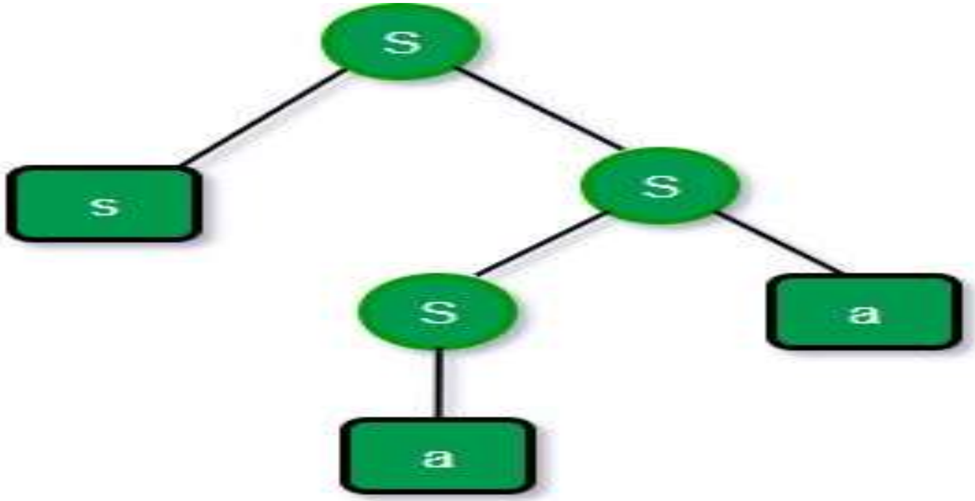
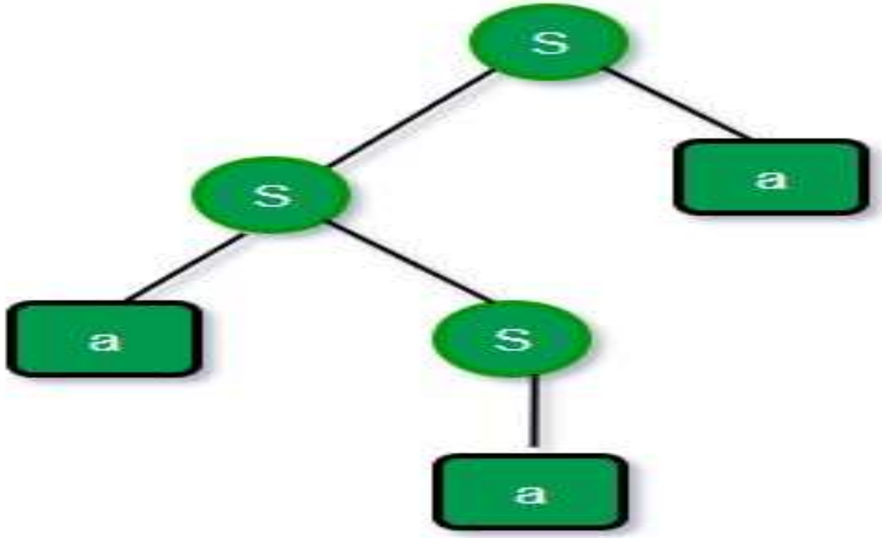
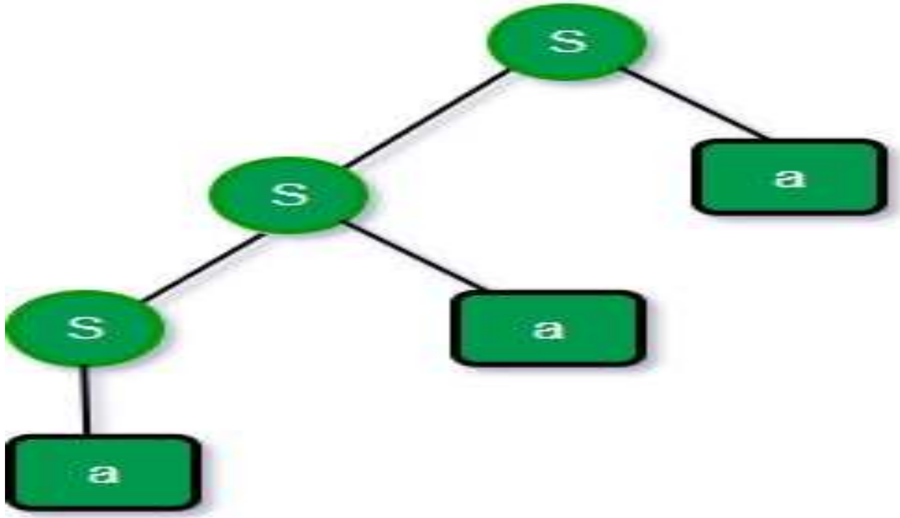
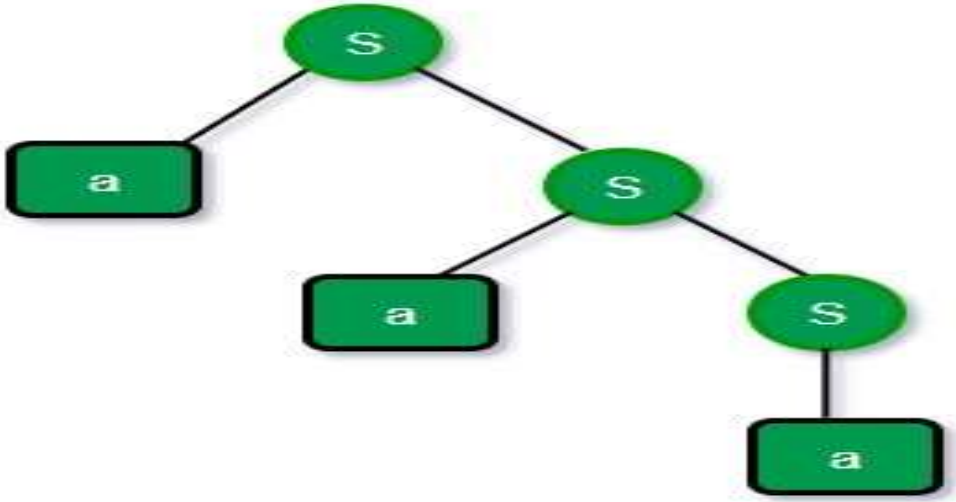
S -> aS | Sa | a

