segma ← language ⇒ finite or infinite

$\Sigma = \{a, b\}$ set of symbols

* finite

L_1 = set of all strings of length 2

= {aa, bb, ab, ba}.

L_2 = set of " " " 3

= {aaa, bbb, aba, aab, abb, baa, bba, bab}

* infinite

L_3 = " " " " where each one starts with a.

= {a, aa, aaa, aba, abb, ---}.

Power of Σ

Σ^0 = set of all symbols of length "0" ⇒ $\Sigma^0 = \{\epsilon\}$ epsilon

Σ^1 = set of all symbols " " "1" ⇒ {a, b}

Σ^2 = " " " " " " "2" = {aa, bb, ab, ba}.

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots \cup \Sigma^n$

= {ε} ∪ {a, b} ∪ {aa, ba, ab, bb} --- ⇐ infinite automata

Contains all possible strings that is finite or infinite.

CFG (context free grammer)

CFG is used to generate language which is called context free language (CFL).

* CFG is defined by 4-tuples = {V, Σ, S, P}, where:-

1- V = set of variables or non-terminals.

2- Σ = set of terminal symbols.

3- S = start symbol.

4- Production Rule.

$A \to a$

$a = \{V \cup \Sigma\}^*$ and $A \in V$

EX:- For generating a language that generates equal number of a's and b's in form $a^n b^n$ so CFG will like:

$G = [(S, A), (a, b), (S \to aAb, A \to aAb \mid \epsilon)]$

$S \to aAb$

$\to aaAbb$

$\to aaaAbbb$

$\to aaa\epsilon bbb$

$= a^3 b^3$

Ex:- $S \to aSB$

$S \to aB$

$B \to b$

$\Sigma = \{a, b\}, \; V = \{S, B\}$.

get the derivation of aabb

$S \Rightarrow aSB$

$\Rightarrow aaBB$

$\Rightarrow aabB$

$\Rightarrow aabb$

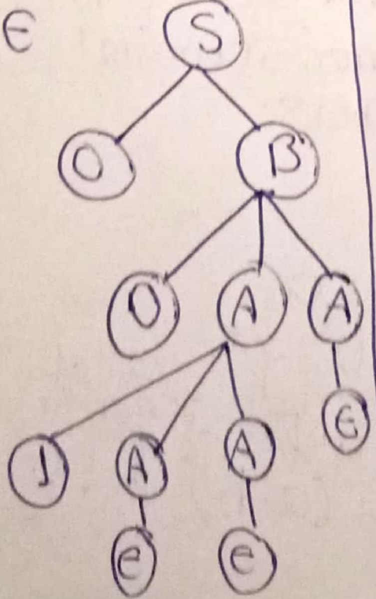
Derivation Tree

It is an ordered tree that graphically represents the semantic information of strings derived from a context free grammar.

$S \to 0B$

$A \to 1AA \mid \epsilon$

$B \to 0AA$



left most derivation is obtained by applying production to the leftmost variable in each step

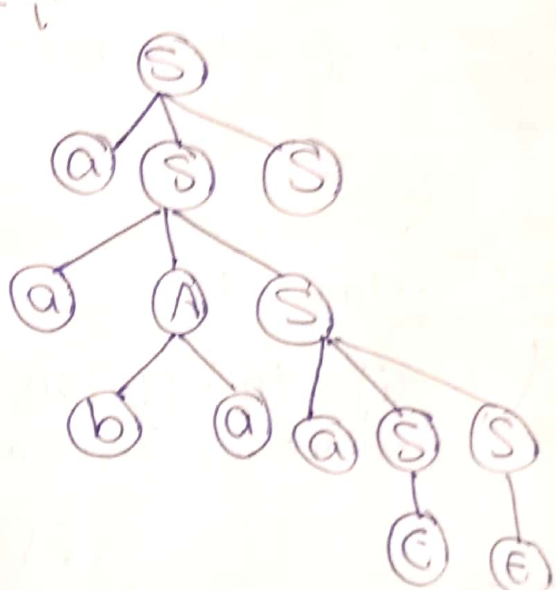
Right most derivation is obtained by applying production to the right most variable in each step

