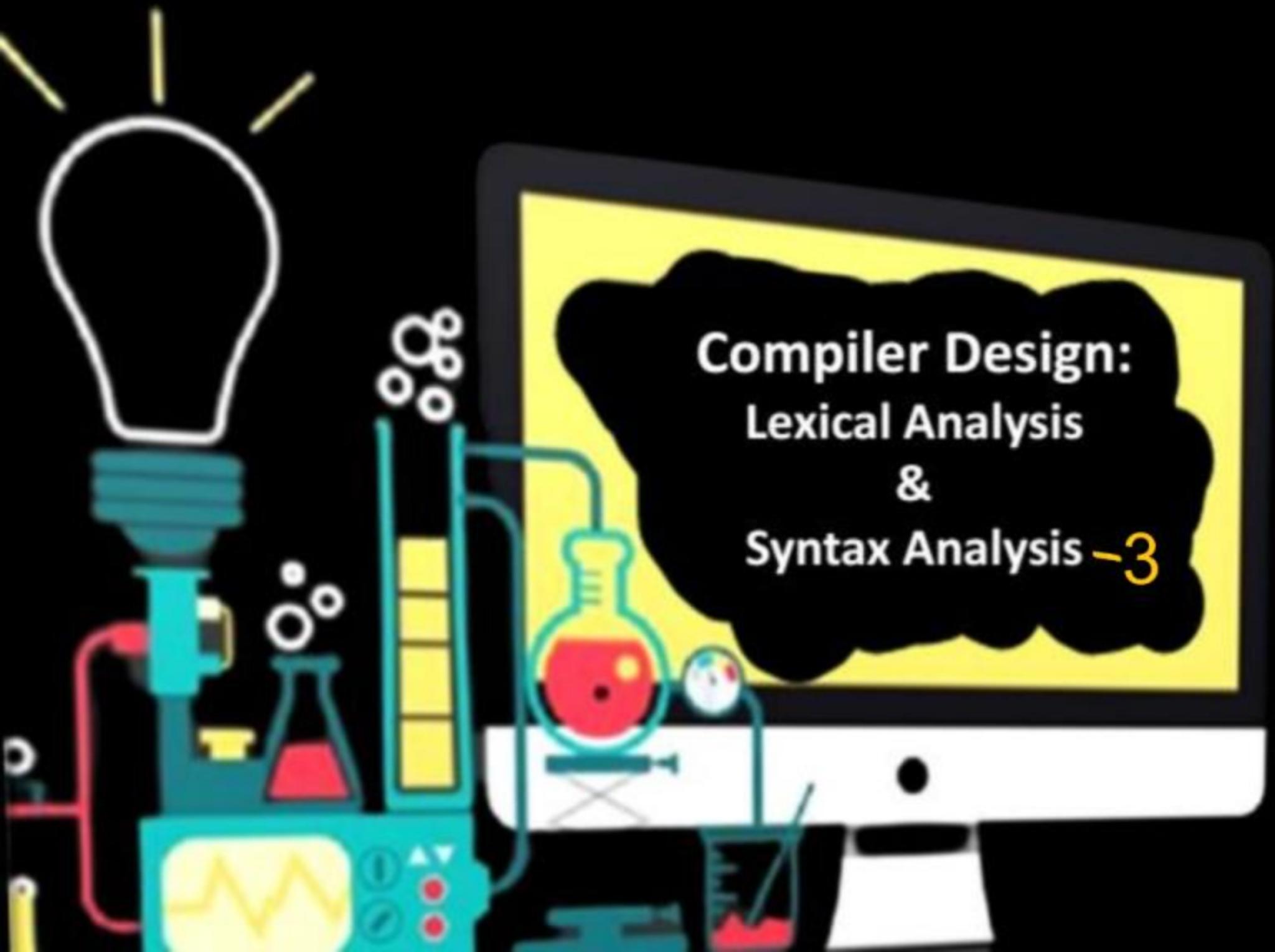


# CS & IT Engineering



Deva sir

## Topics to be covered:

- Lexical Analysis 
  - practice
  - GATE PYQ's
- Syntax Analysis

# Lexical Analysis

10  
= 13 tokens

```
void main( )  
{  
    printf("x=%d", x);  
}
```

⇒ Semantic Error

Find no. of tokens.

