1. Give a context-free grammar for each of the following languages.

a) L= {w | w contains even number of 0's}

= $S \rightarrow 1S|0T|\varepsilon$

$T \rightarrow 0S|1T$

b) L={w | w contains twice as many 1s as 0s}

= $S \rightarrow SS|A|B|C$

$A \rightarrow A011|0A11|01A1|011A|\varepsilon$

$B \rightarrow B110|1B10|11B0|110B|\varepsilon$

$C \rightarrow C101|1C01|10C1|101C|\varepsilon$

c) L={w | w contains even number of 0s and 1s}

= $S \rightarrow 0X|1Y|\varepsilon$

$X \rightarrow 0S|1Z$

$Y \rightarrow 1S|0Z$

$Z \rightarrow 0Y|1X$

d) L={w | where each 0's is followed by at least as many 1's}

= $S \rightarrow AS|\varepsilon$

$A \rightarrow 0A1|1A|\varepsilon$

e) L(G) = { $a^i b^j c^k$ | i, j, k ≥ 0 and i=j or i=k}. ∑={a,b,c}

= $S \rightarrow AC|\ S'$
$A \rightarrow aAb|\varepsilon$
$C \rightarrow cC|\varepsilon$
$S' \rightarrow aBc|B$
$B \rightarrow bB|\varepsilon$

f)) L(G) = { $a^i b^j c^k$ | j > i+k}. ∑={a,b,c}

= $S \rightarrow ABC$

$A \rightarrow aAb|\ \varepsilon$

$B \rightarrow bB|\ b$

$C \rightarrow bCc|\ \varepsilon$

g) $L(G) = \{ a^n b^m \mid 0 < n < m < 3n\}. \sum=\{a,b\}$

= $S \longrightarrow aSbb|aSbbb|Zb$
$Z \longrightarrow aZb|ab$

h) L(G) = set of all strings w over {a, b} such that w is not palindrome.

= $Y \longrightarrow aYa|bYb|aZb|bZa$
$Z \longrightarrow aZ|bZ|\varepsilon$

i) $L=\{w \mid w = w^R$ AND $|w|$ is even, w is a palindrome$\}$

= $S \longrightarrow A0A|B1B|\varepsilon$
$A \longrightarrow 1A|0A|\varepsilon$
$B \longrightarrow 1B|0B|\varepsilon$

j) $L(G) = \{ a^i b^j c^k \mid i, j, k \geq 0$ and $i=j$ or $j=k\}. \sum=\{a,b,c\}$

= $S \longrightarrow AC|\ S'$
$A \longrightarrow aAb|C$
$C \longrightarrow cC|\varepsilon$
$S' \longrightarrow A'B$
$A' \longrightarrow aA'|\varepsilon$
$B \longrightarrow bBb|A'$

k) $L(G) = \{ a^n b^m c^m d^{2n} \mid n \geq 0, m > 0\}$

= $S \longrightarrow aBdd|A$
$A \longrightarrow aSdd|\varepsilon$
$B \longrightarrow bBc|bc$

l) L= {w | w contains at least 4 a's}

= $S \longrightarrow RaRaRaRaR$
$R \longrightarrow bR|aR|\varepsilon$

2. What does the following CFGs do?

a) $S \longrightarrow ZSZ|0$
$Z \longrightarrow 0|1$

= L={w | the length of w is odd and its middle is 0}

b) $S \longrightarrow 0E0|1E1|\varepsilon$
$E \longrightarrow 1E|0E|\varepsilon$

= L={w | w starts and ends with the same symbol}

c) $S \rightarrow AB$
$A \rightarrow 0A1 | \varepsilon$
$B \rightarrow 1B | \varepsilon$

= $L(G) = \{0^m 1^{m+n} \mid n, m \geq 0\}$ over the terminals {0,1}

d) $S \rightarrow \varepsilon | 1S1S1S0S \mid 1S1S0S1S \mid 1S0S1S1S \mid 0S1S1S1S$

