Date & Time of Examination - 01/12/2021 @ 9:30 AM

Examination Roll No. - 18312911011

Name of Program - B.tech (IT & MI)

Semester / Year - VII Semester / IV year

Unique Paper Code - 911711

Title of Paper - Computer Language, design and Engineering

Q.4

Difference between

(i) Ambiguous Grammar & Non-Ambiguous Grammar

Ambiguous Grammar	Non-Ambiguous Grammar
1) In ambiguous grammar, there exists more than one derivation tree of the content free grammar	A content-free grammar is called unambiguous grammar if there exists only one parse tree
2) In ambiguous grammar, the leftmost & rightmost derivations are not same	2) The leftmost & rightmost derivations are same
3) Amount of non-terminals is less	3) Amount of non-terminals are more
4) Length of parse tree is short	4) Length of parse tree is large
5) Speed of derivation of a tree is faster here	5) Speed of derivation of a tree is slower
6) Contains Ambiguity	6) Does not contain Ambiguity
7) Ex: $S \rightarrow S+S/S*S/S/a$	7) Ex: $X \rightarrow AB$ $A \rightarrow Aa/a$ $B \rightarrow b$

(ii)

Leftmost Derivation / Right Most Derivation

Left Most Derivation	Right Most Derivation
1) Obtained by applying Production to the left most variable in each step	1) Obtained by applying production to the right most variable in each step
2) Always the leftmost non-terminal which is expanded	2) Always the rightmost non-terminal which is expanded
3) Let any set of production rules in a content free grammar be $X \to X+X \mid X*X \mid X \mid a$ over an alphabet $\{a\}$ Then the leftmost derivation of the string "$a+a*a$" may be $X \to X+X \to a+X \to$ $a+X*X \to a+a*X$ $\to a+a*a$	3) The right most derivation for the same content free grammar (same example) for the above string "$a+a*a$" may be $X \to X*X \to X*a$ $\to X+X*a \to$ $X+a*a \to a+a*a$

(iii)

Deterministic / Non-Deterministic Grammar

Deterministic Grammar	Non-Deterministic Grammar
1) Grammar without any common prefix in any productions from same left hand side is known as deterministic grammar	1) Grammar with any common prefix between atleast two different productions from the same left side in the grammar
2) It is not time consuming during parsing	2) It is very time consuming as a lot of back tracking is required.
3) It is more suitable for predictive or top down parsing because of its deterministic nature	3) Need to convert into deterministic grammar through left factoring to make it suitable for predictive parsing
4) Ex: S → AaB / BA A → a/b B → d/e (No common Prefix)	4) Ex: S → aSb / aA / b A → aB / a (Common Prefix a) B → b