[Question 1]

Consider the following CFG:

$S \rightarrow \_aS \mid ba$

Prove that this generates the language defined by the regular expression $a{*}ba$

Given $S \rightarrow \_aS \mid ba$

Let the language generated by the CFG be L_{CFG}

Let the language defined by the regular expression be L_{a*ba}

To show $L_{CFG} = L_{a*ba}$, we must prove that

1. $L_{CFG} \subseteq L_{a*ba}$	*Every string generated by the CFG is also in the language* $a * ba$
2. $L_{a*ba} \subseteq L_{CFG}$	*Every string in the language* $a * ba$ *can be generated by the CFG is also*

1.

Σ: **Terminal(s)**	a, b
V: **Non-terminal(s)**	S
P: **Production Rule(s)**	P1. $S \Rightarrow \_aS$ $\ldots \Rightarrow \_aaS$ $\ldots \Rightarrow \_aaa \ldots aaS$ *will generate words with arbitrary number of a's* P2. $S \Rightarrow ba$ *will generate just the word ba*

Therefore, any string generated by L_{CFG} will be in the form $a * ba$

2.

$a *$ can be generated with the production P1: $S \Rightarrow \_aS$

ba can be generated with the production P2: $S \Rightarrow ba$

Therefore, any string generated by L_{a*ba} can be generated by $a * ba$ L_{CFG}

Thus,

$L_{CFG} = L_{a*ba}$

[Question 2]

Find CFGs for the following languages over the alphabet $\Sigma = \{a\ b\}$:

All words that do not have the substring ab.

Σ: **Terminal(s)**	a, b
V: **Non-terminal(s)**	$\boldsymbol{S}$, StartsWithA , StartsWithB , RemainingAsAfterB
P: **Production Rule(s)**	**P1.** $\boldsymbol{S} \Rightarrow \Lambda$ **P2.** $\boldsymbol{S} \Rightarrow$ StartsWithA **P3.** $\boldsymbol{S} \Rightarrow$ StartsWithB …will generate the empty string Λ or A or StartsWithB
	P4. StartsWithA $\Rightarrow$ ***a***StartsWithA **P5.** StartsWithA $\Rightarrow$ ***a***StartsWithB **P6.** StartsWithA $\Rightarrow \Lambda$ …will generate strings starting with a, followed by more a's. or strings starting with a, followed by StartsWithB. or the empty string Λ
	P7. StartsWithB $\Rightarrow$ ***b***RemainingAsAfterB **P8.** StartsWithB $\Rightarrow$ ***b*** …will generate strings starting with b, followed by more b's *or* strings starting with b, followed by RemainingAsAfterB. *or just the word b*
	P9. RemainingAsAfterB $\Rightarrow$ ***a***RemainingAsAfterB **P10.** RemainingAsAfterB $\Rightarrow \Lambda$ …will generate strings starting with b, followed by more a's. or the empty string Λ

$a*$ can be generated with the production PROD 1: $S \Rightarrow \_aS$
ba can be generated with the production PROD 2: $S \Rightarrow ba$
Therefore, any string generated by L_{a*ba} can be generated by $a*ba$ L_{CFG}

Thus,
L_{CFG} guarantees that no generated string contains the substring "ab."

Question 3

Investigate each of the CFGs provided and decide whether the word *abba* is generated by the given CFGs.

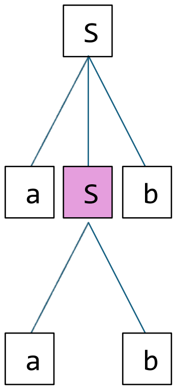
In the case where *abba* is not generated a brief discussion why a particular CFG does not generate *abba*.

If abba is indeed generated, then draw the corresponding syntax tree illustrating the generation of *abba*.

1. *CFG 1:* $S \rightarrow aSb \mid ab$

Σ: **Terminal(s)**	$a,\ b$
V: **Non-terminal(s)**	$\boldsymbol{S}$
P: **Production Rule(s)**	P1: $S \Rightarrow aSb$
	P2: $S \Rightarrow ab$

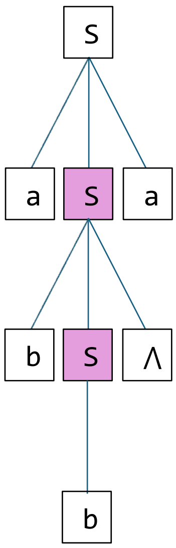
Production Rule	**Terminal(s) generated**
PROD 1: $S \Rightarrow aSb$	a
PROD 1: $S \Rightarrow aSb$	b
PROD 1: $S \Rightarrow aSb$	b
PROD 2: $S \Rightarrow ab$	a



2. *CFG 2:* $S \rightarrow aS \mid bS \mid a$

Σ: **Terminal(s)**	$a,\ b$
V: **Non-terminal(s)**	$\boldsymbol{S}$
P: **Production Rule(s)**	P1: $S \Rightarrow aS$
	P2: $S \Rightarrow bS$
	P3: $S \Rightarrow a$

Production Rule	**Terminal(s) generated**
PROD 3: $S \Rightarrow a$	a
PROD 2: $S \Rightarrow bS$	bb
PROD 3: $S \Rightarrow a$	a

