

Context-Free Grammar Introduction

Definition – A context-free grammar (CFG) consisting of a finite set of grammar rules is a quadruple $(\mathbf{N}, \mathbf{T}, \mathbf{P}, \mathbf{S})$ where

- \mathbf{N} is a set of non-terminal symbols.
- \mathbf{T} is a set of terminals where $\mathbf{N} \cap \mathbf{T} = \mathbf{NULL}$.
- \mathbf{P} is a set of rules, $\mathbf{P}: \mathbf{N} \rightarrow (\mathbf{N} \cup \mathbf{T})^*$, i.e., the left-hand side of the production rule \mathbf{P} does not have any right context or left context.
- \mathbf{S} is the start symbol.

Example

- The grammar $(\{A\}, \{a, b, c\}, P, A)$, $P: A \rightarrow aA, A \rightarrow abc$.
- The grammar $(\{S, a, b\}, \{a, b\}, P, S)$, $P: S \rightarrow aSa, S \rightarrow bSb, S \rightarrow \epsilon$
- The grammar $(\{S, F\}, \{0, 1\}, P, S)$, $P: S \rightarrow 00S \mid 11F, F \rightarrow 00F \mid \epsilon$

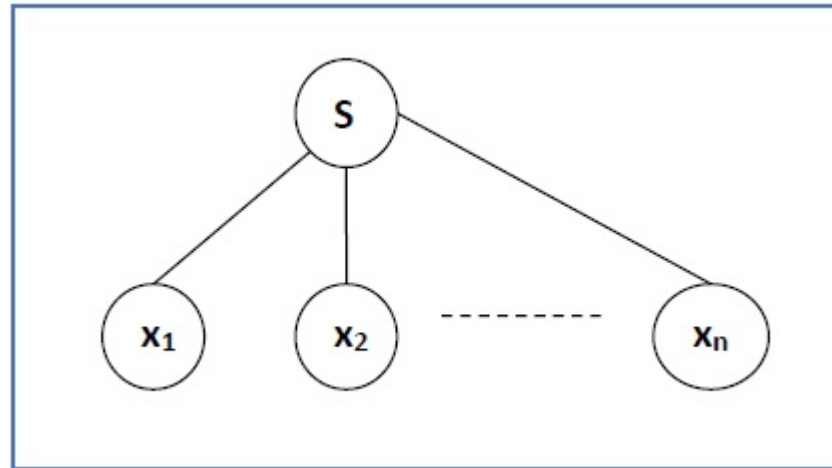
Generation of Derivation Tree

A derivation tree or parse tree is an ordered rooted tree that graphically represents the semantic information a string derived from a context-free grammar.

Representation Technique

- **Root vertex** – Must be labeled by the start symbol.
- **Vertex** – Labeled by a non-terminal symbol.
- **Leaves** – Labeled by a terminal symbol or ϵ .

If $S \rightarrow x_1x_2 \dots x_n$ is a production rule in a CFG, then the parse tree / derivation tree will be as follows –



There are two different approaches to draw a derivation tree –

Top-down Approach –

- Starts with the starting symbol **S**
- Goes down to tree leaves using productions

Bottom-up Approach –

- Starts from tree leaves
- Proceeds upward to the root which is the starting symbol **S**

Derivation or Yield of a Tree

The derivation or the yield of a parse tree is the final string obtained by concatenating the labels of the leaves of the tree from left to right, ignoring the Nulls. However, if all the leaves are Null, derivation is Null.

