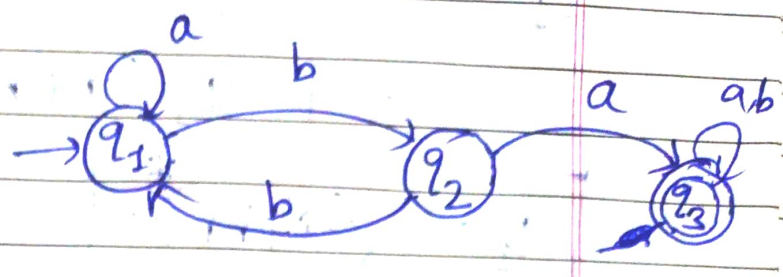
TE_CE_V_501_QP3 Solution

Q.1) Arden's Theorm.

δ	a	b
→ q_1	q_1	q_2
q_2	q_3	q_1
* q_3	q_3	q_3



--- 2 marks

Statement: Arden's theorm states that if P and Q are regular expressions over an alphabet then the equation,

$R = Q + RP$ has the unique sol^n

$$\boxed{R = QP^*}$$ provided that ϵ is not contained in P.

Similarly for the equation,

$R = PR + Q$ the solution is,

$$\boxed{R = P^*Q}$$

Consider state q_1

$$q_1 = q_1a + q_2b + \epsilon \quad \ldots (1)$$

$$q_2 = q_1b \quad \ldots (2)$$

$$q_3 = q_3a + q_2a + q_3b \quad \ldots (3)$$

Step 2 :- Substitute q_2 into q_1

$$q_1 = \epsilon + q_1 a + q_1 bb$$

$$= \epsilon + q_1(a+bb)$$

$$\boxed{q_1 = (a+bb)^*}$$

find q_2

$$q_2 = q_1 \cdot b$$

$$\boxed{q_2 = (a+bb)^* b}$$

Substituting q_2 in q_3

$$\therefore q_3 = (a+bb)^* ba + q_3(a+b)$$

$$\boxed{\therefore q_3 = (a+bb)^* ba(a+b)^*}$$ final state

↳ Deriving RE = 3 marks

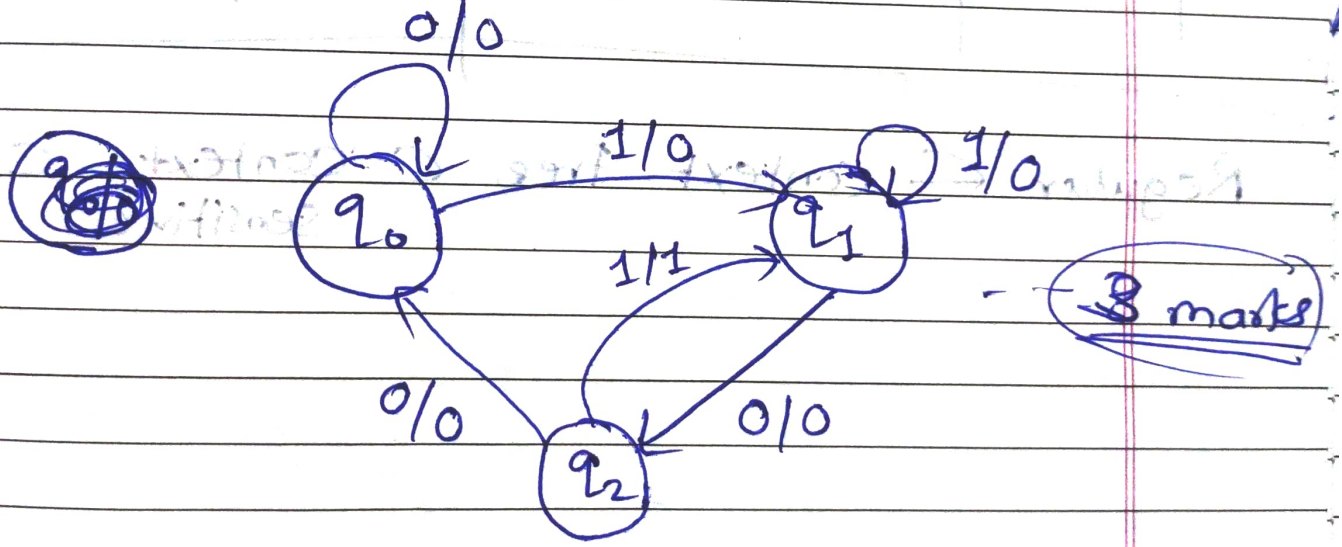
B) Mealy machine for detecting 101

q_0 = initial state
q_1 = last symbol read was 1
q_2 = last symbol read were 10

Transition Table

	0	1
q_0	$q_0/0$	$q_1/0$
q_1	$q_2/0$	$q_1/0$
q_2	$q_0/0$	$q_1/1$

2 marks



3 marks

Output for input string 1010101

It produces 0010101

The outputs are 1 at positions 3, 5, and 7 which are exactly the points where the last three i/p symbols are 101.