Grammar Check AMBIGUOUS OR NOT









Grammar Check AMBIGUOUS OR NOT

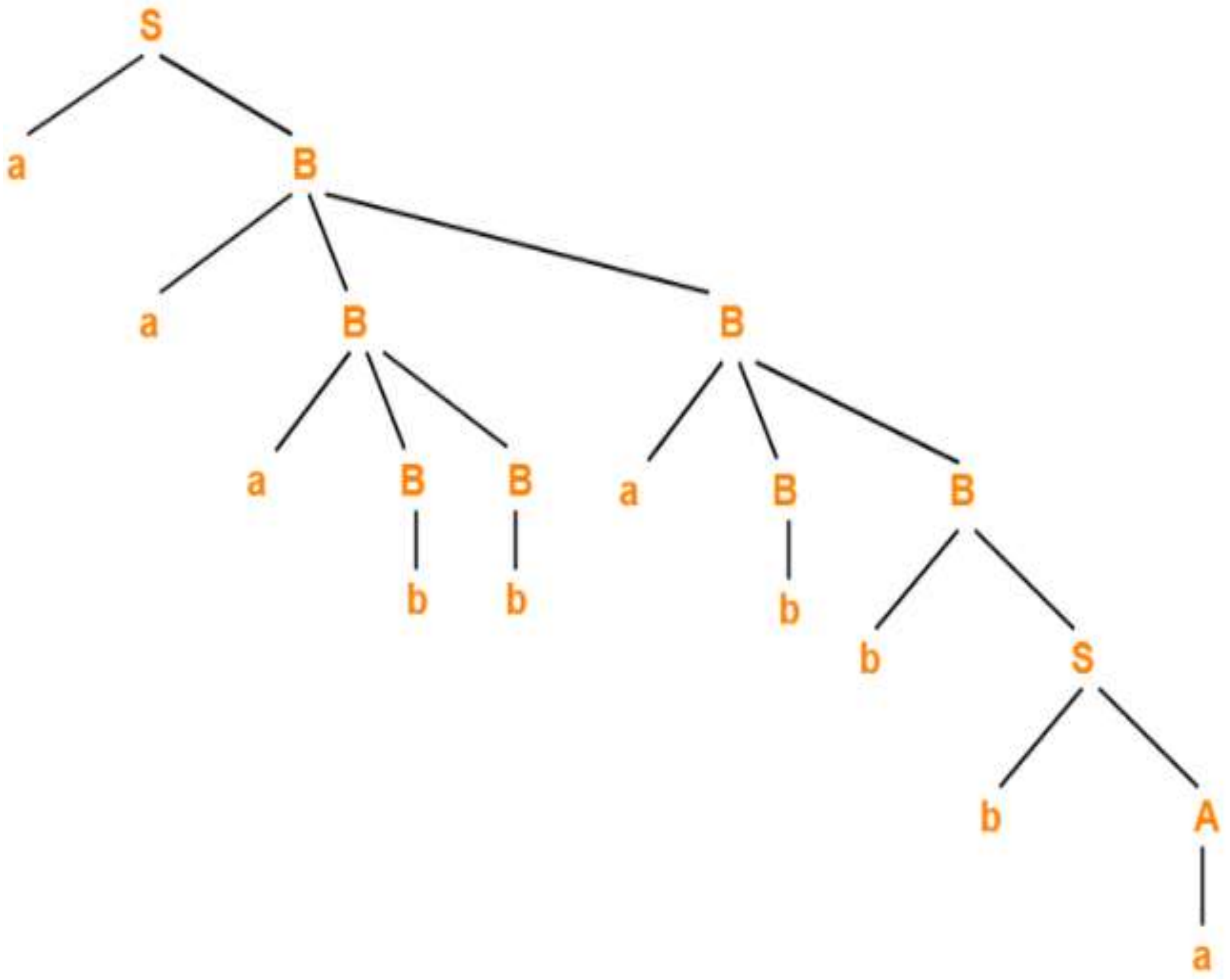
1. S → aB / bA

   S → aS / bAA / a

   B → bS / aBB / b

- Let us consider a string w = aaabbabbba
- Now, let us derive the string w using leftmost derivation.

S → a**B**
→ aa**B**B (Using B → aBB)
→ aaa**B**BB (Using B → aBB)
→ aaab**B**B (Using B → b)
→ aaabb**B** (Using B → b)
→ aaabba**B**B (Using B → aBB)
→ aaabbab**B** (Using B → b)
→ aaabbabb**S** (Using B → bS)
→ aaabbabbb**A** (Using S → bA)
→ aaabbabbba (Using A → a)



Left Recursion Examples

S->Sab/Scd/Sef/g/h

$$S \rightarrow \overset{\alpha_1}{S\,ab} \;/\; \overset{\alpha_2}{S\,cd} \;/\; \overset{\alpha_3}{S\,ef} \;/\; \overset{\beta_1}{g} \;/\; \overset{\beta_2}{h}$$

$$S \rightarrow gS' / hS'$$

$$S' \rightarrow \epsilon / abS' / cdS' / efS'$$

Left Recursion Examples

$\mathbf{A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \ldots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n}$