11 tokens

```
x = a+++++=**b;
```

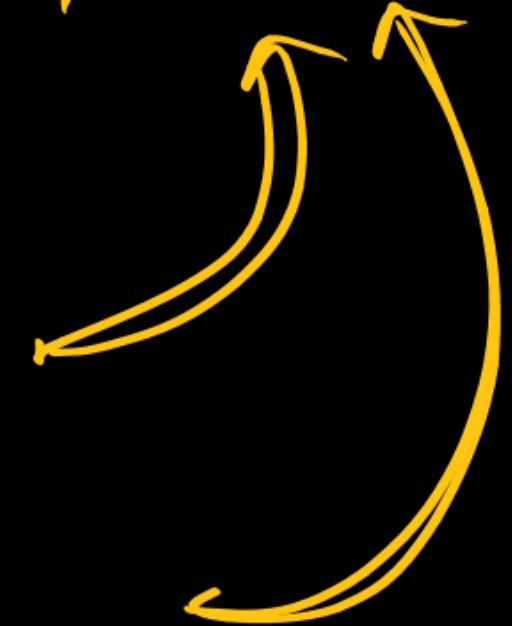
12

```
int **p; → 5 tokens
```

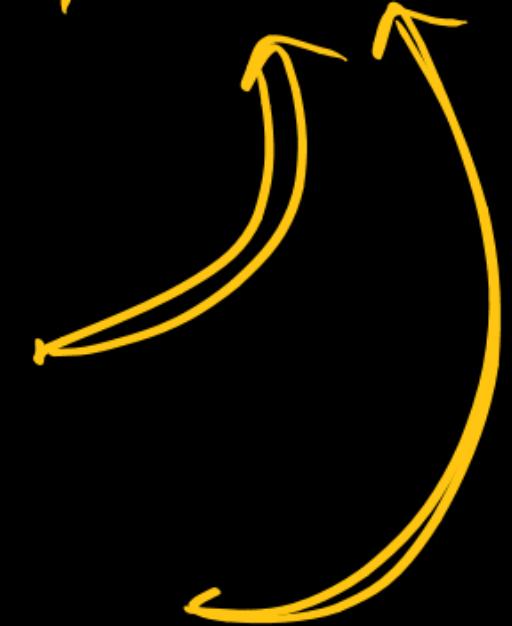
13

```
x++; → 3 tokens
```

14

int 12xy;  $\Rightarrow$  lexical error

15

int x = 029;

16

int x = 0XA<sup>x</sup>X23;

