



CS & IT ENGINEERING

Compiler Design

Lexical Analysis & Syntax Analysis



Lecture No. 3



By- DEVA Sir

TOPICS TO BE COVERED

Tokens

Find Tokens

Lexical Errors

- I) **Keywords** : ① int, float, double, char, long, short, void, signed, unsigned, volatile, const, struct, union, enum, typedef ② if, else, switch, case, default, while, do, for, break, continue, return, goto ④ sizeof()

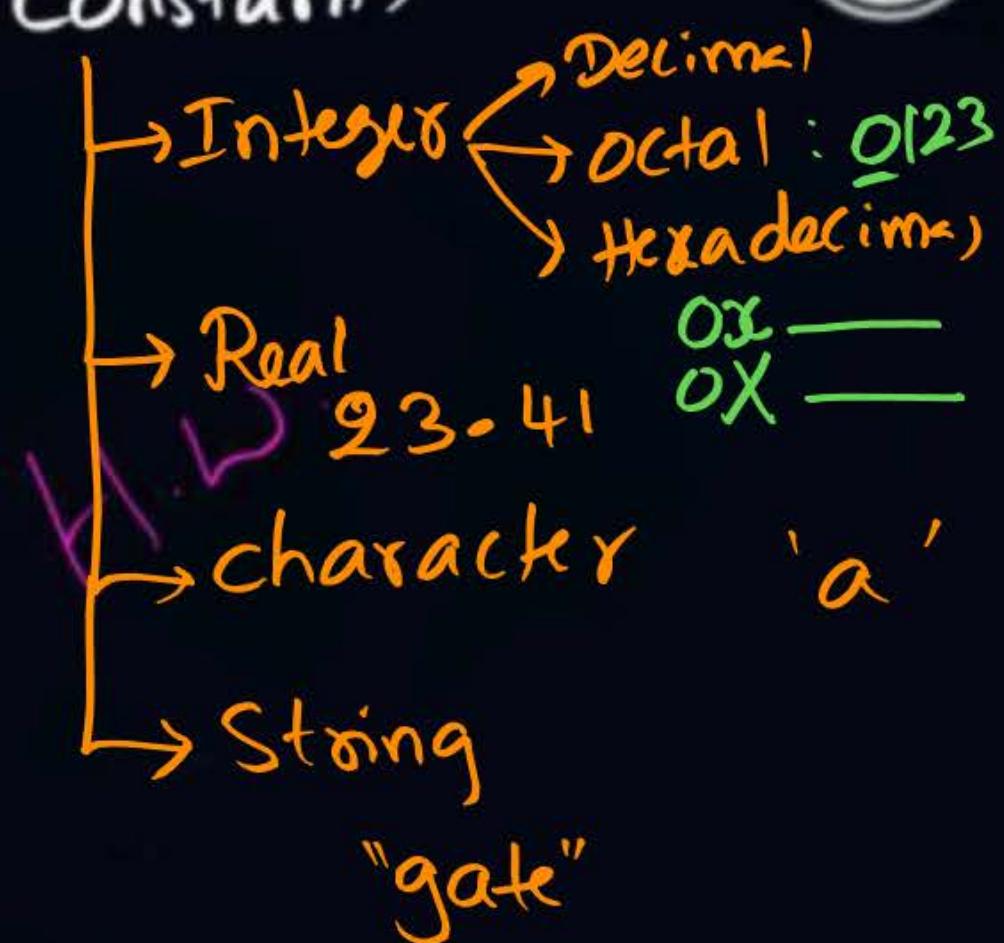
II) Operators

Unary: !, ~, +, -, ++, --, *, &, . . .	Assignment: =, +=, -=, *=, ...
Arithmetic: +, -, *, /, %	
Relational: <, <=, >, >=, ==, !=	
Bitwise: &, , ^, <<, >>	
Logical: &&,	

III) Identifiers

- Name given to variable | function
- comprises letters, digits, underscore
- Should not start with digit
- Should not be a keyword

IV) Constants



V) Special Tokens

; { } () : , .

Find the TOKENS in the following codes.

①

int x ;

,,3

int → Keyword
x → Identifier
; → Special TOKEN

②

int float ;

No lexical error
 But Syntax error

int ✓
float ✓
; ✓ = 3

③

int gate123 ; = 3

④

int | 123gate ;

✓

Invalid sequence

Lexical Error

⑤

int x = "gate" ; = 5

String Constant

⑥

int x = 025 ; = 5

Octal

Octal: 0 to 7

⑦

int x = 029 ;