$\mathbf{A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \ldots \mid \beta_n A'}$

$\mathbf{A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \ldots \mid \alpha_m A' \mid \in}$

$\mathbf{A \rightarrow A\alpha \mid \beta}$

To the following:

$\mathbf{A \rightarrow \beta A'}$

$\mathbf{A' \rightarrow \alpha A' \mid \in}$

Left Recursion Examples

$A \rightarrow A\alpha$ / β
$A \rightarrow \beta A'$
$A' \rightarrow \alpha A'$ / ϵ

- $A \rightarrow ABd$ / Aa / a
- $B \rightarrow Be$ / b

- $A \rightarrow aA'$
- $A' \rightarrow BdA'$ / aA' / ϵ
- $B \rightarrow bB'$
- $B' \rightarrow eB'$ / ϵ

Left Recursion Examples

$A \rightarrow A\alpha$ / β
$A \rightarrow \beta A'$
$A' \rightarrow \alpha A'$ / ϵ

- $E \rightarrow E + E$ / $E \times E$ / a

-

- $E \rightarrow aA$
- $A \rightarrow +EA$ / xEA / ϵ

Left Recursion Examples

$A \rightarrow A\alpha$ / β
$A \rightarrow \beta A'$
$A' \rightarrow \alpha A'$ / ϵ

