Module - 3.

Definitions :

- A grammar G is 4-tuple or quadruple $G = (V, T, P, S)$ where

  V is set of variables or non-terminals

  T is set of terminals

  P is set of productions. Each productions is of the form $\alpha \rightarrow \beta$ where α is a string in $(V \cup T)^+$ (no ε) hence ε cannot occur on left side of any production.
  β is a string in $(V \cup T)^*$ hence includes ε also.

  S is the start symbol.

Grammar from Finite Automata.

1) Obtain grammar to generate string consisting of any number of a's.

S is ~~start~~ final state $\Rightarrow$ $S \to \varepsilon$

$\delta(S, a) = S$ $\Rightarrow$ $S \to aS$

$G = (V, T, P, S)$

where

$V = \{S\}$

$T = \{a\}$

$P = \{S \to aS \mid \varepsilon\}$

$S = S$

2) Obtain grammar to generate strings consisting of atleast one a.

<u>Method 1</u>

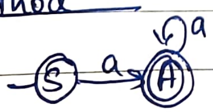
$L = \{a, aa, aaa, aaaa, \ldots\}$

$S \to a$.

$S \to aS$.

$S \to a \mid aS$.

<u>Method 2</u>



$\delta(S, a) = A$ $\Rightarrow$ $S \to aA$

$\delta(A, a) = A$ $\Rightarrow$ $A \to aA$

A is final state $\Rightarrow$ $A \to \varepsilon$

$S \to aA$
$A \to aA | \varepsilon$

3) Obtain grammar to generate string consisting of any number of a's and b's.

$\delta(S, a) = S$
$\delta(S, b) = S$
S is final state

$S \to aS$
$S \to bS$
$S \to \varepsilon$.

$G = (V, T, P, S)$
where $V = \{S\}$
$T = \{a, b\}$
$P = \{ S \to aS | bS | \varepsilon \}$.
$S = S$

4. Obtain grammer to generate strings consisting of atleast two a's.

$L = \{aa, aaa, aaaa, \ldots\}$

$S \to aa$ string length 2
$S \to aS$. string length >2

$G = (V, T, P, S)$.
where $V = \{S\}$
$T = \{a\}$
$P = \{S \to aS | aa\}$
$S = S$

5) Obtain grammar to generate string consisting of even number of a's

$L = \{ \varepsilon, aa, aaaa, aaaaaa, \ldots \}$

$S \rightarrow aaS$ string length of 2, 4, 6, ...

$S \rightarrow \varepsilon$ string length 0 or to terminate

$G = (V, T, P, S)$

where $V = \{S\}$

$T = \{a\}$

$P = \{ S \rightarrow aaS \mid \varepsilon \}$

$S = S$.

6) Obtain grammar to generate string consisting of multiple of three a's

$L = \{ \varepsilon, aaa, aaaaaa, aaa\,aaa\,aaa, \ldots \}$.

$S \rightarrow \varepsilon$

$S \rightarrow aaaS$.

$G = (V, T, P, S)$

where $V = \{S\}$

$T = \{a\}$

$P = \{ S \rightarrow aaaS \mid \varepsilon \}$

$S = S$

7) Obtain grammar to generate strings of a's & b's such that string length is multiple of 3.

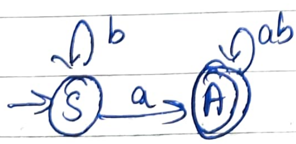
$L = \{ aaa, aba, aab, abb, baa, bab, bba, bbb, aaaaaa, \ldots \}$.

$S \to \varepsilon$
$S \to AAAS$ $\quad |x| =$ multiple of 3
$A \to a|b$.

$G = (V, T, P, S)$
where $V = \{S, A\}$
$T = \{a, b\}$
$P = \{ S \to AAAS / \varepsilon,$
$A \to a|b \}$.
$S = S$.

8) Obtain grammar to generate string consisting of any number of a's & b's with at least one a.

