ABC

FINAL EXAM ANSWER SHEET

Serial No. 116165

National University
Of Computer & Emerging Sciences

SEMESTER: ☐ SPRING ☐ SUMMER ☐ FALL 20____

Please Tick Appropriate Box

Course Title Theory of Automata Course Code CS3005.

Roll No. 20K-0247 Section E Date 6/06/2022

No. of continuation sheets attached ____

INSTRUCTIONS FOR CANDIDATES

- Write Question No. In the middle of the line using thick tipped pen.
- Use only blue or black pen to write your answers.
- Answers written using pencil will not be checked.
- Pencil is only allowed to draw diagrams or write program code.
- Cell Phone is not Allowed.

(THIS ANSWER BOOK CONTAINS PAGE NOS. 1-22)

Q./Part No.	Marks
Q. - 1	10
Q. - 2	10
Q. - 3	10
Q. - 4	10
Q. - 5	10
Q. - 6	10
Q. - 7	10
Q. - 8	2 ~~4~~ 24
Q. - 9	19.5
Q. - 10	

Q./Part No.	Marks
Q. - 11	
Q. - 12	
Q. - 13	
Q. - 14	
Q. - 15	
Q. - 16	
Q. - 17	
Q. - 18	
Q. - 19	
Q. - 20	

Marks Obtained: ~~91.5~~ 93.5

Total Marks: 100

Invigilator's Signature

Examiner's Signature

Date

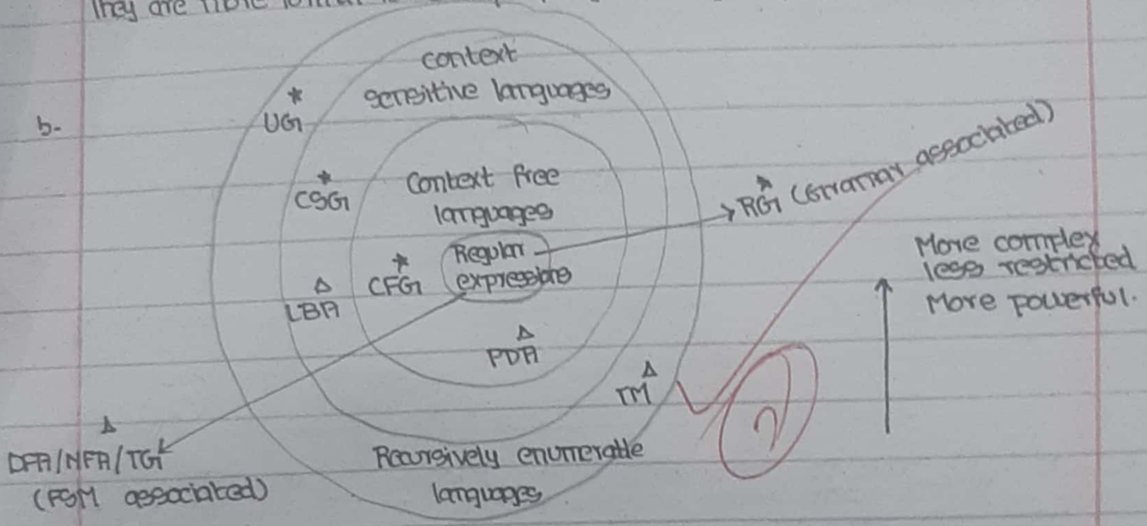
116165

Rough Work

QUESTION 1:

a- semantic languages are human understandable language that are represented in the form of strings, words and alphabets. They are informal. Whereas syntatic languages are syntax restricted languages that generates errors in case of invalid syntax. They are more formal for example programming languages.

