Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE
1		
	a)	$S \rightarrow (S)S \mid \epsilon$
	b)	$L = \{0^n 1^m / n >= 0$ and $m = n$ or $m = 2n\}$ is a CFL but not a DCFL.
	c)	Given $S \rightarrow aSa / aa / bb / aAB$ $A \rightarrow AbA$ $B \rightarrow b$ A derives no string over terminals. Hence, A is useless. Removing A, we get $S \rightarrow aSa / aa / bb$ $B \rightarrow b$ Since B is not reachable from S. Therefore, B is useless. Removing B, we get $S \rightarrow aSa / aa / bb$.
	d)	Given $S \rightarrow aSa / bSb / a / b / \epsilon$ Removing $S \rightarrow \epsilon$, we get $S \rightarrow aSa / bSb / a / b / aa / bb$

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

e) 7-tuple representation of PDA is
$M = (Q, \Sigma, \tau, \delta, q_0, z_0, F)$
where Q is finite set of states
Σ is — " — ~~symbol~~ input symbols
τ is — " — stack symbols
δ maps $Q \times \Sigma \cup \{\epsilon\} \times \tau$ to $Q \times \tau^*$
q_0 is initial state
z_0 is bottom of stack symbol
F is final state / set of final states.

f) 7-tuple representation of TM is
$M = (Q, \Sigma, \tau, \delta, q_0, B, F)$
where Q is finite set of states
Σ is — " — input symbols
τ is — " — tape symbols
δ maps $Q \times \tau$ to $Q \times \tau \times \{L, R\}$
q_0 is initial state
B is blank symbol of tape.
F is finite set of final states (which is also a halt state)

g) Given an arbitrary algorithm with an input, determining whether the ~~gol~~ given algorithm will halt or not is a decidable problem which is Halting Problem. In other words, will given TM halt on input or in infinite loop.

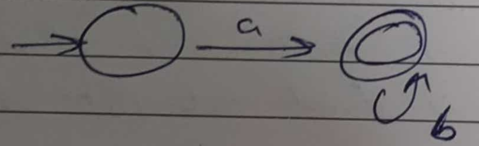
Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks | Question No. | START WRITING HERE

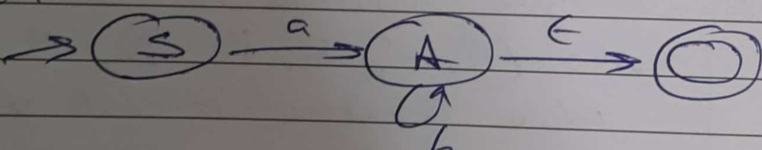
h) i) NPDA is as powerful as DPDA - FALSE
ii) NTM is as powerful as DTM - TRUE

Q2

(a) Right Linear grammar for ab^* can be obtained from its DFA.



Consider

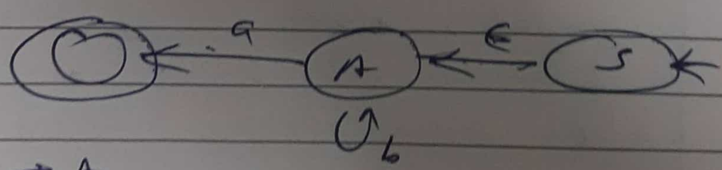


Right Linear Grammar is

$S \rightarrow aA$
$A \rightarrow bA / \epsilon$

$\boxed{OR}$ $S \rightarrow aA / a$
$A \rightarrow bA / b$

For left linear grammar, we reverse ~~the~~ direction of all edges & travel from final to start as



$S \rightarrow A$
$A \rightarrow Ab / a$ $\boxed{OR}$ $S \rightarrow Sb / a$

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

2b Given grammar: $S \to 0B \mid 1AA$

$A \to 0 \mid 0S \mid 1AA$

$B \to 1 \mid 1S \mid 0BB$

Derivation for "00110101"