$S \to a$ $\quad$ atleast one a
$S \to aS$ $\quad$ } any length of a's & b's
$S \to bS$



$\delta(S, a) = A \Rightarrow S \to aA$
$\delta(S, b) = S \Rightarrow S \to bS$.
$\delta(A, a) = A \Rightarrow A \to aA$
$\delta(A, b) = A \Rightarrow A \to bA$.
A is final state $\Rightarrow A \to \varepsilon$.

$G = (V, T, P, S)$
where $V = \{S, A\}$
$T = \{a, b\}$
$P = \{ S \to aA / bS \; ; \; A \to aA / bA / \varepsilon \}$
$S = S$

9) Obtain grammar to generate strings consisting of any number of a's & b's with atleast one b.

Similar to Q8)

$S \to bA / aS$
$A \to aA / bA / \epsilon$

$G = (V, T, P, S)$
where $V = \{S, A\}$
$T = \{a, b\}$
$P = \{ S \to bA/aS \quad A \to aA/bA/\epsilon \}$
$S = S$.

10) Obtain grammar to generate strings consisting of any number of a's and b's with atleast one a or atleast one b.

$S \to a/b$ terminating clause
$S \to aS$ any no of a
$S \to bS$ any no of b

(define grammar for all)

11) Obtain grammar to accept the following languages:

(i) $L = \{ w : |w| \bmod 3 > 0$ where $w \in \{a\}^* \}$
for mod 3 > 0 remainder is 1 or 2.
$S \to a / aa$. minimum string that can be accepted.
$S \to aaaS$. 3 a's with 1 or 2 will not be multiple.

$S \to a / aa / aaaS$.

classmate

or

$S \to aA$
$A \to aB$
$B \to aS$
$A \to \varepsilon$
$B \to \varepsilon$

$\Rightarrow$

$S \to aA$
$A \to aB / \varepsilon$
$B \to aS / \varepsilon$.

Grammar from Regular Expression.
Instead of DFA we can write RE & convert to grammar

12) Obtain grammar to generate strings of a's and b's having a substring ab.

substring ab $\Rightarrow$ $\underbrace{(a+b)^*}_{A}\, ab\, \underbrace{(a+b)^*}_{A}$.

A ab A.

$S \to AabA$.
$A \to aA / bA / \varepsilon$.

13) Obtain grammar to generate strings of a's & b's ending with string ab.

RE $= (a+b)^* ab$.

$S \to Aab$
$A \to aA / bA / \varepsilon$.

Date
Page

14) Obtain grammar to generate strings of a's and b's starting with ab.

$RE = ab\cdot(a+b)^*$

$S \rightarrow abA$
$A \rightarrow aA / bA / \varepsilon$

15) Obtain grammar to generate the following language
$L = \{w : n_a(w) \bmod 2 = 0$ where $w \in \{a,b\}^*\}$

$$RE = (\underset{S}{b^*}\, a\, \underset{S}{b^*}\, a\, \underset{S}{b^*})^*$$

for b^* : $S \rightarrow bS / \varepsilon$.

$S \rightarrow SaSaS$

$\Rightarrow$ $S \rightarrow \varepsilon | bS | SaSaS$.

Grammars for other languages

Obtain grammar for the following languages:

16) $L = \{a^n b^n : n \geq 0\}$.

equal number of a's and b's in string. (including null)

$S \rightarrow aSb / \varepsilon$.

17) $L = \{a^n b^n : n \geq 1\}$.
Equal no of a's & b's in string (excluding null)

$S \rightarrow ab / aSb$ $\boxed{\text{or}}$ $S \rightarrow aSb / bSa / \varepsilon$

18) $L = \{a^{n+1} b^n : n \geq 0\}$.

One extra a compared to b.

$S \to a \mid aSb$

22) $L = \{a^n b^{n+1} ; n \geq 0\}$.

One extra b compared to a $\quad S \to b \mid aSb$.

23) $L = \{a^n b^{n+2} : n \geq 0\}$

string which has 2 b's more than no of a's.

$S \Rightarrow bb \mid aSb$

24) $L = \{a^n b^{2n} : n \geq 0\}$.

No of b's in string is twice as much as a's.

