1. Convert the following grammar to Chomsky normal form:

$S \to AB \mid BCS$
$A \to aA \mid C$
$B \to bbB \mid b$
$C \to cC \mid \lambda$

$S \to AB \mid BT_1$
$T_1 \to CS$
$A \to A'A \mid C$
$B \to B'T_2 \mid B'$
$T_2 \to B'B$

$\to$

$C \to C'C \mid \lambda$
$A' \to a$
$B' \to b$
$C' \to c$

$\to$

$S \to AB \mid BT_1$
$T_1 \to CS$
$A \to A'A \mid C$
$B \to B'T_2 \mid B'$
$T_2 \to B'B$
$C \to C'C \mid \lambda$
$A' \to a$
$B' \to b$
$C' \to c$

2. Show that all the symbols of the grammar are useful

$S \to A \mid CB$
$A \to C \mid D$
$B \to bB \mid b$
$C \to cC \mid c$
$D \to dD \mid d$

Construct an equivalent grammar G_C by removing the chain rules from the grammar. Show that G_C contains useless symbols.

* A terminal is useful if it occurs in a string in the language of G.
→ variable must occur in a sentential form of the grammar.
→ every symbol occuring in the sentential form must be capable of deriving a terminal string

Sample iterations
$S \to A \to (C) \to (cC) \to c(c()) \to cc(c) \to ccc$ ✓
$S \to A \to D \to (dD) \to d(d) \to dd$ ✓
$S \to CB \to (c)B \to C(b) \to cb$ ✓
$B \to b;\ C \to c;\ D \to d$

All symbols derive a terminal string

Term	Prev
{B, C, D}	∅
{B, C, D, A}	{B, C, D}
{B, C, D, A, S}	{B, C, D, A}
{B, C, D, A, S}	{B, C, D, A, S}

Removing Chain rules
chain(S) = {S, A, C, D, B}
chain (A) = {C, D}
chain (B) = {B}
chain (C) = {C}
chain (D) = {D}

$\longrightarrow$ G_C:
$S \to C \mid D \mid CB \mid cC \mid c \mid dD \mid d \mid bB \mid b$
$A \to cC \mid c \mid dD \mid d$
$B \to bB \mid b$
$C \to cC \mid c$
$D \to dD \mid d$

Term
{S, B, C, D} - Here, even though A generates terminal strings, it is consumed by G_C's S rule and isn't substituted anywhere else. Therefore A is useless. Formally, B, C, D are not useless because they generate terminal strings, but intuitively we can see that their presence is redundant.

3. Give the upper diagnol matrix produced by the CYK algorithm when run with the Chomsky normal form grammar:

$S \to AT \mid AB$
$T \to XB$
$X \to AT \mid AB$
$A \to a$
$B \to b$

the input strings abbb and aabbb
(page 2)

3. V_{ij} - all nonterminals that generate j symbols of the string, starting from symbol i.

$V_{1,1}$ abbb - a $V_{1,1} = \{A\}$
$V_{2,1}$ - b $V_{2,1} = \{B\}$
$V_{3,1}$ - b $V_{3,1} = \{B\}$

$V_{1,5}$

V \ j	1	2	3	4	5
i 1	{A}	{S,X}	{A,T}	{S,X}	
2	{B}	∅	{B}		
3	{B}	∅			
4	{B}				
5					

abbb ✓

$V_{1,4}$ $\boxed{abbb}$
$V_{1,1}$ $V_{2,3}$
$V_{1,2}$ $V_{3,2}$
$V_{1,3}$ $V_{4,1}$

abbb
$V_{1,2}$
$V_{1,1}$; $V_{2,1}$

$V_{1,3}$
abbb
$V_{1,1}$ $V_{2,2}$
$V_{1,2}$ $V_{3,1}$

$V_{2,3}$
abbb
$V_{2,1}$ $V_{3,2}$
$V_{2,2}$ $V_{4,1}$

aabbb
$V_{1,1}, V_{2,1}, \ldots V_{5,1}$
{A} {A} ... {B}

V \ j	1	2	3	4	5
i 1	{A}	∅	{B}	{S,X,B}	{B,T}
2	{A}	{S,X}	{A,T}	{S,X}	
3	{B}	∅	{B}		
4	{B}	∅			
5	{B}				

aabbb
$V_{1,2}$
$V_{1,1}$; $V_{2,1}$

aabbb
$V_{1,3}$
$V_{1,1}$ $V_{2,2}$
$V_{1,2}$ $V_{3,1}$

aabbb
$V_{1,5}$
$V_{1,1}$ $V_{2,4}$
$V_{1,2}$ $V_{3,3}$
$V_{1,3}$ $V_{4,2}$
$V_{1,4}$ $V_{5,1}$

aabbb
$V_{1,4}$
$V_{1,1}$ $V_{2,3}$
$V_{1,2}$ $V_{3,2}$
$V_{1,3}$ $V_{4,1}$

It would follow given the CYK output that aabbb is not a derivable string from the given grammar, as the start symbol S is not included for $V_{1,5}$.