- $E \rightarrow E + T$ / T
- $T \rightarrow T$ x F / F
- $F \rightarrow id$

- $E \rightarrow TE'$
- $E' \rightarrow +TE'$ / ϵ
- $T \rightarrow FT'$
- $T' \rightarrow xFT'$ / ϵ
- $F \rightarrow id$

Left Recursion Examples

$A \rightarrow A\alpha \;/\; \beta$

$A \rightarrow \beta A'$

$A' \rightarrow \alpha A' \;/\; \epsilon$

$S \rightarrow (L) \;/\; a$

$L \rightarrow L\, ,\, S \;/\; S$

- $S \rightarrow (L) \;/\; a$
- $L \rightarrow SL'$
- $L' \rightarrow\, ,SL' \;/\; \epsilon$

Left Recursion Examples

$A \rightarrow A\alpha \;/\; \beta$

$A \rightarrow \beta A'$

$A' \rightarrow \alpha A' \;/\; \epsilon$

- $S \rightarrow S0S1S \;/\; 01$

- $S \rightarrow 01A$
- $A \rightarrow 0S1SA \;/\; \epsilon$

Left Recursion Examples

$A \rightarrow A\alpha$ / β

$A \rightarrow \beta A'$

$A' \rightarrow \alpha A'$ / $\in$

- $S \rightarrow A$

  $A \rightarrow Ad$ / Ae / aB / ac

  $B \rightarrow bBc$ / f

$S \rightarrow A$

$A \rightarrow aBA'$ / acA'

$A' \rightarrow dA'$ / eA' / $\in$

$B \rightarrow bBc$ / f

Left Recursion Examples

$A \rightarrow A\alpha$ / β

$A \rightarrow \beta A'$

$A' \rightarrow \alpha A'$ / $\in$

$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \ldots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n$

$\rightarrow \beta_1 A' \mid \beta_2 A' \mid \ldots \mid \beta_n A'$

$A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \ldots \mid \alpha_m A' \mid \in$

Left Recursion Examples

$A \rightarrow A\alpha$ / β

$A \rightarrow \beta A'$

$A' \rightarrow \alpha A'$ / $\in$

- $A \rightarrow AA\alpha$ / β
- $A \rightarrow \beta A'$
- $A' \rightarrow A\alpha A'$ / $\in$

Example

The production set

S => Aα | β

A => Sd

after applying the above algorithm, should become

S => Aα | β

A => Aαd | βd

and then, remove immediate left recursion using the first technique.

A => βdA’

A' => αdA' | ε

Now none of the production has either direct or indirect left recursion.

Left Recursion Examples

A → Aα / β
A → βA’
A’ → αA’ / ∈

1. A → Ba / Aa / c

   B → Bb / Ab / d

- **Step-01:**

First let us eliminate left recursion from A → Ba / Aa / c

Eliminating left recursion from here, we get-

A → BaA’ / cA’

A’ → aA’ / ∈

Now, given grammar becomes-

A → BaA’ / cA’

A’ → aA’ / ∈

B → Bb / Ab / d

1. A → Ba / Aa / c

   B → Bb / Ab / d

- **Step-02:**
- Substituting the productions of A in B → Ab, we get the following grammar-
- A → BaA’ / cA’
- A’ → aA’ / ∈
- B → Bb / BaA’b / cA’b / d
- **Step-03:**

Left Recursion Examples

Now, eliminating left recursion from the productions of B, we get the following grammar-

- A → BaA’ / cA’
- A’ → aA’ / ∈
- B → cA’bB’ / dB’
- B’ → bB’ / aA’bB’ / ∈

$$\mathbf{A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \ldots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n}$$

$$\mathbf{A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \ldots \mid \beta_n A'}$$

$$\mathbf{A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \ldots \mid \alpha_m A' \mid \in}$$

This is the final grammar after eliminating left recursion.

