CS & IT ENGINEERING

Compiler Design

Lexical & Syntax Analysis

Lecture No. 4

By- DEVA Sir

01 CFG

02 Ambiguous & Unambiguous CFGs

03 Elimination of Left Recursion

04 Left Factoring

05 First & Follow Computation

CFG : $V \longrightarrow (VUT)^*$

$S \rightarrow \cdot\ \cdot\ -$

$A \rightarrow \cdot\ -\ \cdot\ \cdot$

~~$aA \rightarrow \cdot\ -\ \cdot\ -$~~

~~$SA \rightarrow \cdot\ -\ -$~~

Example :

start

$S \rightarrow Ab$

$A \rightarrow a$

✓

start

$S \rightarrow SS \mid \varepsilon$

✓

start

$E \rightarrow E+E \mid a$

✓

Derivations of a String:

Linear
1) Left Most Derivation (LMD)
2) Right Most Derivation (RMD)

Non Linear
3) Parse Tree (Derivation Tree)

Sentential Form: Any sequence of symbols derived from START

Derivative step (step) ✓

LMD ✓

RMD ✓

Leaf node ✓

Non Leaf node ✓

Parse Tree ✓

How many steps to derive a string?

(Length of derivation)

⇓

No. of substitutions takes place

$S \to AB$

$A \to a$

$B \to b$

String = ab

LMD :

3 Length of derivation
3 steps
3 Substitutions

S

$S \to AB$ (Left most)

AB

$A \to a$ (Left most non-ter)

aB

$B \to b$

ab

LMD

S

A

B

LMD order : SAB

AB
aB
ab
} Left sentential forms

$S \to AB$

$A \to a$

$B \to b$

String = ab

RMD :

RMD order : SBA

AB
Ab
ab
} Right sentential forms

RMD

$S \Rightarrow AB \Rightarrow Ab \Rightarrow ab$

LMD: "In every sentential form, **left most** nonterminal is substituted to derive a string"

RMD: In every sentential form, **Right most** nonterminal is substituted to derive a string.