b-



c- Concatenation of two RL's

Union of two RL's

Intersection of two RL's

Reverse of a RL

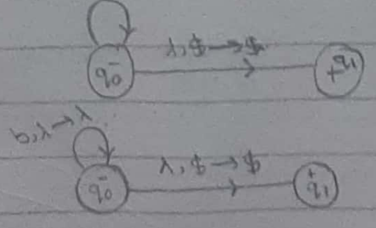
Closure of a RL

Complement of a RL

d- Intersection of two CFL's

Complement of a CFL

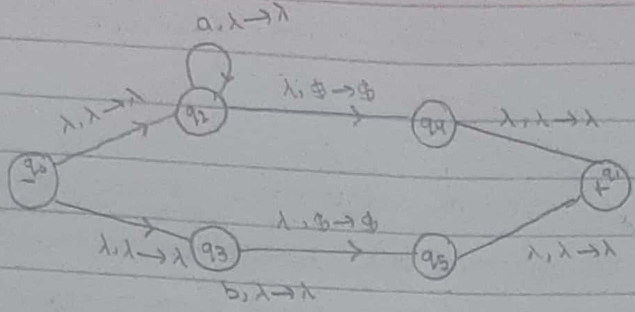
e- a^*



b^*

Union:

$ initially in stack.



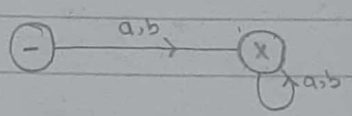
QUESTION 2.

a-

	a	b
- A	A	B
B	A	E
C	B	E
D	D	C
E	A	E

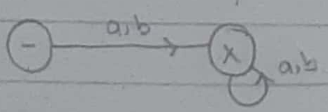
It has no final state, not accepting an

thus can be minimized directly to:

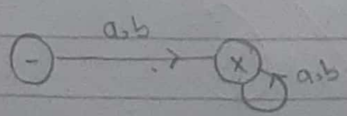


b- As DFA1 have no final state so the union of DFA1 and DFA2 will o
It will accept strings accepted by DFA2 as DFA 1 is not acceptin
DFA2. Refer same DFA2 as given in question paper.

c- As DFA1 have no final state, not accepting anything so the intersecti
DFA1 and DFA2 will be a DFA not accepting anything.



d- As DFA1 have no final state, not accepting anything so the concaten
DFA1 and DFA2 will be a DFA not accepting anything.



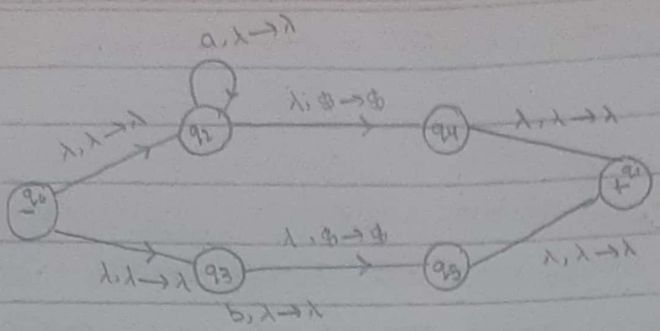
e-

	a	b
± Z_1 ≡ A	A ≡ Z_1	B ≡ Z_2
Z_2 ≡ B	A ≡ Z_1	E ≡ Z_3
Z_3 ≡ E	A ≡ Z_1	E ≡ Z_3

As the initial and final states are same of DFA1, the closure of DFA 1 will

Rough Work

Union:

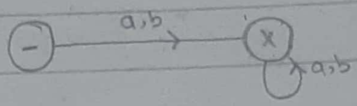


$ initially in stack.

QUESTION 2.

		a	b
a-	-A	A	B
	B	A	E
	C	B	E
	D	D	C
	E	A	E

It has no final state, not accepting any string, thus can be minimized directly to:



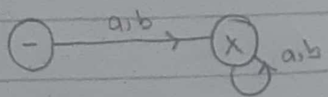
a, b

-s

then minimized

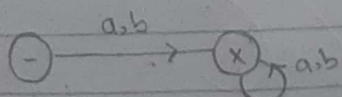
b- As DFA1 have no final state so the union of DFA1 and DFA2 will only be DFA2. Refer same DFA2 as given in question paper.
It will accept strings accepted by DFA2 as DFA 1 is not accepting anything

c- As DFA1 have no final state, not accepting anything so the intersection of DFA1 and DFA2 will be a DFA not accepting anything.



a,b

d- As DFA1 have no final state, not accepting anything so the concatenation of DFA1 and DFA2 will be a DFA not accepting anything.