17

in t x ;  $\Rightarrow$  4 tokens

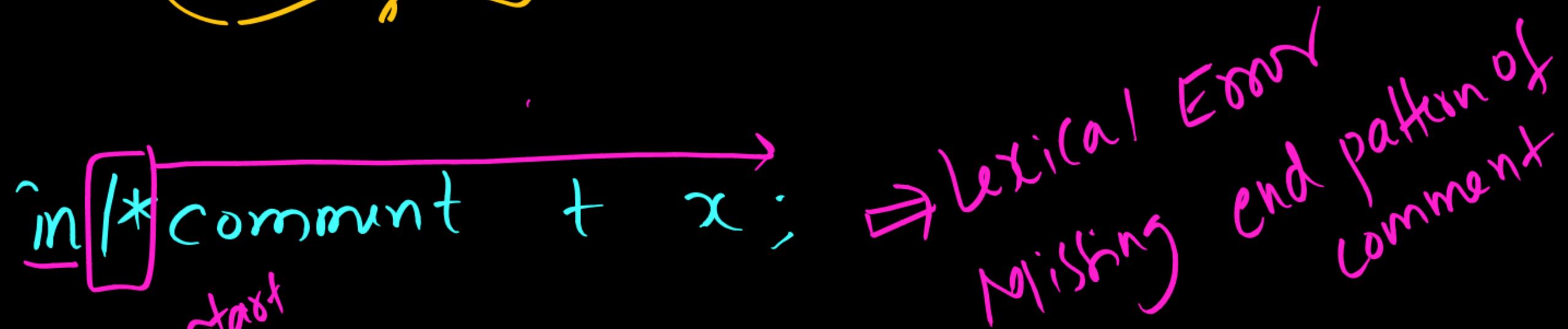
18

intx = ;  $\Rightarrow$  2 tokens