Invalid sequence

Lexical
 Error

⑧

int x = 0xabc ; = 5 tokens

Hexadecimal

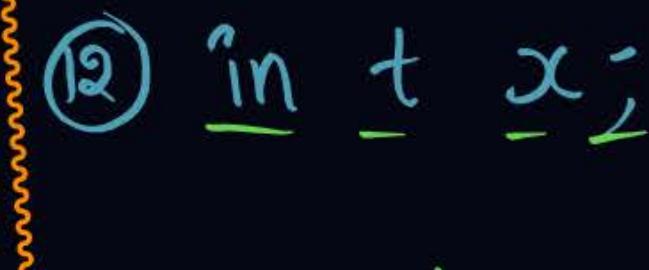
⑨

`int 23=45 ;`

→ 5 tokens
No lexical error
But syntax error }
Compilation Error

⑩ 
= 3

⑪ 
= 4

⑫ 
= 4

(13)

 $x = \underline{+} \underline{y};$
 $\Rightarrow 5$

(14)

 $x = \underline{+} \underline{y};$
 $\Rightarrow 5$

(15)

 $x \boxed{+} \underline{=} \underline{y};$
 $\Rightarrow 4$

(16)

 $x \boxed{=} \boxed{x} \underline{+} \underline{+} \underline{y};$
 $\Rightarrow 7$

(17)

 $x = \underline{*} \underline{*} \underline{y};$
 $\Rightarrow 6$

(18)

 $x = y \boxed{/ * \text{comment}};$

begin → search for end rule
 is missing

 Lexical
Error

(19)

 $x = y \boxed{/ * \text{comment}} / * \text{comment} * \boxed{/};$

= 4 tokens
 ignore
 begin → search for end rule

(20)

 $x = y \underline{\text{comment}} * \boxed{|};$
 \Rightarrow

Note:

How many tokens ?