= L={w | w contains thrice as many 1s as 0s}

e) $S \rightarrow aSbb \mid aSb \mid \varepsilon$

= $L(G) = \{a^n b^m \mid 2n \geq m \geq n \geq 0\}$ over the terminals {0,1}

3. Convert the following Regular expressions to a CFG.

a) a(b|c*)

= $S \rightarrow aX$

$X \rightarrow b | C$

$C \rightarrow Cc | \varepsilon$

b) 0*1(0 + 1)*

= $S \rightarrow A1B$

$A \rightarrow 0A | \varepsilon$

$B \rightarrow 0B \mid 1B \mid \varepsilon$

c) (a + b)*(a* + (ba)*)

= $V \rightarrow WX$

$W \rightarrow aW$

$W \rightarrow bW$

$W \rightarrow \varepsilon$

$X \rightarrow Y$

$X \rightarrow Z$

$Y \rightarrow aY$

$Y \rightarrow \varepsilon$

$Z \rightarrow baZ$

Z ⟶ ε

d) (a+b)* aa (a+b)*

= S⟶ AaaA

A⟶ aA|bA| ε

e) a* + a(a|b)*

= S⟶ X|Y

X⟶ aX| ε

Y⟶ aZ

Z⟶ aZ|bZ| ε

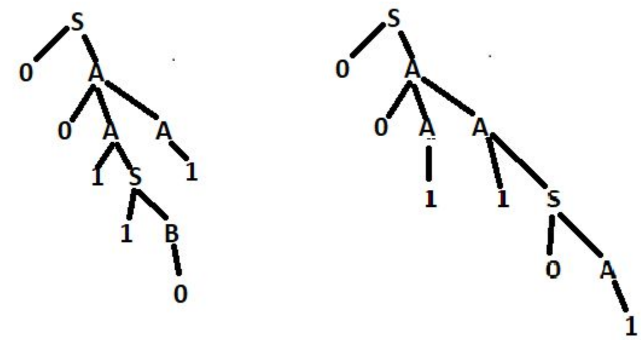
4. Consider the following context-free grammar ∑={0,1}. Give leftmost and rightmost derivations for the following strings and check parse-tree ambiguity.

a) S ⟶ 0A | 1B
A ⟶ 0AA| 1S | 1
B ⟶ 0S |1BB| 0

Strings: 001101

leftmost derivation:	**rightmost derivation:**
S ⟶ 0A ⟶00AA ⟶001A ⟶0011S ⟶00110A ⟶001101	S ⟶ 0A ⟶00AA ⟶00A1 ⟶001S1 ⟶0011B1 ⟶001101

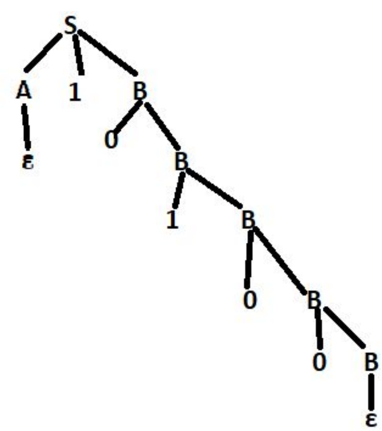
we can find two parse trees for this grammar, so the grammar is ambiguous.



b) S ⟶ A 1 B
A ⟶ 0A | ε
B ⟶ 0B |1B| ε

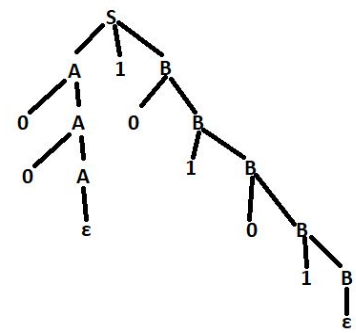
Strings: 10100, 0010101

= for string 10100:

leftmost derivation:	**rightmost derivation:**	**Parse Tree:**
S ⟶ A1B **⟶ ϵ1B** **⟶ 10B** **⟶ 101B** **⟶ 1010B** **⟶ 10100ϵ** **⟶ 10100**	**S⟶ A1B** **⟶ A10B** **⟶ A101B** **⟶ A1010B** **⟶ A10100** **⟶ ϵ10100** **⟶ 10100**	

for string 0010101:

leftmost derivation:	**rightmost derivation:**	**Parse Tree:**
S⟶ A1B **⟶ 0A1B** **⟶ 00A1B** **⟶ 00ε1B** **⟶ 001B** **⟶ 0010B**	**S⟶ A1B** **⟶ A10B** **⟶ A101B** **⟶ A101ε** **⟶ 0A101** **⟶ 00A101**	