Ex:- aabaa

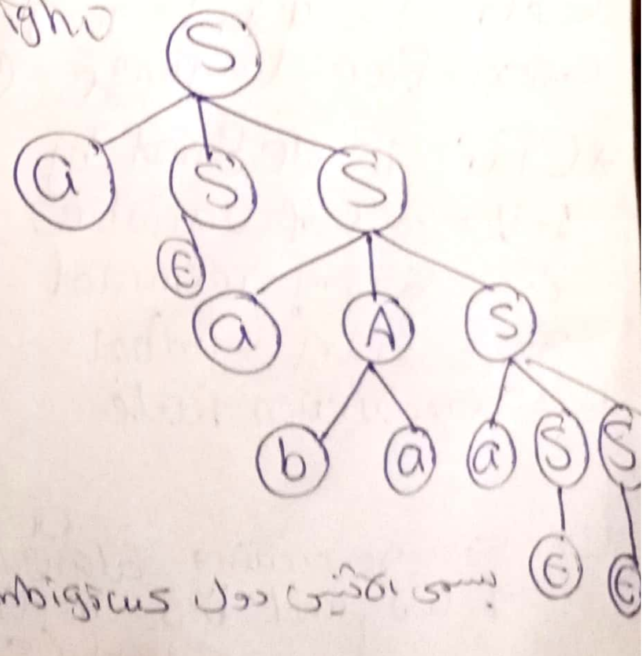
$S \to aAS \mid aSS \mid \epsilon$

$A \to SbA \mid ba$

For left



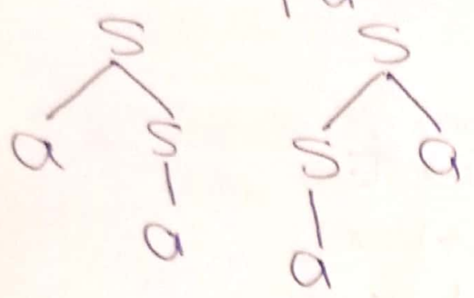
For right



ambiguous يسمى الاتنين دول

Ex:- check if the following expressions over the given grammer is ambiguous or not:

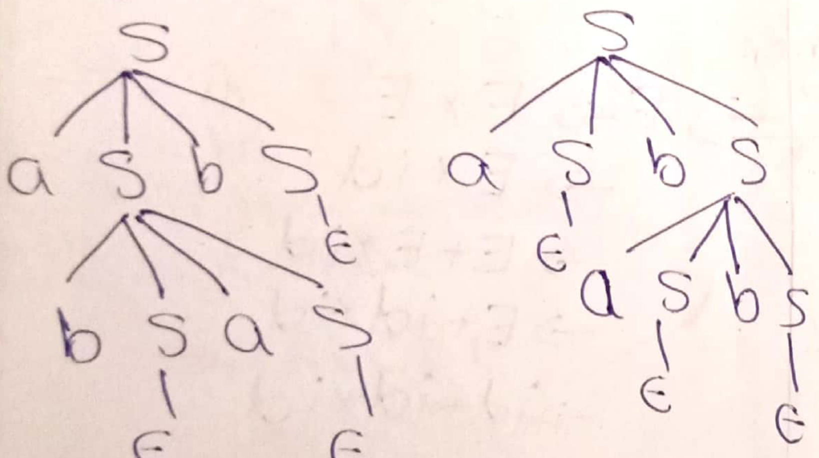
$S \to aS$ $\quad S = aa$
$\mid Sa$
$\mid a$