$S \to \varepsilon \mid aSbb$ $\quad$ { for every a's there is 2 b's }

25) $L = \{0^m 1^m 2^n \mid m \geq 1$ and $n \geq 0\}$.

no of 0's & 1's is same with atleast one and no of 2 is zero or more.

$\underbrace{0^m 1^m}_{A} \underbrace{2^n}_{B}$ $\quad$ where A = no of zero's followed by 1's.

$S \to AB$.
$A \to 01 \mid 0A1$
$B \to \varepsilon \mid 2B$ $\quad$ (zero or more 2's).

classmate
Date
Page

26) $L = \{ 0^i 1^j \mid i \neq j, i \geq 0 \text{ and } j \geq 0 \}$.

no of 0's is not equal to no of 1's.

$S \rightarrow 0S1$ (generates $0^n 1^n$)
$S \rightarrow A$ (generates more 0's than 1's)
$S \rightarrow B$ (generates more 1's than 0's)
$A \rightarrow 0A \mid 0$ (atleast one zero)
$B \rightarrow 1B \mid 1$. (atleast one one)

27) Ob $L = \{ a^{n+2} b^m \mid n \geq 0 \text{ and } m > n \}$

$m>n$	
n	m
0	1 or more
1	2 or more
2	3 or more

for $n = 0$ $a^2 b^{\text{1 or more}}$ $\Rightarrow$ $aabb^*$

$L = \{ \underbrace{aabb^*}_{a^n b^n}, \underbrace{aaabbb^*}_{a^n b^n}, \underbrace{aaaabbbb^*}_{a^n b^n}, \ldots \}$

$S \rightarrow aAB$.
$A \rightarrow aAb \mid ab$ (generates $a^n b^n$)
$B \rightarrow bB \mid \varepsilon$ (generate 0 or more b's)

28) $L = \{ a^n b^m \mid n \geq 0, m > n \}$.

for $n=0, m \geq 1$ $\Rightarrow \varepsilon bb^*$
$n=1, m \geq 2$ $\Rightarrow abbb^*$
$n=2, m \geq 3$ $\Rightarrow aabbbb^*$

$L = \{ bb^*, abbb^*, aabbbb^*, \ldots \}$.

classmate

Date
Page

$S \to AB$

$A \to aAb \mid \varepsilon$ (generate $a^n b^n$)

$B \to bB \mid b$ (more b's than a)

Q.29) $L = \{ a^n b^{n-3} \mid n \geq 3 \}$.

for : $n = 3$, $n = 4$, $n = 5$, $n = 6$

$L = \{aaa, aaaab, aaaaabb, aaaaaabbb, \ldots\}$.

It is observed that string aaa followed by $a^n b^n$.

$S \to aaaA$

$A \to aAb \mid \varepsilon$.

30) $L = \{ a^n b^m c^k \mid n + 2m = k$ for $n \geq 0, m \geq 0\}$

$L = \{ a^n b^m c^{n+2m} \}$ substituting for k.

$L = \{ a^n b^m c^n c^{2m} \mid n \geq 0, m \geq 0 \}$ (splitting c^{n+2m})

$\Rightarrow L = \{ w \mid a^n b^m c^{2m} c^n$ where $n \geq 0, m \geq 0 \}$.

for $b^m c^{2m}$: $A \to bAcc \mid \varepsilon$

where A is present between a^n & c^n.

$S \to aSc \mid A$. (to generate a^n & c^n first)

Productions $\Rightarrow$ $S \to aSc \mid A$

$A \to bAcc \mid \varepsilon$.

classmate
Date
Page

Leftmost Derivation.