LMD : $S \to 0B$

$\Rightarrow 00BB$

$\Rightarrow 001B$

$\Rightarrow 0011S$

$\Rightarrow 00110B$

$\Rightarrow 001101S$

$\Rightarrow 0011010B$

$\Rightarrow 00110101$

RMD

$S \to 0B$

$\Rightarrow 00BB$

$\Rightarrow 00B1S$

$\Rightarrow 00B10B$

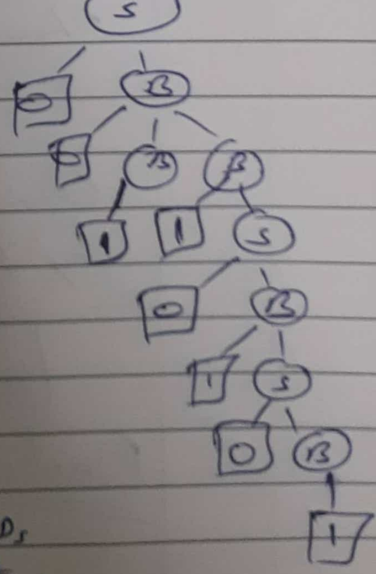
$\Rightarrow 00B101S$

$\Rightarrow 00B1010B$

$\Rightarrow 00B10101$

$\Rightarrow 00110101$

Parse Tree / Derivation Tree



PS: Grammar is Ambiguous and Hence multiple LMDs, RMDs & Parse Trees could exist.

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

2c

Given CFG

$S \to AA \mid 0$

$A \to AS \mid 1$

Consider $A \to AS \mid 1$

Apply Lemma 2 [$\because A \to A\alpha \mid \beta$ becomes
$A \to \beta \mid \beta A'$
$A' \to \alpha \mid \alpha A'$]

Therefore,

$\boxed{A \to 1 \mid 1A'}$

$A' \to S \mid SA'$

Consider $S \to AA \mid 0$

Apply Lemma 1 [$\because A \to B\alpha$ & $B \to \beta$
becomes $A \to \beta\alpha$]

Therefore,

$\boxed{S \to 1A \mid 1A'A \mid 0}$

Consider $\boxed{A' \to S \mid SA'}$

by Lemma 1, we get

$\boxed{A' \to 1A \mid 1A'A \mid 0 \mid 1AA' \mid 1A'AA' \mid 0A'}$

Final GNF

$S \to 1A \mid 1A'A \mid 0$

$A \to 1 \mid 1A'$

$A' \to 1A \mid 1A'A \mid 0 \mid 1AA' \mid 1A'AA' \mid 0A'$

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE
	3 a	Detect ambiguity in $S \to SaS / SbS / c$

Consider string "cacbc"

LMD1

$S \to SaS$
$\Rightarrow caS$
$\Rightarrow caSbS$
$\Rightarrow cacbS$
$\Rightarrow cacbc$

LMD2

$S \to SbS$
$\Rightarrow SaSbS$
$\Rightarrow caSbS$
$\Rightarrow cacbS$
$\Rightarrow cacbc$

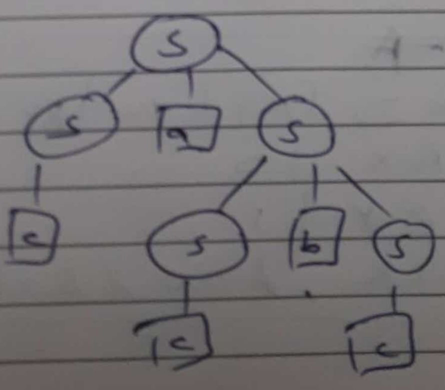
RMD1

$S \to SaS$
$\Rightarrow SaSbS$
$\Rightarrow SaSbc$
$\Rightarrow Sacbc$
$\Rightarrow cacbc$