ambiguous

$S \to aSbS$
$\mid bSaS$
$\mid \epsilon$

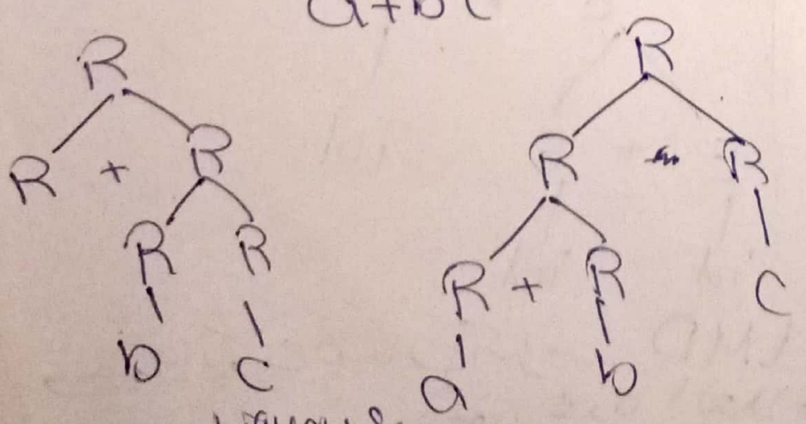
$w = abab$



ambiguous

$R \to R+R \,/\, RR \,/\, R^* \,/\, a \,/\, b \,/\, c$

$a+bc$



ambiguous

Ex: 2 $E \to E + E$
$\quad / E * E$
$\quad / id$

get the derivation of the following exp
id + id * id

$E \to E + E$
$\to id + E$
$\to id + E * E$
$\to id + id * E$
$\to id + id * id$

LMD → left most der

Another LMD ⟹

$E \to E * E$
$\to E + E * E$
$\to id + E * E$
$\to id + id * E$
$\to id + id * E$

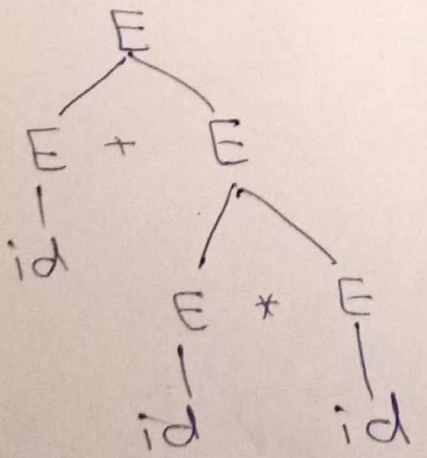
$E \to E + E$
$\to E + E * E$
$\to E + E * id$
$\to E + id * id$
$\to id + id * id$

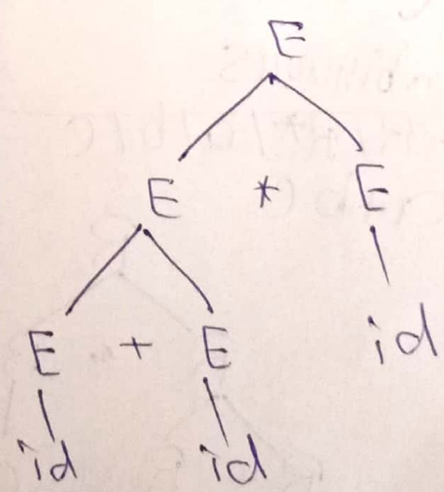
RMD → Right most der.

Another RMD ⟹

$E \to E * E$
$\to E * id$
$\to E + E * id$
$\to E + id * id$
$\to id + id * id$

Parse Tree For LMD





وعندما يكون هناك اكثر من LMD و RMS و Parse لنفس exp
تسمى هذه العلاقة ambiguous.