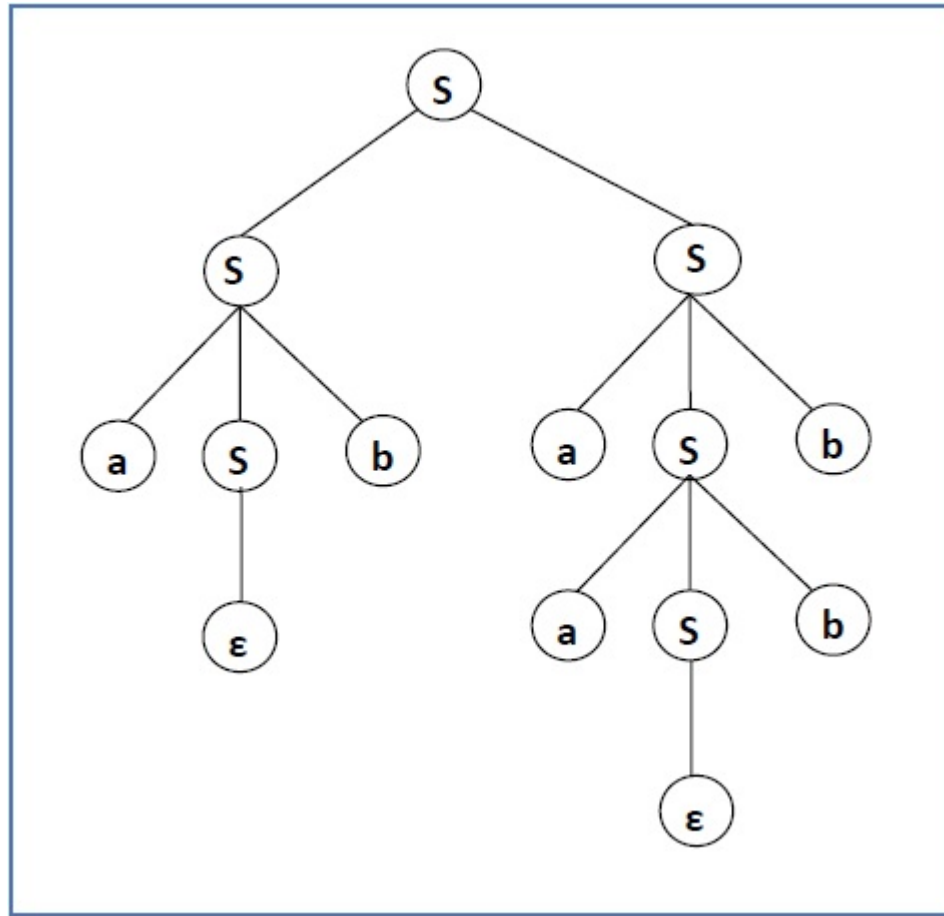
Example

Let a CFG $\{N,T,P,S\}$ be

$$N = \{S\}, T = \{a, b\}, \text{ Starting symbol} = S, P = S \rightarrow SS \mid aSb \mid \varepsilon$$

One derivation from the above CFG is “abaabb”

$$S \rightarrow SS \rightarrow aSbS \rightarrow abS \rightarrow abaSb \rightarrow abaaSbb \rightarrow abaabb$$



Sentential Form and Partial Derivation Tree

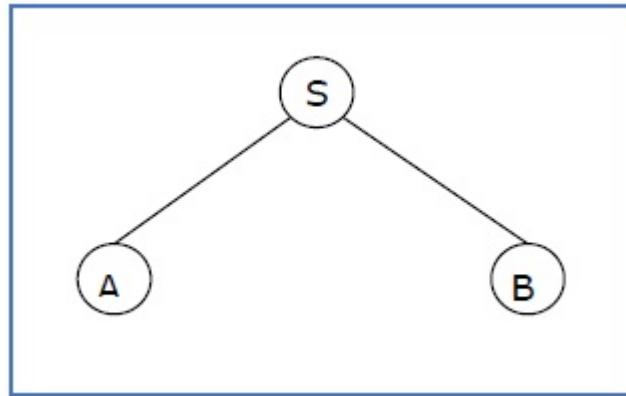
A partial derivation tree is a sub-tree of a derivation tree/parse tree such that either all of its children are in the sub-tree or none of them are in the sub-tree.

Example

If in any CFG the productions are –

$$S \rightarrow AB, A \rightarrow aaA \mid \epsilon, B \rightarrow Bb \mid \epsilon$$

the partial derivation tree can be the following –



If a partial derivation tree contains the root S, it is called a **sentential form**. The above sub-tree is also in sentential form.

