Context Free Grammar

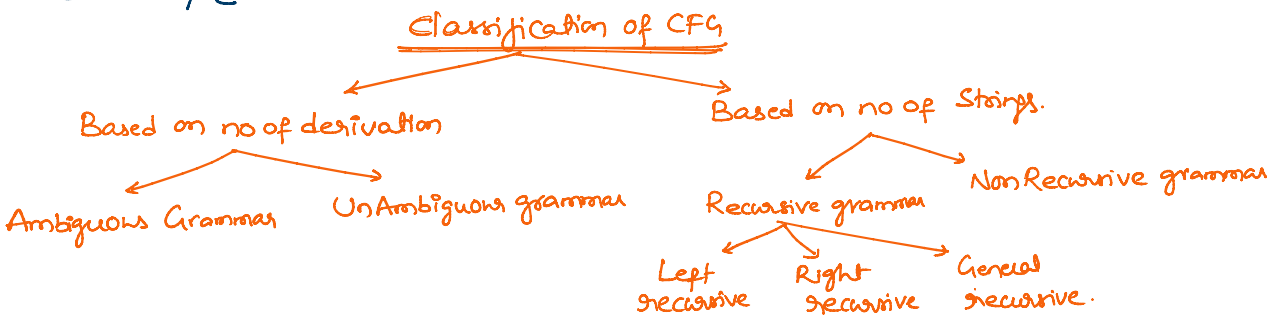
Wednesday, March 30, 2022 5:53 PM

Syntax Analysis It groups tokens together into syntactic structures according the grammar.

Context-free grammar :- A grammar $G = (V, T, P, S)$ is said to be CFG if all productions in P are of the form $A \to \alpha$ where A is single nonterminal and $\alpha \in (V \cup T)^*$

V ← set of non terminals
T ← set of terminals.
S ← starting nonterminal.

Example:
$S \to Aa$
$A \to BaC$
$C \to ab / \epsilon$

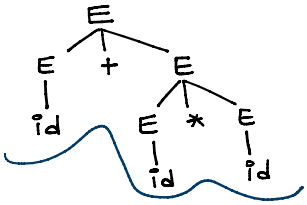


Derivation :- Derivation provides a means of deriving (generating) the sentences of a language.

Ex:-
$E \to E+E$
$E \to E*E$
$E \to id$

input: id + id * id

$E \to \underline{E}+E$
$\to id + \underline{E}$
$\to id + E * \underline{E}$
$\to id + \underline{E} * id$
$\to id + id * id$



Parse tree / Derivation tree.

Derivation of Strings can be done in two ways.

① **Leftmost derivation (LMD) :-** A derivation is said to be leftmost if in each step the leftmost nonterminal is replaced.

Note: Top-down parsers use Leftmost derivation.

② **Rightmost derivation (RMD) :-** A derivation is said to be rightmost if in each step the rightmost nonterminal is replaced.

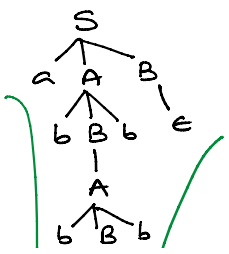
Note: Bottom-up parser use rightmost derivation in reverse.

Q:- $S \to aAB$
$A \to bBb$
$B \to A / \epsilon$

String: abbbb

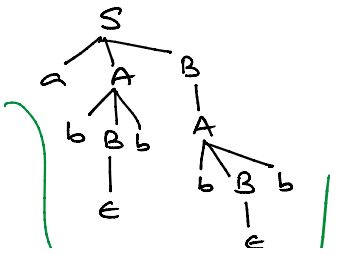
LMD

$S \to a\underline{A}B$
$\to abB\underline{b}B$
$\to ab\underline{A}bB$
$\to abbB\underline{b}bB$
$\to abbbbB$
$\to abbbb$

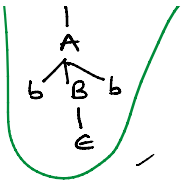


RMD

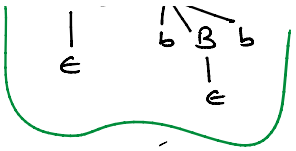
$S \to aA\underline{B}$
$\to aA\underline{A}$
$\to aAbB\underline{b}$
$\to a\underline{A}bb$
$\to abB\underline{b}bb$