Parser

Top-Down Parser (TDP) | Bottom-Up Parser (BUP)

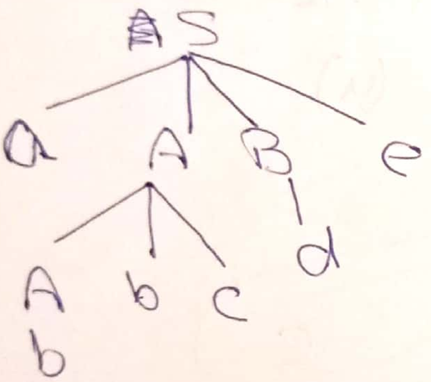
Ex: check if the following string can be generated from the following grammer using parse tree:-

$S \rightarrow aABe$ abbcde

$A \rightarrow Abc / b$

$B \rightarrow d$.

Top-Down



left most derivation

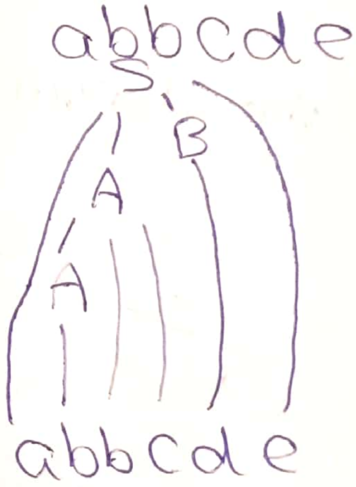
$S \Rightarrow aABe$

$\Rightarrow aAbcBe$

$\Rightarrow abbcBe$

$\Rightarrow abbcde$

Bottom-up



Right most derivation

$S \Rightarrow aABe$

$\Rightarrow aAde$

$\Rightarrow aAbcde$

$\Rightarrow abbcde$

EX1:- let G be the Grammar:-

$S \rightarrow abSc \mid A$

$A \rightarrow cAd \mid cd$

a) Give a drivation of ababccddcc

$S \Rightarrow abSc$

$\Rightarrow ababScc$

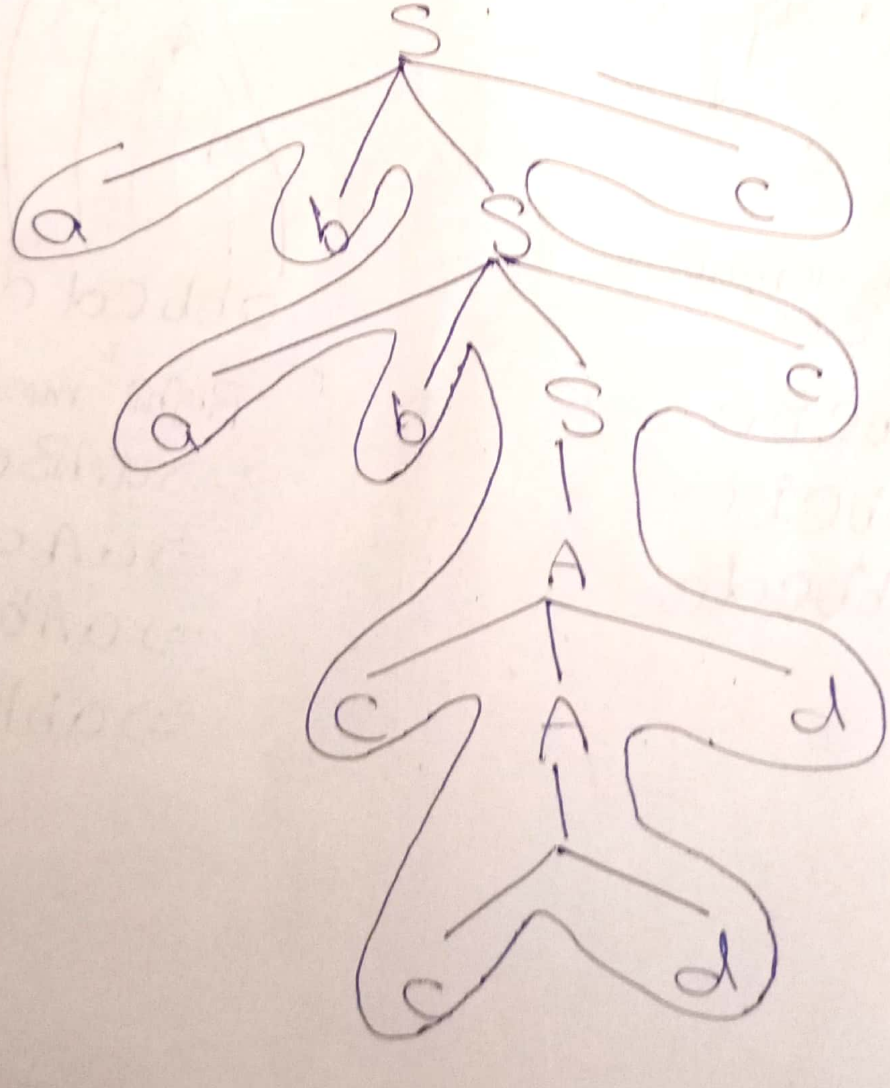
$\Rightarrow ababAcc$

$\Rightarrow ababcAdcc$

$\Rightarrow ababccddcc$

left most derivation تسمى وهكذا

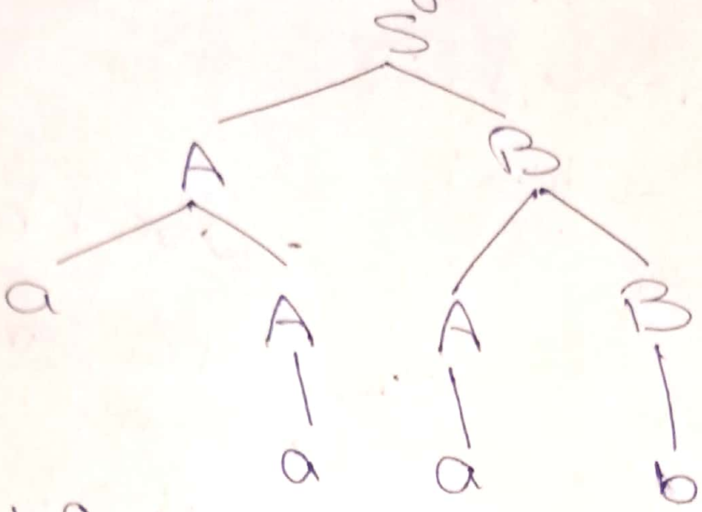
b) build the derivation tree of (a)



Scanned with CamScanner

c) Give the notation of L(G)