a, b

		a	b
e-	± Z_1 ≡ A	A ≡ Z_1	B ≡ Z_2
	Z_2 ≡ B	A ≡ Z_1	E ≡ Z_3
	Z_3 ≡ E	A ≡ Z_1	E ≡ Z_3

As the initial and final states are now same, the closure of DFA1 will be DFA1.

ambiguity: If odd a's are not allowed with b so:

a(aa)* b (aa)* a

2

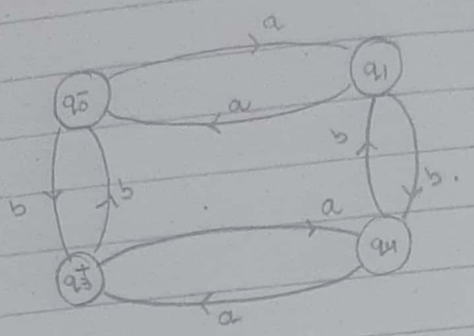
QUESTION 3:

a- $[(a+b)a]^*(a+b+\lambda)$.

b- $[b^*a(b^*ab^*ab^*)^*]\,b\,[b^*a(b^*ab^*ab^*)^*]$.

2

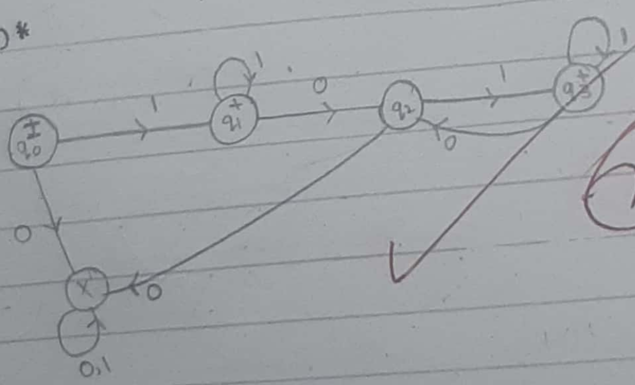
c-



2

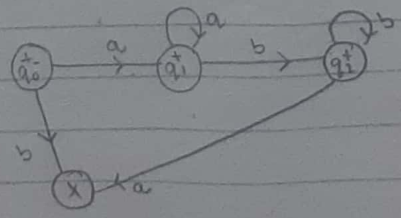
~~d- Its DFA is not possible as its a CFG.~~

e- $(1+101)^*$



2

d- As alphabets are not having c,d so we can only consider $a^i b^j$ and there is no condition on i and j. hence DFA will be:



b* only ?

1.5 + 0.3

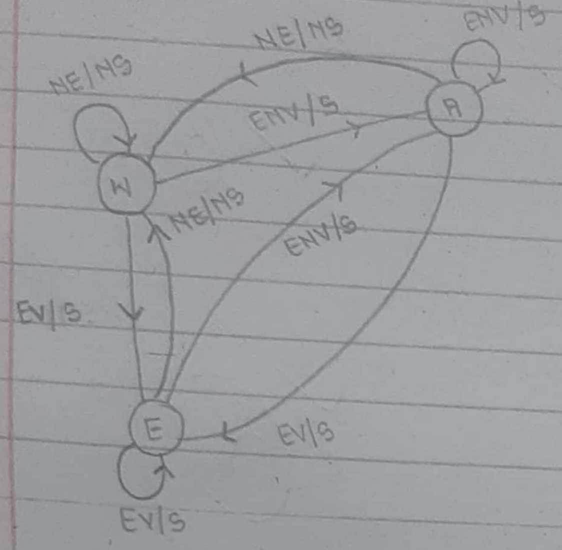
wrong but its ok

and if we want to consider c,d also at any how so its DFA is not possible.