Left Recursion Examples

$A \rightarrow A\alpha \ / \ \beta$

$A \rightarrow \beta A'$

$A' \rightarrow \alpha A' \ / \ \in$

2. X → XSb / Sa / b

   S → Sb / Xa / a

- **Step-01:**
-
- First let us eliminate left recursion from X → XSb / Sa / b
-
- Eliminating left recursion from here, we get-
- X → SaX' / bX'
- X' → SbX' / ∈
-
- Now, given grammar becomes-
- X → SaX' / bX'
- X' → SbX' / ∈
- S → Sb / Xa / a

2. X → XSb / Sa / b

   S → Sb / Xa / a

- **Step-02:**
- Substituting the productions of X in S → Xa, we get the following grammar-
- X → SaX' / bX'
- X' → SbX' / ∈
- S → Sb / SaX'a / bX'a / a
- **Step-03:**

Left Recursion Examples

- Now, eliminating left recursion from the productions of S, we get the following grammar-
- X → SaX' / bX'
- X' → SbX' / ∈
- S → bX'aS' / aS'
- S' → bS' / aX'aS' / ∈
- This is the final grammar after eliminating left recursion.

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \dots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

$$\rightarrow \beta_1 A' \mid \beta_2 A' \mid \dots \mid \beta_n A'$$

$$A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \dots \mid \alpha_m A' \mid \in$$

Left Recursion Examples

$A \rightarrow A\alpha \,/\, \beta$

$A \rightarrow \beta A'$

$A' \rightarrow \alpha A' \,/\, \epsilon$

$S \rightarrow Aa \,/\, b$

$A \rightarrow Ac \,/\, Sd \,/\, \epsilon$

- **Step-01:**

- First let us eliminate left recursion from
- $S \rightarrow Aa \,/\, b$
- This is already free from left recursion.

Left Recursion Examples

$S \rightarrow Aa \,/\, b$

$A \rightarrow Ac \,/\, Sd \,/\, \epsilon$

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \dots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

$$\rightarrow \beta_1 A' \mid \beta_2 A' \mid \dots \mid \beta_n A'$$

$$A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \dots \mid \alpha_m A' \mid \epsilon$$

- **Step-02:**

- Substituting the productions of S in $A \rightarrow Sd$, we get the following grammar-
- $S \rightarrow Aa \,/\, b$
- $A \rightarrow Ac \,/\, Aad \,/\, bd \,/\, \epsilon$

- **Step-03:**

- Now, eliminating left recursion from the productions of A, we get the following grammar-
- $S \rightarrow Aa \,/\, b$
- $A \rightarrow bdA' \,/\, A'$
- $A' \rightarrow cA' \,/\, adA' \,/\, \epsilon$

Left Recursion Examples

A → Aα / β

A → βA’

A’ → αA’ / ∈

- Left Factoring Examples.

A->αβ1 | αβ2

A -> αA’
A’ -> β1 | β2

- A → aAB / aBc / aAc
  - A → aA’
  - A’ → AB / Bc / Ac
  - A → aA’
  - A’ → AD / Bc
  - D → B / c
- S → bSSaaS / bSSaSb / bSb / a
  - S → bSS’ / a
  - S’ → SaaS / SaSb / b
  - S → bSS’ / a
  - S’ → SaA / b
  - A → aS / Sb

- Left Factoring

A->αβ1 | αβ2
A -> αA'
A' -> β1 | β2

- S → iEtS / iEtSeS / a
- E → b

- S → iEtS / iEtSeS / a
- E → b

- S → iEtSS' / a
- S' → eS / ∈
- E → b

- Left Factoring

A->αβ1 | αβ2

A -> αA'

A' -> β1 | β2

- $S \rightarrow aSSbS$ / $aSaSb$ / abb / b
  - $S \rightarrow aS'$ / b
  - $S' \rightarrow SSbS$ / $SaSb$ / bb
  - $S \rightarrow aS'$ / b
  - $S' \rightarrow SA$ / bb
  - $A \rightarrow SbS$ / aSb

- Left Factoring

$A \to \alpha\beta 1 \mid \alpha\beta 2$

$A \to \alpha A'$

$A' \to \beta 1 \mid \beta 2$

$S \to a / ab / abc / abcd$

- $S \to aS'$
- $S' \to b / bc / bcd / \epsilon$

- $S \to aS'$
- $S' \to bA / \epsilon$
- $A \to c / cd / \epsilon$

- $S \to aS'$
- $S' \to bA / \epsilon$
- $A \to cB / \epsilon$
- $B \to d / \epsilon$