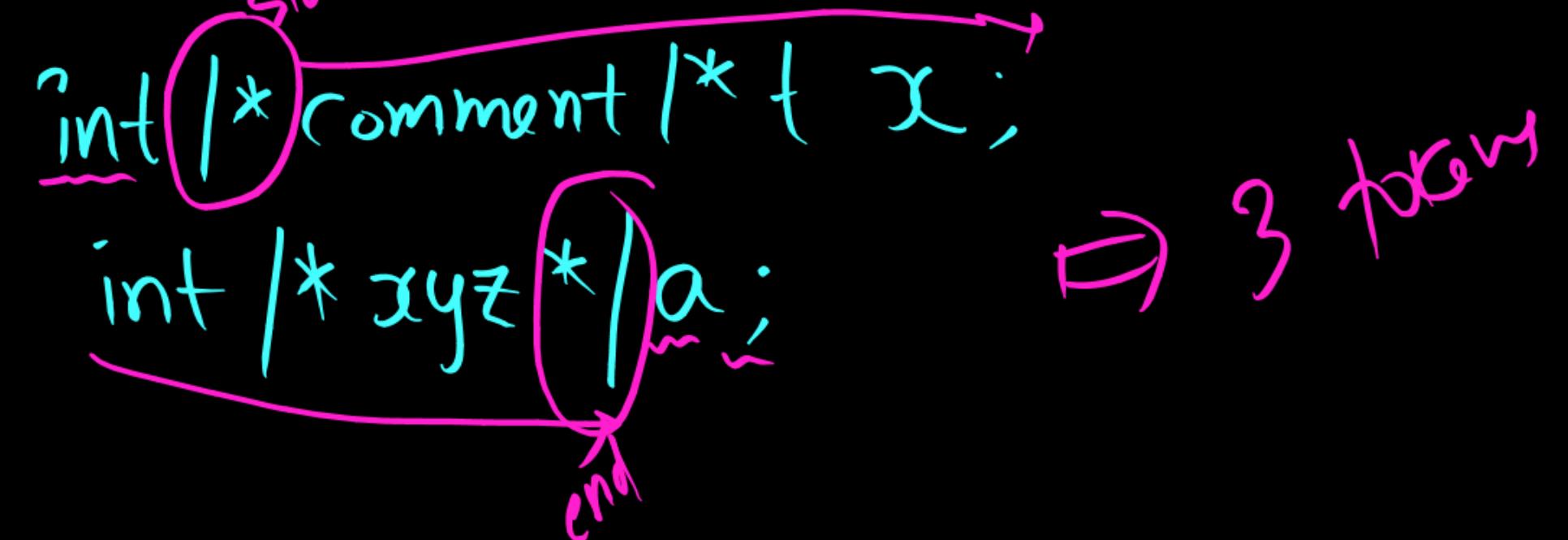
19



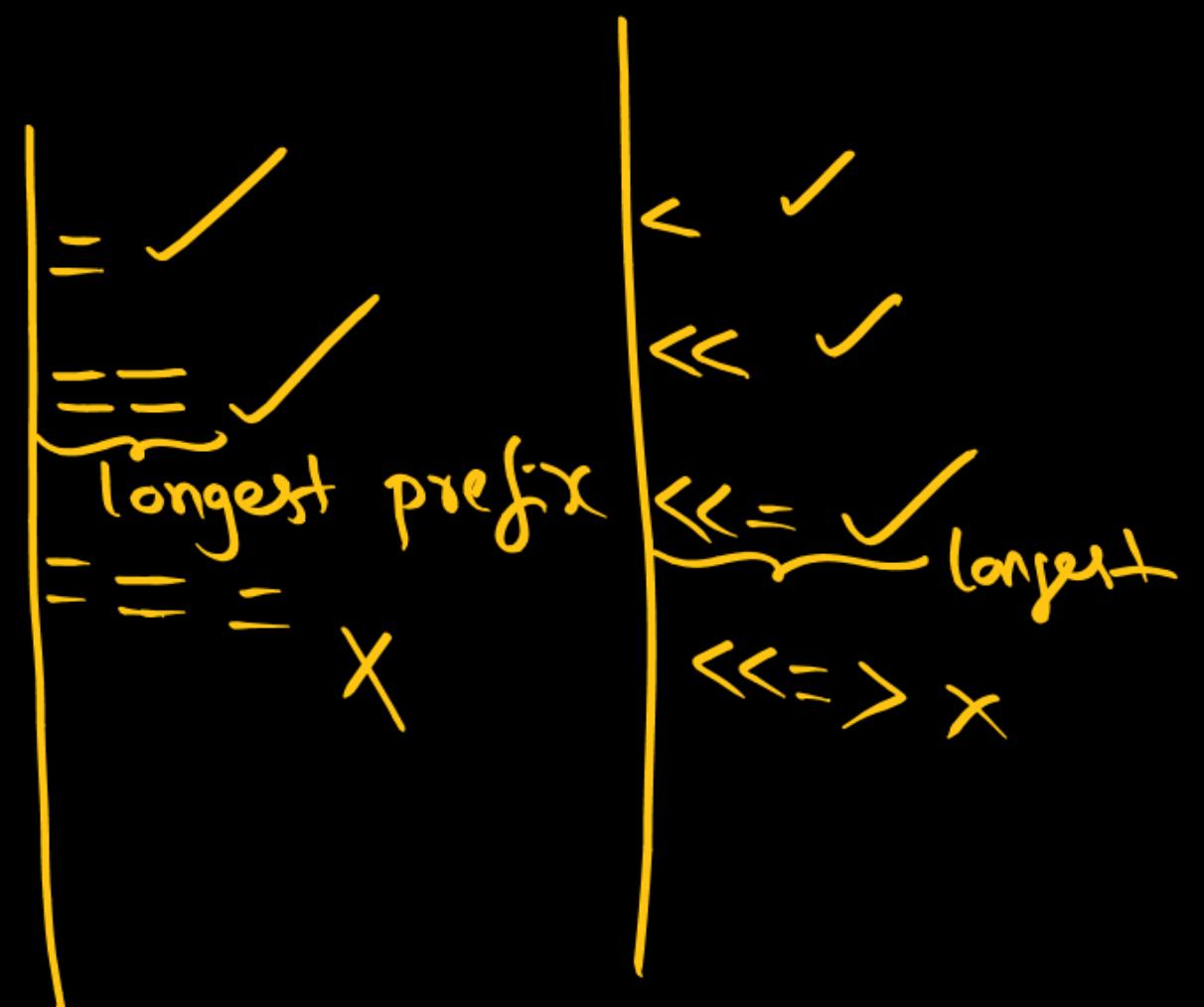
20



21



22

 $x ==+<= > == y ; \Rightarrow 