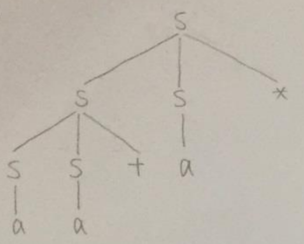
Q1:

(1) $S \xrightarrow{lm} SS* \xrightarrow{lm} SS+S* \xrightarrow{lm} aS+S* \xrightarrow{lm}$
$aa+S* \longrightarrow aa+a*$

(2) $S \xrightarrow{rm} SS* \xrightarrow{rm} Sa* \xrightarrow{rm} SS+a*$
$\xrightarrow{rm} Sa+a* \xrightarrow{rm} aa+a*$

(3)

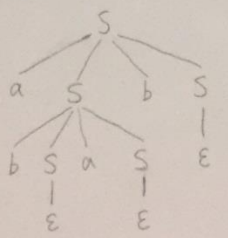


(4) Although it has one leftmost derivation and one rightmost derivation, but the two derivations only match one parsing tree. so it is unambiguous

Q2. To get the parsing tree, we have to get the derivation first.

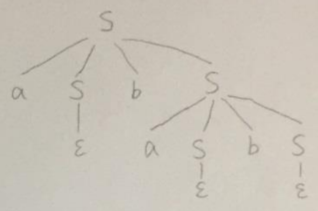
(1) 1° The first way to build the left-most parsing tree is as follows:

$S \xrightarrow{lm} aSbS \xrightarrow{lm} a\underline{bSaS}bS \xrightarrow{lm} ab\varepsilon a\varepsilon b\varepsilon$
$\longrightarrow abab$. Then we can get the parsing tree



2° The second way to build the left-most parsing tree is as follows:

$S \xrightarrow{lm} aSbS \xrightarrow{lm} a\varepsilon b\underline{S} \xrightarrow{lm} ab\underline{aSbS}$
$\xrightarrow{lm} aba\varepsilon b\varepsilon \xrightarrow{lm} abab$. Then we can get:



(2) right-most derivation:

$S \xrightarrow{rm} aSbS \xrightarrow{rm} aSb\varepsilon \xrightarrow{rm} aSb \longrightarrow a\,\underline{bSaS}\,b$
$\xrightarrow{rm} abSa\varepsilon b \xrightarrow{rm} abSab \xrightarrow{rm} ab\varepsilon ab$
$\xrightarrow{rm} abab$.

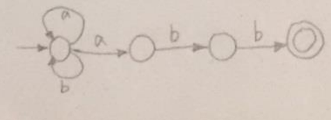
OR:

$S \xrightarrow{rm} aSbS \xrightarrow{rm} aSb\,\underline{aSbS} \xrightarrow{rm} aSbaSb\varepsilon$
$\xrightarrow{rm} aSbaSb \xrightarrow{rm} aSba\varepsilon b \xrightarrow{rm} aSbab$
$\xrightarrow{rm} a\varepsilon bab \xrightarrow{rm} abab$

Remember that:

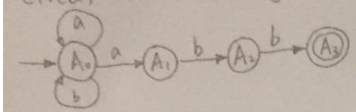
if a grammar is unambiguous, then a given sentence has a unique leftmost derivation, a unique rightmost derivation, and both derivations correspond to the same unique parse tree.

Q3. From Lecture note 2, we know the NFA for $(a|b)^*abb$ is as follows:



Step 1:

Let's fill A_i $(i=0,1,\cdots n)$ into the state circle, then we can get:



Step 2:

According the inputs and the arrows, we can get some products as follows:

$A_0 \rightarrow aA_0 / bA_0 / aA_1$

means "or"

$A_1 \rightarrow bA_2$

$A_2 \rightarrow bA_3$

Step3:

Here, A_0 is the start state, A_3 is the end state, for the end state A_3, we should add a product $A_3 \rightarrow \varepsilon$ into it.

Then we can get the context-free grammar:

$A_0 \rightarrow aA_0 / bA_0 / aA_1$

$A_1 \rightarrow bA_2$

$A_2 \rightarrow bA_3$

$A_3 \rightarrow \varepsilon$

You should remember that:

① Each RE can be described by the context-free grammar, but the reverse is not.

② Grammar is function-stronger than RE.

③ DFA is a special form of NFA.