CSE2002 Theory of Computation And Compiler Design   DA   WINTER 2021-22

by

Sharadindu Adhikari, 19BCE2105

Q1.

(a) Alphabet, $\Sigma = \{0, 1\}$

(a) DFA for $L = \{ w \mid$ number of 0's in w is in the form (E) $5i+3$, for some natural number $i \}$

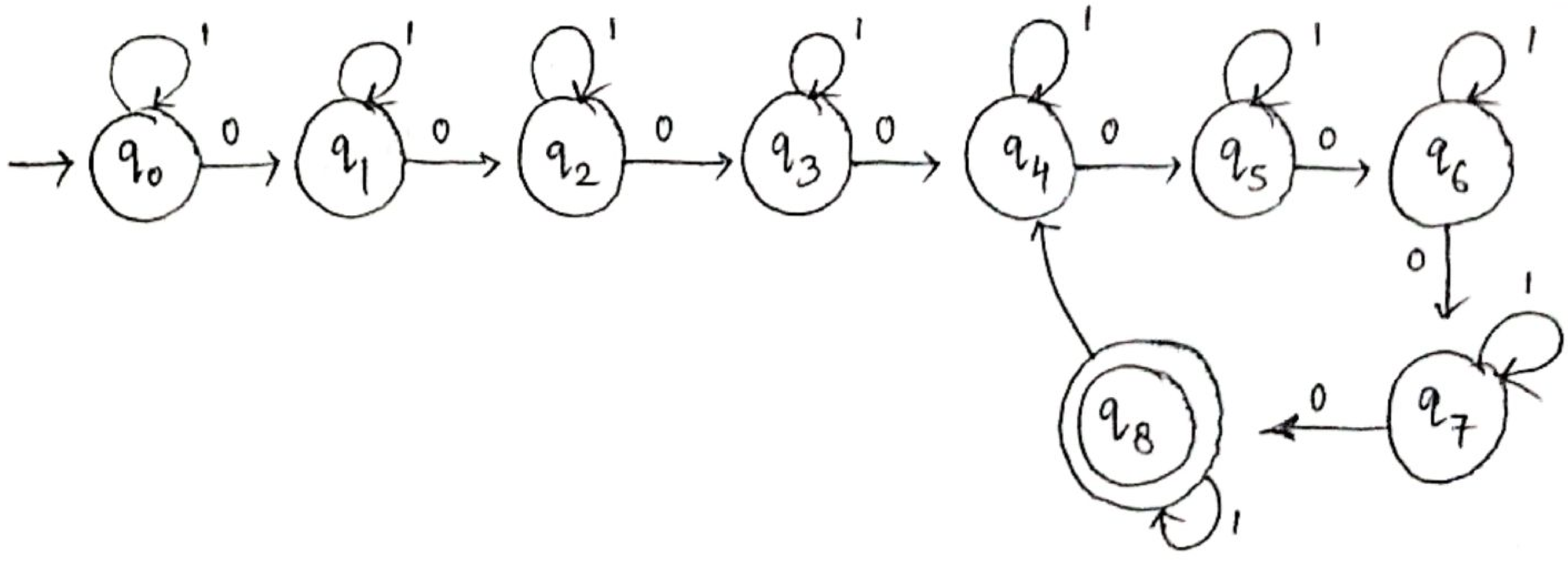
values of i can be 1, 2, 3, 4, 5, ... ⇒ (natural nos.)

No. of 0's in string, $|w|_0$ = 8, 13, 18, 23, 28, 33, ...

$5(1)+3$, $5(2)+3$, $5(3)+3$ ....

No. of 1's in string, $|w|_1$ = can be any number (not mentioned)

So, DFA is :



$q_0 \rightarrow$ initial state ; $q_8 \rightarrow$ final state

Q1. (b)

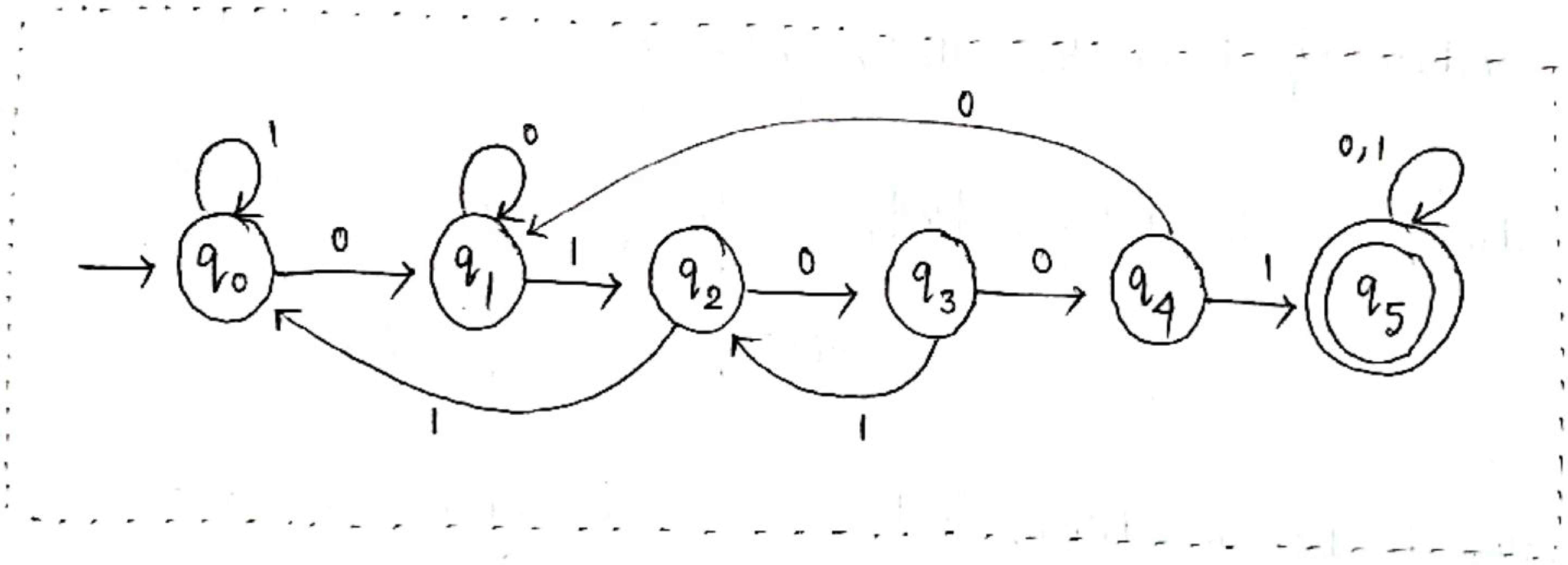
DFA for L = { w | w has 01001 as a substring }.