9 \text{ tokens}$ 

23

6-2018

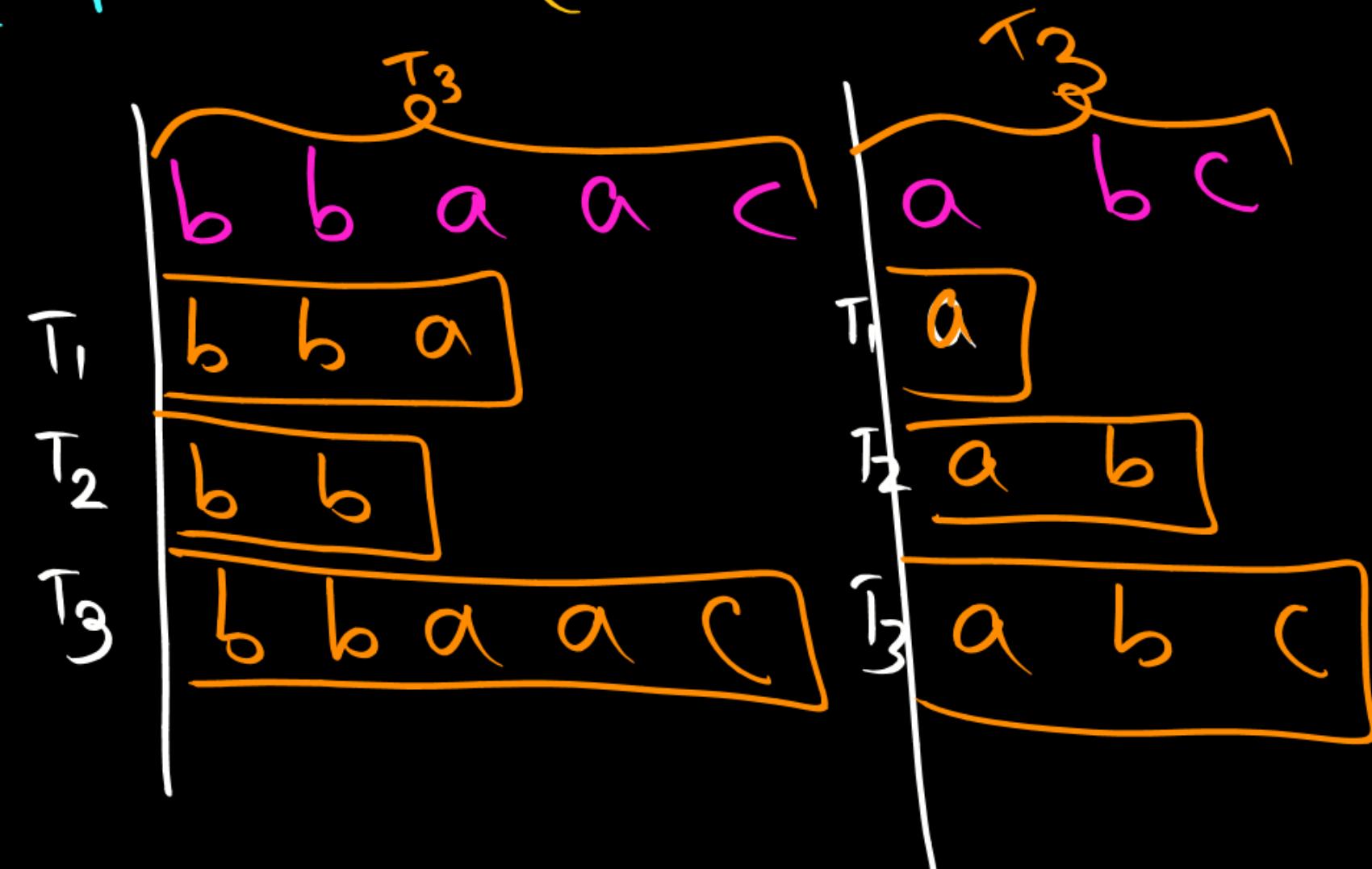
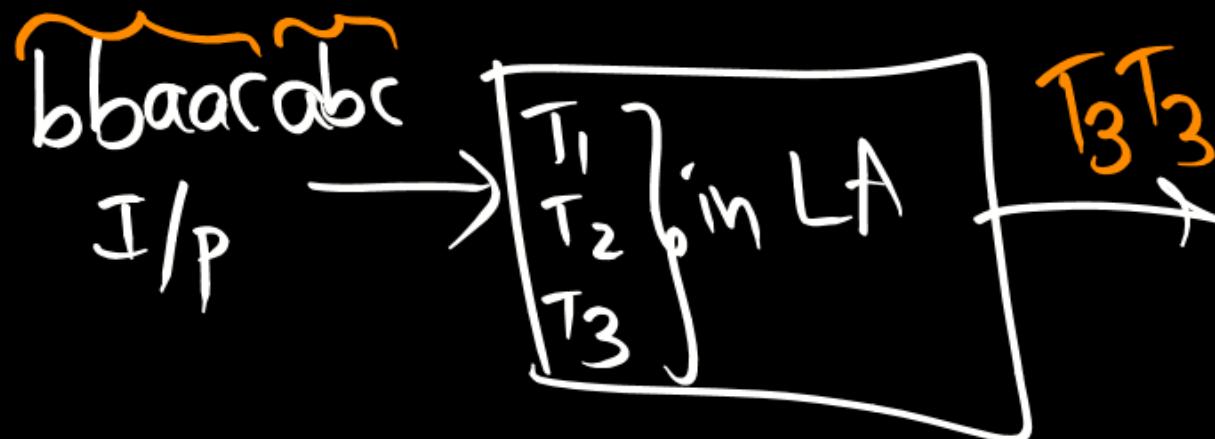
PW