c) Chomsky Hierchy ... 5 marks

Proposed by Noam Chomsky in 1956, classifies formal grammars (languages) into four types

Type	Grammar	Language	Production Rules	M/C
0	Unrestricted	R.E	$\alpha \to \beta$	TM
1	Context-Sensitive	context sensitive	$\alpha A \beta \to \alpha \gamma \beta$	LBA
2	Context-free	context free	$A \to \gamma$	PDA
3	Regular	Regular	$A \to aB$ or $A \to a$ or $A \to Ba$ or $A \to a$	DFA or NFA

Regular $\subset$ context-free $\subset$ context sensitive $\subset$ Recursively Enumerable

D) PDA is 7 tuple $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ — 2 marks

finite
Q = set of states
Σ = i/p alphabet
Γ = stack alphabet
$\delta = Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma$ finite subset of $Q \times \Gamma^*$ is δ

Two ways in which PDA accepts a lang.

(1) Acceptance by final state — 1.5 marks

The PDA accepts a string if after reading the entire i/p, it is a final state. The contents of stack does not matter.

$$L(P) = \{ \omega \in \Sigma^* \mid (q_0, \omega, Z_0) \vdash^* (q, \epsilon, \gamma) \}$$
$q \in F$ and $\gamma \in \Gamma^*$

(2) Acceptance by empty stack — 1.5 marks

The PDA accepts a string if after reading the entire i/p, the stack is completely empty. final state does not matter, so F is not needed.

$$N(P) = \{ \omega \in \Sigma^* \mid (q_0, \omega, Z_0) \vdash^* (q, \epsilon, \epsilon) \text{ for some } q \in Q \}$$

(E) Variants of TM.

(i) Multitrack TM - 2 marks
Def^n & Diagram of each type

(ii) Multitape TM - 2 marks

(iii) Universal TM - 1 mark

(F) PCP

i	A_i	B_i
1	1	111
2	10111	10
3	10	0

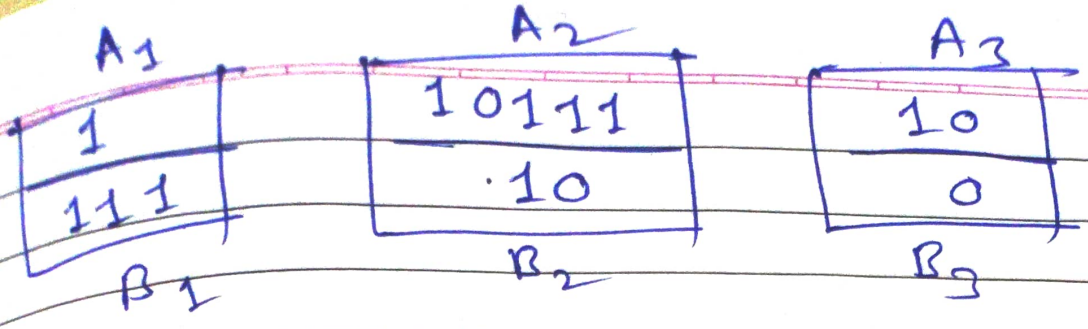
Def^n - 2 marks

An instace of PCP consists of two lists of non-empty strings over an alphabet Σ with same no of strings in each list.

$$A = \{A_1, A_2 \cdots A_n\} \quad B = \{B_1, B_2 \cdots B_n\}$$

Question is whether a given instance has solution or not

PCP = undecidable - no algorithm can decide it for every instance



① $\frac{A_2}{B_2} = \frac{10111}{10}$

② $\frac{A_1}{B_1} \quad \frac{101111}{10111}$

③ $\frac{A_1}{B_1} = \frac{10111111}{10111111}$

④ $\frac{A_3}{B_3} = \frac{101111110}{101111110}$

3 mates

So the sequence 2, 1, 1, 3 has solution

Q. 2 A)

(1) RE for language that contains strings with 01 as substring over $\Sigma = \{0,1\}$

$$RE = (0+1)^* 01 (0+1)^*$$

Examples 01, 1011, 0001, 110100

accepted strings

rejected strings = 0, 1, 10, 11000 etc

(2) RE for language ends with 'or' third symbol from right end is 0.

$$(0+1)^* 0 (0+1)(0+1)$$

e.g. 000, 011, 001, 111

Q.2) B)