$\rightarrow abbBbbB$
$\rightarrow abbbbB$
$\rightarrow abbbb$



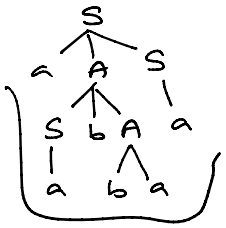
$\rightarrow aAbb$
$\rightarrow abBbbb$
$\rightarrow abbbb$



Q:- $S \rightarrow aAS/a$
$A \rightarrow SbA/SS/ba$
input: aabbaa

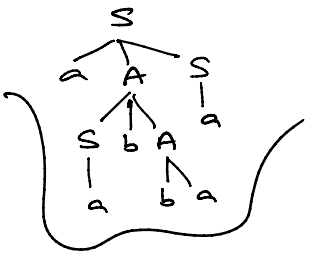
LMD

$S \rightarrow aAS$
$\rightarrow aSbAS$
$\rightarrow aabAS$
$\rightarrow aabbaS$
$\rightarrow aabbaa$



RMD

$S \rightarrow aAS$
$\rightarrow aAa$
$\rightarrow aSbAa$
$\rightarrow aSbbaa$
$\rightarrow aabbaa$

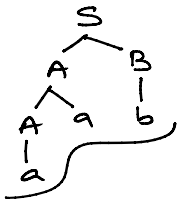


Ambiguous Grammar:- A grammar is ambiguous if it produces more than one leftmost or more than one rightmost derivation for the same sentence.

Ex:- $S \rightarrow AB/aaB$
$A \rightarrow a/Aa$
$B \rightarrow b$
input: aab

LMD-1

$S \rightarrow AB$
$\rightarrow AaB$
$\rightarrow aaB$
$\rightarrow aab$

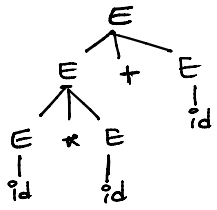


LMD-II

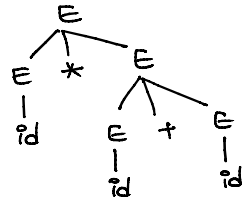
$S \rightarrow aaB$
$\rightarrow aab$

the grammar is Ambiguous.

Q:- Check is the following grammar ambiguous?
$E \rightarrow E+E/E*E/id$



input: id * id + id



input: id * id + id

Ambiguous grammar.

Note:- Testing Ambiguity of CFG is undecidable.

⇒ If the leftmost and rightmost symbol in the RHS if same as LHS then such grammars are called Ambiguous Grammar.

Imp

$E \rightarrow E+E$

Ambiguous

Q:- Consider the following CFG.
$E \rightarrow E+E / (E*E) / id$

I. which of the following terminal strings has more one parsetree when passed according to the above grammar.

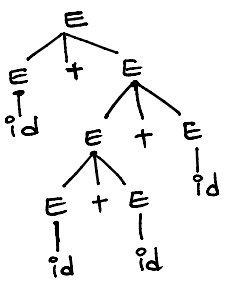
~~a)~~ id + id + id + id

~~b)~~ id + (id * (id * id))

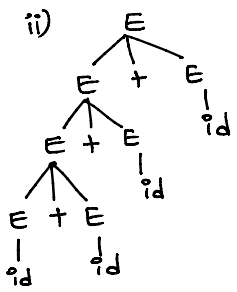
~~c)~~ (id * (id * id)) + id

~~d)~~ ((id * id + id) * id)

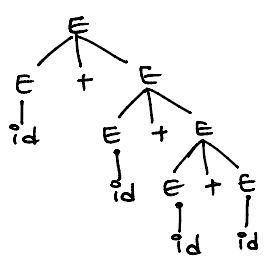
(i)



(ii)



(iii)