4. Construct a grammar G' that contains no left-recursive rules and is equivalent to

$S \to A \mid C$
$A \to Aab \mid AaC \mid B \mid a$
$B \to Bb \mid Cb$
$C \to cC \mid c$

① $S \to A \mid C$
② $A \to Aab \mid AaC \mid B \mid a$
$A \to BZ_1 \mid aZ_1 \mid B \mid a$
$Z_1 \to abZ_1 \mid aZ_1C \mid ab$
③ $B \to Bb \mid Cb$
$B \to CbZ_2 \mid Cb$
$Z_2 \to bZ_2 \mid b$
$C \to cC \mid c$

⇒

$S \to A \mid C$
$A \to BZ_1 \mid aZ_1 \mid B \mid a$
$Z_1 \to abZ_1 \mid aZ_1C \mid ab$
$B \to CbZ_2 \mid Cb$
$Z_2 \to bZ_2 \mid b$
$C \to cC \mid c$

5. Convert the chomsky normal form grammar to Greibach normal form. Process the variables according to the order S, A, B, C, D.
1 2 3 4 5

1 $S \to AB$ ✓
2 $A \to BB \mid CC$
3 $B \to AD \mid CA$
4 $C \to a$
5 $D \to b$

$B \to AD$ is out of order
$B \to BBD \mid CCD \mid CA$

In Order
$S \to AB$
$A \to BB \mid CC$
$B \to BBD \mid CCD \mid CA$
$C \to a$
$D \to b$

Removing direct left recursion
$B \to BBD \mid CCD \mid CA$ $v_1 = CCD$
$v_2 = CA$
$u_1 = B$
$B \to v_1Z \mid v_2Z \mid v_1 \mid v_2$
$Z \to u_1Z \mid u_1$

$B \to CCDZ \mid CAZ \mid CCD \mid CA$
$Z \to BZ \mid B$

$S \to AB$
$A \to BB \mid CC$
$B \to CCDZ \mid CAZ \mid CCD \mid CA$
$C \to a$
$D \to b$
$Z \to BZ \mid B$

Our final grammar yields

$S \to aCDZBB \mid aAZBB \mid aCDBB \mid aACDBB \mid aCB$
$A \to aCDZB \mid aAZB \mid aCDB \mid aAB \mid aC$
$B \to aCDZ \mid aAZ \mid aCD \mid aA$
$C \to a$
$D \to b$
$Z \to aCDZZ \mid aAZZ \mid aCDZ \mid aAZ \mid aCDZ \mid aAZ \mid aCD \mid aA$

$S \to AB \to (BB)B \to (CCDZ)BB \to (a)CDZBB$ $S \to aCDZBB$
$S \to AB \to (BB)B \to (CAZ)BB \to (a)AZBB$ $S \to aAZBB$
$S \to AB \to (BB)B \to (CCD)BB \to (a)CDBB$ $S \to aCDBB$
$S \to AB \to (BB)B \to (CA)CDBB \to (a)ACDBB$ $S \to aACDBB$
$S \to AB \to (CC)B \to (a)CB$ $S \to aCB$

A RULES
$A \to BB \to (CCDZ)B \to (a)CDZB$ $A \to aCDZB$
$A \to BB \to (CAZ)B \to (a)AZB$ $A \to aAZB$
$A \to BB \to (CCD)B \to (a)CDB$ $A \to aCDB$
$A \to BB \to (CA)B \to (a)AB$ $A \to aAB$
$A \to CC \to (a)C$ $A \to aC$

B RULES
$B \to CCDZ \to (a)CDZ$ $B \to aCDZ$
$B \to CAZ \to (a)AZ$ $B \to aAZ$
$B \to CCD \to (a)CD$ $B \to aCD$
$B \to CA \to (a)A$ $B \to aA$

C and D rules
$C \to a$, $D \to b$

Z

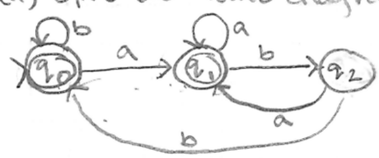
Z rules
$Z \to (CCDZ)Z \to (a)CDZZ$ $Z \to aCDZZ$
$Z \to (CAZ)Z \to (a)AZZ$ $Z \to aAZZ$
$Z \to (CCD)Z \to (a)CDZ$ $Z \to aCDZ$
$Z \to (CA)Z \to (a)AZ$ $Z \to aAZ$
+ B RULES