Pumping Lemma Theorm :-

Statement and conditions = 2 marks

$L = \{0^n 1^n \mid n \geq 0\}$ is not regular

$L = \{01, 0011, 000111, \ldots\}$

$w = 0011$ $|w| = n$ any +ve int

Divide string into 3 parts,

$\underset{u}{0}\underset{v}{0}\underset{w}{1}1$

3 marks for proof

① cond^n is True because

$|uv| \leq n$

$|001| \leq 4$

② $|v| \geq 1$

$|01| \geq 1$

③ uv^iw $i \geq 0$ $\in L$

$i = 0$, $(0)^0(1)^0 = \epsilon$ empty string $\in L$

$i = 1$

$a(a^* b)^1 b \in L$

$u\ v\ w$

$i = 2$

$a(a^* b)^2 b$

$aababb \notin L$

contradiction so the given lang. is non-regular

(2) Decision properties of Regular languages (Any five properties)

All closure properties of regular languages

Kleen's closure $L^* \in L$ Regular

Concatenation $L1 \cdot L2$ is regular

Union $L1 \cup L2$ Union

Intersection $L1 \cap L2$ —

Complement $L^* = L$

5 marks

Q. 3 A)

Transition Table

δ	a	b	c	ϵ
$\rightarrow q_0$	$\{q_0\}$	ϕ	ϕ	$\{q_1\}$
q_1	ϕ	$\{q_1\}$	ϕ	$\{q_2\}$
$* q_2$	ϕ	ϕ	$\{q_2\}$	ϕ

Step 1 ϵ - closure

ϵ-closure$(q_0) = \{q_0, q_1, q_2\} = A$

ϵ-closure$(q_1) = \{q_1, q_2\} = B$

ϵ-closure$(q_2) = \{q_2\} = C$

2 marks

Step 2 subset construction

4 marks

$A = \{q_0, q_1, q_2\}$

$\{q_0, q_1, q_2\}$

a: $\delta \in \{q_0\}$ closure $= A$

b: $\delta = \{q_1\} \cdot \{q_1, q_2\} = B$

c: $\delta \in \{q_2\}$ closure of (q_2) C

$B = \{q_1, q_2\}$

$a = \phi \rightarrow D$ ~~dead~~ dead state

$b = \delta = \{q_1\}$ closure $= \{q_1, q_2\} = B$

$c = \delta \{q_2\}$ closure $= C$

$C = \{q_2\}$

$a = \phi \rightarrow D$

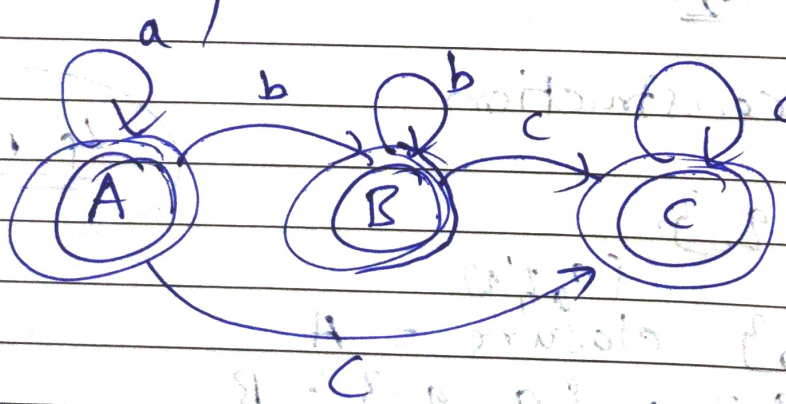
$b = \phi \rightarrow D$

$c = \phi \{q_2\} \rightarrow C$

NFA - DFA Conversion

DFA Transition Table → 4 marks

	a	b	c
*A	A	B	C
*B	D	B	C
*C	D	D	C
D	D	D	D



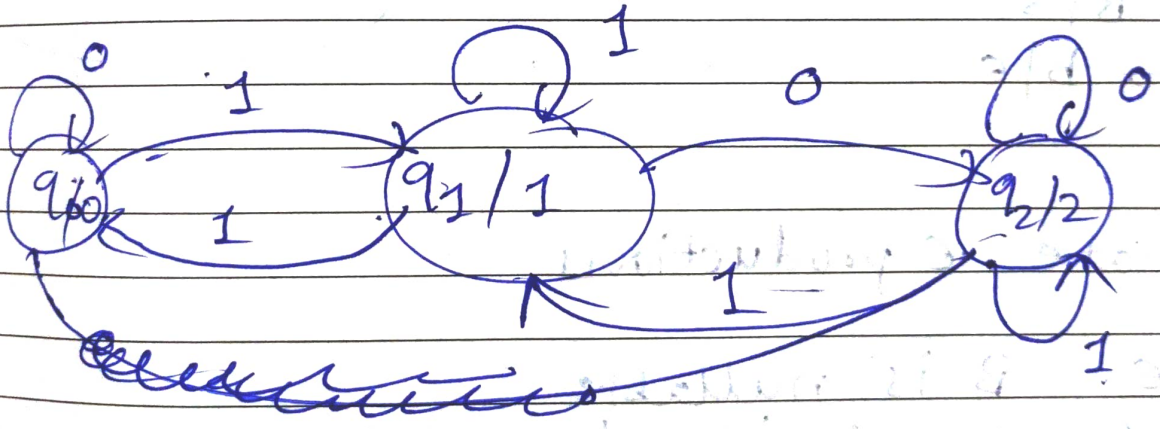
Q.3 B)

Moore machine remainder mod 3 of the binary number.