$$T_1 : a ? (b|c)^* a = (\varepsilon + a)(b+c)^* a$$

$$T_2 : b ? (a|c)^* b = (\varepsilon + b)(a+c)^* b$$

$$T_3 : c ? (b|a)^* c = (\varepsilon + c)(b+a)^* c$$

Input : bbaaacabc



24

P  
W

$$T_1 = a^*bc$$

$$T_2 = b^*ac$$

Input :  $\frac{aabcb}{T_1} \frac{bc}{T_1} \frac{bbac}{T_2}$

O/P :  $T_1 T_1 T_2$

# How to design Lexical Analyzer?

Problem : Identifier

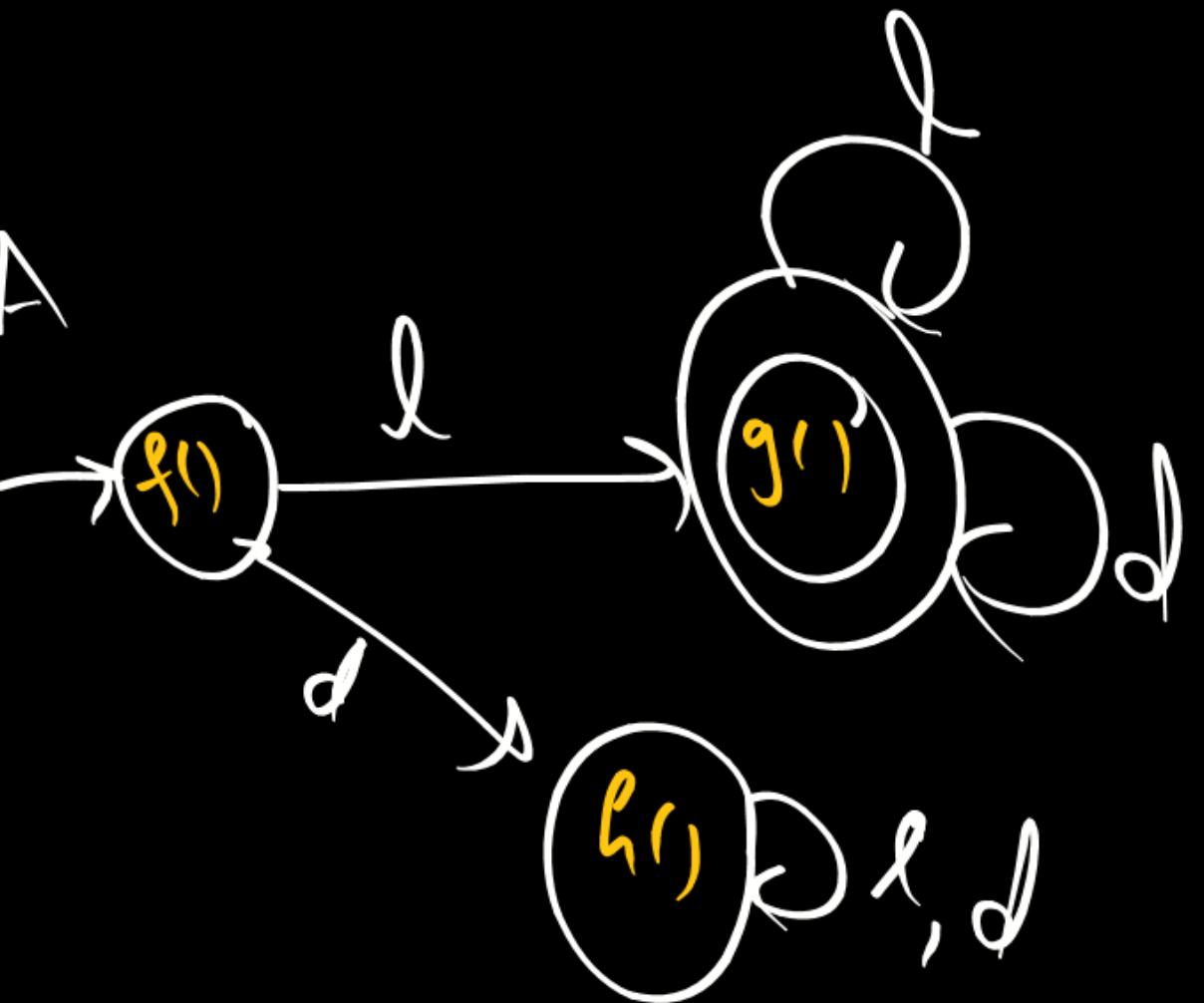
```
char ch;
f( )
{
    ch=getchar();
    if(ch=='l')
        g();
    if(ch=='d')
        h();
    if(ch=='newline')
        STOP
}
```

Step 1 : Write a <sup>(RegEx)</sup> pattern for Identifier

$$l(l+d)^*$$

Step 2 : construct DFA

Step 3 : Write a program  
(Implement LA)



Which of the following is used in Lexical Analyzer?