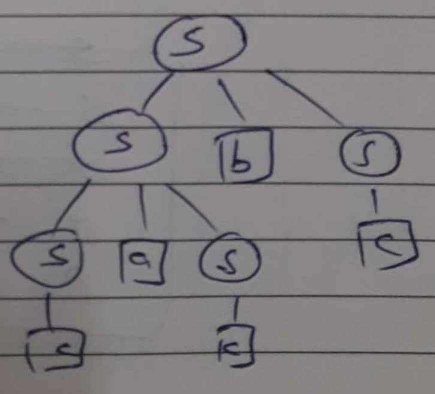
RMD2

$S \to SbS$
$\Rightarrow Sbc$
$\Rightarrow SaSbc$
$\Rightarrow Sacbc$
$\Rightarrow cacbc$

ParseTree1 / DT1



ParseTree2 / DT2



Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE
	3b	

CFG to CNF

Consider grammar

$S \to XYX$
$X \to 0X \mid \epsilon$
$Y \to 1Y \mid 1$

Removing $X \to \epsilon$, we get.

$S \to XYX \mid XY \mid YX \mid Y$
$X \to 0X \mid 0$
$Y \to 1Y \mid 1$

Removal of Unit Production ($S \to Y$), we get

$S \to XYX \mid XY \mid YX \mid 1Y \mid 1$
$X \to 0X \mid 0$
$Y \to 1Y \mid 1$

Final CNF

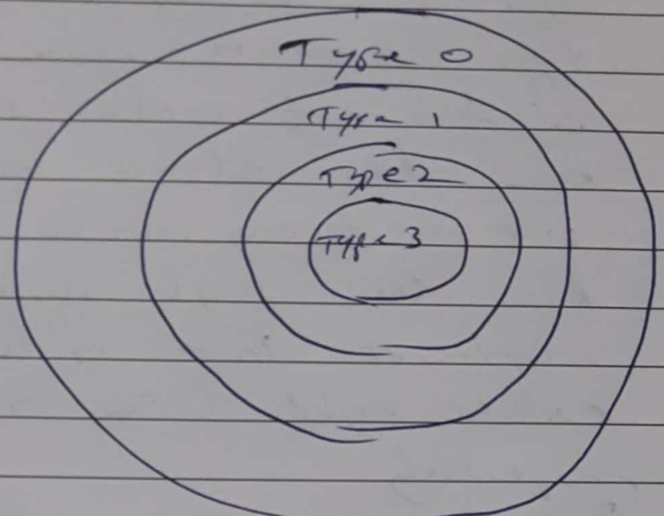
~~$S \to XA$~~

$$\boxed{\begin{array}{l} S \to XD \mid XY \mid YX \mid BY \mid 1 \\ X \to AX \mid 0 \\ Y \to BY \mid 1 \\ D \to YX,\ A \to 0\ \&\ B \to 1 \end{array}}$$

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

3 c Chomsky ~~theory~~ Hierarchy :



Chomsky has classified grammers and arranged them in a hierarchical order, ~~...~~ calling them Type 0 to Type 3. Such that every grammer of higher type (say Type 2) is always belongs to grammer of lower type (ie Type 0 + 1). These grammar classes are named as.

Type 3 — Regular grammar
Type 2 — Context Free Grammar
Type 1 — Context Sensitive Grammar
Type 0 — Unrestricted Grammar.

These grammars when represented as $G = (V, T, P, S)$ follow a set of production rules. These rules are such that, every rule of lower type is applicable to higher type..

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

The rules are.

For Type 3 : Every production is of type $A \rightarrow \alpha$ where α does not contain start symbol and has atmost one Variable(V) with 0 to any number of Terminals(T^*). Also, the single variable must be rightmost in α or leftmost in α, called Right Linear & Left Linear respectively.

Example:

OR

$S \rightarrow aA \mid a$ $\qquad$ $S \rightarrow Aa$

$A \rightarrow bA \mid b$ $\qquad$ ~~$A \rightarrow bA \mid \epsilon$~~

$A \rightarrow Ab \mid \epsilon$

For Type 2 : Every production is of type $A \rightarrow \alpha$ where α does not contain start symbol but can contain any number of variables & terminals in any order (ie $(V+T)^*$)

Hence $A \rightarrow (V+T)^*$ where A is a single variable only.