W = x 01001 x ; x can be $\{0, 1\}^*$

w contain 01001 as substring in every string

Language, L = { 01001, 1101001, 10101001, ... }

So, the DFA is :



Here, $q_0 \rightarrow$ initial state

$q_5 \rightarrow$ final state

L = { w|w has 01001 as a substring }

Q2.

(a) To prove $\{a^p$, where $p = n^2 \mid n >= 1\}$ is regular.

I'll use pumping lemma to prove it.

Let the pumping length be P.

The condition on the length is that it should be

$|s| \geq P.$

$s \longrightarrow$ length of the string

I've to divide the string in 3 parts : xyz, and the conditions are :

(i) $xy^iz \in A$, every $i \geq 0$

(ii) $|y| > 0$

(iii) $|xy| \leq p$

Given that $L = \{a^{p^2} : p \geq 1\}$

Assume that L is a regular language.

Let p be the pumping length for L,

$\therefore \forall w \in L \wedge |w| \geq p = m_0^2, m_0 \in z^+$

$w = xyz : |y| \neq 0, |xy| \leq p \;\&\; (\forall x \geq 0)(xyz \in L)$

Let $|y| = \alpha$, $|xy| = \beta$

$\therefore |z| = |xyz| - |xy|$

if $|xyz| = n^2$, then $|z| = n^2 - \beta$

we know, $xy^iz \in L \ \forall i \geq 0$

$\therefore |xy^iz| = m_1^2$ for some i & $m_1 \in \mathbb{Z}^+$

$\therefore (n^2 - \beta) + (i-1)\alpha + \beta = m_1^2, \quad i \geq 1$

$\Rightarrow n^2 + \alpha_i - \alpha = m_1^2$

Since $n^2 + \alpha i - \alpha$ is a perfect square $\forall i \geq 1$,

Let $i - 1 = n\alpha$

$\therefore n^2 + \alpha^2 n^2 = n^2(\alpha^2 + 1)$

$\because \alpha^2 + 1$ can't be a perfect square,

$n^2(\alpha^2+1)$ can't be a perfect square.

$\therefore \exists\, i \geq 1 : n^2 + \alpha i - \alpha$ is not a perfect square

& $xy^iz \notin L$

This contradicts my assumption that L is a regular language.

Therefore, L is NOT a regular language.

Assume, e.g., the pumping length be 3.

so, $a^9 \Rightarrow \underbrace{aa}_{x}\ \underbrace{aaaaa}_{y}\ \underbrace{aa}_{z}$. so, $x \to aa$

$y \to aaaaa$

$z \to aa$

We can take any i so ~~that~~ $i = 2$.

~~Q~~2. ~~(b)~~ Contd.

i) $xy^iz = aa(aaaaa)^2aa$

$= aa\ aa\ aaaaa\ aaaaa$

We can see in the above string, it's length is 14, which is not a square of a natural number, and according to the first condition, the xy^iz should be a string that comes under the given language, but it is false here.

We can say that this is NOT a regular language.

Let's check other conditions for confirmation.

ii) $|y| > 0$ and $|y| = 5$, so this condition is true.

iii) $|xy| \leq p$, here $p=3$ and $|xy| = 7$, so this condition also contradicts and hence it is proved that this is NOT a regular language.

Q2.