- Left Factoring

$A \to \alpha\beta_1 \mid \alpha\beta_2$

$A \to \alpha A'$

$A' \to \beta_1 \mid \beta_2$

$$S \rightarrow aAd \,/\, aB$$

$$A \rightarrow a \,/\, ab$$

$$B \rightarrow ccd \,/\, ddc$$

-

- $S \rightarrow aS'$
- $S' \rightarrow Ad \,/\, B$
- $A \rightarrow aA'$
- $A' \rightarrow b \,/\, \epsilon$
- $B \rightarrow ccd/ddc$

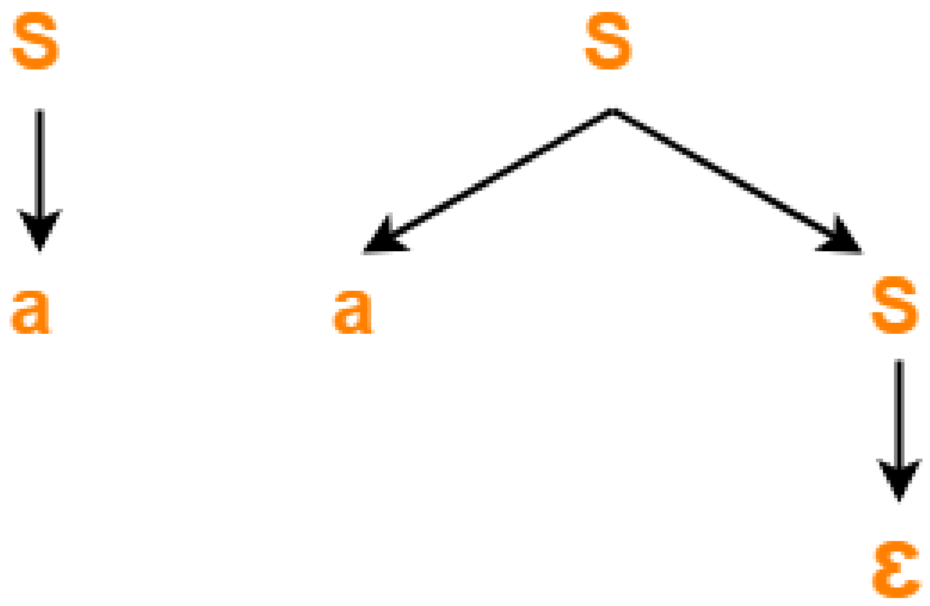
- Left Factoring

$A \rightarrow \alpha\beta 1 \mid \alpha\beta 2$

$A \rightarrow \alpha A'$

$A' \rightarrow \beta 1 \mid \beta 2$

Relationship Between Left Recursion, Left Factoring & Ambiguity-Example-01: Ambiguous Grammar With Left Factoring-



Ambiguous Grammar with Left Factoring

Consider the following grammar-

$S \rightarrow aS / a / \in$

Clearly, this grammar has left factoring.

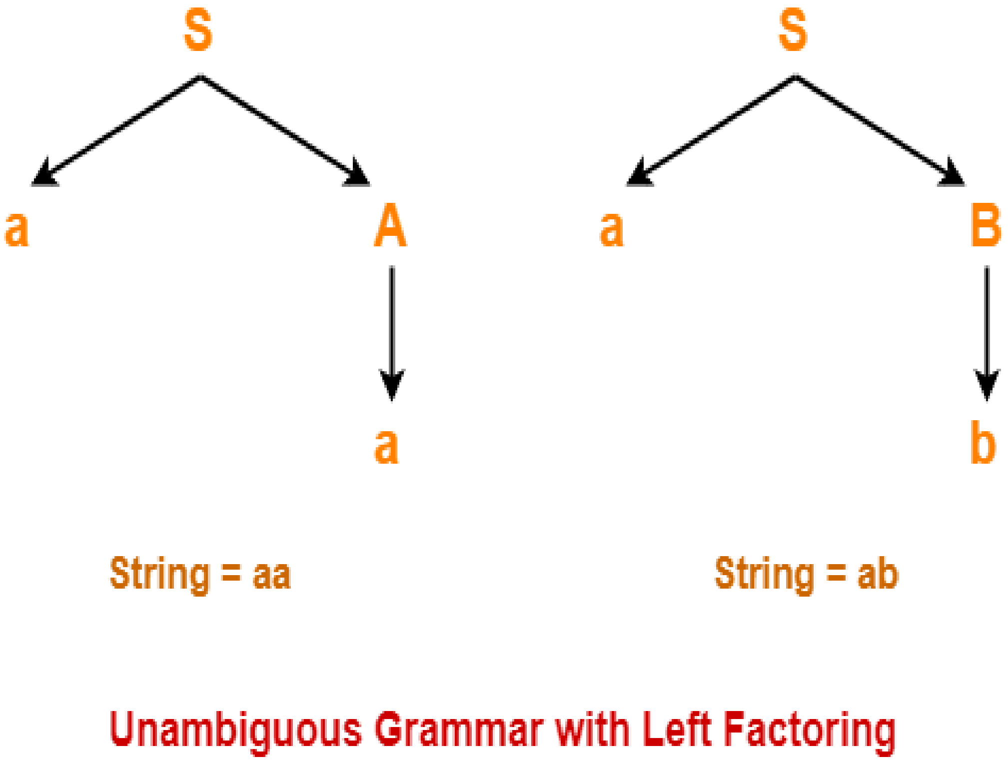
Now, let us draw parse trees for the string