Next remainder / Transitions $= (2r + b) \bmod 3$

Current state	0	1	X(0/1)
→ q_0	$q_0(2\cdot0+0=0)$	$q_1(2\cdot0+1)=1$	0
q_1	$q_2(2)$	$q_0(2\cdot1+1)=3=0$	1
q_2	$q_1(2\cdot2+0)=4=1$	$q_2(2\cdot2+1)=5=2$	2

3 marks



Conversion to Mealy m/c

Current state	Input 0	Input 1
q_0	$q_0/0$	$q_1/1$
q_1	$q_2/2$	$q_0/0$
q_2	$q_1/1$	$q_2/2$

Date
Page No.

Same ~~exple~~ example

$w = 1010$

Output is 1221

Moore machine output is ~~01221~~ 01221

Mealy machine output is 1221

Q.4 A)

Given Grammar,

$S \rightarrow ASA \mid aB$
$A \rightarrow B \mid S$
$B \rightarrow b \mid \epsilon$

6 marks

① Eliminate ϵ productions

$B \rightarrow \epsilon$ B is nullable
$A \rightarrow B$ so A is also nullable

S not nullable,

Nullable set is $\{A, B\}$

for each production add every version obtained by deleting nullable variable

Page No. Date

and Drop $B \rightarrow \epsilon$

therefore,

$S \rightarrow ASA \mid SA \mid AS \mid aB \mid a$

$A \rightarrow B \mid S$

$B \rightarrow b$

Step 2 :- Eliminate Unit productions

$A \rightarrow B$ — replace with productions of B, $A \rightarrow b$
$A \rightarrow S$ — replace with productions of S,

$A \rightarrow ASA \mid SA \mid AS \mid aB \mid b$

therefore,

$S \rightarrow ASA \mid SA \mid AS \mid aB \mid a$

$A \rightarrow b \mid ASA \mid SA \mid AS \mid aB \mid a$

$B \rightarrow b$

Step 3 :- Eliminate useless symbols

$B \rightarrow b$, $A \rightarrow b$, $S \rightarrow a$ reachable

No symbol is useless.

4 marks

Conversion to CNF

$S \rightarrow AD_1 | SA | AS | C_aB | a$

$A \rightarrow AD_1 | SA | AS | C_aB | a | b$

$B \rightarrow b$

$D_1 \rightarrow SA$

$C_a \rightarrow a$

Q.4B) Given grammar,

$S \rightarrow aB | bA$

$A \rightarrow a | aS | bAA$

$B \rightarrow b | bS | aBB$

G generates strings over $\Sigma = \{a, b\}$ with equal no. of a's and b's

A is variable for strings with one extra a

B is variable for strings with one extra b

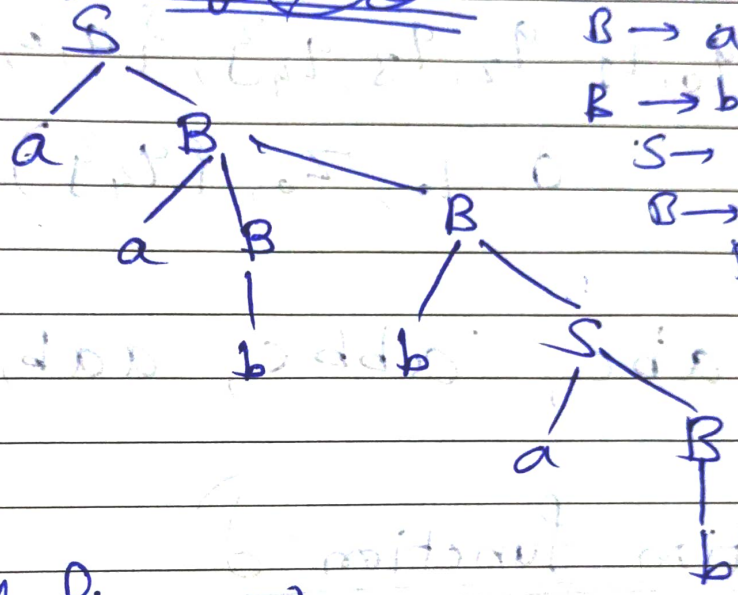
(1) LMD

$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aabBB$
$\Rightarrow aabbS$
$\Rightarrow aabbaB$
$\Rightarrow aabbab$