In the derivation process if a leftmost variable is replaced at every step, then the derivation is said to be leftmost

Eg:

① derive id + id * id from production rules
$E \rightarrow E+E \mid E-E \mid E*E \mid E/E \mid E.E \mid E \wedge E \mid id.$

$E \rightarrow E + E$
$\rightarrow id + E$
$\rightarrow id + E * E$
$\rightarrow id + id * E$
$\rightarrow id + id * id$

② derive 'aaabbabbba' from
$S \rightarrow aB \mid bA$
$A \rightarrow aS \mid bAA \mid a$
$B \rightarrow bS \mid aBB \mid b.$

$S \rightarrow aB$	$(S \rightarrow aB)$
$\rightarrow aaBB$	$(B \rightarrow aBB)$
$\rightarrow aaaBBB$	$(B \rightarrow aBB)$
$\rightarrow aaabBB$	$(B \rightarrow b)$
$\rightarrow aaabbB$	$(B \rightarrow b)$
$\rightarrow aaabbaBB$	$(B \rightarrow aBB)$
$\rightarrow aaabbabB$	$(B \rightarrow b)$
$\rightarrow aaabbabbS$	$(B \rightarrow bS)$
$\rightarrow aaabbabbbA$	$(S \rightarrow bA)$
$\rightarrow aaabbabbba.$	$(A \rightarrow a)$

Rightmost derivation.

In the derivation process if a right most variable is replaced at every step, then the derivation is said to be rightmost.

Eg :

① derive id+id *id from

$E \rightarrow E+E \mid E-E \mid E*E \mid E/E \mid E\hat{}E \mid id.$

$E \rightarrow E+E$

$\rightarrow E+E*E$

$\rightarrow E+E*id$

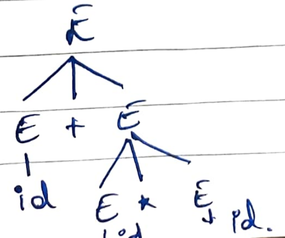
$\rightarrow E+id*id$

$\rightarrow id+id*id.$

Derivation Tree (Parse Tree)

Let $G = (V, T, P, S)$ be a CFG (context free grammar). The tree is derivation tree with following properties.

1. The root has the lable S
2. Every vertex has a lable which is in $(V \cup T \cup \epsilon)$
3. Every leaf node has label from T and an interior vertex has a label from V.
4. If vertex is labeled A & if $X_1, X_2, \ldots, X_n$ are all children of A from left, then $A \rightarrow X_1 X_2 X_3 \ldots X_n$ must be a production in P.



Ambiguous grammar.

Let $G = (V, T, P, S)$ be a context free grammar. A grammar G is ambiguous if and only if there exists at least one string $w \in T^*$ for which two or more different parse tree exists by applying either the leftmost derivation or right most derivation.

1) Consider the grammar shown below from which any arithmetic expression can be obtained.

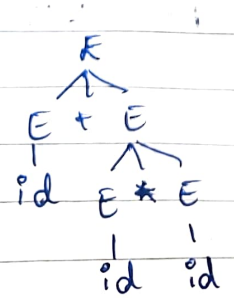
$E \rightarrow E+E$
$E \rightarrow E-E$
$E \rightarrow E*E$
$E \rightarrow E/E$
$E \rightarrow E^\wedge E$
$E \rightarrow id$

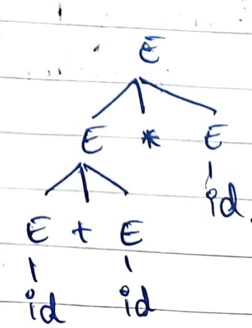
Show that the grammar is ambiguous.

Using leftmost derivation, the string id + id * id can be derived in two ways,

$E \rightarrow E + E$
$\rightarrow id + E$
$\rightarrow id + E * E$
$\rightarrow id + id * E$
$\rightarrow id + id * id$

$E \rightarrow E * E$
$\rightarrow E + E * E$
$\rightarrow id + E * E$
$\rightarrow id + id * E$
$\rightarrow id + id * id$





Since, the two parse trees are different for the same sentence id + id * id by applying leftmost derivation, the grammar is ambiguous.

2) Is the following grammar ambiguous?

$S \rightarrow aS \mid X$

$X \rightarrow aX \mid a$

For string 'aaaa' using leftmost derivation.