For a String : LMD & RMD need not be same

$S \to a$

String = a

S
⇓
a

LMD & RMD

$S \to AB$
$A \to a$
$B \to b$

LMD:	RMD:
S	S
⇓	⇓
AB	AB
⇓	⇓
aB	AL
⇓	⇓
ab	ab

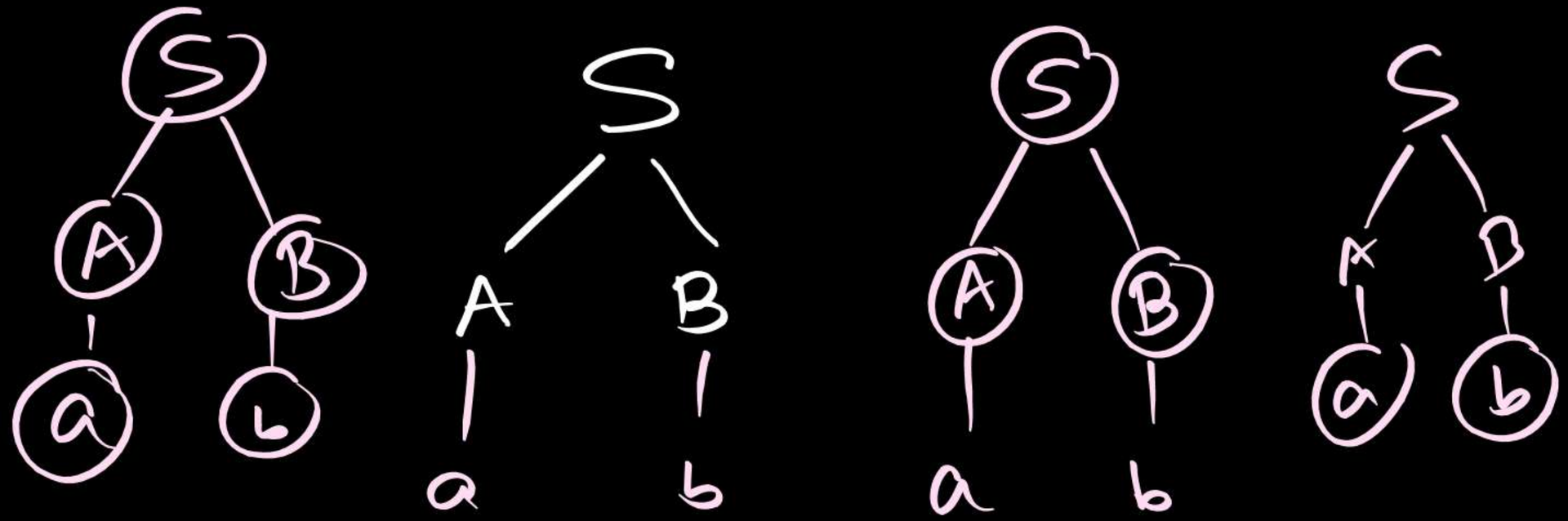
LMD & RMD are different

Parse Tree (Derivation Tree)

$S \rightarrow AB$

$A \rightarrow a$

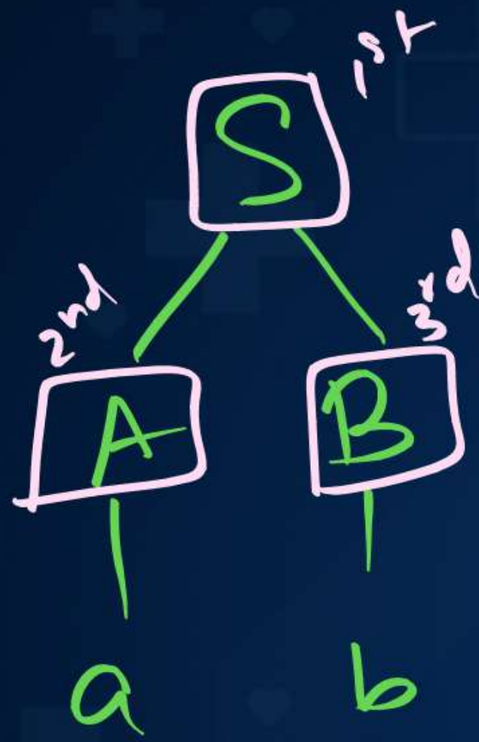
$B \rightarrow b$



Leaf nodes : a, b

Non leaf nodes : S, A, B

LMD:

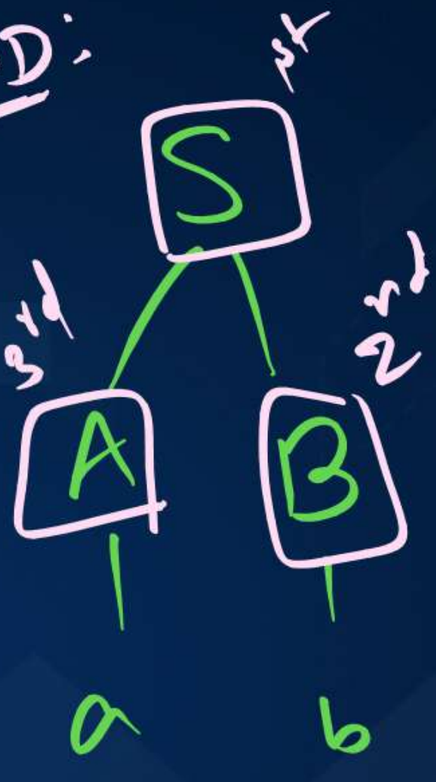


W = ab

S, A, B

LMD order

RMD:



