**AESTIVIAL TOC ASSIGNMENT**

|  |  |  |
| --- | --- | --- |
| **Q1**  **Parameter** | **Context Free Grammar** | **Regular Grammar** |
| Type | Type-2 | Type-3 |
| Recognizer | Push-down automata. | Finite State Automata |
| Rules | Productions are of the form: A->B;  A∈N(Non-Terminal)  B∈V\*(Any string) | Productions are of the form: V –> VT / T (left-linear grammar) (or) V –> TV /T (right-linear grammar) |
| Restriction | Less than Regular Grammar | More than any other grammar |
| Right-hand Side | The right-hand side of production has no restrictions. | The right-hand side of production should be either left linear or right linear. |
| Set Property | Super Set of Regular Grammar | Subset of Context Free Grammar |
| Intersection | Intersection of two CFL need not be a CFL | Intersection of two RG is a RG. |
| Complement | They are not closed under complement | Closed under complement |
| Range | The range of languages that come under CFG is wide. | The range of languages that come under RG is less than CFG. |
| Examples | S –> AB;A –> a;B –> b | S -> aS | bS | ∊ |

**Q2**

Leftmost and Rightmost Derivation of a String

* **Leftmost derivation** − A leftmost derivation is obtained by applying production to the leftmost variable in each step.
* **Rightmost derivation** − A rightmost derivation is obtained by applying production to the rightmost variable in each step.

**Example**

Let any set of production rules in a CFG be

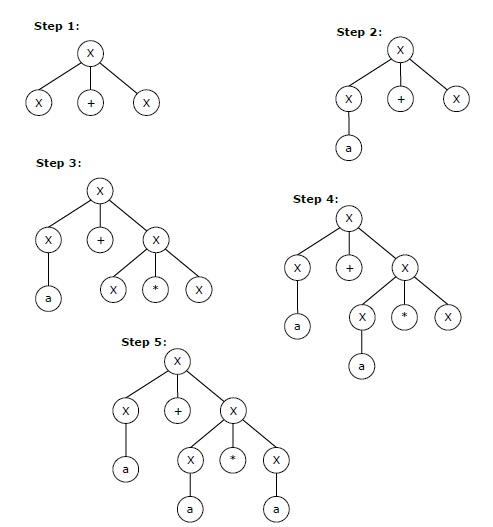
X → X+X | X\*X |X| a

over an alphabet {a}.

The leftmost derivation for the string **"a+a\*a"** may be −

X → X+X → a+X → a + X\*X → a+a\*X → a+a\*a

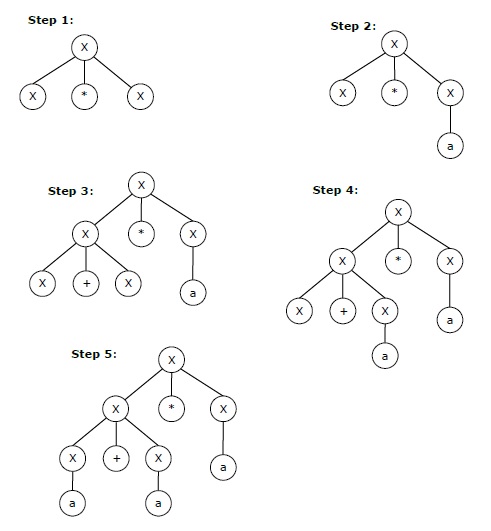
The stepwise derivation of the above string is shown as below −



The rightmost derivation for the above string **"a+a\*a"** may be −

X → X\*X → X\*a → X+X\*a → X+a\*a → a+a\*a

The stepwise derivation of the above string is shown as below −



**Q3**

If a context free grammar **G** has more than one derivation tree for some string **w ∈ L(G)**, it is called an **ambiguous grammar**. There exist multiple right-most or left-most derivations for some string generated from that grammar.

## **Problem**

Check whether the grammar G with production rules −

X → X+X | X\*X |X| a

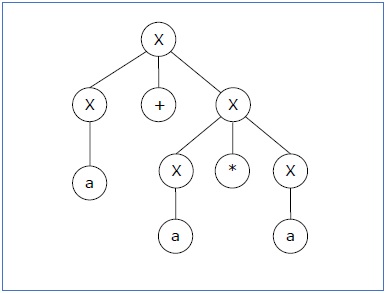
is ambiguous or not.

## **Solution**

Let’s find out the derivation tree for the string "a+a\*a". It has two leftmost derivations.

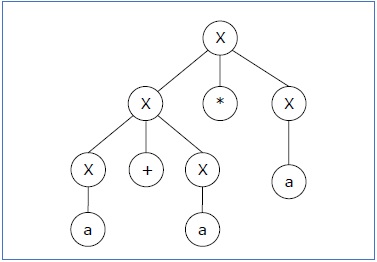
**Derivation 1** − X → X+X → a +X → a+ X\*X → a+a\*X → a+a\*a

**Parse tree 1** −



**Derivation 2** − X → X\*X → X+X\*X → a+ X\*X → a+a\*X → a+a\*a

**Parse tree 2** −



Since there are two parse trees for a single string "a+a\*a", the grammar **G** is ambiguous.

**Q4**

Pushdown Automata is a finite automata with extra memory called stack which helps Pushdown automata to recognize Context Free Languages.

A Pushdown Automata (PDA) can be defined as :

Q is the set of states

∑is the set of input symbols

Γ is the set of pushdown symbols (which can be pushed and popped from stack)

q0 is the initial state

Z is the initial pushdown symbol (which is initially present in stack)

F is the set of final states

δ is a transition function which maps Q x {Σ ∪ ∈} x Γ into Q x Γ\*. In a given state, PDA will read input symbol and stack symbol (top of the stack) and move to a new state and change the symbol of stack.

**Q5**

## **Lemma**

If **L** is a context-free language, there is a pumping length **p** such that any string **w ∈ L** of length **≥ p** can be written as **w = uvxyz**, where **vy ≠ ε**, **|vxy| ≤ p**, and for all **i ≥ 0, uvixyiz ∈ L**.