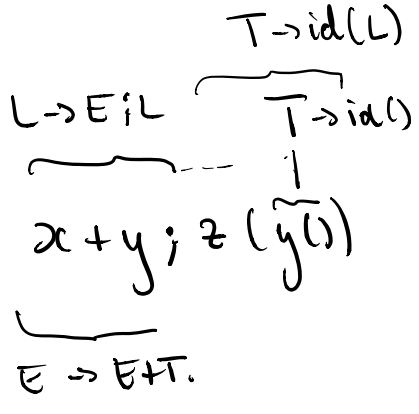
a) i)



Goal $\underset{lm}{\Rightarrow}$ L $\underset{lm}{\Rightarrow}$ E ; L $\underset{lm}{\Rightarrow}$ E+T ; L $\underset{lm}{\Rightarrow}$ T+T ; L $\underset{lm}{\Rightarrow}$

id+T ; L $\underset{lm}{\Rightarrow}$ id+id ; L $\underset{lm}{\Rightarrow}$ id+id ; E $\underset{lm}{\Rightarrow}$ id+id ; T

$\underset{lm}{\Rightarrow}$ id+id ; id(L) $\underset{lm}{\Rightarrow}$ id+id ; id(E) $\underset{lm}{\Rightarrow}$ id+id ; id(T) $\underset{lm}{\Rightarrow}$

id+id ; id(id())

ii) We need to eliminate left recursion and do left factoring. There is no indirect left recursion, so we only need to fix direct ones.

$E \rightarrow E+T \mid T$

becomes.

$E \rightarrow TE'$

$E' \rightarrow +TE'$

For left factoring:

$L \to E;L$
$L \to E$
$\Big\} \to$ $L \to EL'$
$L' \to ;L \mid \varepsilon$

$T \to id$
$T \to id()$
$T \to id(L)$
$\Big\} \to$ $T \to idT'$
$T' \to \varepsilon \mid () \mid (L)$
$\Big\} \to$ $T \to idT'$
$T' \to \varepsilon \mid (T''$
$T'' \to) \mid L)$

b) 1. $G \to L$
2. $L \to LP$
3. $L \to P$
4. $P \to (P)$
5. $P \to ()$

Parse the string (())()

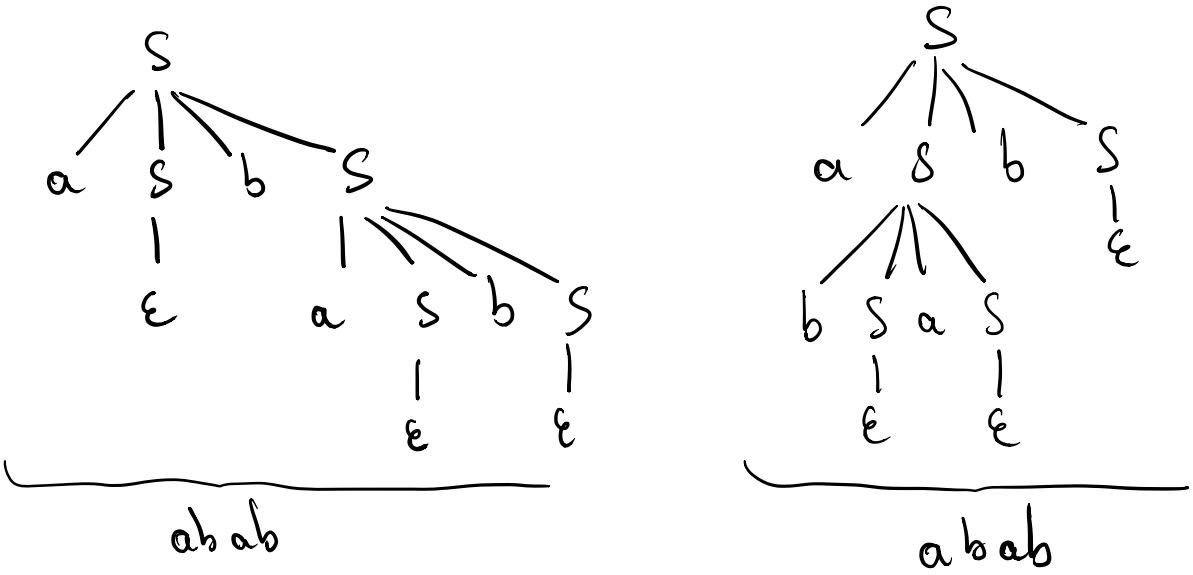
Step	Stack	Input	Action
0	0	(())()	S3
1	03	())()	S6
2	036	))()	S10
3	03610	)()	R5
4	035	)()	S8
5	0358	()	R4
6	02	()	R3
7	01	()	S3
8	013	)	S7
9	0137	$	R5
10	014	$	R2
11	01	$	accept.

c) No, because $b \in$ FIRST (bA) and $b \in$ FIRST (XY) and they must be disjoint.

d) $S \to aSbS$

$S \to bSaS$

$S \to \varepsilon$



By definition, if a string has more than 1 parse tree, the grammar is ambiguous.