Rough Work

QUESTION 4:

a-

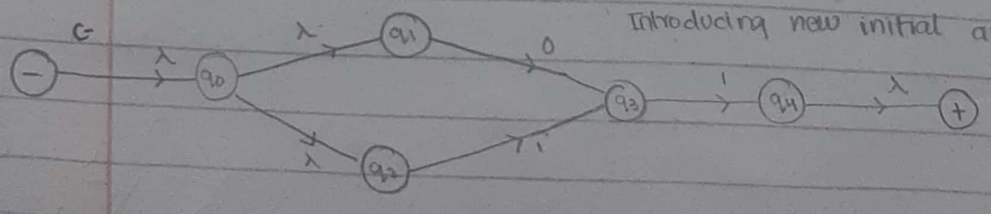


b- NFA1 is already in DFA as we have no transition of lambda, multiple transitions of same alphabets from a single state or missing transition of any alphabet from a state. It will have transition table:

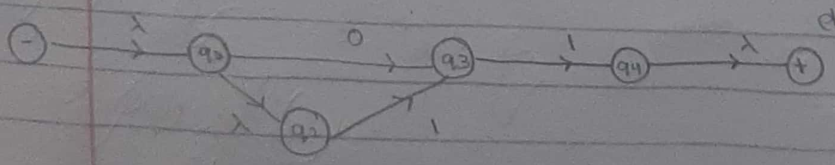
Q	δ(Q,a)	δ(Q,b)
q0	q1	q3
q1	q2	q4
q2	q1	q1
q3	q2	q4
q4	q4	q4

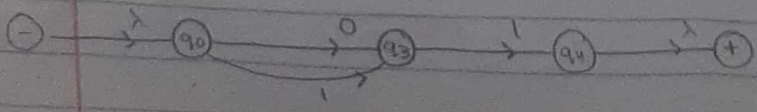
c-

Introducing new initial and final states.



eliminating state q1



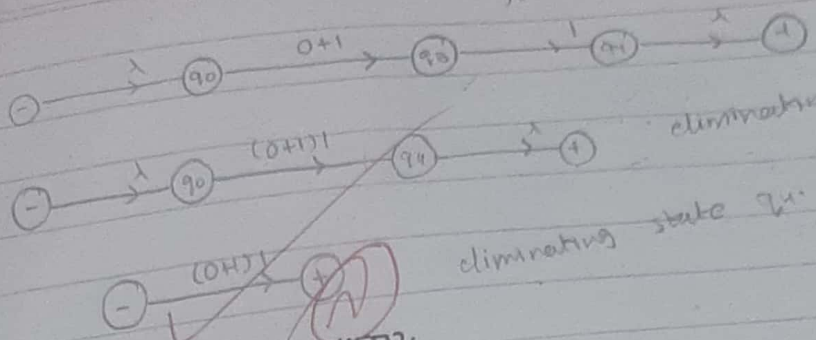


eliminating state q2

116165

Rough Work

Joining transitions from state q0 to q3



eliminating state q3

eliminating state q4.

2

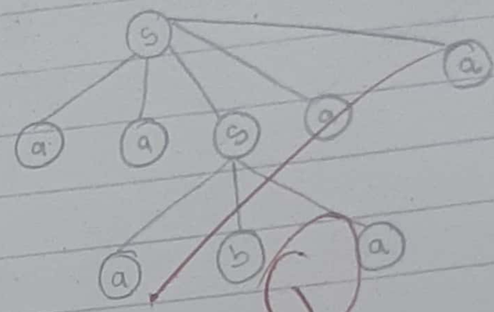
Hence RE is (0+1)1 for NFA2.

QUESTION 5.

b- S → aaSaa | aba

2

f- aaabaaa



1

c- S → AB
B → bBc | bB | bc
A → aA | a

2

aaa. No CFG or RG, so no derivation tree.

e- S → S1 | S2
S1 → aaSaa | aba
S2 → AB
B → bBc | bB | bc
A → aA | a

2

a- S → S S1 | λ
S1 → aa | bb | ABA | λ
A → ab | ba
B → aaB | bbB | λ

there you go bar is here

2

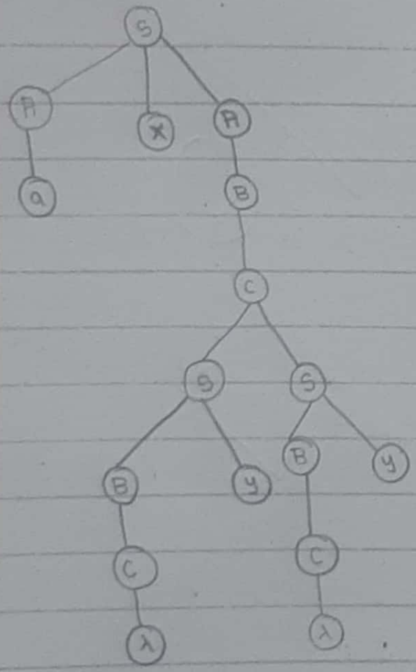
d- Its not context free or regular language therefore its RG or CFG doesnt exist