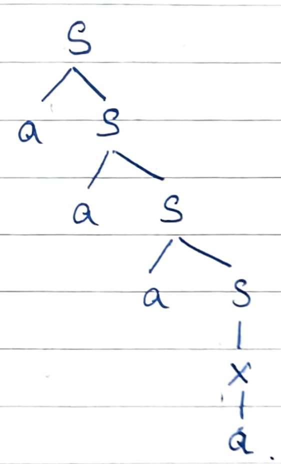
$S \rightarrow aS$ $(S \rightarrow aS)$

$\rightarrow aaS$ $(S \rightarrow aS)$

$\rightarrow aaaS$ $(S \rightarrow aS)$

$\rightarrow aaaX$ $(S \rightarrow X)$

$\rightarrow aaaa$ $(X \rightarrow a)$



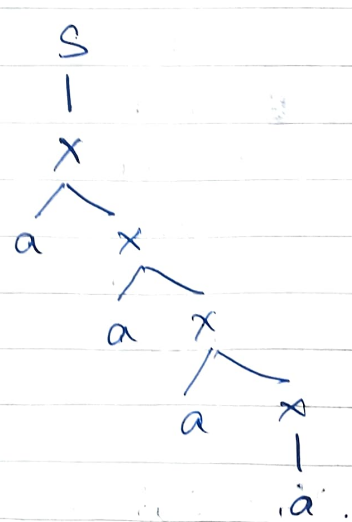
$S \rightarrow X$ $(S \rightarrow X)$

$\rightarrow aX$ $(X \rightarrow aX)$

$\rightarrow aaX$ $(X \rightarrow aX)$

$\rightarrow aaaX$ $(X \rightarrow aX)$

$\rightarrow aaaa$ $(X \rightarrow a)$



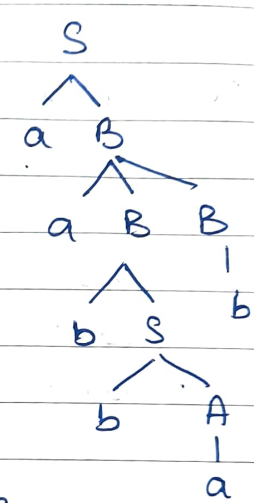
Since, there are two left most derivations for the same sentence aaaa, & different parse trees, the given grammar is ambiguous.

3) Is the following grammar ambiguous

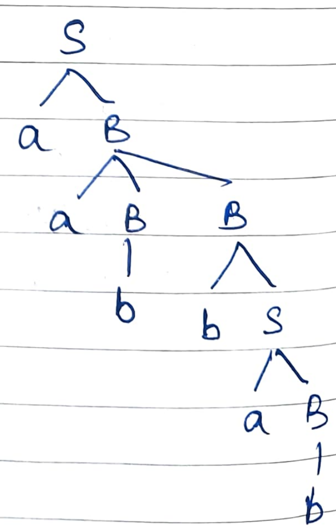
$S \rightarrow aB \mid bA$
$A \rightarrow aS \mid bAA \mid a$
$B \rightarrow bS \mid aBB \mid b$

Consider the string aabbab for derivation.

$S \rightarrow aB$ $(S \rightarrow aB)$
$\rightarrow aaBB$ $(B \rightarrow aBB)$
$\rightarrow aabSB$ $(B \rightarrow bS)$
$\rightarrow aabbAB$ $(S \rightarrow bA)$
$\rightarrow aabbaB$ $(A \rightarrow a)$
$\rightarrow aabbab$ $(B \rightarrow b)$.



Same string can be obtained by applying leftmost derivation in different way.



$S \rightarrow aB$ $(S \rightarrow aB)$
$\rightarrow aaBB$ $(B \rightarrow aBB)$
$\rightarrow aabB$ $(B \rightarrow b)$
$\rightarrow aabbS$ $(B \rightarrow bS)$
$\rightarrow aabbaB$ $(S \rightarrow aB)$
$\rightarrow aabbab$ $(B \rightarrow b)$