Example

$S \rightarrow aSa \mid bSb \mid aa \mid bb \mid a \mid b$

OR

$S \rightarrow (S)S \mid \epsilon$

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

For Type 1: Every production is of type
$\alpha \to \beta$ where
α and β both can be $(V+T)^*$
but $|\beta| \geq |\alpha|$ and β should not contain the start symbol.

Ex: $S \to aA \mid bB \mid abA$
$aA \to ac$
$bA \to bd$
$B \to e$

For Type 0: Every production is of type
$\alpha \to \beta$ with only restriction that start symbol should not be derived from other variables.
Hence β should not contain S.
Rest all productions are permitted in Type 0.

Ex
$S \to aA \mid bAB$
$aA \to c$
$bA \to \epsilon$
$B \to baA \mid aA$

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

4a

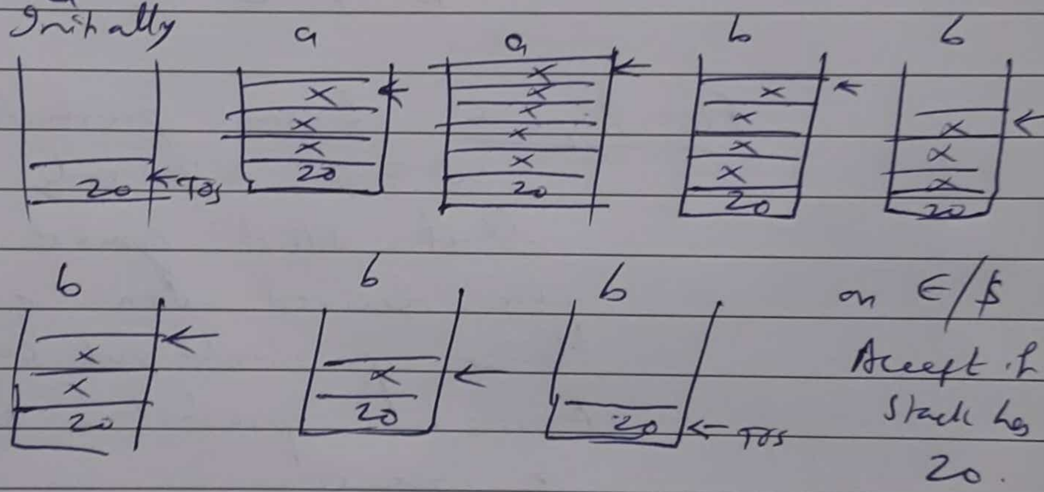
PDA for $L = \{a^n b^{2n+1} \mid n > 0\}$

Approach :

PUSH 3 X for first 'a' as input
PUSH 2 X for all remaining a's
POP an X for every 'b' in i/p.

Starting with Z_0 in stack, if stack has Z_0 when string ends, then string belongs to language.

Ex : aabbbbb



7-tuple representation of the PDA

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$\Sigma = \{a, b\}$, $\Gamma = \{X, Z_0\}$

$Q = \{q_0, q_1, q_2\}$ $F = \{q_2\}$

q_0 : PUSH XXX on 'a' if TOS is Z_0
PUSH XX on 'a' if TOS is X
POP X and change state on 'b'

q_1 : POP X for every b

q_2 : Accept.

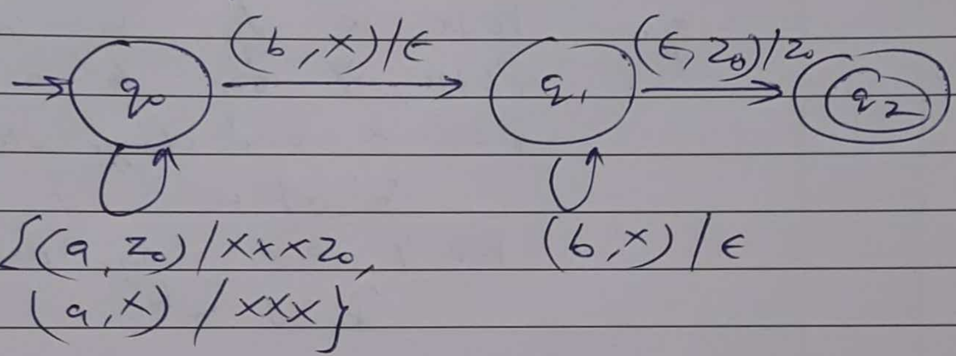
Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