1

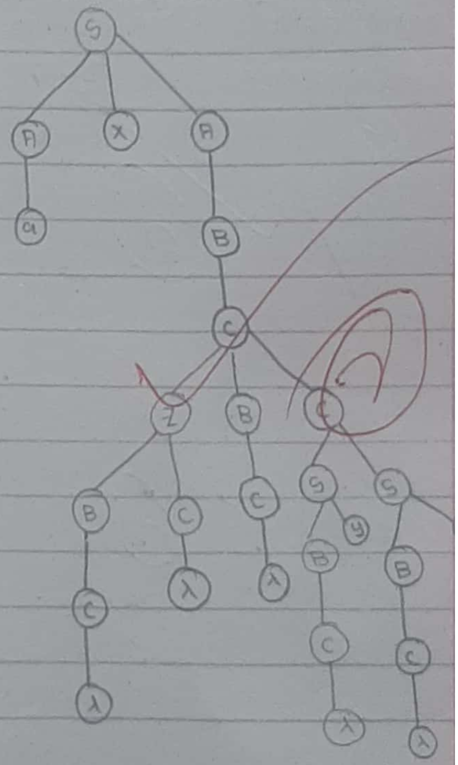
QUESTION 6:

a- No useless productions. CFG will be same.

b- One derivation:



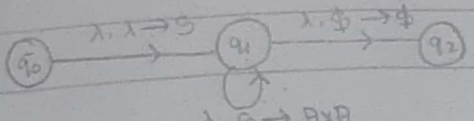
Second derivation:



Two possible derivation trees, hence CFG is ambiguous.