P
W

① ++ → 1

② ** → 2

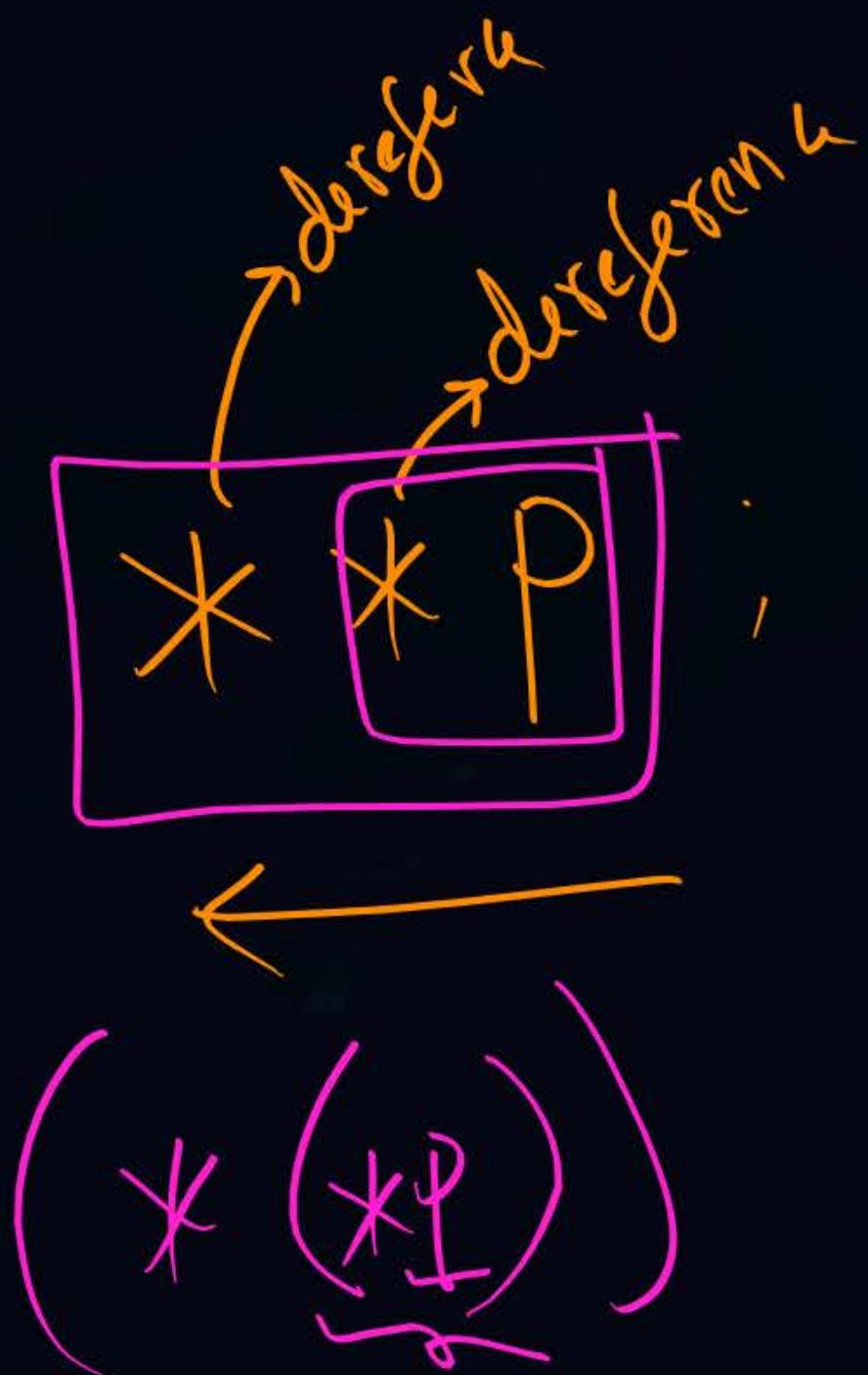
③ *** → 3

④ += → 1

⑤ =+ → 2

⑥ <= → 1

⑦ != → 1



+ =
- =
* =
/ =
% =
< =
> =
&& =
|| =
& =
| =

(21)

 $x = A[20];$ X

(22)

 $x = \text{fun}(2, 3);$ = 9

(23)

 $x = \text{printf} ("gate = %d", \text{rank});$ = 9

(24)

 $x = \text{scanf} ("%d%d", x, y);$ = 11

(25)

 $x = y$
- - -
 $= 3$ No lexical Err
(Syntax Err)

(26)

 $x = 'a';$
 $= 4$ begin
,end

(27)

 $x = '\backslash n';$
 $= 4$

(28)

 $x = \boxed{abc};$
begin

lexical err

(29)

 $x = "abc";$
 $= 4$

(30)