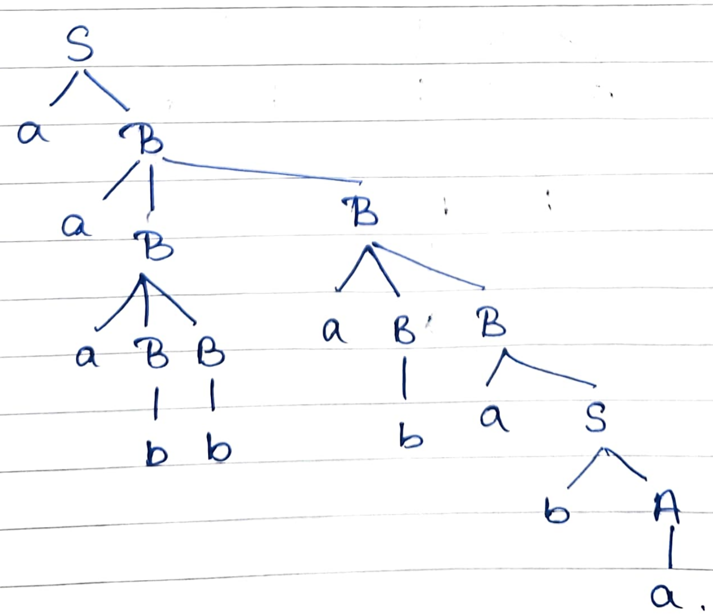
Note that there are two parse trees for string 'aabbab' by applying leftmost derivation and so, the given grammar is ambiguous.

4) Obtain the string aaabbabbba by applying leftmost derivation & the parse tree for the grammar shown below. Is it possible to obtain the same string again by applying leftmost derivation but by selection different productions?

$$S \rightarrow aB \mid bA$$
$$A \rightarrow aS \mid bAA \mid a$$
$$B \rightarrow bS \mid aBB \mid b.$$

leftmost derivation for the string aaabbabbba is shown below:

$S \rightarrow aB$	$(S \rightarrow aB)$
$\rightarrow aaBB$	$(B \rightarrow aBB)$
$\rightarrow aaaBBB$	$(B \rightarrow aBB)$
$\rightarrow aaabBB$	$(B \rightarrow b)$
$\rightarrow aaabbB$	$(B \rightarrow b)$
$\rightarrow aaabbaBB$	$(B \rightarrow aBB)$
$\rightarrow aaabbabB$	$(B \rightarrow b)$
$\rightarrow aaabbabbS$	$(B \rightarrow bS)$
$\rightarrow aaabbabbbA$	$(S \rightarrow bA)$
$\rightarrow aaabbabbba.$	$(A \rightarrow a)$



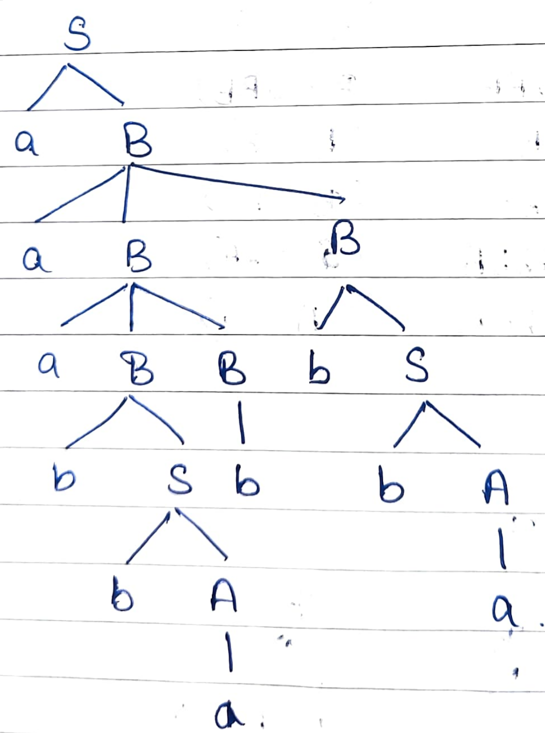
classmate

Date

Page

The leftmost derivation for the same string aaabbabbba but by applying different set of productions is shown below:

$S \to aB$	$(S \to aB)$
$\to aaBB$	$(B \to aBB)$
$\to aaaBBB$	$(B \to aBB)$
$\to aaabSBB$	$(B \to bS)$
$\to aaabbABB$	$(S \to bA)$
$\to aaabbaBB$	$(A \to a)$
$\to aaabbabB$	$(B \to b)$
$\to aaabbabbS$	$(B \to bS)$
$\to aaabbabbbA$	$(S \to bA)$
$\to aaabbabbba.$	$(A \to a)$



5) Is the grammar ambiguous?