c-

⟶



$\lambda, S \to AxA$

$\lambda, S \to By$

$\lambda, S \to zC$

$\lambda, A \to B$

$\lambda, A \to a$

$\lambda, B \to C$

$\lambda, B \to b$

$\lambda, C \to SS$

$\lambda, C \to c$

$\lambda, C \to \lambda$

$\lambda, C \to ZBC$

$\lambda, Z \to aBZ$

$\lambda, Z \to BC$

$\lambda, Z \to Z$

$x, x \to \lambda$

$y, y \to \lambda$

$z, z \to \lambda$

$a, a \to \lambda$

$b, b \to \lambda$

$c, c \to \lambda$

2

d- $\delta(q_0, \lambda, \lambda) = (q_1, S)$

$\delta(q_1, \lambda, S) = \{(q_1, AxA), (q_1, By), (q_1, zC)\}$

$\delta(q_1, \lambda, A) = \{(q_1, B), (q_1, a)\}$

$\delta(q_1, \lambda, B) = \{(q_1, C), (q_1, b)\}$

$\delta(q_1, \lambda, C) = \{(q_1, SS), (q_1, c), (q_1, \lambda), (q_1, ZBC)\}$

$\delta(q_1, \lambda, Z) = \{(q_1, aBZ), (q_1, BC), (q_1, Z)\}$

$\delta(q_1, a, a) = \{(q_1, \lambda)\}$

$\delta(q_1, b, b) = \{(q_1, \lambda)\}$

$\delta(q_1, c, c) = \{(q_1, \lambda)\}$

$\delta(q_1, x, x) = \{(q_1, \lambda)\}$

$\delta(q_1, y, y) = \{(q_1, \lambda)\}$

$\delta(q_1, z, z) = \{(q_1, \lambda)\}$

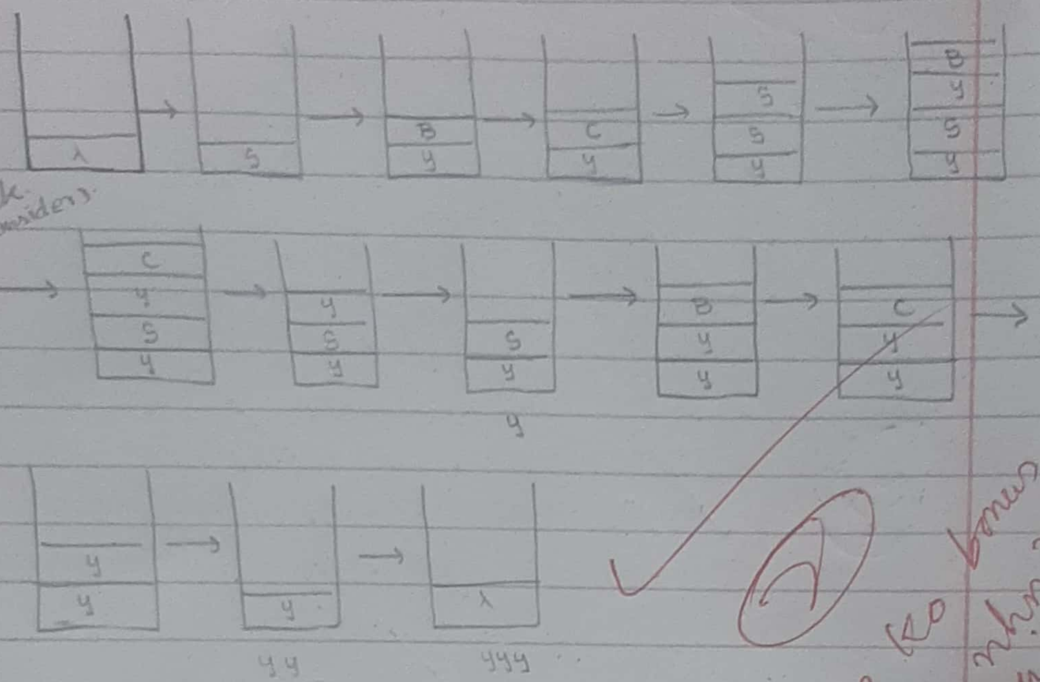
$\delta(q_1, \lambda, \lambda) = (q_2, \lambda)$

2

e- →

Rough Work

add $ at bellow of every stack (Just to consider)



2

AP ko bonus nhi milega paper is sheet !!

QUESTION 7:

(1) Remove D and then B→D as D is useless (cannot derive word completely).

a- S → AB | S → AB
A → aAb | λ | A → aAb | λ
B → bBc | λ | D | B → bBc | λ
C → cC | λ | C → cC | λ
D → aDc