 $x = y + + + + z ;$

- - - - -
= 8 tokens

③1

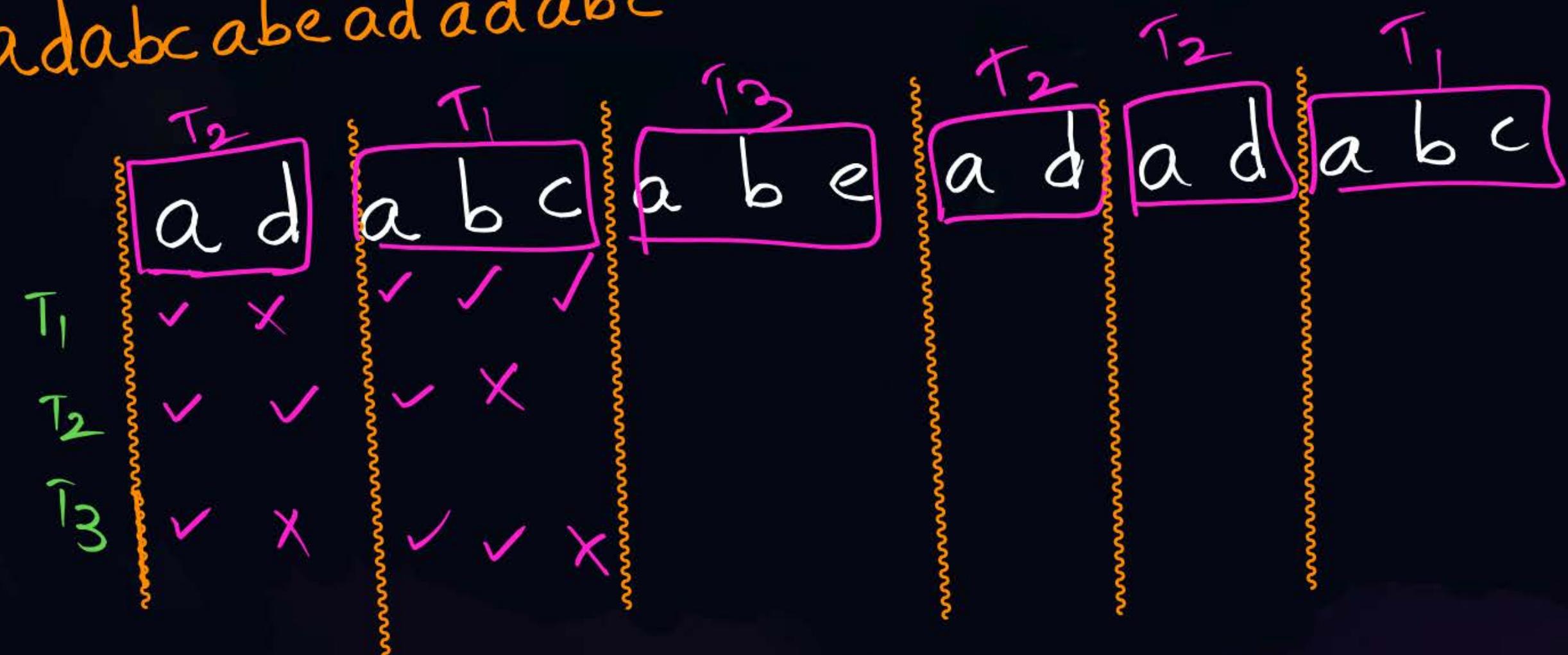
$$T_1 = abc$$

$$T_2 = ad$$

$$T_3 = abe$$

$$O/P = T_2 \bar{T}_1 T_3 \bar{T}_2 T_2 \bar{T}_1$$