δ maps $Q \times \Sigma \cup \{\epsilon\} \times \Gamma \to Q \times \Gamma^*$

$\delta(q_0, a, Z_0) = (q_0, XXXZ_0)$
$\delta(q_0, a, X) = (q_0, XXX)$
$\delta(q_0, b, X) = \cancel{(q_1, \epsilon)}\ (q_1, \epsilon)$
$\delta(q_1, b, X) = (q_1, \epsilon)$
$\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$

Graphical representation:



4b) NPDA for $L = \{ww^R / w$ belongs to $(a+b)^*\}$
Assume w^R is reverse of w. Hence valid string is "abaaaaba".
Approach: PUSH w & POP on w^R.

"abaa aaba"
PUSH   POP & verify.

For every 'a' of w PUSH X
— " — 'b' of w PUSH Y
— " — 'a' of w^R POP and verify that TOS is X
— " — 'b' of w^R — " — Y.
on ε, TOS must be Z_0 (Bottom of Stack).

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

Transition from w to w^R is non-deterministic.
It is some 'a' with TOS X or
some 'b' with TOS Y.

7-tuple representation of the PDA.

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
$\Sigma = \{a, b\}$, $\Gamma = \{X, Y, z_0\}$
$Q = \{q_0, q_1, q_2\}$ $F = \{q_2\}$

q_0: PUSH X for 'a' in w
PUSH Y for 'b' in w
POP X and change state on first 'a' of w^R
POP Y and change state on first 'b' of w^R

q_1: POP X on every 'a' in w^R
POP Y — " — 'b' — " —
Till TOS is z_0 on end of string.

q_2: Final state.

~~δ~~ δ maps $Q \times \Sigma \cup \{\epsilon\} \times \Gamma$ to a set of $Q \times \Gamma^*$

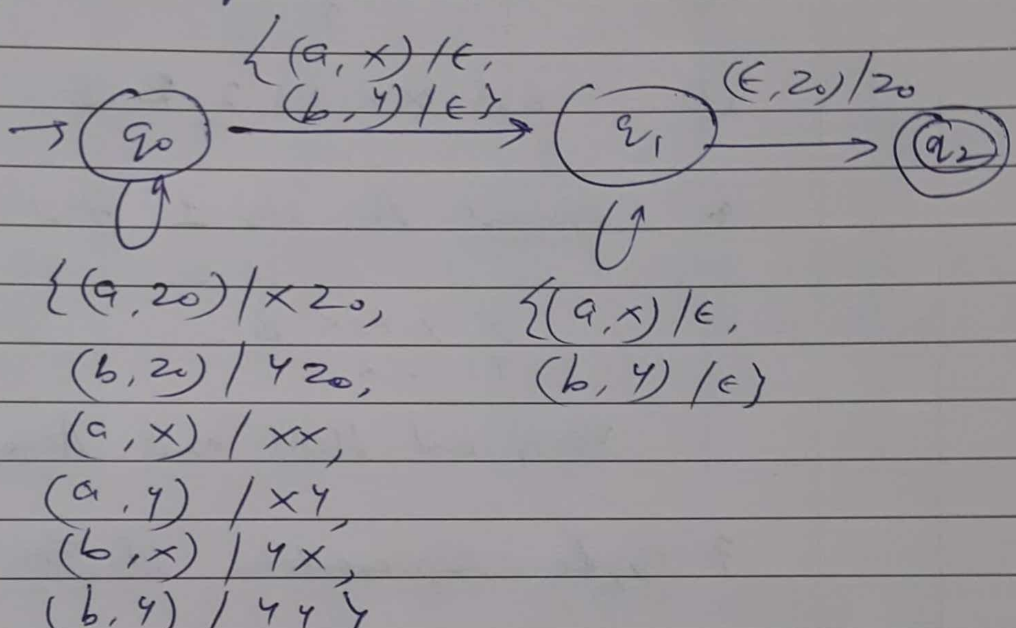
$\delta(q_0, a, z_0) = \{(q_0, Xz_0)\}$
$\delta(q_0, b, z_0) = \{(q_0, Yz_0)\}$
$\delta(q_0, a, Y) = \{(q_0, XY)\}$
$\delta(q_0, b, X) = \{(q_0, YX)\}$
$\delta(q_0, a, X) = \{(q_0, XX), (q_1, \epsilon)\}$
$\delta(q_0, b, Y) = \{(q_0, YY), (q_1, \epsilon)\}$