w = a

Clearly,

- Two different parse trees exist for the string
- w = a.
- Therefore, the grammar is ambiguous.

Relationship Between Left Recursion, Left Factoring & Ambiguity-Unambiguous Grammar With Left Factoring



Consider the following grammar-

S → aA / aB

A → a

B → b

Clearly, this grammar has left factoring.

The language generated by this grammar consists of only two strings

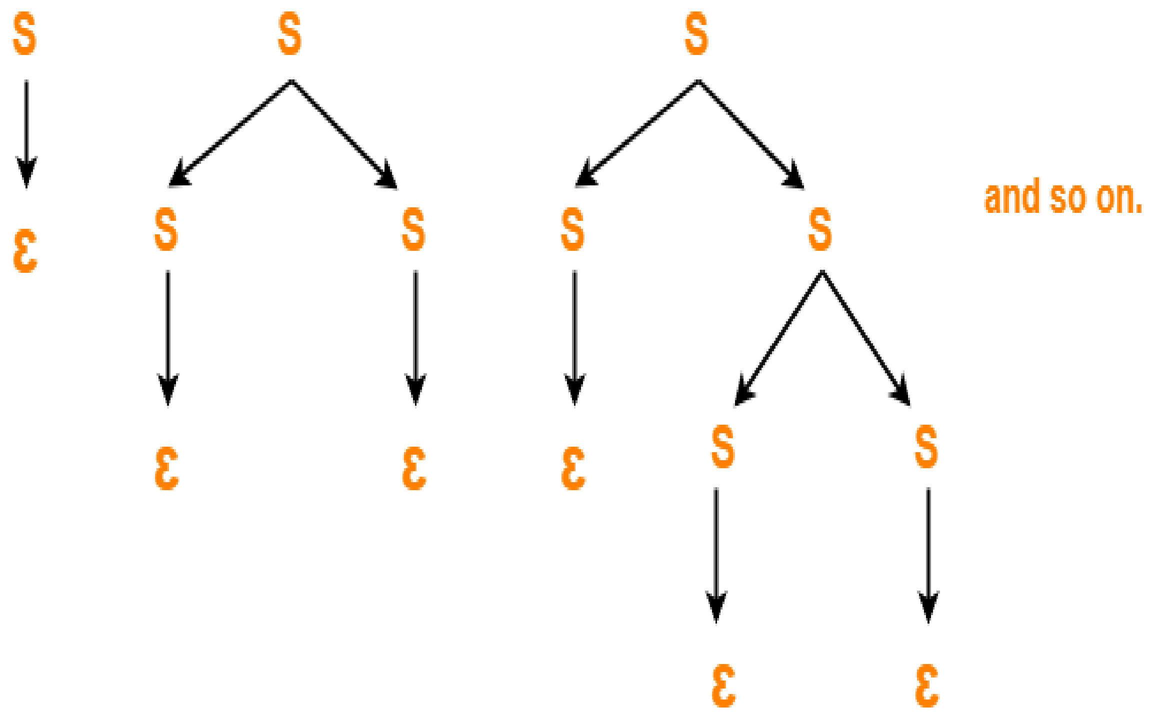
L(G) = { aa , ab}.