(2) RMD

$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aaBbS$
$\Rightarrow aaBbaB$
$\Rightarrow aaBbab$
$\Rightarrow aabbab$

Parse Tree ⇒ ~~Right most~~

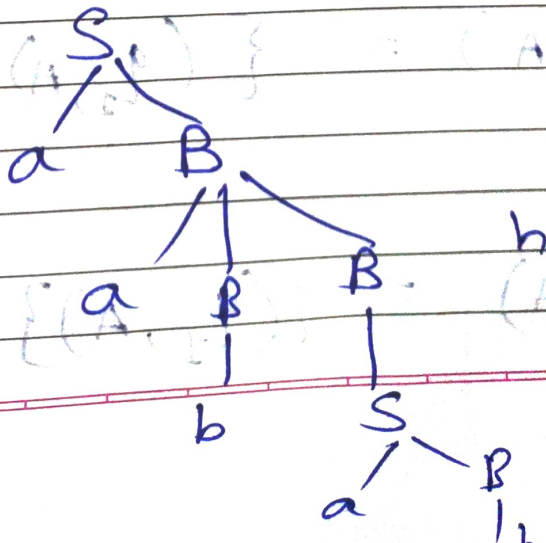


Rules used are
$S \to aB$,
$B \to aBB$
$B \to bS$
$S \to aB$
$B \to b$
$B \to b$
} Seq. of P.T

Parse tree ~~left most~~

Rules used are

$S \to aB$, $B \to aBB$, $B \to b$, $B \to S$, $S \to aB$, $B \to b$



Both Parse trees are having same Yield but expands differently so Grammar is ambiguous

Q. 5) A)

Design PDA = $L = \{ a^n b^m c^n \mid m, n \geq 1 \}$

acceptance by final state.

Push an A for every a, skip over the b's without looking the stack (at least one b is required)

$$P = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b, c\}, \{A, Z_0\}, \delta, q_0, Z_0, \{q_4\})$$

$L = \{ abc, aabbc, aab, cc, aaabbbccc \}$

Transition function δ — 5 marks

1) $\delta(q_0, a, Z_0) = \{(q_0, AZ_0)\}$

2) $\delta(q_0, a, A) = \{(q_0, AA)\}$

3) $\delta(q_0, b, A) = \{(q_1, A)\}$ (first b requires only one a already read)