S, B, A

RMD order

Length of derivation

=

No. of Substitutions in LMD / RMD

or

=

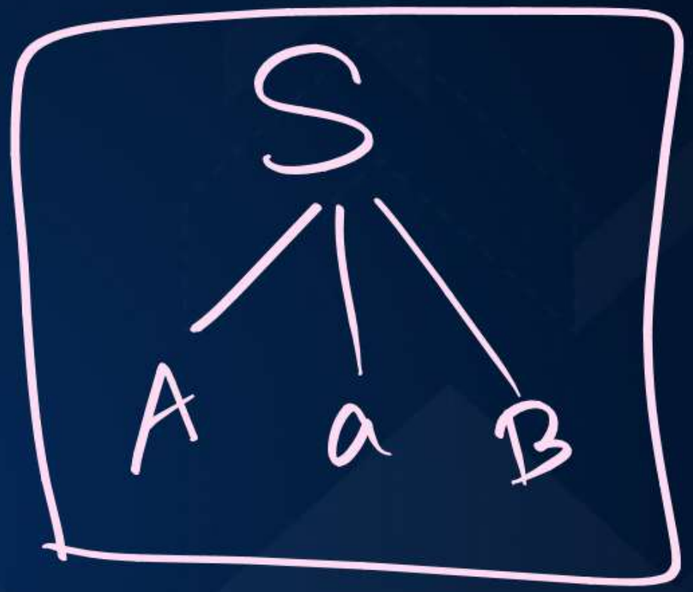
No. of non leaf nodes in Parse Tree

=

No. of steps

$S \rightarrow AaB$

produle
Rule
production Rule



Production tree

Parent
child
Sibling

No. of Parse Trees ?

$S \to A \mid a$

$A \to a \mid b$

$\omega = a$

Length of derivation is confused either 2 or 1

Length = 2

Length = 1

2 derivations

2 Parse Trees
2 LMDs
2 RMDs

For any string:

No. of Parse Trees = No. of LMDs

= No. of RMDs

= No. of derivations

Length of derivation

=

No. of steps to derive a string

No. of derivations

=

How many LMDs?

How many RMDs?

How many different parse trees?

Length of derivation can be asked

only when you have one derivation.

Types of CFGs [Based on No. of derivations]

① Ambiguous CFG

Atleast one string can be derived with more than one parse tree

$\exists w \in L(G) \ni >1$ P.T.

② Unambiguous CFG

Every string has only 1 Parse Tree
(1 derivation)
(1 LMD)
(1 RMD)

$\forall w \in L(G) \ni 1$ P.T.

① $S \rightarrow ab$

ab : 1 PT

Unambiguous CFG

② $S \rightarrow a \mid b$

a ⇨ 1 PT S → a

b ⇨ 1 PT S → b

Unambiguous CFG

③ $S \rightarrow SS \mid \varepsilon$

$\varepsilon \Rightarrow$ Infinite Parse Trees

≥ 1 PT

Ambiguous CFG

$\varepsilon = \varepsilon\varepsilon = \varepsilon\varepsilon\varepsilon = \varepsilon\varepsilon\varepsilon\varepsilon$

④ $S \rightarrow Ab \mid ab \mid \varepsilon$

$A \rightarrow \varepsilon \mid a \mid b$

$\varepsilon \Rightarrow 1$ PT

$b \Rightarrow 1$ PT

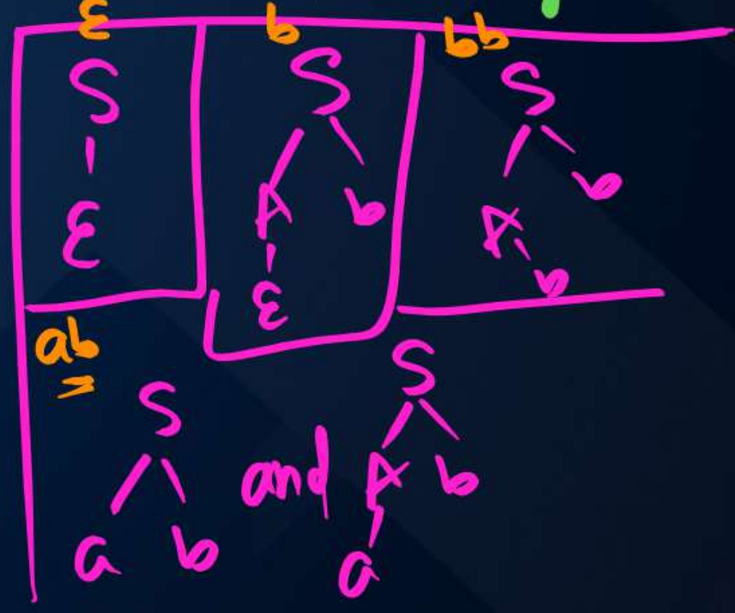
$ab \Rightarrow 2$ PTs

$bb \Rightarrow 1$ PT

Q1) How many strings are generated ? $\Rightarrow$ 4 strings

Q2) Is CFG Ambiguous ? $\Rightarrow$ Yes

Q3) Is CFG Unambiguous ! $\Rightarrow$ No



(5)