(b) Given,

$$exprs := exprs + expr \mid exprs * expr \mid expr$$

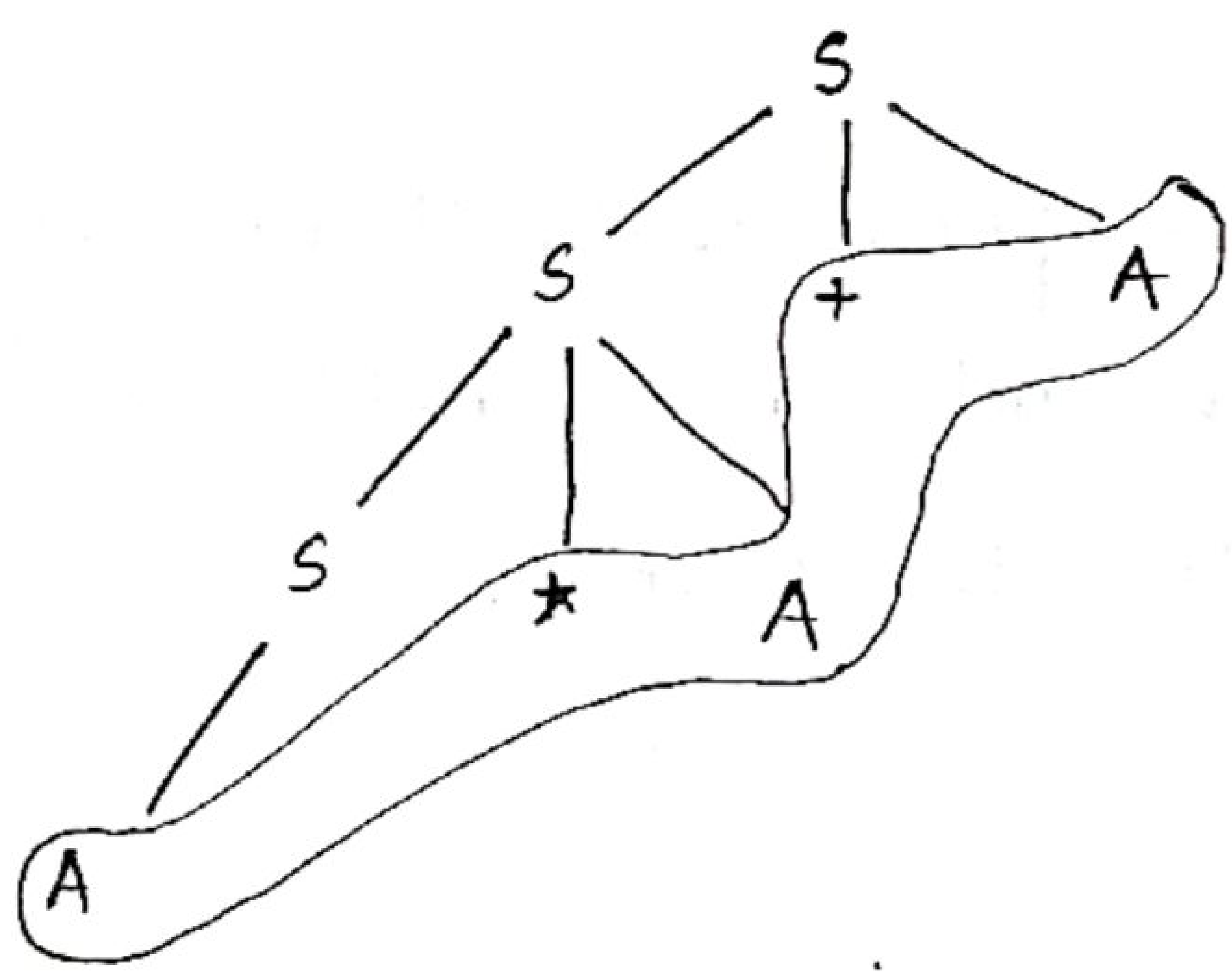
$$expr := X$$

For convenience,

Let $exprs \rightarrow S$, and $expr \rightarrow A$

So, $S \rightarrow S+A \mid S*A \mid A$; $A \rightarrow x$

parse tree is as follows :



∴ only string accepted

$\Rightarrow A * A + A$

$\Rightarrow x * x + x$

Left derivation	Right derivation
$S \rightarrow S + A$	$S \rightarrow S + A$
$S \rightarrow S * A + A$	$S \rightarrow S + x$
$S \rightarrow A * A + A$	$S \rightarrow S * A + x$
$S \rightarrow x * A + A$	$S \rightarrow S * x + x$
$S \rightarrow x * x + A$	$S \rightarrow A * x + x$
$S \rightarrow x * x + x$	$S \rightarrow x * x + x$

We can not make more than 1 parse tree of the given grammar, even for left derivation or right derivation. There is only one ~~left~~ way: left or right derivation to find a particular string in the given language.

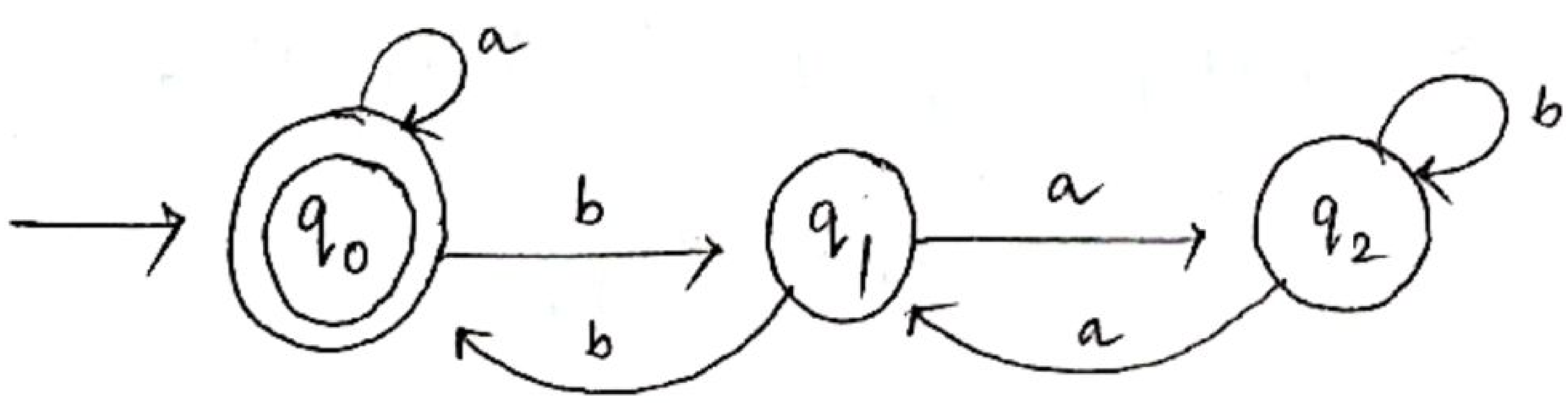
This grammar is, thus, unambiguous.

If we are making a parse tree, it can only expand in the left direction and on the right there is only A, and it would lead to x only.

— * —

19BCE2105

Q3. To find the regular expression accepted by the given automation



$$q_0 = q_1 a + q_1 b + \epsilon$$

$$q_1 = q_0 b + q_2 a$$

$$q_2 = q_2 b + q_1 a$$

$$q_2 = q_2 b + q_1 a \quad \{ q_1 = (q_2 a + q_0 b) \}$$

$$q_2 = q_2 b + q_2 aa + q_0 ba$$

$$q_2 = q_2 (b + aa) + q_0 ba \quad \{\text{Arden's Theorem}\}$$

$$q_2 = q_0 ba (b + aa)$$

$$q_1 = q_0 ba (b + aa)^n + q_0 b$$

$$q_0 = \epsilon + q_0 a + (q_0 ba (b + aa)^n + b) b$$

$$q_0 = \epsilon + q_0 (a + (ba (b + aa)^n + b)$$

$$q_0 = \epsilon (a + (ba (b + aa)^n + b) b)^n$$

$$q_0 = (a + (ba (b + aa)^n + b) b)^n$$

Q4.