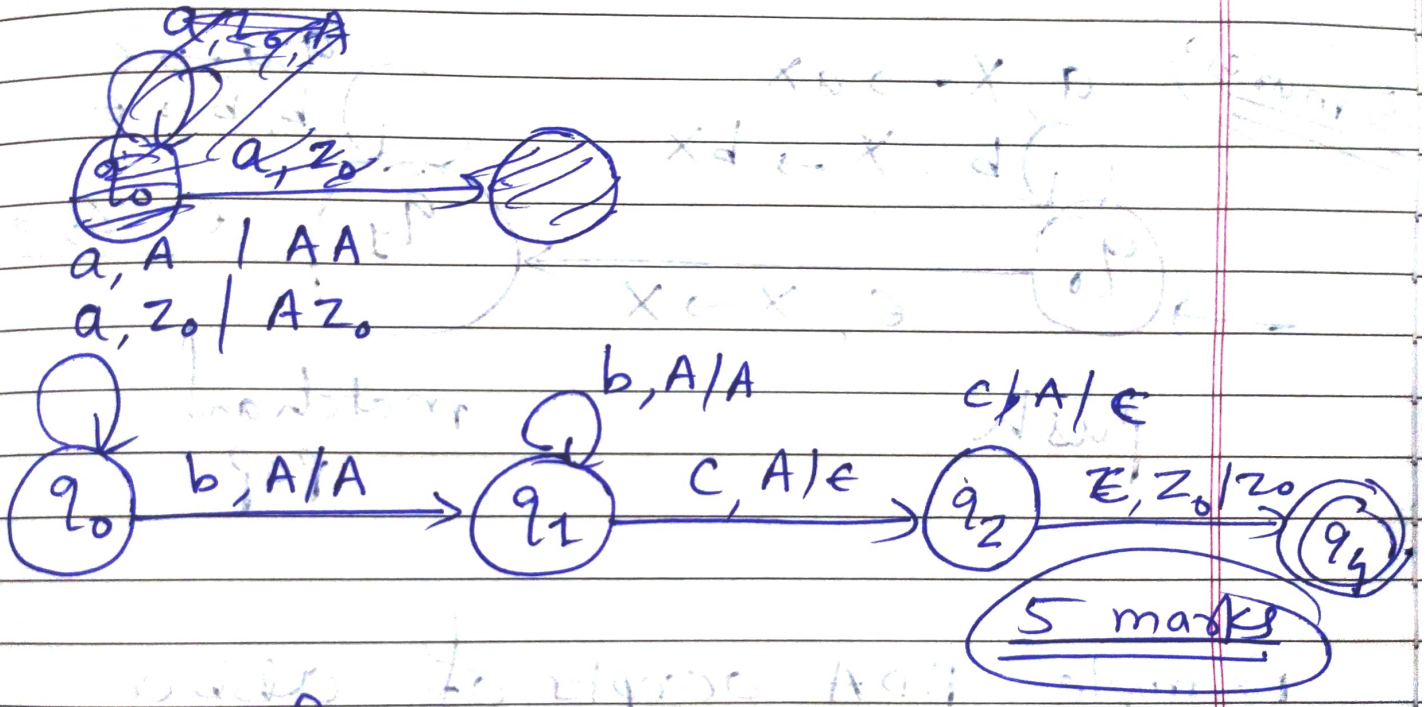
4) $\delta(q_1, b, A) = \{(q_1, A)\}$

Date / Page No

5) $\delta(q_1, c, A) = \{(q_2, \epsilon)\}$ first c requires at least one b already read

6) $\delta(q_2, c, A) = \{(q_2, \epsilon)\}$

7) $\delta(q_2, \epsilon, Z_0) = \{(q_4, Z_0)\}$



5 marks

Acceptance of string

$(q_0, aabcc, Z_0)$

$= (q_0, abcc, AZ_0)$

$= (q_0, bcc, AAZ_0)$

$= (q_1, cc, AAZ_0)$

$= (q_2, c, AZ_0)$

$= (q_2, \epsilon, Z_0)$ by rule 7

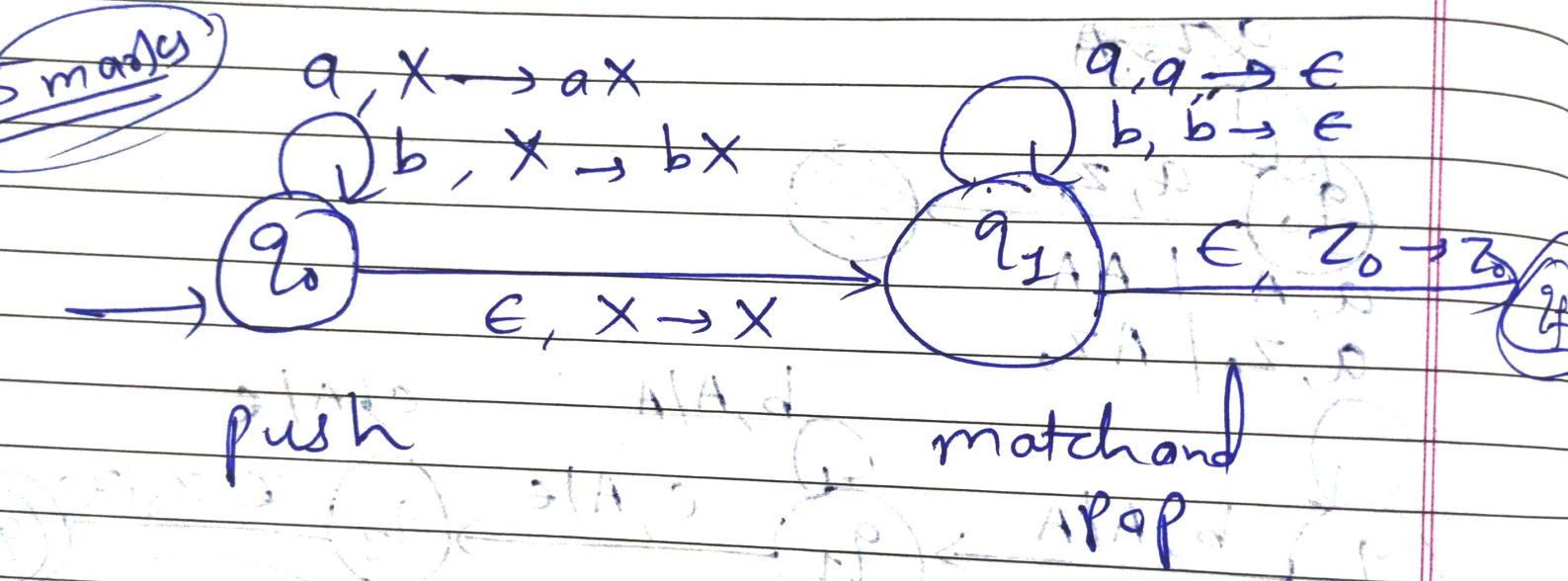
PDA accepts aabcc string

Q.5 B) $L = \{ w w^R \mid w \in \{a,b\}^* \}$

$L = \{ \epsilon, aa, bb, abba, baab, aabbaa \cdots \}$

language consists of even length palindr

We construct non-deterministic PDA beca
PDA must guess the midpoint



Example PDA accepts of abba

$w = ab \quad w^R = ba$

State	Remaining input	Stack top
q_0	abba	Z_0
q_0	bba	aZ_0
q_0	ba	baZ_0
q_1	ba	aZ_0
q_1	a	
q_1	ϵ	Z_0
q_f	ϵ	Z_0