- A) Regular Expression
- B) Finite Automata
- C) Regular Grammar
- D) None

Revision:

Character stream

Token stream

Lexical Error

Longest prefix Rule

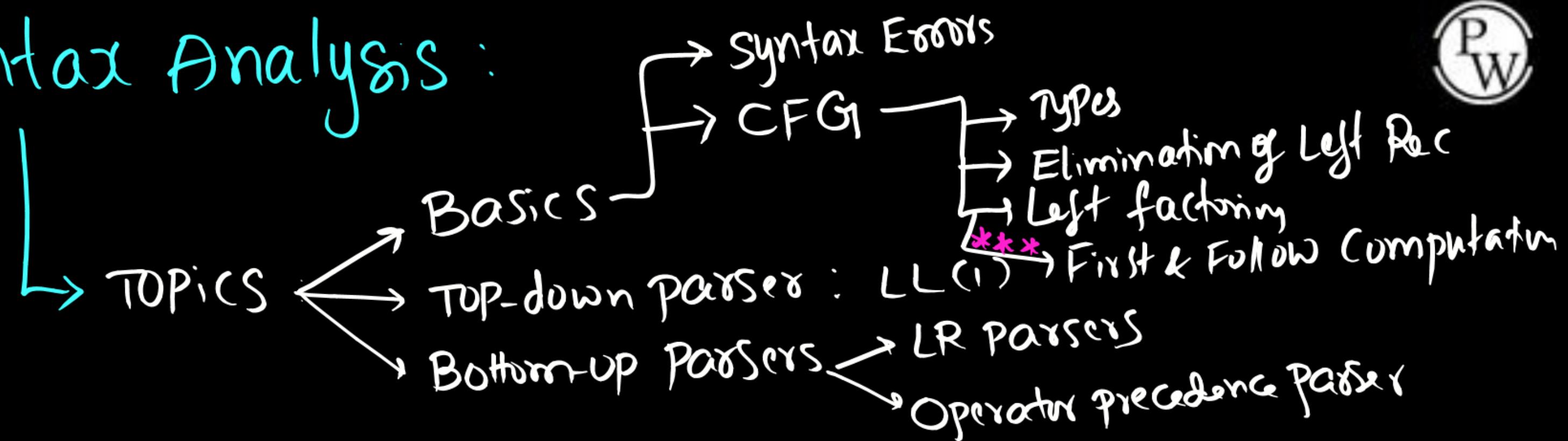
Token produces

Token, comment, whitespace

Regular Expression

FA

# Syntax Analysis :



# Find Syntax Errors.



①

```
int x;
```

No Syntax Error

No Error

②

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

Syntax Err

Expecting Identifier ,

③

```
x, y;
```

No Syntax Err

Semantic Err  
x & y are not declared

④

```
x=y;
```

No Syntax Err

# Find Syntax Errors.

⑤

if( );      | Syntax Error

⑥

while( );

⑦

for( ; ; );      No syntax Err  
                  Infinite loop

\*\*⑧

x y;      No syntax Err  
                  Semantic Err

⑨

if(23);

No error

10

```
typedef int x;  
x y;
```

≡

int y;

No error

(II)  
G-202

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

Which of the following  
throws error?

- A) Lexical Analysis
- B) Syntax "
- C) Semantic "
- D) M/C dependent Optimizer

12

```
Void main(  
{  
}
```

⇒ Syntax Error

13

```
for(x,y,z);
```

⇒ Syntax Error

for has fixed syntax  
(mismatched chrs)

\* \* \* (14) ~~semantic error~~  
~~f6()~~ is not defined

```
void main()
{
    foo(2,3,4);
}
```

function call  
Because no external file #include

(15)

```
Void main()
{
    for (2;3;4);
}
```

No compilation error

\* \* \* (16)

```
#include<stdio.h>
void main()
{
    foo(2,3,4);
}
```

compiler will ignore  
Linker error

(17)