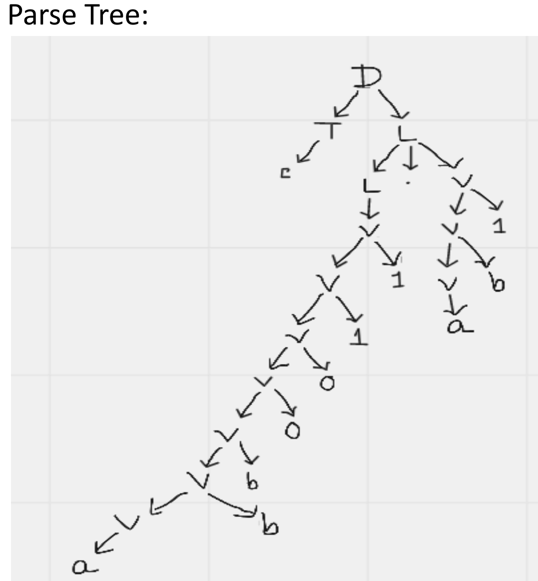
→ 00101B **→ 001010B** **→ 0010101B** **→ 0010101ε** **→ 0010101**	**→ 001A101** **→** **0010ε101** **→ 0010101**	

The grammar is unambiguous since only one parse tree is possible for every string.

c) D → TL
T → c | Tc
L → L.V | V
V → a | b | 0 | 1| Va | Vb | V0 | V1

Strings: cabb0011.ab1 (Rightmost derivation)

=

rightmost derivation **D→ TL** **→ TL.V** **→ TL.V1** **→ TL.Vb1** **→ TL.ab1** **→ TV.ab1** **→ TV1.ab1** **→ TV11.ab1** **→ TV011.ab1** **→ TV0011.ab1** **→ TVb0011.ab1** **→ TVbb0011.ab1** **→ Tabb0011.ab1** **→ cabb0011.ab1**	Parse Tree: 

The grammar is unambiguous since only one parse tree is possible for every string.

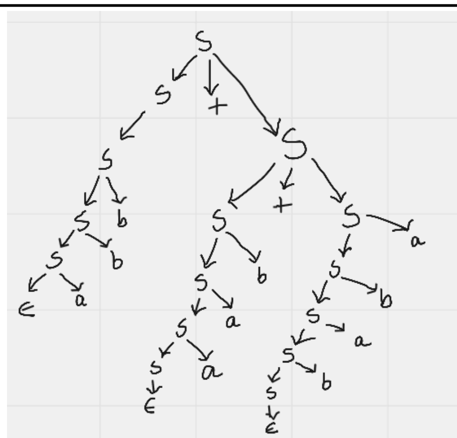
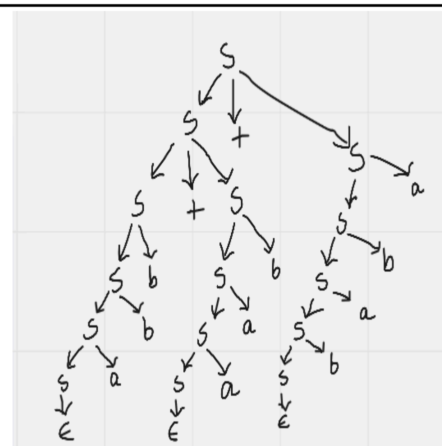
d) $S \rightarrow S + S$