6. Let M be the deterministic finite automaton

$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$F = \{q_0\}$

δ	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_0

(a) Give the state diagram of M



b) Trace the computation of M that processes babaab

$[q_0, babaab]$ q_0
$[q_0, abaab]$ q_0
$[q_1, baab]$ q_1
$[q_2, aab]$ q_2
$[q_1, ab]$ q_1
$[q_1, b]$ q_1
$[q_2, \lambda]$ q_2

babaab is not a valid string as it does not end in the final state q_0

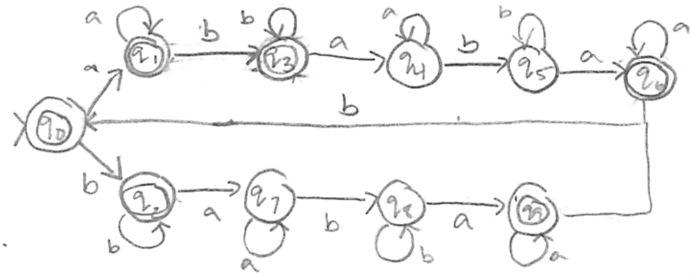
d) Give a regular expression for the language accepted if both q_0 and q_1 are accepting states

$$(b^*a^+)^* \cup (b^*a^+(ba)^+)^* \cup R_1$$

c) Give a regular expression for L(M)

$$R_1 = (b^*a^+b^+)^*(b^*a^+(ba)^+b)^*(b^*a^+(ba)^+b)^*$$

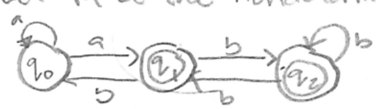
7. Build a DFA that accepts the language: The set of strings over $\{a, b\}$ that contain an even number of substrings ba.



λ is a legal, even $\emptyset$ string
aabb is a legal, even $\emptyset$ string
bbb is a legal, even $\emptyset$ string
bbbaaabbbaaabbb

Given that $\emptyset$ (ba)'s is a legal substring; $Q = \{q_0 .. q_9\}$
$F = \{q_0, q_1, q_2, q_3, q_6, q_9\}$
$\Sigma = \{a, b\}$

8. Let M be the nondeterministic finite automaton



$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$F = \{q_1, q_2\}$

a) Construct the transition table of M.

δ	a	b	λ^*
q_0	$\{q_0, q_1\}$	$\emptyset$	q_0
q_1	$\emptyset$	$\{q_0, q_2\}$	q_1
q_2	$\emptyset$	$\{q_1, q_2\}$	q_2

b) Trace all the computations of the string aabb in M.

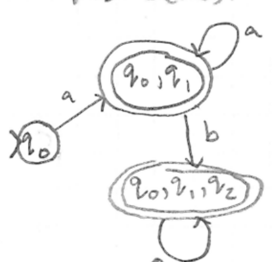
1		2		3		4		5
$[q_0, aabb]$	q_0	$[q_0, aabb]$	q_0	$[q_0, aabb]$	q_0	$[q_0, aabb]$	q_0	$[q_0, aabb]$
$\vdash [q_0, abb]$	q_0	$\vdash [q_0, abb]$	q_0	$\vdash [q_0, abb]$	q_0	$\vdash [q_0, abb]$	q_0	$\vdash [q_1, abb]$
$\vdash [q_0, bb]$	q_0	$\vdash [q_1, bb]$	q_1	$\vdash [q_1, bb]$	q_1	$\vdash [q_1, bb]$	q_1	
		$\vdash [q_2, b]$	q_2	$\vdash [q_2, b]$	q_2	$\vdash [q_0, b]$	q_0	
		$\vdash [q_2, \lambda]$	q_2	$\vdash [q_1, \lambda]$	q_1			

c) Is aabb in L(M)? Yes, aabb gets processed to λ and ends in F states q_2 and q_1 in traces 2 & 3.

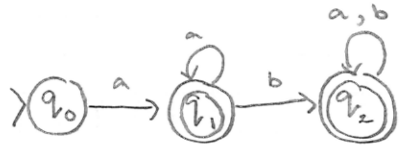
d) Give a regular expression for L(M).
$(a^+) \cup (a^+b)^*(a^+b^+)(a^+b)^*$

e) Construct a DFA that accepts L(M).

δ	$a\lambda^*$	$b\lambda^*$
>0	0,1	$\emptyset$
0,1	0,1	0,1,2
0,1,2	0,1,2	0,1,2

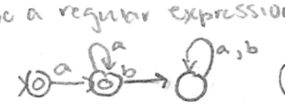


$\Rightarrow$



$(a^+ b^*)^*$

d) Give a regular expression for the language accepted if q_0 and q_1 are accepting states

For DFA  $(a^+)^*$ if $q_{0,1}$ are accepting states