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

$$\delta(q_1, a, X) = (q_1, \epsilon)$$
$$\delta(q_1, b, Y) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$$

Graphical representation.



5a TM for language $\{0^n 1^{2n} 2^n \mid n >= 0\}$

Approach:

Step1: Make leftmost '0' a Blank (B)

Step2 = Replace 2 left '1's by X

Step3 = Make rightmost '2' a Blank (B)

Step4: Repeat from step 1 till tape has no more '0's. On X/B, go to step 5

Step5: Verify that tape has all X or only Blank

Step6: Accept & Halt State.

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

Consider string 0011111 22
↑

① . B 0 1 1 1 1 2 2 B
↑

② B 0 X X 1 1 2 2 B
↑

③ B 0 X X 1 1 2 B B
↑

4 Repeat from step 1 till all X

X X X X B
↑ →

verify and Halt and Accept if all X

7-tuple representation of TM.

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

$\Sigma = \{0, 1, 2\}$, $\Gamma = \{0, 1, 2, X, B\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$

~~F = {q8}~~ $F = \{q_7\}$

q_0: Make leftmost '0' Blank.

q_1: Move right for all '0's till '1' & Make one 1 'X'

q_2: Make 2^{nd} 1 'X'

q_3: Move right till last symbol (rightmost)

q_4: Make rightmost '2' Blank

q_5: Move left for all symbols till left Blank

q_6: Verify all X & Halt in q_7 to Accept

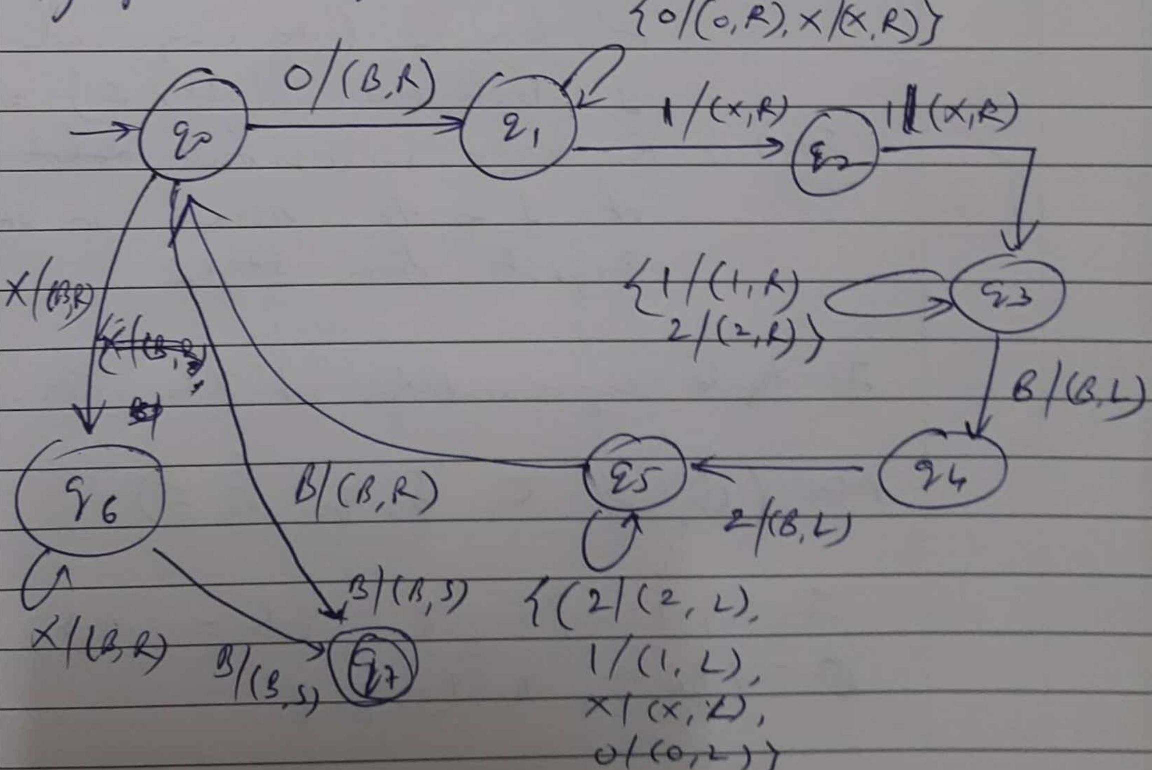
Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