Now, let us draw parse trees for these strings-

Clearly,

- A unique parse tree exists for both the strings.
- Therefore, the grammar is unambiguous.

Relationship Between Left Recursion, Left Factoring & Ambiguity-Ambiguous Grammar With Left Recursion-



Ambiguous Grammar with Left Recursion

Consider the following grammar-

S → SS / ∈

Clearly, this grammar has left recursion.

Now, let us draw parse trees for the string

w = ∈

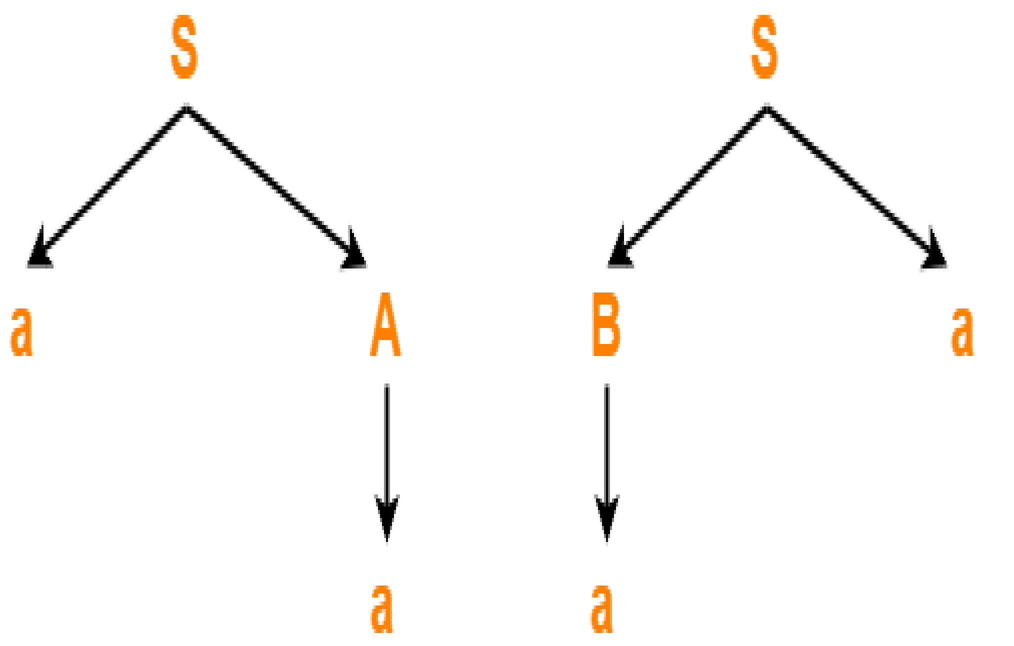
Clearly,

- Infinite parse trees exist for the string
- w = ∈.
- Therefore, the grammar is ambiguous.

- Consider the following grammar-
  - $S \rightarrow Sa \,/\, \in$

- Clearly, this grammar has left recursion.
- A unique parse tree exists for all the strings that can be generated from the grammar.
- Therefore, the grammar is unambiguous.

Example-05: Ambiguous Grammar Without Left Recursion & Without Left Factoring-



Ambiguous Grammar Without Left Recursion & Without Left Factoring

Consider the following grammar-

$S \rightarrow aA \ / \ Ba$

$A \rightarrow a$

$B \rightarrow a$

Clearly, this grammar has neither left recursion nor left factoring.

Now, let us draw parse trees for the string

w = aa

Clearly,

- Two different parse trees exist for the string w = aa.
- Therefore, the grammar is ambiguous.

Example-06: Unambiguous Grammar With Both Left Recursion & Left Factoring-

Consider the following grammar-

$S \rightarrow Sa \,/\, \epsilon \,/\, bB \,/\, bD$

$B \rightarrow b$

$D \rightarrow d$

Clearly, this grammar has both left recursion and left factoring.

A unique parse tree exists for all the strings that can be generated from the grammar.

Therefore, the grammar is unambiguous.

To gain better understanding about relationship between left recursion, left factoring and ambiguity-