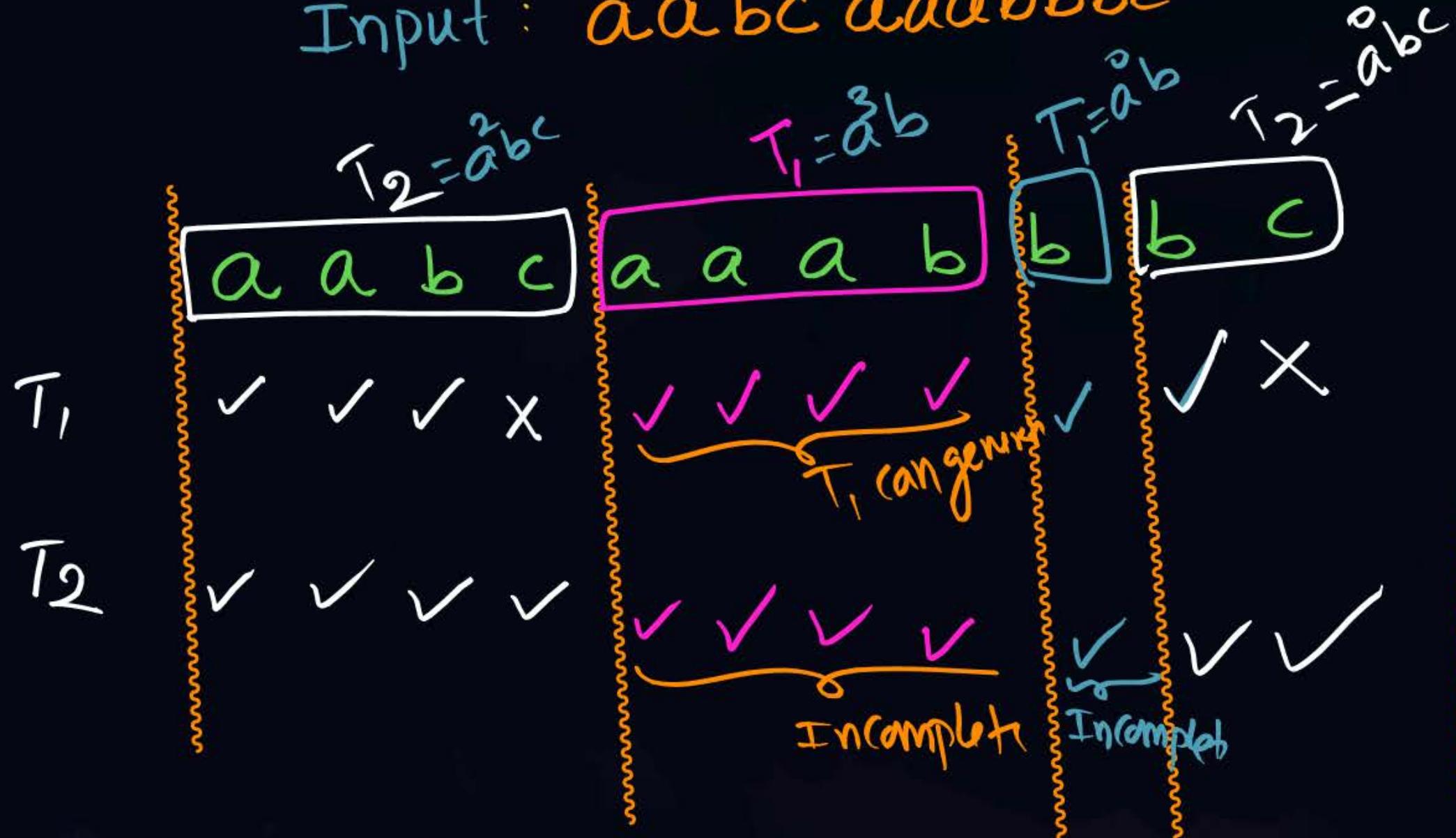
Input: adabcabeadadabc



(32)

$$\left. \begin{array}{l} T_1 = a^* b \\ T_2 = a^* b c \end{array} \right\} \text{Output} = T_2 T_1 T_1 T_2$$