$S \rightarrow AB \mid ab$

$A \rightarrow Aa \mid a$

$B \rightarrow Bb \mid b$

Do you have any string having >1 PT?

ab

>1 PT

Ambiguous CFG

⑥ $S \rightarrow AB$

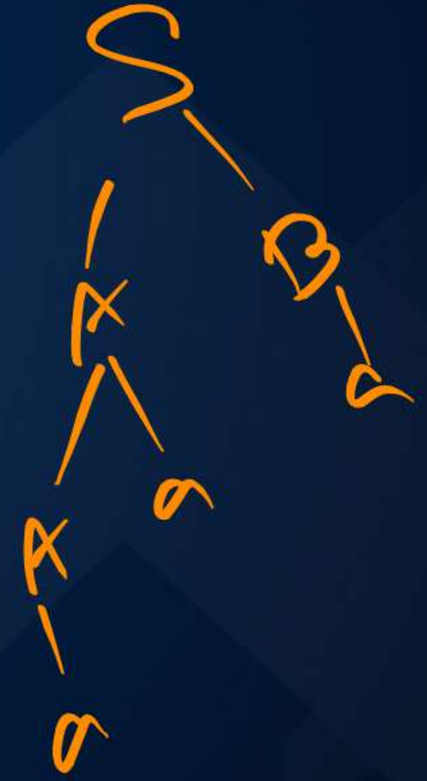
$A \rightarrow Aa \mid a$

$B \rightarrow Ba \mid a$

Ambiguous CFG

$aa \Rightarrow 1$ PT

$aaa \Rightarrow 2$ PTs



(6) $S \rightarrow AB$

$A \rightarrow Aa \mid a$

$B \rightarrow Ba \mid a$

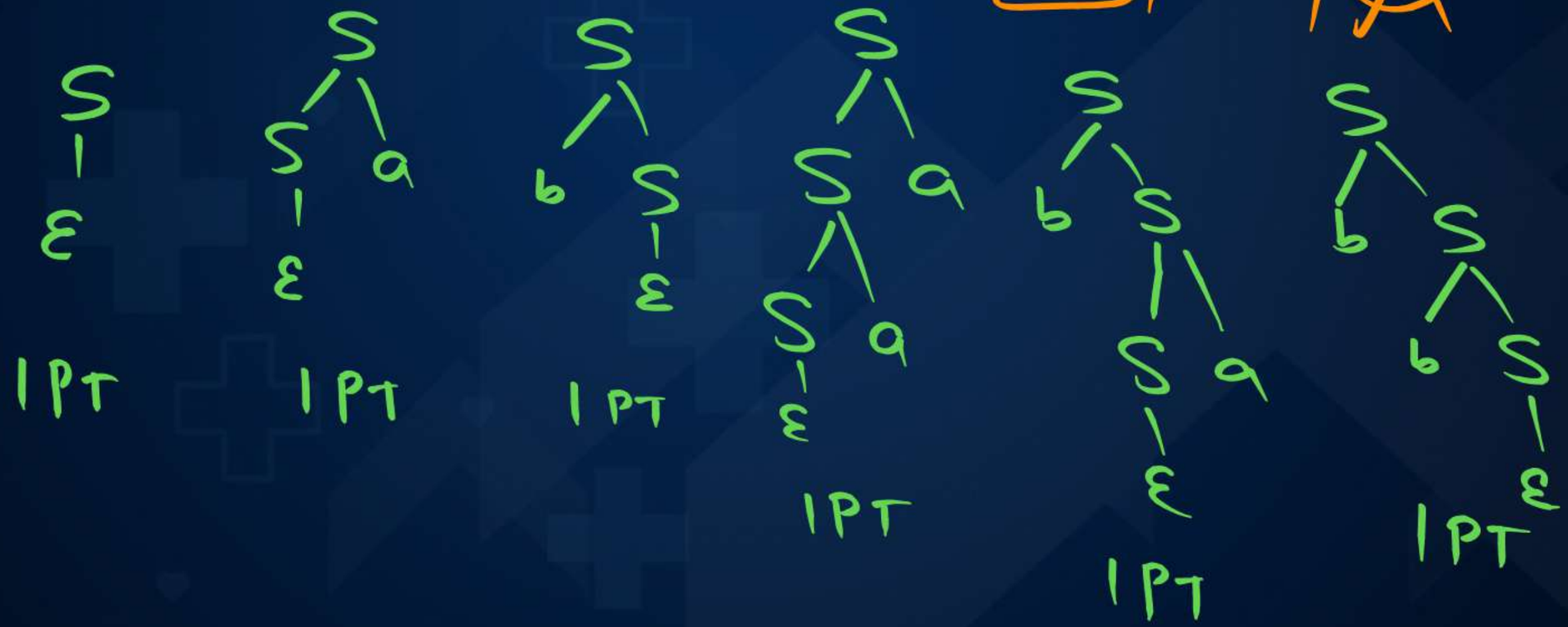
How many parse trees for a^4 ?