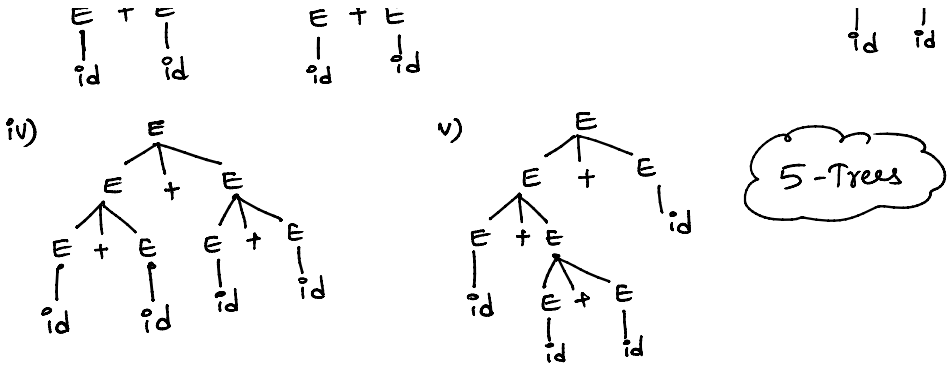


II. For the correct answer to the

~~c)~~ ((id*id+id)*id)

II. for the correct answer to the above question how many parse trees are possible?

~~a)~~ 5 b) 4 c) 3 d) 2



2007 (2+2)

Q:- Consider the following CFG

$S \to bA / aB$

$A \to a / aS / bAA$

$B \to b / bS / aBB$

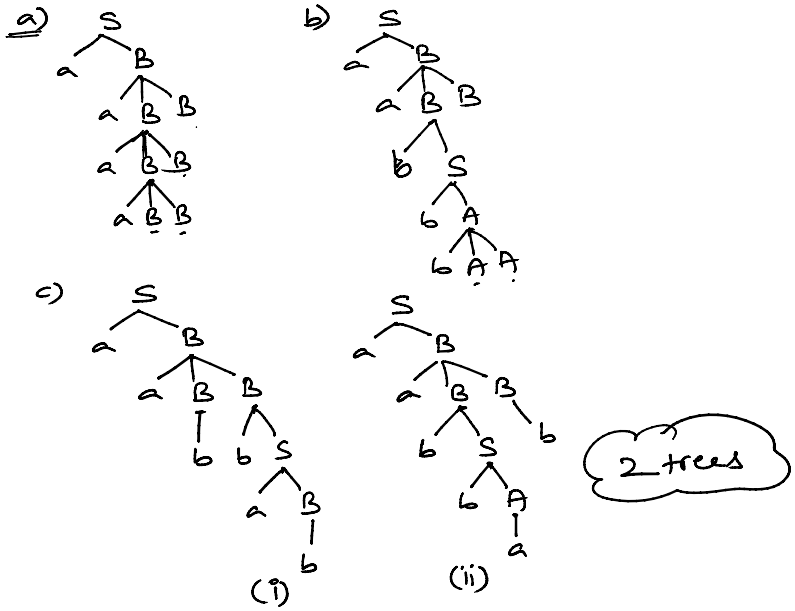
I. which of the following string is generated by the grammar.

~~a) aaaabb~~ ~~b) aabbbb~~

~~c) aabbab~~ ~~d) abbbba~~

II. for the correct answer to the above question how many parsetrees are possible?

a) 1 ~~b) 2~~ c) 3 d) 4

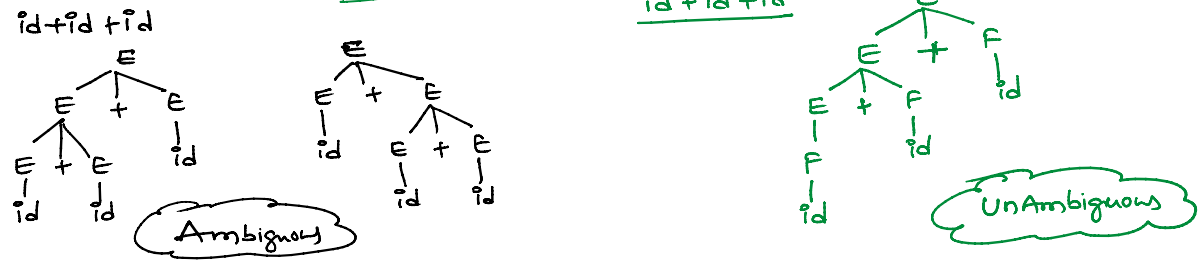


Note ① No Algorithm exist to verify ambiguity of CFG.

② No Algorithm exist to convert ambiguous grammars into unambiguous.

③ By trial and error method Some of the ambiguous grammars can be converted to unambiguous.

Ex:- $E \to E+E / id \Rightarrow E \to E+F / F$, $F \to id$



④ By redefining the precedence and associativity of operators Some of the ambiguous grammars can be converted into unambiguous