$S \to abS \mid acS \mid c$

(a) As it is start state, and has ϵ on right of the 's',

so follow (s) = ~~[illegible]~~ $\{ \$ \}$

(b) parsing table :

	First ( )	Follow ( )
S	$\{a, c\}$	$\{ \$ \}$

parse table construction :

	a	b	c	$
S	$S \to abS$ $S \to acS$		$S \to c$	

As we can see that there is more than 1 entry for non-terminal production to a terminal state, then it is NOT an LL (1) grammar.

Q4.

(c) Parse the string "acacc" = w [if it follows LL(1) by removing $S \rightarrow abS$]

stack	input	production
S \$	acacc \$	$S \rightarrow acS$
acS \$	acacc \$	$S \rightarrow acS$
	(pop a and c; two pop operations)	
$\boxed{S\ \$}$	$\boxed{acc\ \$}$	
acS \$	acc \$	$S \rightarrow c$
	(pop a and c; two pop operations)	
$\boxed{S\ \$}$	$\boxed{c\ \$}$	
c\$	c\$	
	(pop operation)	
\$	\$	
	(pop the stack) : The string is accepted	

19BCE2105

Q5.

$S \rightarrow Sa$

$S \rightarrow Sc$

$S \rightarrow c$

Augmented grammar :

0. $S' \rightarrow S\$$.
1. $S \rightarrow Sa$
2. $S \rightarrow Sc$
3. $S \rightarrow c$

Symbol	first ( )	follow ( )
S	{c}	{\$}
S'	{c}	{a, c, \$}

19BCE2105

Q5 (b)

	Action						Go to
state	+	*	(	)	num	$	S
0			shift 2		shift 1		3
1	reduce S→num	reduce S→num				reduce S→num	
2			shift 6		shift 4		5
3	shift 14	shift 15				Accept	
4	Reduce S→num	Reduce S→num		Reduce S→num			
5	shift 10	shift 11		shift 7			
6			shift 6		shift 4		8
7	Reduce S→(s)	Reduce S→(s)				Reduce S→(s)	
8	shift 10	shift 11		shift 9			
9	Reduce S→(s)	Reduce S→(s)		Reduce S→(s)			
10			shift 6		shift 4		12
11			shift 6		shift 4		13

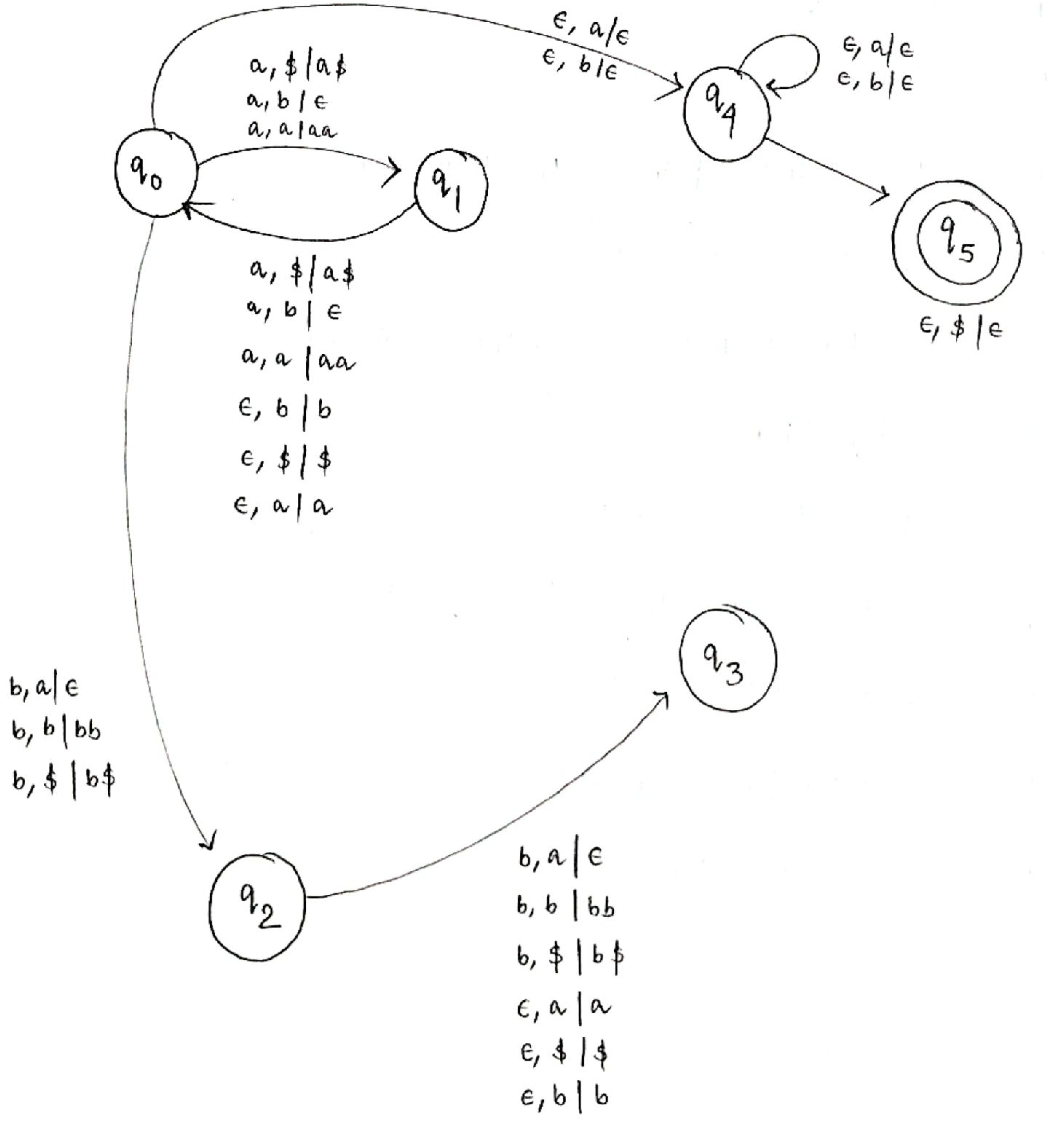
state	Action: +	*	(	)	num	$	Go to: S
12	shift 10 reduce S→S*S	shift 11 reduce S→S*S		Reduce S→S*S			
13	shift 10 reduce S→S*S	shift 11 reduce S→S*S		Reduce S→S*S			
14			shift 2		shift 1		16
15			shift 2		shift 1		17
16	shift 14 reduce S→S+S	shift 15 reduce S→S+S			reduce S→S+S		
17	shift 14 reduce S→S*S	shift 15 reduce S→S*S			reduce S→S*S		

