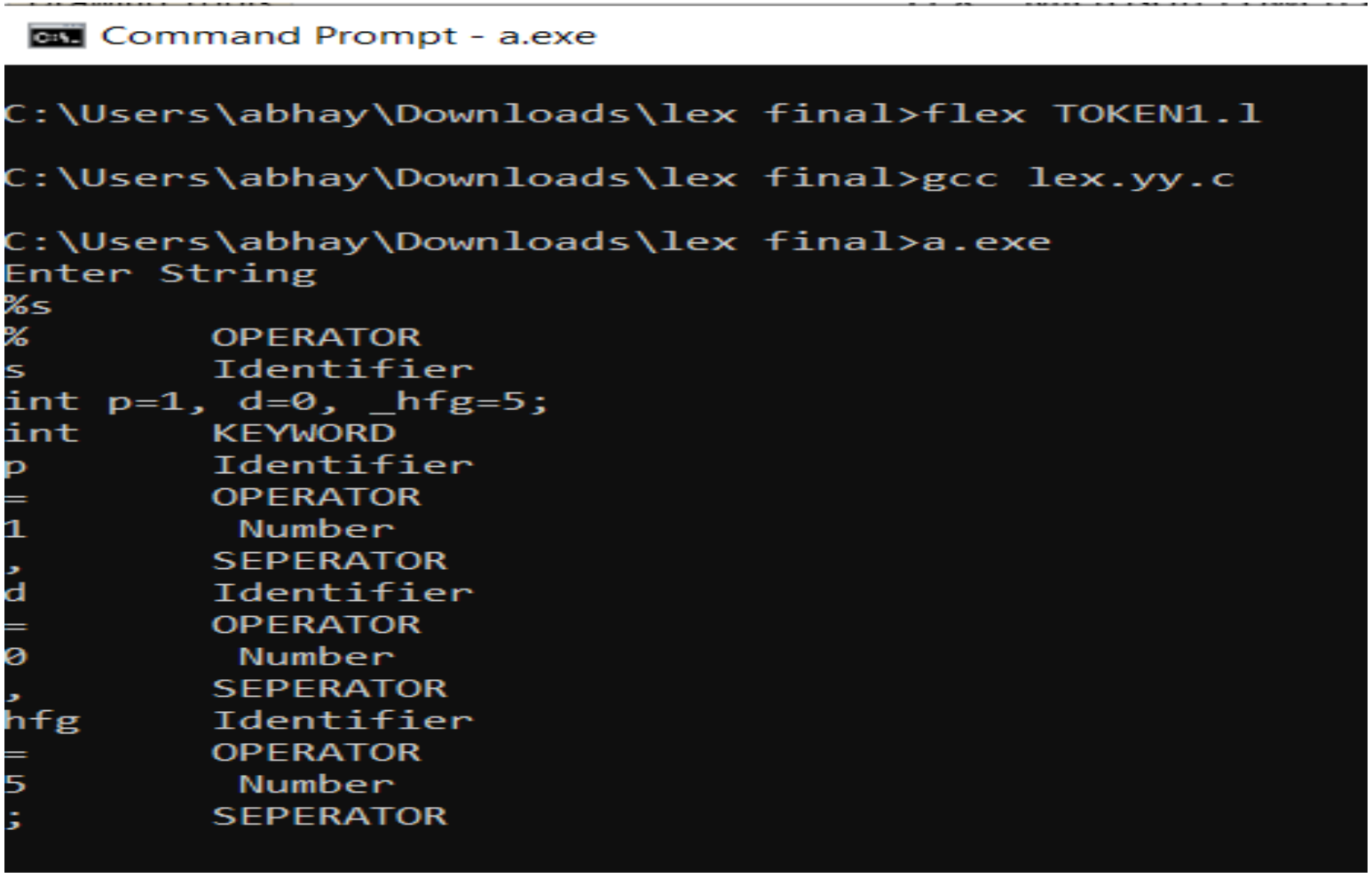
Flex, A fast scanner generator

- flex is a tool for generating tokens
- It read the given input files.
- Flex generates as output a C source file, ‘lex.yy.c’, which defines a routine ‘yylex()’.
- This file is compiled and linked with the ‘-lfl’ library to produce an executable.
- When the executable is run, it analyzes its input for occurences of the regular expressions.
- Whenever it finds one, it executes the corresponding C code.
- Flex is a free and open-source software alternative to lex. It generates lexical analyzers.


Command Prompt
C:\Users\abhay\Downloads\lex final>flex vowelcount.l
C:\Users\abhay\Downloads\lex final>gcc lex.yy.c
C:\Users\abhay\Downloads\lex final>a.exe
Enter String
Abhay Deep Seth
vowel=5 and Consonent=8
C:\Users\abhay\Downloads\lex final>

```
%{
#include<stdio.h>
%}

%%
"auto"|"double"|"int"|"struct"|"break"|"else"|"long"|"switch"|"case"|"enum"|"register"|"typedef"|"char"|"extern"|"re
turn"|"union"|"continue"|"for"|"signed"|"void"|"do"|"if"|"static"|"while"|"default"|"goto"|"sizeof"|"volatile"|"const"|"
float"|"short"|"printf" {printf("%s\tKEYWORD\n",yytext);}
"{"|"}"|";"|","|"("|")"   {printf("%s\tSEPERATOR\n",yytext);}
[0-9]* {printf("%s\t Number \n",yytext);}
"+"|"-"|"/"|"="|"*"|"%"   {printf("%s\tOPERATOR\n",yytext);}
[a-zA-Z][0-9]+|[a-zA-Z]* {printf("%s\tIdentifier\n",yytext);}
.|\n  ;
%%
/*call the yywrap function*/
int yywrap()
{
return 1;
}
/*Auxiliary function*/
/*Driver function*/
int main(void)
{
/*call the yylex function.*/
printf("Enter String \n");
yylex();
return 0;
}
```


Command Prompt - a.exe
C:\Users\abhay\Downloads\lex final>flex TOKEN1.l
C:\Users\abhay\Downloads\lex final>gcc lex.yy.c
C:\Users\abhay\Downloads\lex final>a.exe
Enter String
%s
% OPERATOR
s Identifier
int p=1, d=0, _hfg=5;
int KEYWORD
p Identifier
= OPERATOR
1 Number
, SEPERATOR
d Identifier
= OPERATOR
0 Number
, SEPERATOR
hfg Identifier
= OPERATOR
5 Number
; SEPERATOR

THANKS