5 marks

accepted

Q. 6) A)

5 marks

Design TM that accepts $L = \{ 0^n 1^n 2^n \mid n \geq 1 \}$

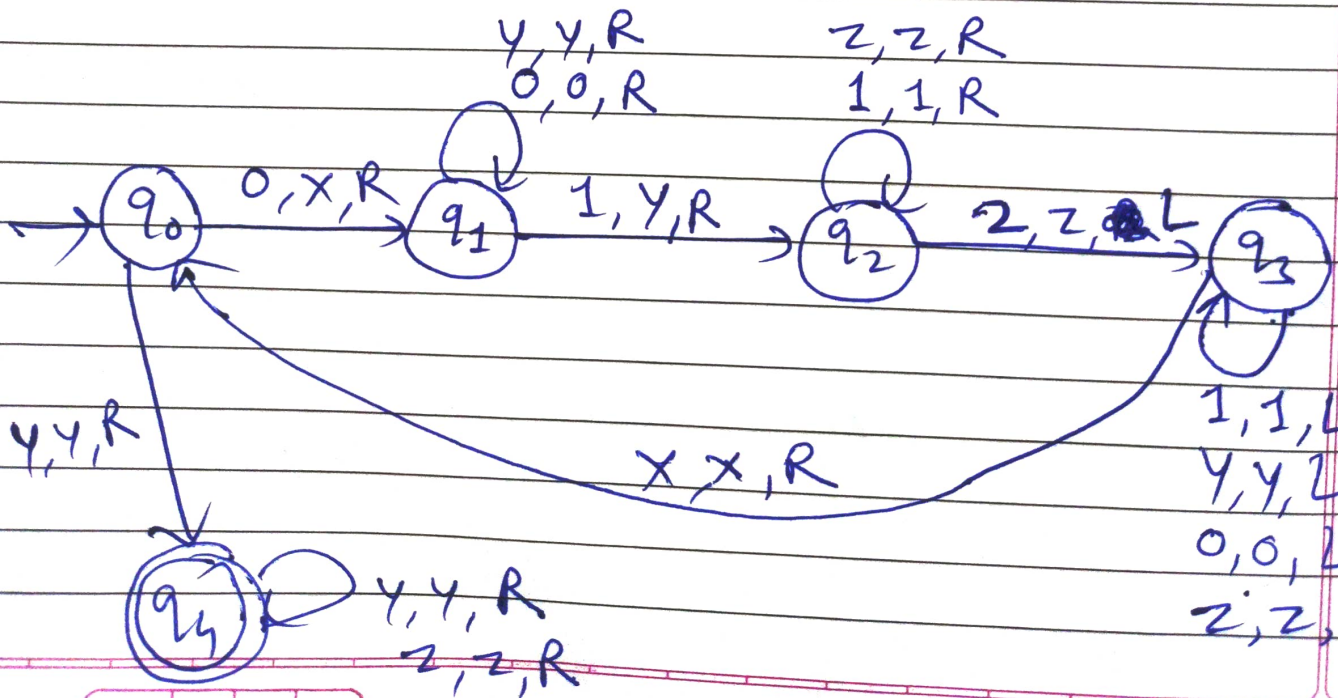
Give Transition diagram. and show seq of configurations of i/p : 012

$L = \{ 0^n 1^n 2^n \mid n \geq 1 \}$

$L = \{ 012, 001122, 000111222 \cdots \}$

i/p tape

B	0	0	1	1	2	2	B	B	B	B



Sequence for i/p $\overset{XYZ}{012}$

Sr. No	State	i/p	Next state	Replace with	move L/R
1	q_0	0	q_1	X	R
2	q_1	1	q_2	Y	R
3	q_2	2	q_1	Z	L
4	q_1	Y	q_0	Y	L
5	q_0	X	q_1	X	R
6	q_1	Y	q_4^*	Y	R
7	q_4^*	Z	q_4^*	Z	R