(2) Remove C as it is useless unreachable.
S → AB
A → aAb | λ
B → bBc | λ

(3) null
Remove A → λ
S → AB | B
A → aAb | ab
B → bBc | λ

(4) null
Remove B → λ
S → AB | B | A | λ
A → aAb | ab
B → bBc | bc

(5) Remove S→B (unit)
S → AB | A | λ | bBc | bc
A → aAb | ab
B → bBc | bc

(6) Remove S→A (unit)
S → AB | λ | bBc | bc | aAb | ab
A → aAb | ab
B → bBc | bc

Simplified CFG

2

b- ① Add $S_1 \to S$

$S_1 \to S$

$S \to AB \mid \lambda \mid bBc \mid bc \mid aAb \mid ab$

$A \to aAb \mid ab$

$B \to bBc \mid bc$

② Remove $S \to \lambda$

$S_1 \to S \mid \lambda$

$S \to AB \mid bBc \mid bc \mid aAb \mid ab$

$A \to aAb \mid ab$

$B \to bBc \mid bc$

③ Remove $S_1 \to S$

$S_1 \to AB \mid bBc \mid bc \mid aAb \mid ab \mid \lambda$

$S \to AB \mid bBc \mid bc \mid aAb \mid ab$

$A \to aAb \mid ab$

$B \to bBc \mid bc$

④ Let $U \to bB$, $V \to aA$, $T \to c$, $F \to b$, $G \to$

$S_1 \to AB \mid UT \mid FT \mid VF \mid GF \mid \lambda$

$S \to AB \mid UT \mid FT \mid VF \mid GF$

$A \to VF \mid GF$

$B \to UT \mid FT$

$U \to FB$

$V \to GA$

$T \to c$

$F \to b$

$G \to a$

→ Resultant CNF

7

6

c- $S \to S\, S_1 \mid \lambda$

$S_1 \to AB \mid UT \mid FT \mid VF \mid GF \mid \lambda$

$S_2 \to AB \mid UT \mid FT \mid VF \mid GF$

$A \to VF \mid GF$

$B \to UT \mid FT$

$U \to FB$

$V \to BA$

$T \to c$

$F \to b$

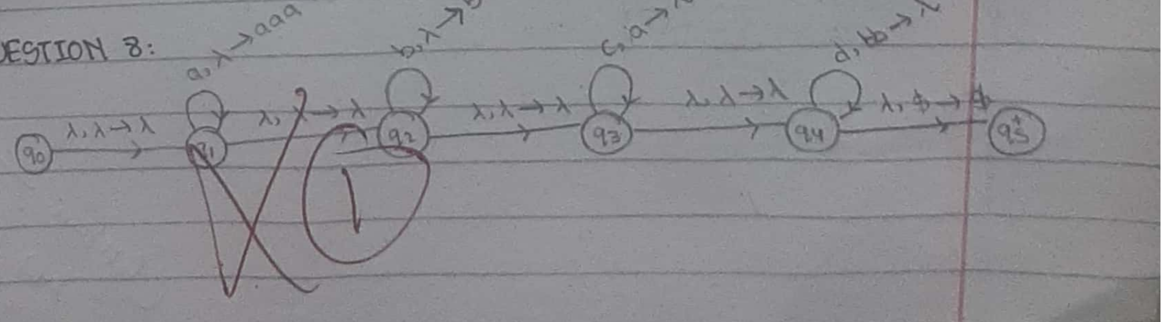
$G \to a$

d- It doesn't remove ambiguity from the CFGs as it only gives chomsky's norm form to CFG. (If CFG can be converted to non-ambiguous form, that's other thing) Also some CFGs are inherently ambiguous. Converting them to CNF will not remove the ambiguity. (Good one)

e- Because it derives the words in less productions. we can see it through derivation tree. There trees will be small hence reducing computational complexities.

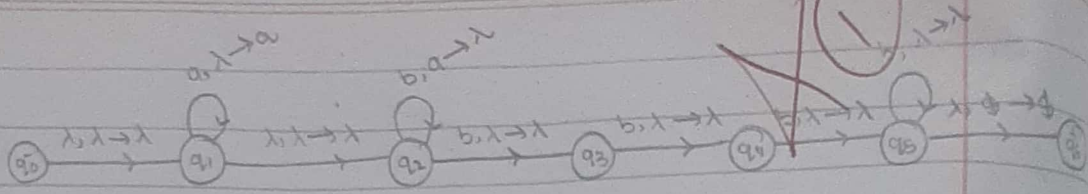
QUESTION 8:

a-



Rough Work

b-



QUESTION 9:

a- This turing machine has infinite loop as it reads a, keep it as a and moves right and reads b, keep it as b and move left so it will continuously move L and R on the loop and halt.

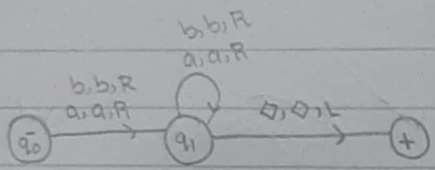
a a b b a ◇

|◇ |a|a|b|b|a|◇

accepted here

2

Error free TM:



Now this TM will read a and b, write a and b and move R respectively unless find ◇ symbol to accept string.

b- accept and halt means when the input is consumed and you are on the final state. Reject and halt means when you enter infinite loop, or you have consumed the input but you are not on the final state so your string is rejected.