$a^4 = aaaa$

⇒ 3 Parse Trees

(7) $S \rightarrow Sa \mid bS \mid \varepsilon$

ε , a , b , aa , ba , bb , ~~ab~~ , . . .

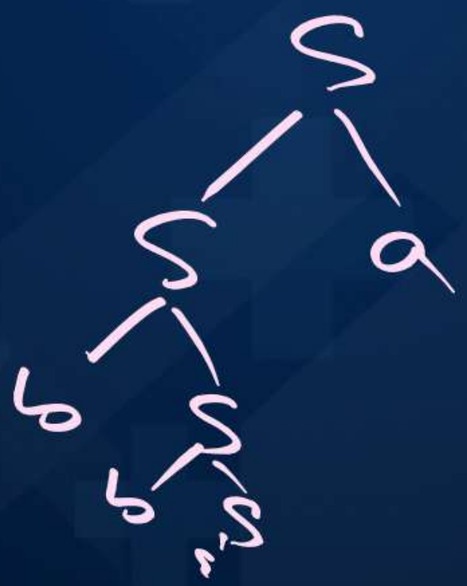


(7) $S \rightarrow Sa \mid bS \mid \varepsilon$

If Same nonterminal is having both Left & Right recursion then CFG is Ambiguous.

Ambiguous

aaa , ~~aab~~ , ~~aba~~ , ~~abb~~ , baa .



2 PTs

Note: Checking Ambiguous & Unambiguous is not having algorithm.

If CFG is generating infinite strings,

How long we verify to decide ?

We may never come up with answer

(8) $S \rightarrow aS \mid Sb \mid \varepsilon$

Left Recursion

last

first

Right Rec

$\Rightarrow$ Ambiguous

LHS → LHS .... (first)

LHS → .... LHS (last)

⑨ $S \rightarrow SaS \mid \varepsilon$

⑩ $S \rightarrow SS \mid a$

⑪ $S \rightarrow Ab$

$A \rightarrow Aa \mid bA \mid c$

Ambiguous CFG

(12) $S \rightarrow (S) \mid \varepsilon$ ⇒ Unambiguous

(13) $S \rightarrow aSb \mid \varepsilon$

(14) $S \rightarrow AB \mid \varepsilon$

$A \rightarrow aA \mid \varepsilon$

$B \rightarrow bB \mid \varepsilon$

⇒ Ambiguous

(15) $S \rightarrow AB$
$A \rightarrow aA \mid \varepsilon$
$B \rightarrow bB \mid \varepsilon$
$\Rightarrow$ Unambiguous

(16) $E \rightarrow E+E \mid a$ $\Rightarrow$ Ambiguous CFG

(17) $E \rightarrow E+E \mid E*E \mid E-E \mid E/E \mid a$

Ambiguous CFG	Unambiguous CFG
Easy to verify	Difficult to verify
Because Some string, >1 PT	Every string, exactly 1 PT

Summary

- Ambiguous & Unambiguous CFG,
- Next: Other Basics.


THANK YOU
GW
SOLDIERS !