$S \rightarrow Sa \mid Sb \mid \varepsilon$

String: abb + aab + baba

Leftmost Derivation:	Rightmost Derivation:
$S \rightarrow S + S$ $\rightarrow Sb + S$ $\rightarrow Sbb + S$ $\rightarrow Sabb + S$ $\rightarrow abb + S + S$ $\rightarrow abb + Sb + S$ $\rightarrow abb + Sab + S$ $\rightarrow abb + Saab + S$ $\rightarrow abb + aab + Sa$ $\rightarrow abb + aab + Sba$ $\rightarrow abb + aab + Saba$ $\rightarrow abb + aab + Sbaba$ $\rightarrow abb + aab + baba$	$S \rightarrow S + S$ $\rightarrow S + Sa$ $\rightarrow S + Sba$ $\rightarrow S + Saba$ $\rightarrow S + Sbaba$ $\rightarrow S + baba$ $\rightarrow S + S + baba$ $\rightarrow S + Sb + baba$ $\rightarrow S + Sab + baba$ $\rightarrow S + Saab + baba$ $\rightarrow S + aab + baba$ $\rightarrow Sb + aab + baba$ $\rightarrow Sbb + aab + baba$ $\rightarrow Sabb + aab + baba$ $\rightarrow abb + aab + baba$

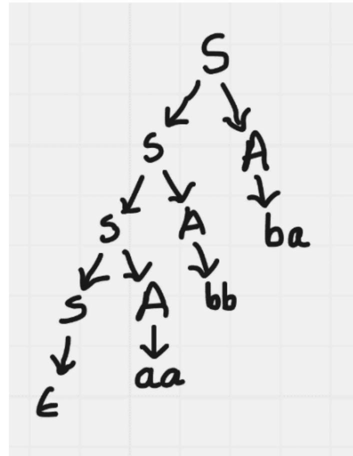
we can find two parse trees for this grammar, so the grammar is ambiguous.

e) $S \rightarrow SA \mid \varepsilon$

$A \rightarrow aa \mid ab \mid ba \mid bb$

String: aabbba

Leftmost Derivation:	Rightmost Derivation:	
S ⟶ SA ⟶SAA ⟶SAAA ⟶aaAA ⟶aabbA ⟶aabbba	S ⟶ SA ⟶Sba ⟶SAba ⟶Sbbba ⟶SAbbba ⟶Saabbba ⟶aabbba	

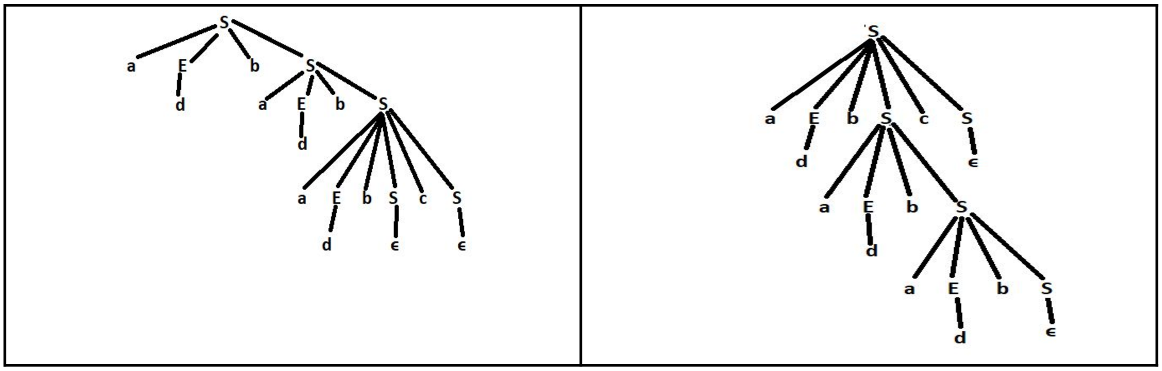
The grammar is unambiguous since only one parse tree is possible for every string.

f) S ⟶ aEbS
S ⟶ aEbScS | ϵ
E ⟶ d

String: adbadbadbc

Leftmost Derivation:	Rightmost Derivation:
S ⟶ aEbS ⟶adbS ⟶adbaEbS ⟶adbadbS ⟶adbadbaEbScS ⟶adbadbadbScS ⟶adbadbadbc	S ⟶ aEbS ⟶aEbaEbScS ⟶aEbaEbScϵ ⟶ aEbaEbSc ⟶ aEbaEbaEbSc ⟶ aEbaEbaEbϵc ⟶ aEbadbadbc ⟶ adbadbadbc