19BCE2105

Q6. (a)

language, $L = \{ w \mid 2\#_a(w) \neq 3\#_b(w),\ w \in \{a,b\}^* \}$,

where, $\#_a(w)$ and $\#_b(w)$ denotes the number of a's and b's occurring in the string w.



Q6. (b) $L_1 = \{0^n 1^m \mid 0 < n \leq m < 2n\}$

G_1 : $S \rightarrow 0S11$
$S \rightarrow T$
$T \rightarrow 0T1$
$T \rightarrow 01$

- Case-1: $n = m$

$S \rightarrow T \rightarrow 0T1 \rightarrow \ldots\ldots$ n-1 times
$\rightarrow 0^{n-1} T 1^{n-1}$
$\rightarrow 0^n 1^n$
$\rightarrow 0^n 1^m$; ~~hence~~ hence proved

$S \rightarrow T$ applied 1 time
$S \rightarrow 0T1$ applied n-1 time
$T \rightarrow 01$ applied 1 time

- Case-2: $m = n+k$, $k > 1$ & $k < n$

$S \rightarrow 0S11 \rightarrow \ldots\ldots$ k times
$\rightarrow 0^k S 1^{2k}$
$\rightarrow 0^k T 1^{2k}$
$\rightarrow 0^{k+1} T 1^{2k+1} \rightarrow \ldots\ldots$ (n-k-1) times
$\rightarrow 0^{n-1} T 1^{n+k-1}$
$\rightarrow 0^n 1^{n+k}$
$\rightarrow 0^n 1^m$; hence proved

$S \rightarrow 0S11$ applied K times

$S \rightarrow T$ applied 1 time

$T \rightarrow 0T1$ applied $(n-K-1)$ times

$T \rightarrow 01$ applied 1 time

Q7.

$S \rightarrow ABaC$

$A \rightarrow BC$

$B \rightarrow b / \lambda$

$C \rightarrow D / \lambda$

$D \rightarrow d$

we've :

/1/ Elimination of null prod^n :-

$B \rightarrow E$

$S \rightarrow ABaC \mid AaC$

$A \rightarrow BC \mid C$

$B \rightarrow B$

$C \rightarrow D \mid \epsilon$

$D \rightarrow d$

$C \to \epsilon$

$S \to ABaC \mid AaC \mid ABa \mid Aa$

$A \to BC \mid c \mid B \mid \epsilon$

$B \to b$

$C \to D$

$D \to d$

$A \to \epsilon$

$S \to ABaC \mid AaC \mid ABa \mid Aa \mid BaC \mid aC \mid Ba \mid a$

$A \to BC \mid c \mid B$

$B \to b$

$C \to D$

$D \to d$

/2/ Elimination of unit production :-

$A \to C$, $A \to B$, and $C \to D$ are unit productions

$S \to ABaC \mid AaC \mid ABa \mid Aa \mid BaC \mid aC \mid Ba \mid a$

$A \to BC \mid D \mid b$

$B \to b$

$C \to d$

$D \to d$

Another unit production, $A \rightarrow 0$ arised.

$S \rightarrow ABaC \mid AaC \mid ABa \mid Aa \mid BaC \mid aC \mid Ba \mid a$

$A \rightarrow BC \mid d \mid b$

$B \rightarrow b$

$C \rightarrow d$

$D \rightarrow d$

∵ D can't be reached from start state, D is useless.

$S \rightarrow ABaC \mid AaC \mid ABa \mid Aa \mid BaC \mid aC \mid Ba \mid a$

$A \rightarrow BC \mid d \mid b$

$B \rightarrow b$

$C \rightarrow d$

/3/ Elimination of single terminals :-

$S \rightarrow ABXC \mid AXC \mid ABX \mid AX \mid BXC \mid XC \mid BX \mid a$

$A \rightarrow BC \mid b \mid d$

$B \rightarrow b$

$C \rightarrow d$

$X \rightarrow a$

19BCE2105

4| Elimination of more than 2 non-terminals together :—

$S \longrightarrow ABZ \mid AZ \mid AY \mid BZ \mid XC \mid BX \mid a$

$A \longrightarrow BC \mid d \mid b$

$B \longrightarrow b$

$C \longrightarrow d$

$X \longrightarrow a$

$Z \longrightarrow XC$

$Y \longrightarrow BX$

$S \rightarrow AU \mid AZ \mid AY \mid BZ \mid XC \mid BX \mid a$

$B \rightarrow b$

$A \rightarrow BC \mid d \mid b$

$C \rightarrow d$

$X \rightarrow a$

$Z \rightarrow XC$

$Y \rightarrow BX$

$U \rightarrow BZ$

This given grammar is the Chomsky Normal Form of the given context free grammar.

Q8.