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

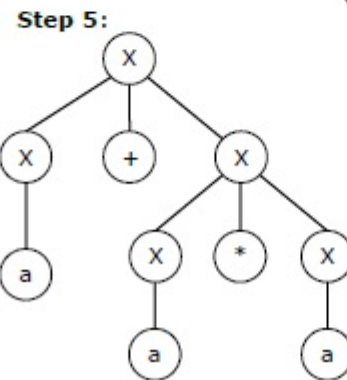
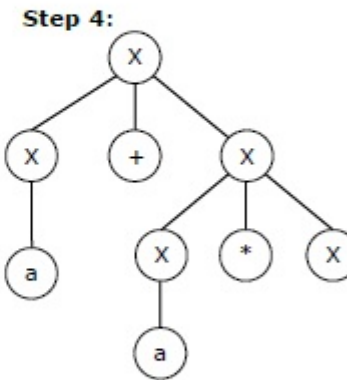
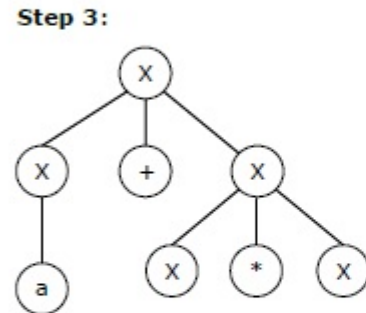
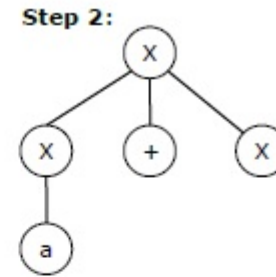
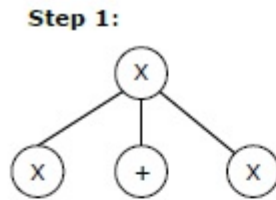
$$X \rightarrow X+X \mid X^*X \mid X \mid a$$

over an alphabet {a}.

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

$$X \rightarrow X+X \rightarrow a+X \rightarrow a + X^*X \rightarrow a+a^*X \rightarrow a+a^*a$$

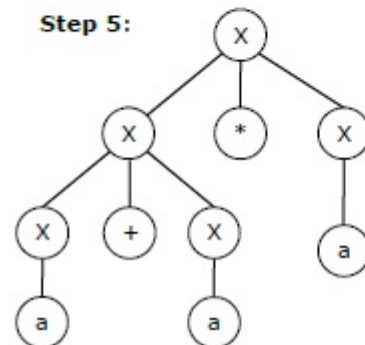
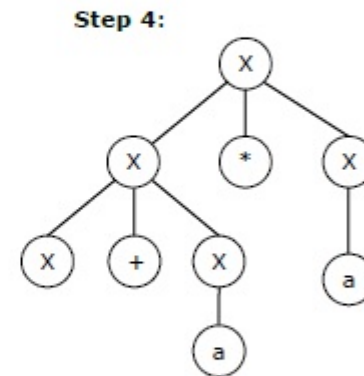
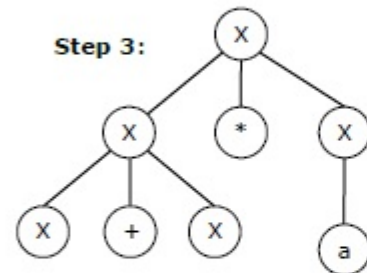
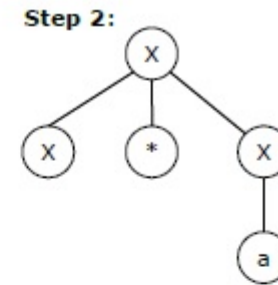
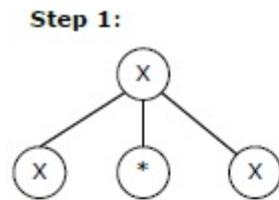
The stepwise derivation of the above string is shown as below –



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

$$X \rightarrow X^*X \rightarrow X^*a \rightarrow X+X^*a \rightarrow X+a^*a \rightarrow a+a^*a$$

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



Left and Right Recursive Grammars

In a context-free grammar G , if there is a production in the form $X \rightarrow Xa$ where X is a non-terminal and 'a' is a string of terminals, it is called a **left recursive production**. The grammar having a left recursive production is called a **left recursive grammar**.

And if in a context-free grammar G , if there is a production in the form $X \rightarrow aX$ where X is a non-terminal and 'a' is a string of terminals, it is called a **right recursive production**. The grammar having a right recursive production is called a **right recursive grammar**.