δ maps $Q \times \Gamma$ to $Q \times \Gamma \times \{L, R, S\}$

$Q \backslash \Gamma$	0	1	2	X	B
q_0	(q_1, B, R)			(q_6, B, R)	(q_7, B, S)
q_1	$(q_1, 0, R)$	(q_2, X, R)		(q_1, X, R)	
q_2		(q_3, X, R)			
q_3		$(q_3, 1, R)$	$(q_3, 2, R)$		(q_4, B, L)
q_4			(q_5, B, L)		
q_5	$(q_5, 0, L)$	$(q_5, 1, L)$	$(q_5, 2, L)$	(q_5, X, L)	(q_0, B, R)
q_6				(q_6, B, R)	(q_7, B, S)
q_7	—	—	—	—	—
~~q_8~~					

graphical representation



Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

5b TM to compute $|m-n|$
Assume both m & n are in Unary (Represented using symbol 1)
ie $\Sigma = \{1, -\}$
Hence, $|4-3|$ can be represented as
1111 - 11 B
↑

Approach:
Step1: Till there is 1 on leftmost ~~cell~~ position of string, make it Blank. On '-', make it Blank & ~~a~~ Halt.
Step2: For every 1 becoming blank on left, move right to make rightmost '1' a blank. If rightmost is not a '1', ~~then~~ but a '-', then replace '-' by '1' (to compensate for the '1' deducted in step 1) and Halt.
Step3: For every successfull ~~deletion~~ replacement of 1 ~~as~~ to Blank in step 2, repeat from step 1.

7-tuple representation of the TM.

$M = (Q, \Sigma, \tau, \delta, q_0, B, F)$

$\Sigma = \{1, -\}$ $\tau = \{1, -, B\}$, $F = \phi$
$Q = \{q_0, q_1, q_2, q_3, q_4\}$