Given, $L = \{a^l b^m c^n \mid l+m=n;\ m,n>0\}$ —— ①

Also, $n = l+m$,

so we can write c^n as $c^{l+m} = c^l \cdot c^m$,

and then eq. ① becomes:

$$L = \{a^l b^m c^l c^m \mid l+m=n;\ m,n>0\}$$

or, $$L = \{a^l b^m c^m c^l \mid l+m=n;\ l\geq 0;\ n,m>0\}$$

Let's take S as $a^l . b^m . c^m . c^l$, and B as $b^m . c^m$.

Thus, the required CFG:

$$S \rightarrow aSc \mid A \ ; \quad A \rightarrow bAc \mid bc$$

Now, the string $a^2b^2c^4$ or, aabbcccc

$S \rightarrow aSc$

$\rightarrow aaScc \quad \{S \rightarrow aSc\}$

$\rightarrow aaAcc \quad \{S \rightarrow A\}$

$\rightarrow aabAccc \quad \{A \rightarrow bAc\}$

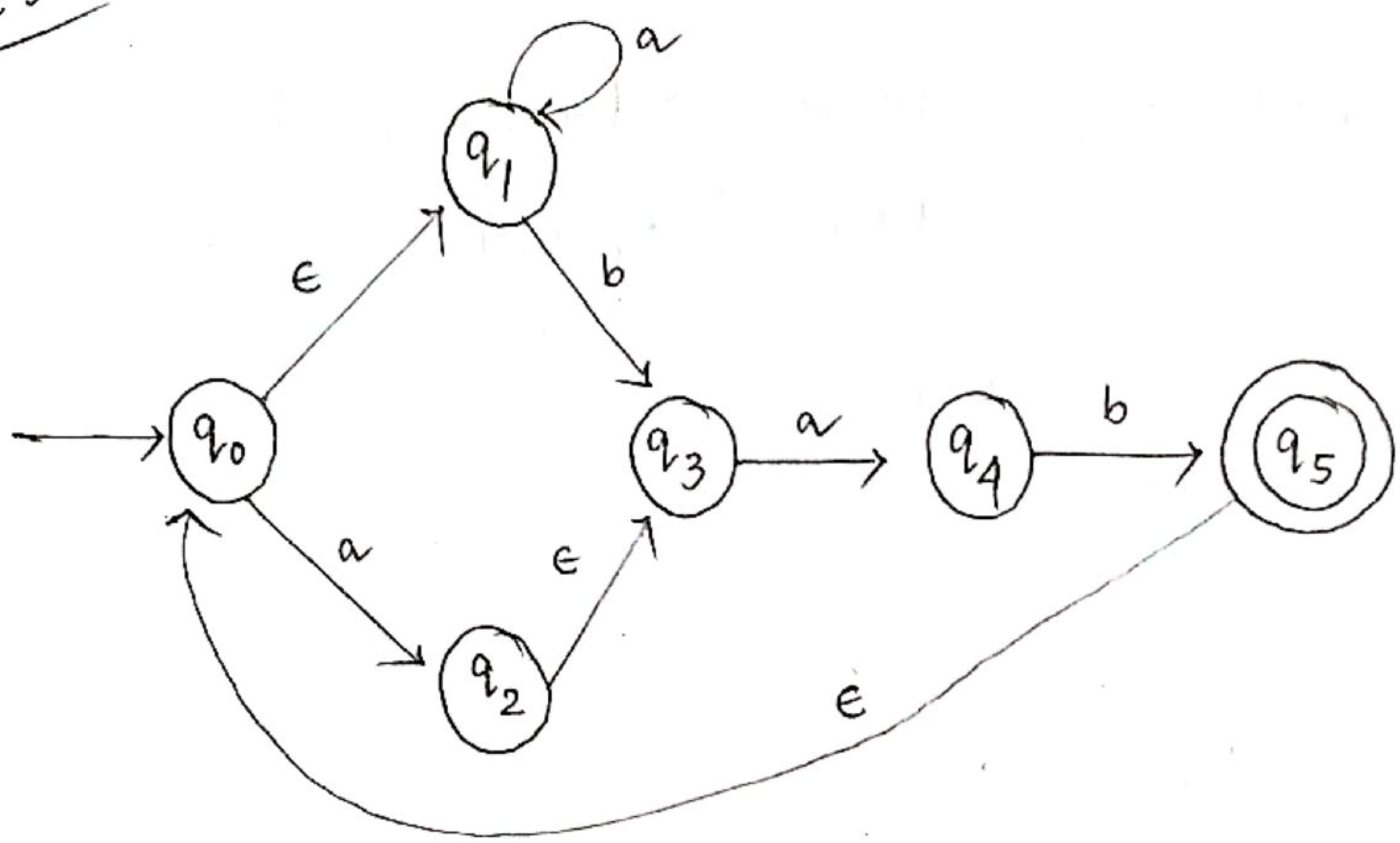
$\rightarrow aabbcccc \quad \{A \rightarrow bc\}$

$\rightarrow a^2b^2c^4$

proved

19BCE2105

Q9.



ϵ - closure (q_0) = $\{q_0, q_1\}$

ϵ - closure (q_1) = $\{q_1\}$

ϵ - closure (q_2) = $\{q_2, q_3\}$

ϵ - closure (q_3) = $\{q_3\}$

ϵ - closure (q_4) = $\{q_4\}$

ϵ - closure (q_5) = $\{q_5, q_0, q_1\}$

We've :

$$\delta'(q_0, a) = \epsilon\text{ - closure }(\delta(\epsilon\text{ - closure }(q_0)), a)$$
$$= \epsilon\text{ - closure }(\delta(q_0, q_1), a)$$
$$= \epsilon\text{ - closure }(q_2, q_1)$$
$$= \{q_2, q_3, q_1\}$$

again,

$$\delta'(q_0, b) = \epsilon\text{-closure}\left(\delta(q_0, q_1), b\right)$$
$$= \epsilon\text{-closure}\left(\phi, q_3\right)$$
$$= \{q_3\}$$