q_4 = final state

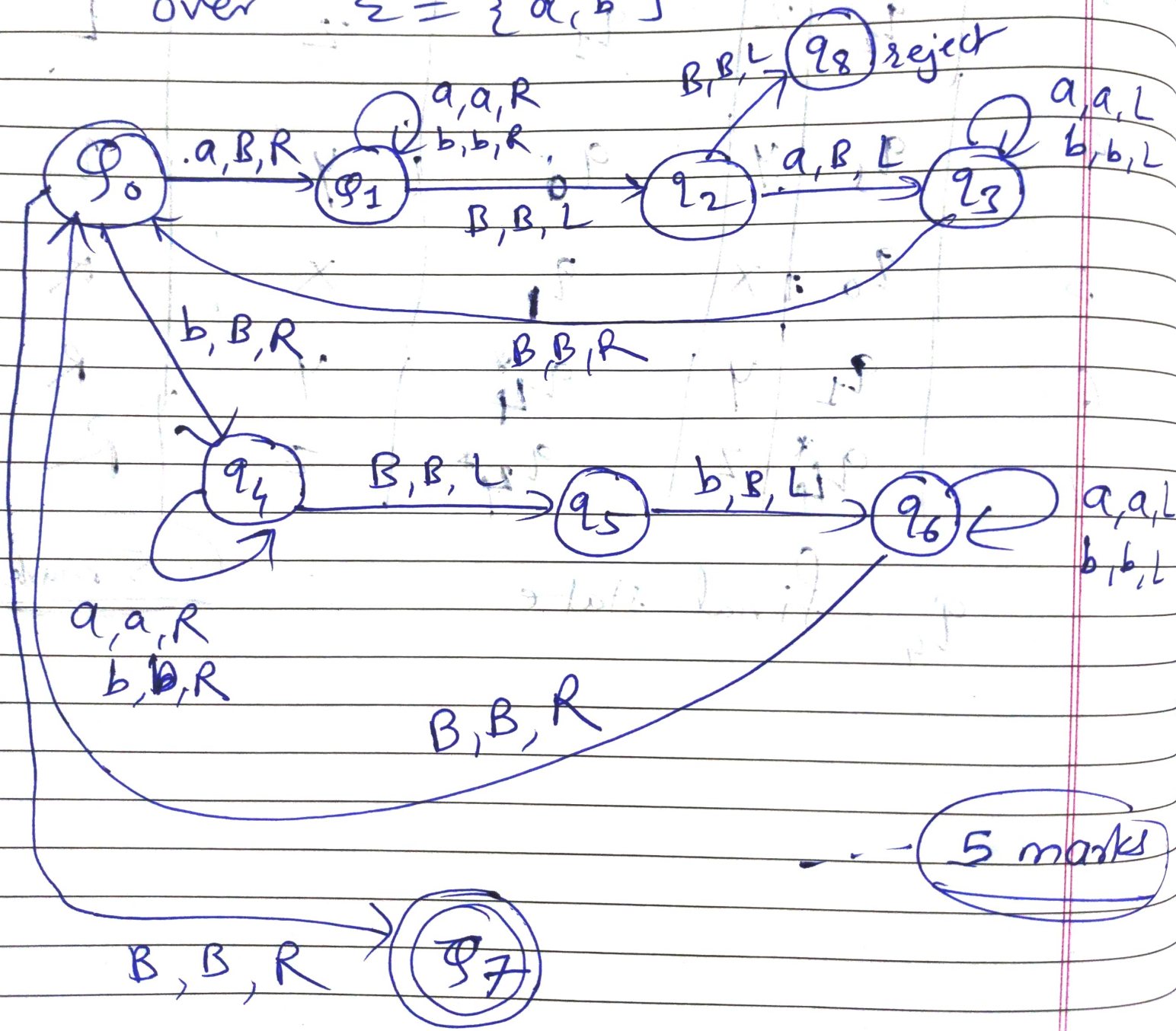
5 marks

Q. 6B)

TM for all set of palindromes

Even palindrome and odd palindrome

over $\Sigma = \{a, b\}$



5 marks

010 input sequence

same for aba

We constructed it for {a, b} alphabet but it remains same for both {0,1} and {a, b}

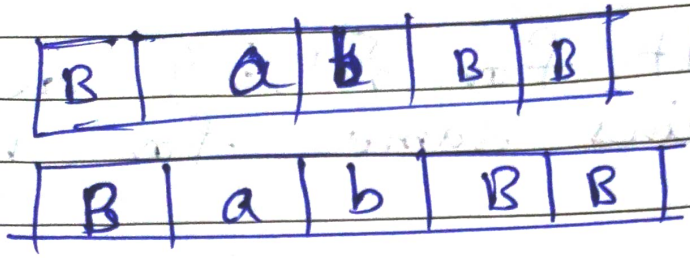
State:

B	a	b	B	B	B

Sr. No	State	i/p	Next state	Replace with	Move L/R
1	q_0	a	q_1	a	R
2	q_1	b	q_1	b	R
3	q_1	B	q_2	B	L
4	q_1	a	q_3	a	L
5	q_3	b	q_3	b	L
6	q_3	a	q_3	a	L
7	q_3	B	q_0	B	R
8	q_0	B	q_7 *	B	R

3 marks

Similarly if input string is same as ab for alphabet {a, b}



State	i/p	Replace with	Next state	Move
q_0	a	a	q_1	R
q_1	b	b	q_1	R
q_1	B	B	q_2	R
q_2	B	B	q_8 (reject)	R

-- 2 marks