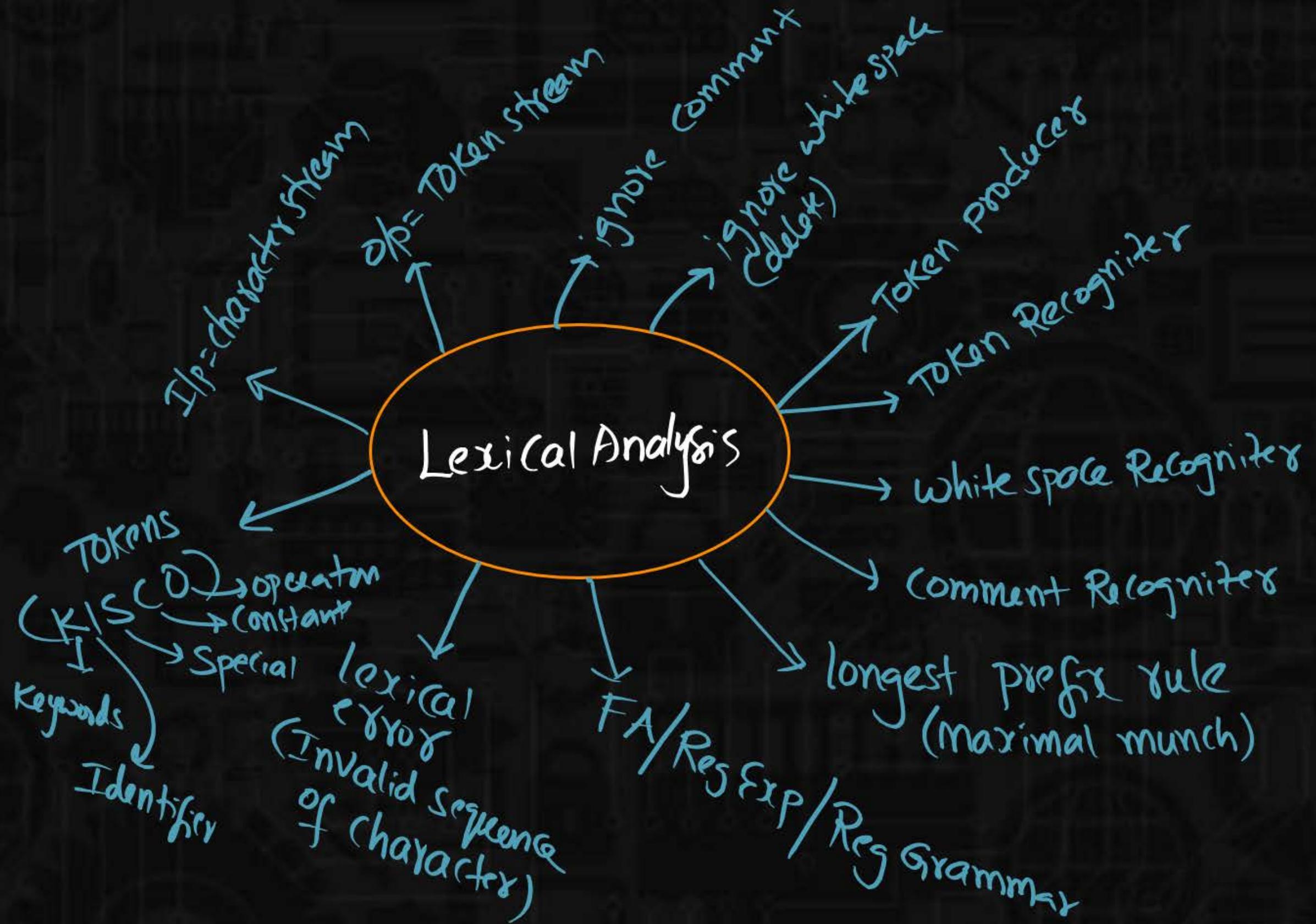
Input : aabc aaabbcc



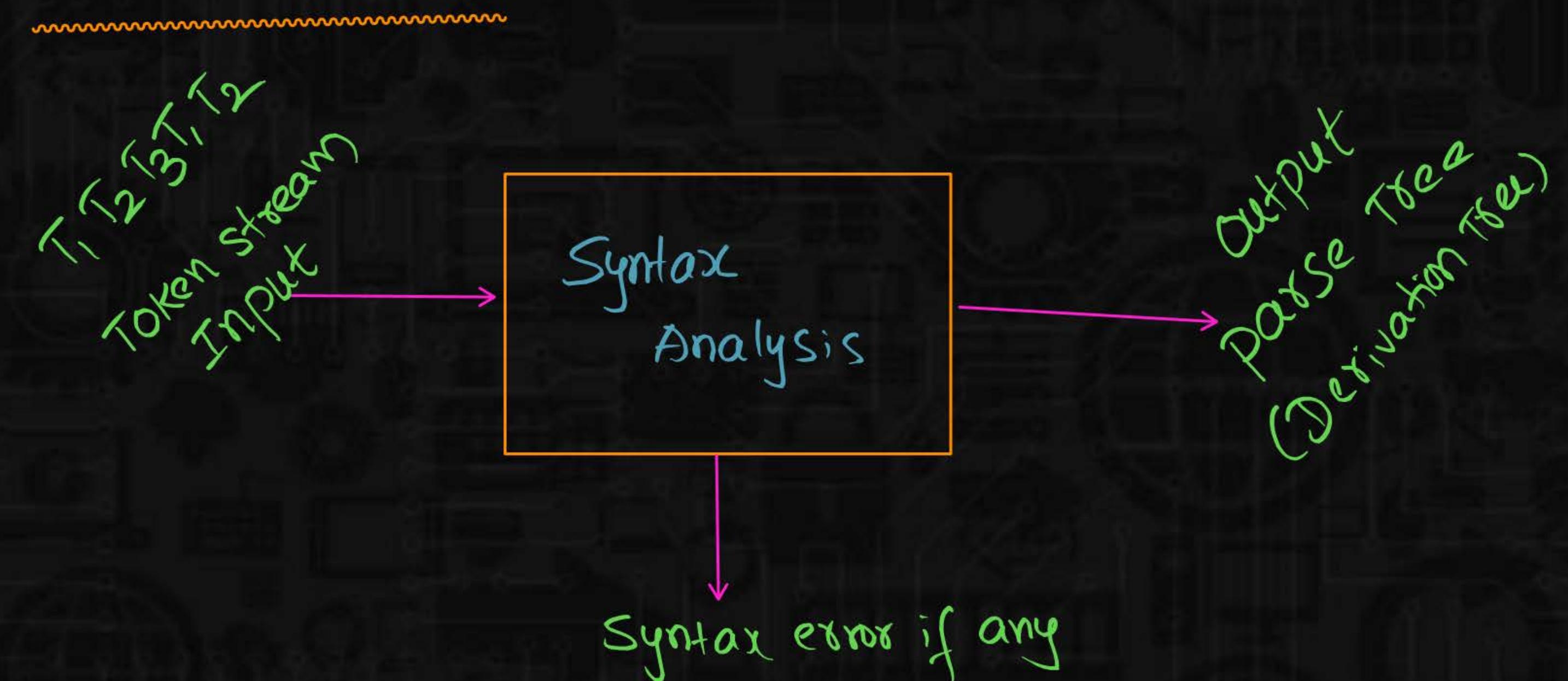
$$\begin{aligned}
 a^* &\rightarrow a^0 = \epsilon \\
 a^* &\rightarrow a^1 = a \\
 a^* &\rightarrow a^2 = aa \\
 a^* &\rightarrow a^3 = aaa \\
 a^* &\rightarrow a^4 = aaaa \\
 &\vdots
 \end{aligned}$$