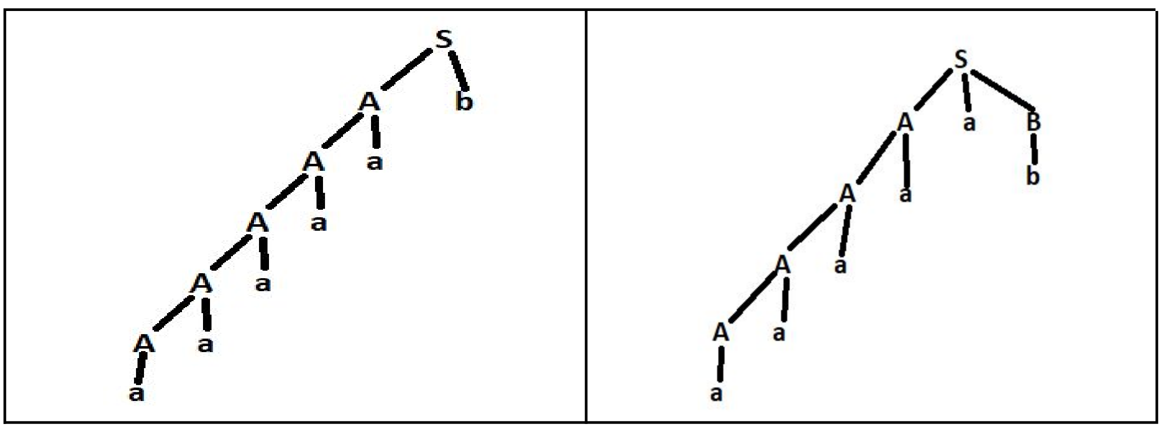
we can find two parse trees for this grammar, so the grammar is ambiguous.



5. Are the following CFGs ambiguous? Are they inherently ambiguous? If not, then give its unambiguous representation.

a) S → Ab | AaB
A → a | Aa
B → b

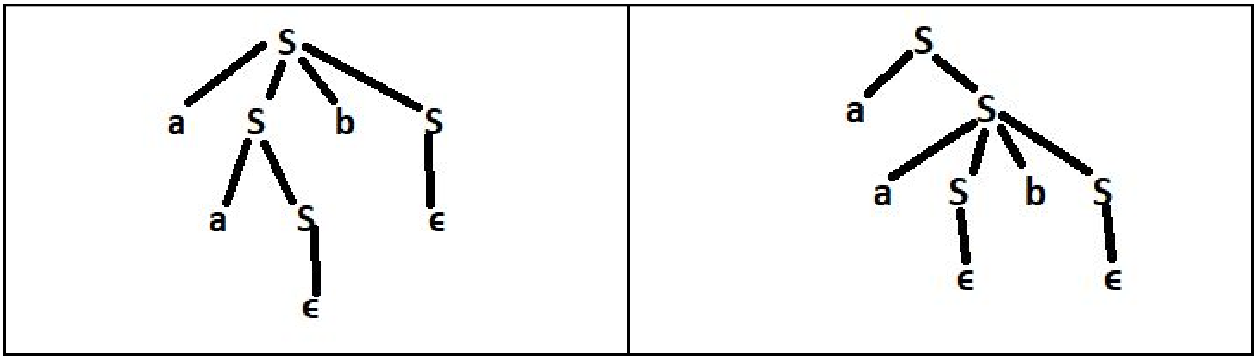
= For the string aaaaab, we can find two parse trees. So the grammar is ambiguous.



The grammar is ambiguous but not inherently ambiguous. So, equivalent unambiguous CFG:
S→ A’b|ab
A’ → A’a|aa

b) S → aS | aSbS | ϵ

= For the string aab, we can find two parse trees. So the grammar is ambiguous.



The grammar is ambiguous but not inherently ambiguous. So, equivalent unambiguous CFG:

S⟶ aS|aTbS|ϵ
T⟶ aTbT|ϵ