Similarly,

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

$\delta'(q_2, b) = \{q_2, q_3\}$

$\delta'(q_4, a) = \{\phi\}$

$\delta'(q_5, b) = \{q_3\}$

$\delta'(q_1, b) = \{q_3\}$

$\delta'(q_3, a) = \{q_4\}$

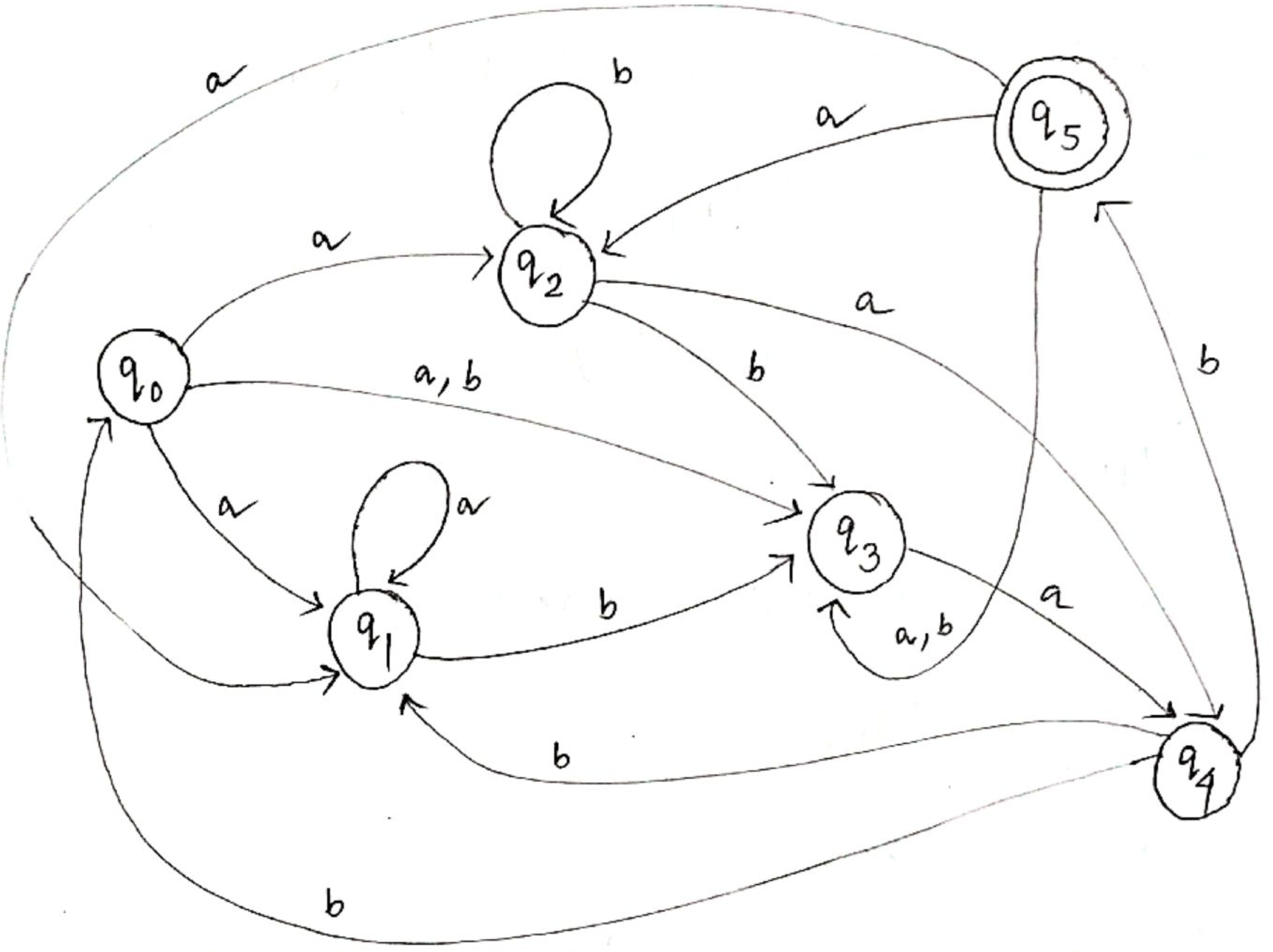
$\delta'(q_4, b) = \{q_5, q_0, q_1\}$

$\delta'(q_2, a) = \{q_4\}$

$\delta'(q_3, b) = \{\phi\}$

$\delta'(q_5, a) = \{q_1, q_2, q_3\}$

Reqd. NFA of the given NFA- ϵ. :-



Now, we've :

$\delta'(q_0, a) = \{q_2, q_3, q_1\}$ (new)

$\delta'(q_0, b) = \{q_3\}$ (new)

$\delta'((q_2, q_3, q_1), a) = \{q_4, q_1\}$ (new)

$\delta'((q_2, q_3, q_1), b) = \{q_2, q_3\}$ (new)

$\delta'(q_3, a) = \{q_4\}$ (new)

$\delta'(q_3, b) = \{\emptyset\}$

$\delta'((q_4, q_1), a) = \{q_1\}$ (new)

$\delta'((q_4, q_1), b) = \{q_5, q_0, q_1, q_3\}$ (new)

$\delta'((q_2, q_3), a) = \{q_4\}$

$\delta'((q_2, q_3), b) = \{q_2, q_3\}$

$\delta'((q_4), a) = \{\phi\}$

$\delta'(q_4, b) = \{q_5, q_0, q_1\}$ (new)

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

$\delta'(q_1, b) = \{q_3\}$

$\delta'((q_5, q_0, q_1, q_2, q_3), a) = \{q_1, q_2, q_3, q_4\}$ (new)