$\overset{*}{\overbrace{a}}$
 Zero or more 'a's

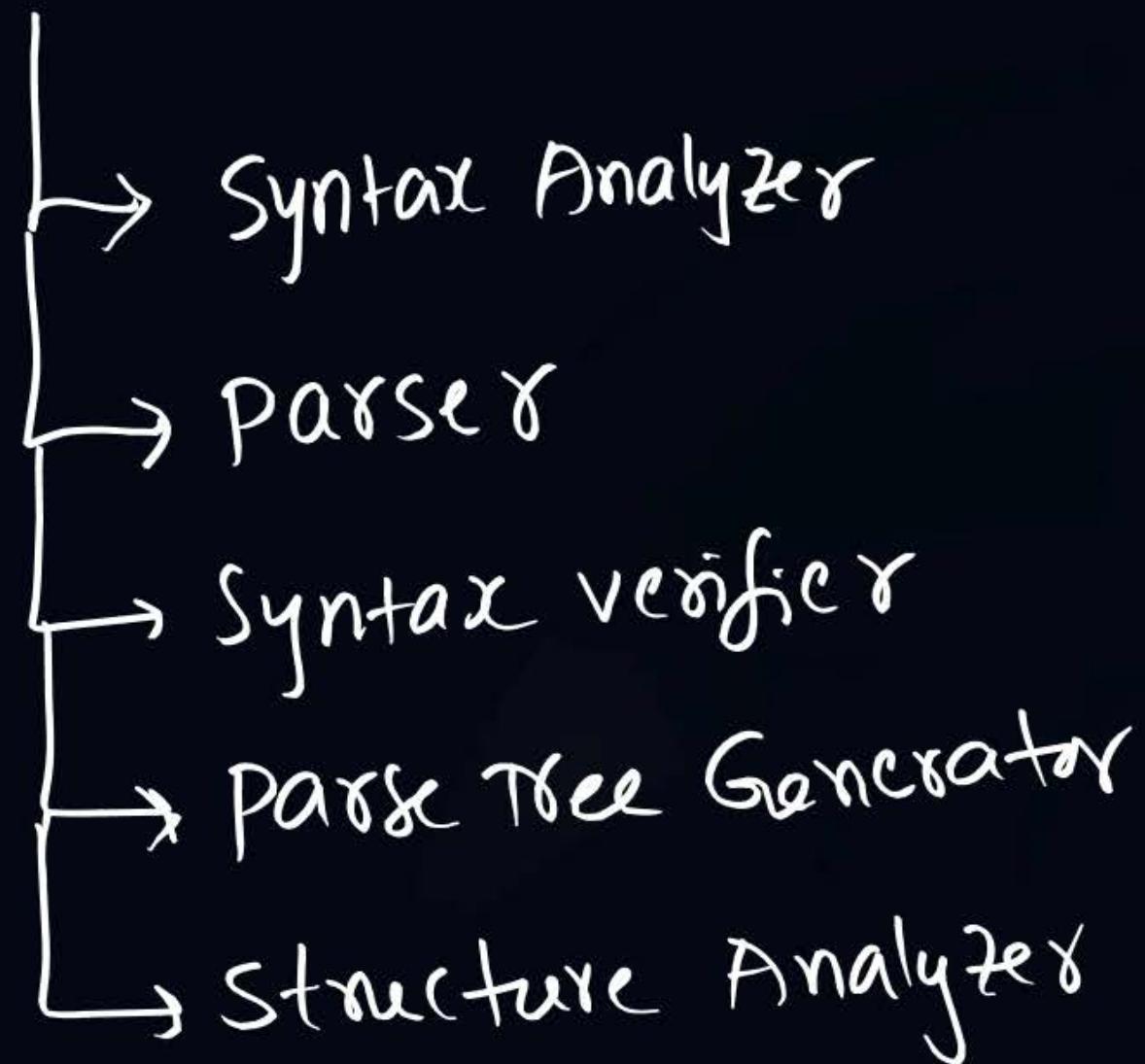
P
W



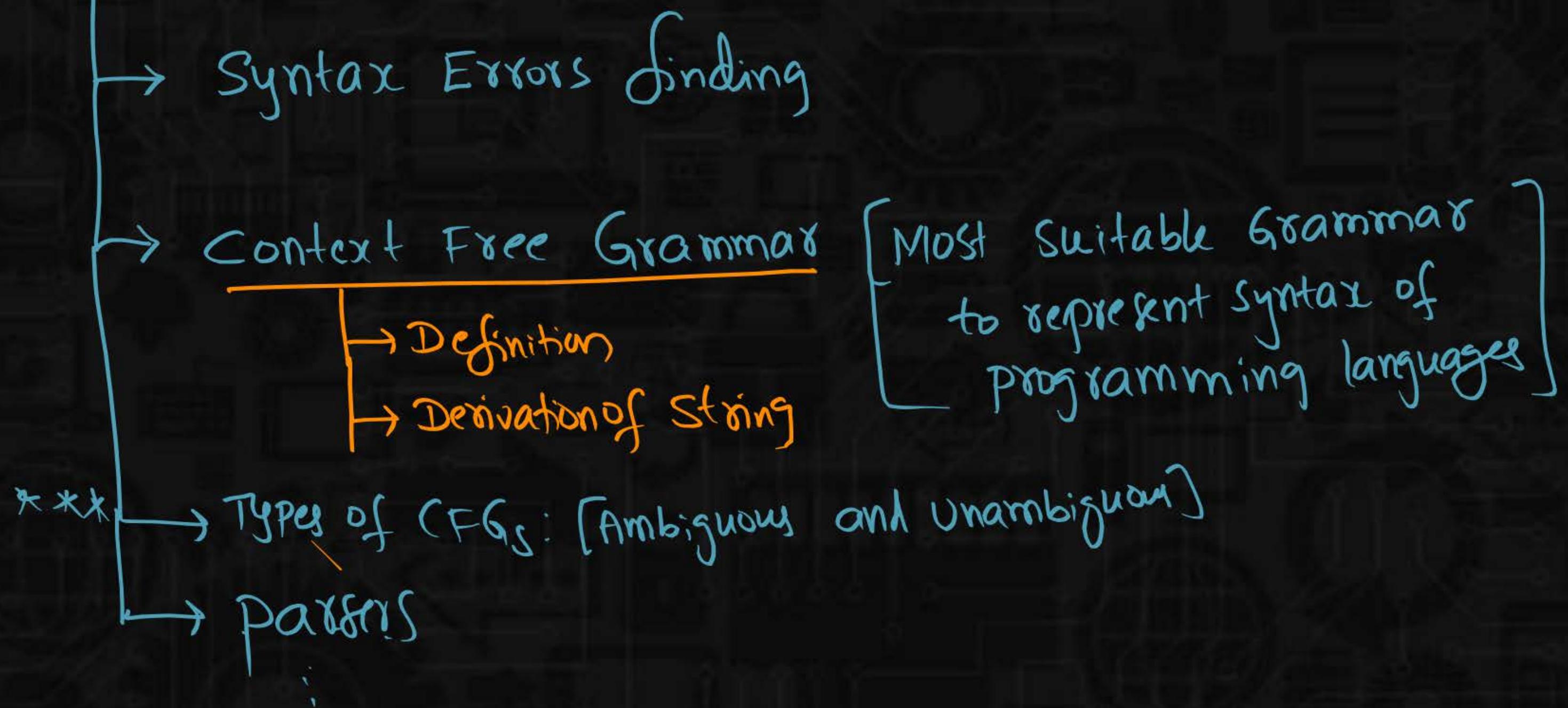
Syntax Analysis :



Syntax Analysis



Syntax Analysis :



Find the Syntax error in following codes.