Ex: let DT as following:-



a) Give a leftmost derivation that generates this DT

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

Rule
Grammar

$S \to AB$
$S \to aAB$
$S \to aaB$
$S \to aaAB$
$S \to aaaB$
$S \to aaab$

b) Give a right most derivation that generates this DT

$S \to AB$
$S \to AAB$
$S \to AAb$
$S \to Aab$
$S \to aAab$
$S \to aaab.$

Ex-let G be the grammar:-

$S \to ASB \mid ab \mid SS$

$A \to aA \mid \lambda$

$B \to bB \mid \lambda$

a) Give the left most derivation of aaabb

$S \to ASB$
$\to aASB$
$\to aaASB$
$\to aaSB$
$\to aaabB$
$\to aaabbB$
$\to aaabb$

b) give the right most derivation of aaabb

$S \to ASB$
$\to ASbB$
$\to \cancel{ASbbB}$
$\to \cancel{ASbb}$
$\to \cancel{AASB}$
$\to \cancel{Aabbb}$
$\to \cancel{aAab}$
$\to ASb$
$\to Aabb$
$\to aAabb$
$\to aaabb$

c) Show that this grammar is ambiguous.

abab

$S \to ASB$
$\to SSB$
$\to abSB$
$\to ababB$
$\to abab$

$S \to SS$
$S \to abS$
$S \to abab$

hence there are two left most derivations so this grammar is ambig