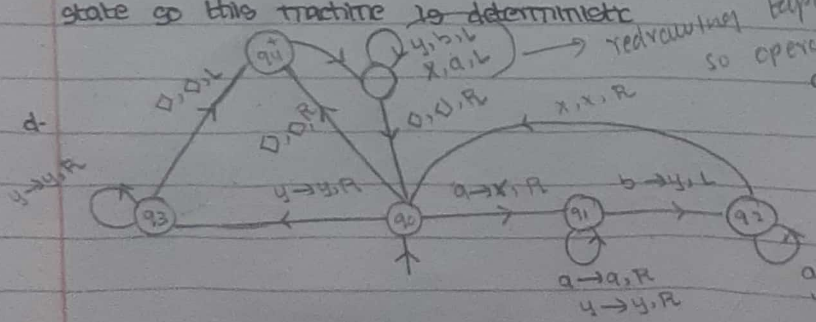
2

c- Turing machines are deterministic in the sense that you cannot have multiple transitions of a single alphabet from a particular state towards other states However missing transitions are allowed of an alphabet from a particular state so this machine ~~is deterministic~~

2

→ redrawing tape so operation can again performed

d-

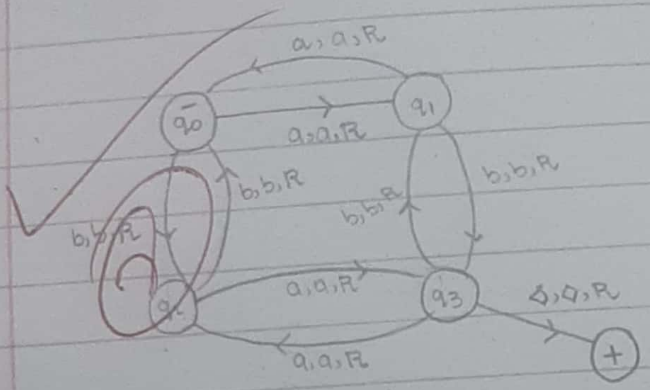


Closure hes issue

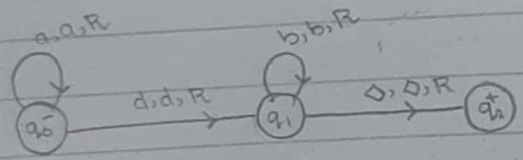
2

e- Yes, it can have, but only in the case when there are no outgoing transitions from those final states.

f-

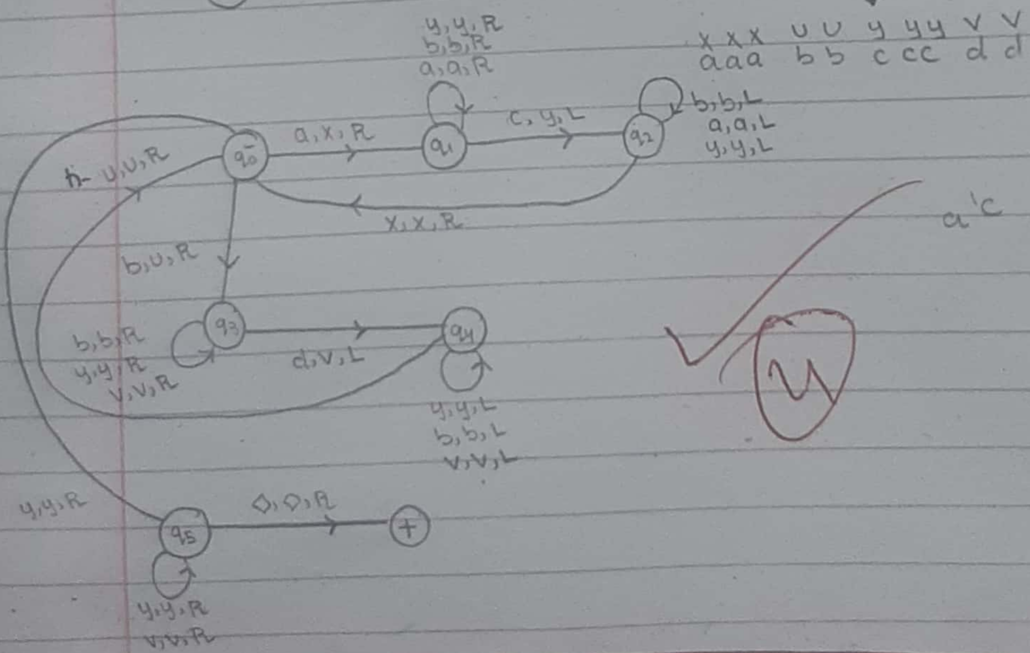


g-



4

h-



a'c

4