$\delta'((q_5, q_0, q_1, q_3), b) = \{q_3\}$

$\delta'((q_5, q_0, q_1), a) = \{q_1, q_2, q_3\}$

$\delta'((q_5, q_0, q_1), b) = \{q_3\}$

$\delta'((q_1, q_2, q_3, q_4), a) = \{q_1, q_4\}$

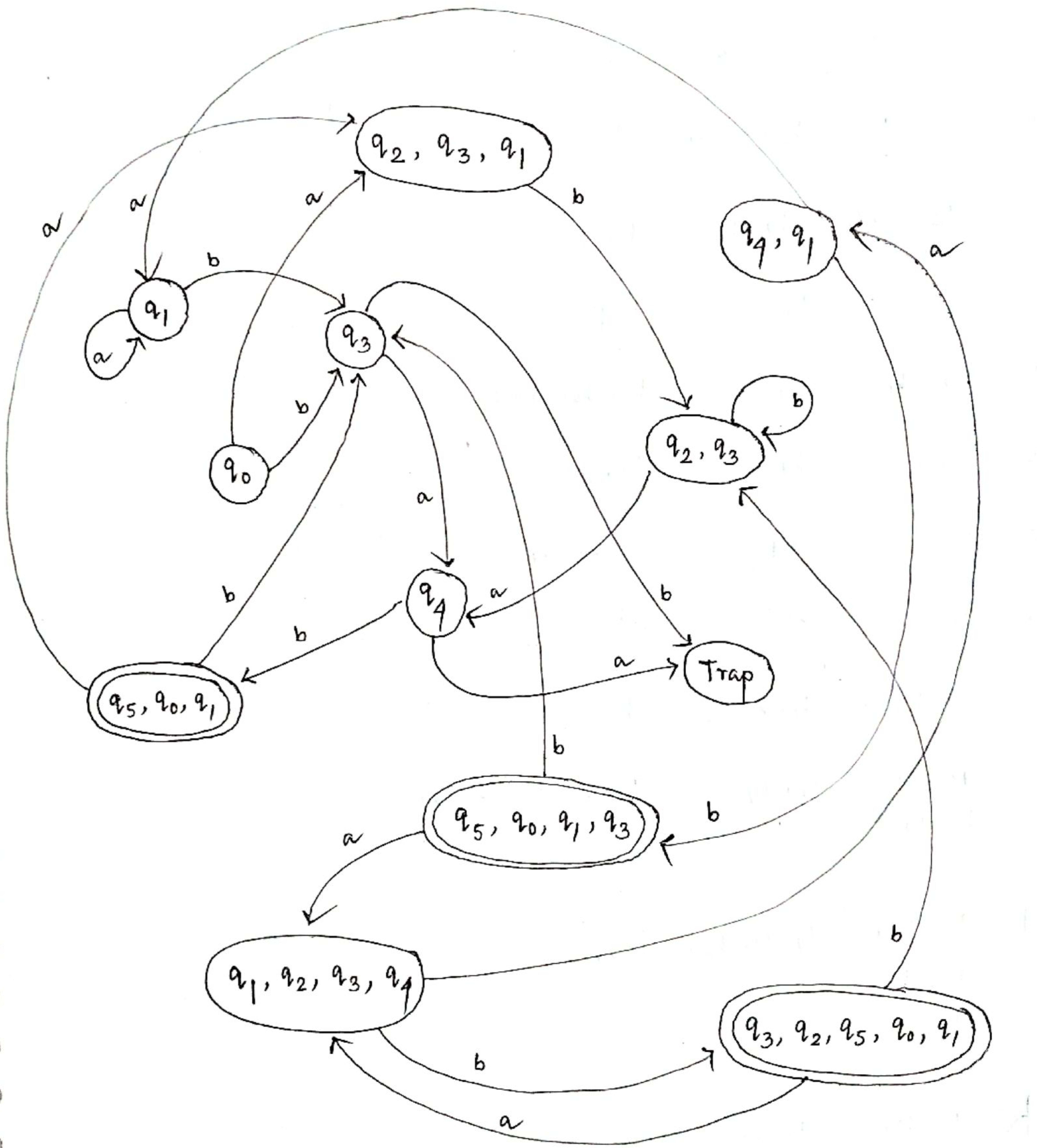
$\delta'((q_1, q_2, q_3, q_4), b) = \{q_3, q_2, q_5, q_0, q_1\}$ (new)

$\delta'((q_3, q_2, q_5, q_0, q_1), a) = \{q_4, q_1, q_2, q_3\}$

$\delta'((q_3, q_2, q_5, q_0, q_1), b) = \{q_2, q_3\}$

19BCE2105

Reqd. DFA of the NFA I just formed in the previous page :—



Now, Minimizing the DFA by Myhill-Nerode Theorem :-

	state	a	b
A	q_0	q_2, q_3, q_1	q_3
B	q_2, q_3, q_1	q_4, q_1	q_2, q_3
C	q_3	q_4	Trap
D	q_4, q_1	q_1	q_5, q_0, q_1, q_3
E	q_2, q_3	q_4	q_2, q_3
F	q_4	Trap	q_5, q_0, q_1
G	q_1	q_1	q_3
H	q_5, q_0, q_1	q_1, q_2, q_3	q_3
I	q_1, q_2, q_3, q_4	q_1, q_4	q_3, q_2, q_5, q_0, q_1
J	q_3, q_2, q_5, q_0, q_1	q_4, q_1, q_2, q_3	q_2, q_3
K	q_5, q_0, q_1, q_3	q_1, q_2, q_3, q_4	q_3
L	Trap	Trap	Trap

19BCE2105

	A	B	C	D	E	F	G	H	I	J	K	L
A	X											
B	✓	X										
C	✓	✓	X									
D	✓	✓	✓	X								
E	✓	✓	✓	✓	X							
F	✓	✓	✓	✓	✓	X						
G	✓	✓	✓	✓	✓	✓	X					
H	✓	✓	✓	✓	✓	✓	✓	X				
I	✓	✓	✓	✓	✓	✓	✓	✓	X			
J	✓	✓	✓	✓	✓	✓	✓	✓	✓	X		
K	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	
L	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	X

As we can see, all the values in the table has been filled, so it is already in its minimized form.

No further minimization is required.