

Lexical Analysis

Tuesday, March 29, 2022 9:28 PM

Lexical Analysis (Scanner)

It scans the stream of characters making up the source program is read from left to right and grouped into tokens, that are sequence of characters having collective meaning. Tokens are of type Keywords, Identifiers, Operators, Constants and Special characters.

Ex:- Total = Sub1 + Sub2 * 2;

Identifier: Total, Sub1, Sub2

Operator: =, +, *

Constant: 2

Special character: ;

Symbol table		
< Total	float	garbage >
< Sub1	float	0.0 >
< Sub2	float	garbage >

Basic functions of lexical Analysis phase

1. Remove the white spaces (tab, newline) and comments.
2. Identify the tokens.
3. Creating storage for identifiers in the symbol table.

Q:- find the number of tokens in the following program code?

① main()
1 2 3
↑ // first program ignored
printf ("Hello Students");
5 6 7 8 9
} 10
Tokens = 10

② While (i > 0)
1 2 3 4 5 6
↑ 7 printf (" %d ", i);
8 9 10 11 12 13 14
} 15 16 17
Tokens = 18

Gate-2000
The number of tokens in the following 'C' statement.

Printf (" i=%d, &i=%u ", i, &i);
1 2 3 4 5 6 7 8 9 10
a) 3 b) 26 c) 10 d) 21.

Q:- which of the following can be recognized as token without depending upon the next input character.

~~a) int~~ ~~b) total~~ ~~c) ++~~ ~~d) all of a,b,c~~

Char integer;
██████████

Types of Lexical errors :-

- i) Unterminated Comments

Absent
x/

vi - i

i) Unterminated Comments

/* -----
beginning

Absent
*/
end

ii) Nested Comment

/* ----- /* ----- */ ----- */
beginning ----- end ----- end

3. Length of the longest identifier:- (<=31 characters)

int interest_and_compound_interest_on_principle_amount;

4. Invalid Identifier:-

int 1stmonthSalary;

5. Illegal Symbol :- int salary = ₹50000;

6. Invalid Constant - int salary = 50,000;

Error Recovery in Lexical Analysis

1. Delete : an extra character. Ex:- int Salary = ₹50000; remove.

2. Insert : missing characters. Ex:- /* ----- */ insert.

3. Transpose :- interchanging of two adjacent characters. Ex:- int 1stSalary ; swap

4. Replace : replacing a character with another one. Ex:- int F1stSalary; replace

"LEX" Tool (FLEX) (JFLEX)

→ Lex is a tool to specify lexical analyzer.

→ Lex tool is called Lex Compiler.

Lex program

- It has 3 parts
1. declaration
 2. Translation rules
 3. Auxiliary procedures

Lex.l → LEX Compiler → lex.yy.c

Tabular representation of Transition diagram Constructed for the regular expression and Standard procedures that use the table to recognize the tokens.

lex.yy.c → C Compiler → a.out lexical Analyzer.

Source prog → a.out → Tokens.

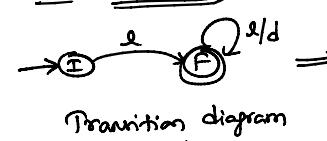
Lex.l

```

ID { letter(letter/digit)* }
% %
{ ID } { yyval=install_id(); return (ID) }
% %
= install_id()
{ //to store identifiers into the symbol table.
}

```

Ex: ID: l(l/d)*



	l	d
I	F	∅
F	F	F

Transition table