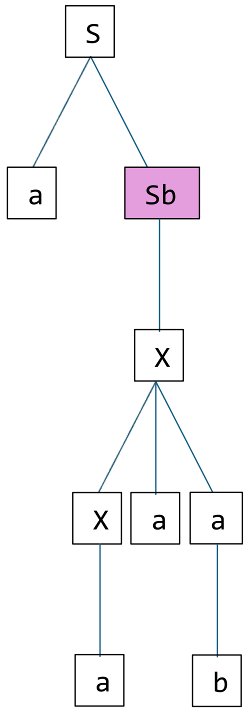


3. *CFG 3* $S \rightarrow aS \mid aSb \mid X$

$X \rightarrow aXa \mid a$

Σ: **Terminal(s)**	a, b
V: **Non-terminal(s)**	$\boldsymbol{S}$, $\boldsymbol{X}$
P: **Production Rule(s)**	P1: $S \Rightarrow aS$
	P2: $S \Rightarrow ab$
	P3: $S \Rightarrow X$
	P4: $X \Rightarrow aXa$
	P5: $X \Rightarrow a$

Production Rule	**Terminal(s) generated**
PROD 1: $S \Rightarrow aS$	a
PROD 1: $S \Rightarrow aSb$	b
PROD 1: $S \Rightarrow aSb$	b
PROD 3: $X \Rightarrow a$	a



4. *CFG 4:* $S \rightarrow aAS | a$
$A \rightarrow SbA | SS | ba$

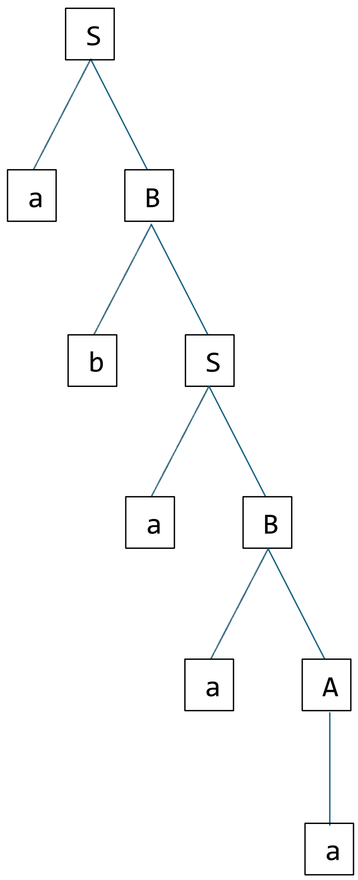
Σ: **Terminal(s)**	a, b
V: **Non-terminal(s)**	$\boldsymbol{S}$ **,** $\boldsymbol{A}$
P: **Production Rule(s)**	P1: $S \Rightarrow aAS$
	P2: $S \Rightarrow a$
	P3: $A \Rightarrow SbA$
	P4: $A \Rightarrow SS$
	P5: $A \Rightarrow ba$

Production Rule	**Terminal(s) generated**
PROD 1: $S \Rightarrow aAS$	a
PROD 4: $A \Rightarrow SS$	aa
PROD 1: $S \Rightarrow aAS$	aab
PROD 1: $S \Rightarrow aAS$	aabb
PROD 2: $S \Rightarrow a$	aabba
PROD 3: $A \Rightarrow ba$	a**abba**b
PROD 2: $S \Rightarrow a$	abba

5. CFG 5: $S \rightarrow aB|bA$
$A \rightarrow a|aS|bAA$
$B \rightarrow b|bS|aBB$

Σ: **Terminal(s)**	$a,\ b$
V: **Non-terminal(s)**	$\boldsymbol{S\ ,\ A\ ,\ B}$
P: **Production Rule(s)**	P1: $S \Rightarrow aB$
	P2: $S \Rightarrow bA$
	P3: $A \Rightarrow a$
	P4: $A \Rightarrow aS$
	P5: $A \Rightarrow bAA$
	P6: $B \Rightarrow b$
	P7: $B \Rightarrow bS$
	P8: $B \Rightarrow aBB$

Production Rule	**Terminal(s) generated**
PROD 1: $S \Rightarrow aB$	a
PROD 7: $B \Rightarrow bS$	ab
PROD 3: $A \Rightarrow a$	abb
PROD 3: $A \Rightarrow a$	abba
PROD 5: $A \Rightarrow bAA$	



Question 4

Convert the grammar below to CNF.

$S \rightarrow aX \mid Yb$

$X \rightarrow ZXZY \mid a$

$Y \rightarrow b \mid bY \mid \Lambda$

$Z \rightarrow a \mid \Lambda$

1. Eliminate ε-productions from the grammar

   $S \rightarrow aX \mid Yb$

   $X \rightarrow ZXZY \mid a$

   $Y \rightarrow b \mid bY$

   $Z \rightarrow a \mid \Lambda$

2. Eliminate any non-terminal that produces a single terminal

   $Y \rightarrow B \mid BY$

   $B \rightarrow b$

   $Z \rightarrow A \mid \Lambda$

   $A \rightarrow a$

3. Ensure productions of the form $A \rightarrow BC$ or $A \rightarrow a$:

   Rewrite $X \rightarrow ZXZY$ as:

   $C \rightarrow FG$

   $F \rightarrow ZX$

   $G \rightarrow ZY$

Chomsky Normal Form.

$S \rightarrow aC \mid Yb$

$C \rightarrow FG$

$F \rightarrow ZX$

$G \rightarrow ZY$

$X \rightarrow a$

$Y \rightarrow B \mid BY$

$B \rightarrow b$

$Z \rightarrow A \mid \Lambda$

$A \rightarrow a$

Question 5

Develop a DPDA accepting the language $L = \{b^{n+1}(ab)a^{n-1} \mid n \geq 2\}$

Define DPDA

$\Sigma = \{a, b\}$	input alphabet
$\Gamma = \{X\ Y\}$	stack alphabet
L	Tape of infinite length containing a string
The states as defined on page 307 of the textbook.	
δ	transition function read b, push read b, push read a, push read a, pop from stack