$S \to AB \mid aaB$

$A \to a \mid Aa$

$B \to b$.

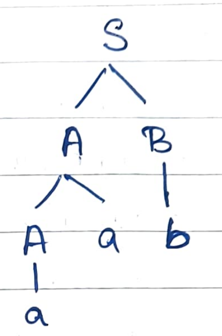
Consider the leftmost derivation for string aab.

$S \to AB$ $(S \to AB)$

$\to AaB$ $(A \to Aa)$

$\to aaB$ $(A \to a)$

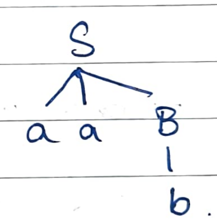
$\to aab$ $(B \to b)$.



Another way of deriving aab is.

$S \to aaB$ $(S \to aaB)$

$\to aab$ $(B \to b)$



Since there are two parse trees for the string aab, the given grammar is ambiguous.

6) Show that the following grammar is ambiguous

$S \rightarrow aSbS$

$S \rightarrow bSaS$

$S \rightarrow \varepsilon$

Consider the leftmost derivation for the string aababb

$S \rightarrow aSbS \quad (S \rightarrow aSbS)$

$\rightarrow aaSbSbS \quad (S \rightarrow aSbS)$.

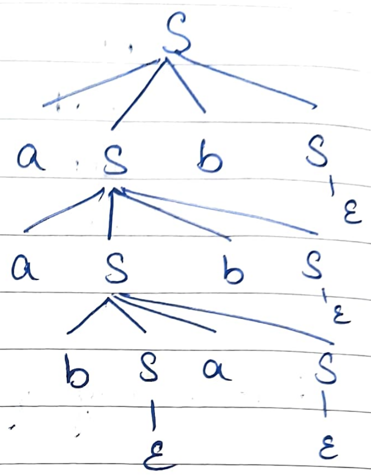
$\rightarrow aabSaSbSbS \quad (S \rightarrow bSaS)$

$\rightarrow aabaSbSbS \quad (S \rightarrow \varepsilon)$

$\rightarrow aababSbS \quad (S \rightarrow \varepsilon)$

$\rightarrow aababbS \quad (S \rightarrow \varepsilon)$

$\rightarrow aababb \quad (S \rightarrow \varepsilon)$



Another way for leftmost derivation

$S \rightarrow aSbS \quad (S \rightarrow aSbS)$

$\rightarrow aaSbSbS \quad (S \rightarrow aSbS)$

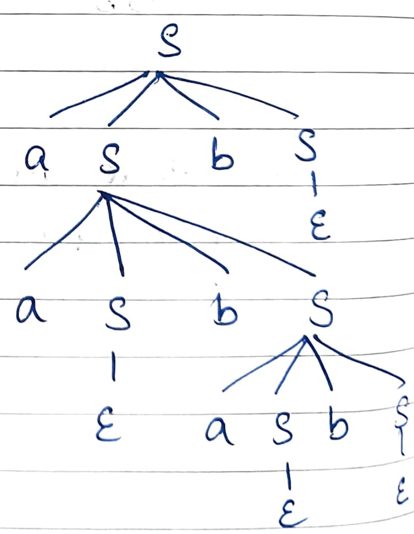
$\rightarrow aabSbS \quad (S \rightarrow \varepsilon)$

$\rightarrow aabaSbSbS \quad (S \rightarrow aSbS)$

$\rightarrow aababSbS \quad (S \rightarrow \varepsilon)$

$\rightarrow aababbS \quad (S \rightarrow \varepsilon)$

$\rightarrow aababb \quad (S \rightarrow \varepsilon)$



Since there are two parse tree for the string aababb, the grammar is ambiguous.