①

```
void main()
{
    int x;
}
```

No error

②

```
void main()
{
    int x,y;
}
```

No error

③

```
int x=y;
```

No syntax Err
Semantic Err
y not declared

④

```
int x, int y;
```

Syntax Error

⑤

```
int x+y;
```

Syntax Error

⑥

```
int y=2;
int x=y;
```

No error

⑦

```
int x,y;
```

```
x=y=2;
```

No error

⑧

```
int x,y;
```

```
x=2, y=2;
```

No error

$x = 2, y = 2;$

1

$(x = 2), (y = 2);$

1

$x = 2;$

$y = 2;$

= (Highest)

, (lowest
precedence)

⑨
No Error

int x ; int y ;

No Syntax Err

⑩
if(2);
 ^
 store

No error

⑪
if();
 ^
 expression missing

Syntax Err

⑫
x y ;
 ^
 No Syntax Err
 (Semantic Err)

⑬
whik(2);

No compilation Err

⑭
while();

 ^
 expression is missed
Syntax Err

⑮
for();
 ^
 It is not correct syntax

Syntax Err

⑯
for(; ;);

No compilation Err

(17)

```
typedef int x;
```

```
x y;
```

meaning
int

No error



```
typedef int Integer;
```

```
Integer y;
```

⑯ for(1,2,3);

⑰ f₈₀(1,2,3);

⑯ for(2;2;2);

⑰ f₈₀(2;2;2);

H.W.

MCQ

Consider the following ANSI C program:

```
int main() {  
    Integer x;  
    Return 0;  
}
```

Which one of the following phases in a seven-phase C complier will throw an error?

[GATE-2021-CS: 1M]

- Lexical analyzer
- Syntax analyzer
- Semantic analyzer
- Machine dependent optimizer

MCQ

Consider the following statements:

- I. Symbol table is accessed only during lexical analysis and syntax analysis.
- II. Compilers for programming languages that support recursion necessarily need heap storage for memory allocation in the run-time environment.
- III. Errors violating the condition 'any variable must be declared before its use' are detected during syntax analysis.

Which of the above statements is/are TRUE?

[GATE-2020-CS: 1M]

- I only
- I and III only
- II only
- None of I, II and III

MCQ

A lexical analyzer uses the following patterns to recognize three tokens T_1, T_2 and T_3 over the alphabet $\{a, b, c\}$.

$$T_1: a? (b|c)^* a$$

$$T_2: b? (a|c)^* b$$

$$T_3: c? (b|a)^* c$$

Note that 'x?' means 0 or 1 occurrence of the symbol x. Note also that the analyzer outputs the token that matches the longest possible prefix.

If the string bbaacabc is processed by the analyzer, which one of the following is the sequence of tokens it outputs?

[GATE-2018-CS: 2M]

- T₁ T₂ T₃
- T₁ T₁ T₃
- T₂ T₁ T₃
- T₃ T₃

MCQ

Match the following according to input from List-I to the compiler phase in the List-II that processes it:

List-I

- (P) Syntax tree
- (Q) Character stream
- (R) Intermediate representation
- (S) Token stream

List-II

- (i) Code generator
- (ii) Syntax analyser
- (iii) Semantic analyser
- (iv) Lexical analyser [GATE-2017-CS: 1M]

P → (ii), Q → (iii), R → (iv), S → (i)

P → (ii), Q → (i), R → (iii), S → (iv)

P → (iii), Q → (iv), R → (i), S → (ii)

P → (i), Q → (iv), R → (ii), S → (iii)

MCQ

Match the following

List-I

P: Lexical analysis

Q: Top down parsing

R: Semantic analysis

S: Runtime environment

List-II

(i) Leftmost derivation

(ii) Type checking

(iii) Regular expressions

(iv) Activation records

[GATE-2016-CS: 1M]

P→(i), Q→(ii), R→(iv), S→(iii)

P→(iii), Q→(i), R→(ii), S→(iv)

P→(ii), Q→(iii), R→(i), S→(iv)

P→(iv), Q→(i), R→(ii), S→(iii)

MCQ

In a compiler, keywords of a language are recognized during

[GATE-2021-CS: 1M]

- parsing of the program
- the code generation
- the lexical analysis of the program
- dataflow analysis

MCQ

Which data structure in a compiler is used for managing information about variables and their attributes?

[GATE-2021-CS: 1M]

- Abstract syntax tree
- Symbol table
- Semantic stack
- Parse table

→ LA ✓

Next : Syntax Analysis :