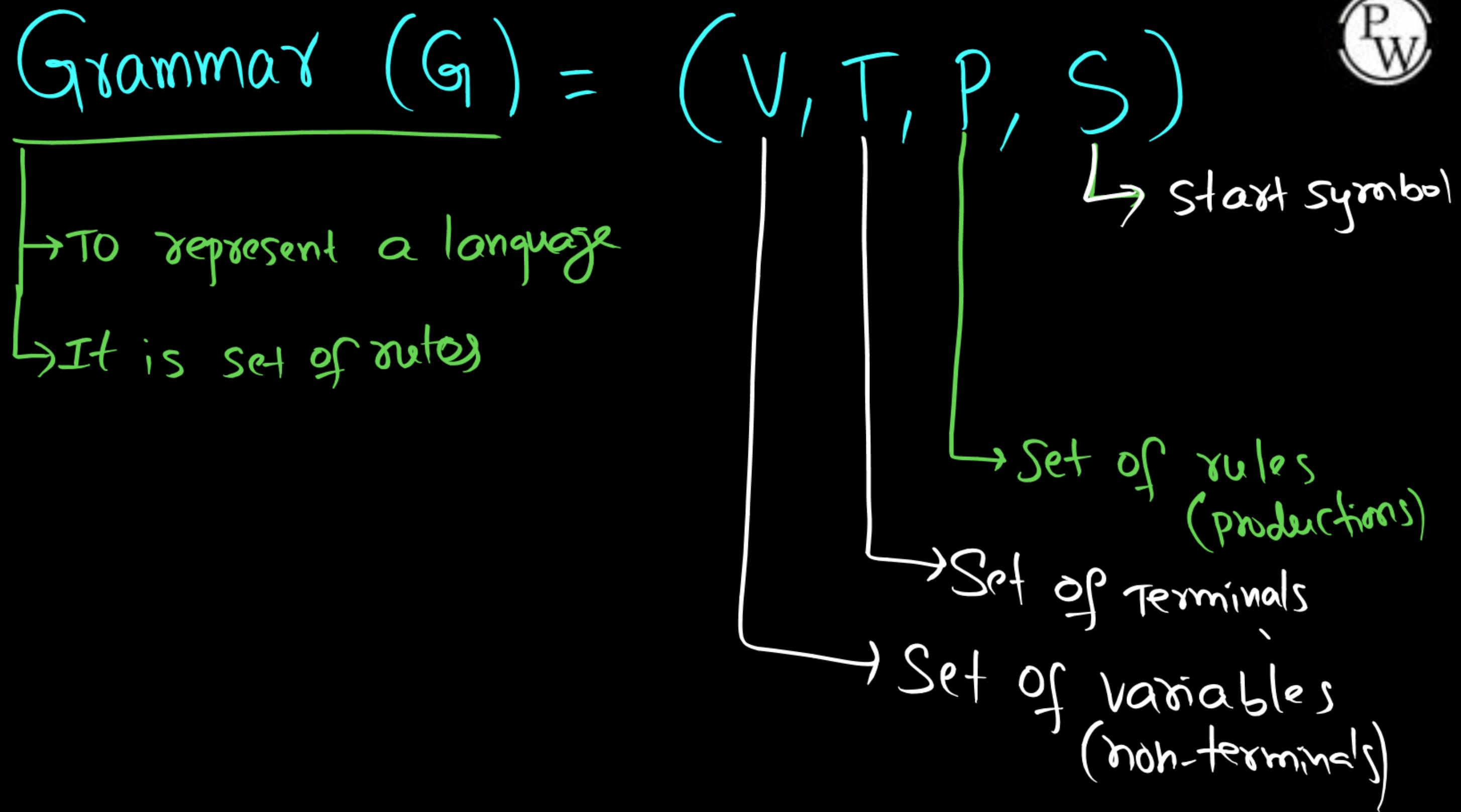
```
#include <P2.C>
Void main()
{
    foo(2,3,4);
}
```

P2.C file => compile later  
=> No error

P2.C file => 1st compile it is

```
void foo(int x, int y, int z)
{
    int a;
    a=x+y+z;
}
```





Context Free Grammar:  
 $(CFG) = (V, T, P, S)$

Each rule in  $P$  is defined as below:

$$V \rightarrow \text{Any}$$
$$V \rightarrow (VUT)^*$$



$\underbrace{LHS}_{\substack{\text{exactly} \\ \text{one} \\ \text{variable}}}$   $\longrightarrow$   $\underbrace{RHS}_{\text{no restriction}}$

$$S = X$$

$$T = \{a, b, c\}$$

$$V = \{X, A, B\}$$

$$P = \left\{ X \rightarrow AaB, X \rightarrow \epsilon, \right. \\ \left. A \rightarrow b, B \rightarrow c \right\}$$

# Why CFG ?

→ It is most suitable grammar  
to represent syntax of <sup>most</sup> programming languages

Program  $\rightarrow$  R main( )B

R  $\rightarrow$  void | int

B  $\rightarrow$  { D; }

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

V = { Program, R, B, D, T }

D  $\rightarrow$  T Id

T = { main, (, ), Void, int, ;, Id, float }

T  $\rightarrow$  int | float

# Summary

- Lexical Analysis ✓
- Syntax Errors ✓
- CFG  
Next

Thank you