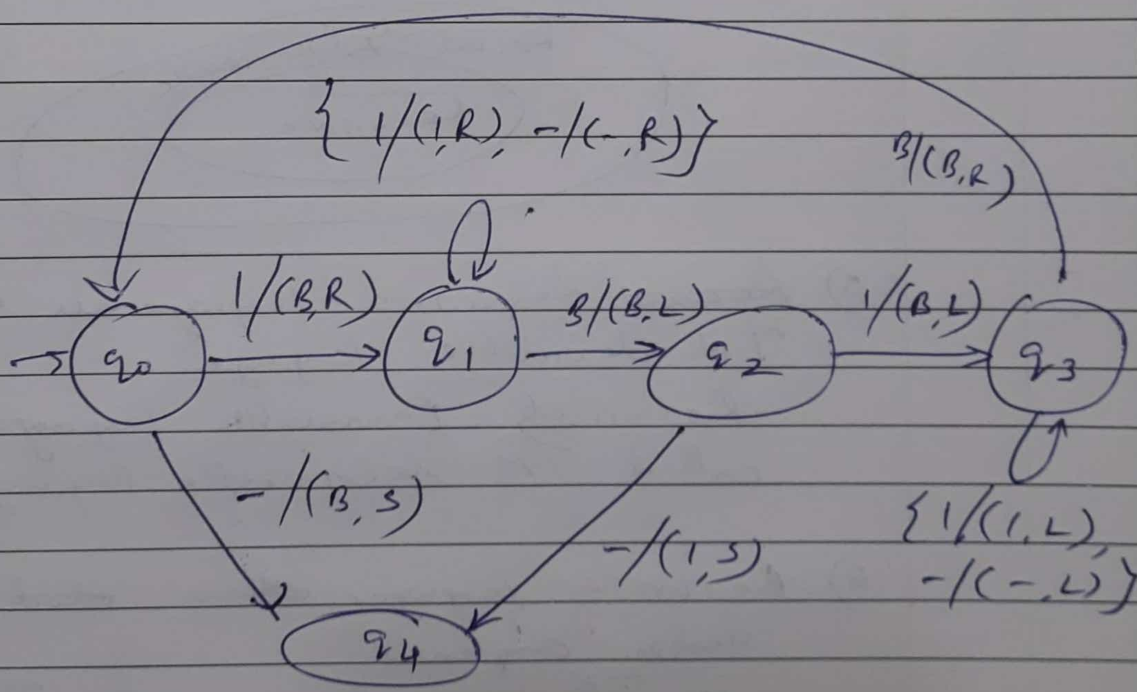
Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

q_0: Make leftmost '1' blank
q_1: Move right
q_2: Make rightmost '1' blank
q_3: Move left
q_4: Halt state

$$\delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{L, R, S\}$$

Q \ Σ	1	−	B
q_0	(q_1, B, R)	(q_4, B, S)	—
q_1	$(q_1, 1, R)$	$(q_1, -, R)$	(q_2, B, L)
q_2	(q_3, B, L)	$(q_4, 1, S)$	—
q_3	$(q_3, 1, L)$	$(q_3, -, L)$	(q_0, B, R)
q_4	—	—	—



Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

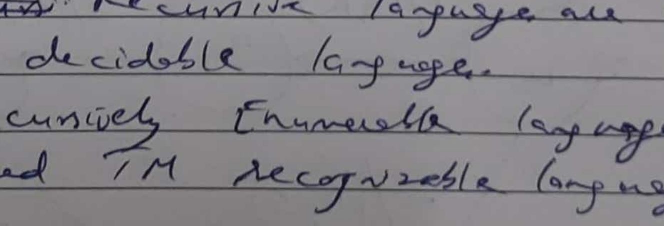
Space for Marks	Question No.	START WRITING HERE

6a Compare Recursive & Recursively Enumerable languages.

1) A language L for which there exist a Turing Machine that halts on every input (valid / invalid) over Σ is a Recursive language

A language L for which there exists a Turing Machine that halt on every valid input over Σ (ie strings over Σ that belong to L) is a Recursively Enumerable language.

2) ~~R~~ All Recursive languages are Recursively Enumerable languages but not vice-versa



3) ~~Recursi~~ Recursive language are also called TM decidable language.

Recursively Enumerable languages are also called TM recognizable language.

4) Recursive languages ~~a~~ are ~~also~~ closed under complement.

ie If L_1 is Recursive then $\overline{L_1}$ is Recursive.

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

Recursively Enumerable languages are not closed under complement

ie If L_2 is Recursively Enumerable language, then $\overline{L_2}$ may / may not be Recursively Enumerable language

5) If there exits a language L_3 which is Recursively Enumerable but not Recursive, then, by definition, $\overline{L_3}$ is not Recursively Enumerable language.

6) Recursive languages are also closed under ~~set diff~~ : set difference but Recursively Enumerable languages are not closed under set difference.

ie If L_4 & L_5 are Recursive then $L_4 - L_5$ & $L_5 - L_4$ are Recursive

But

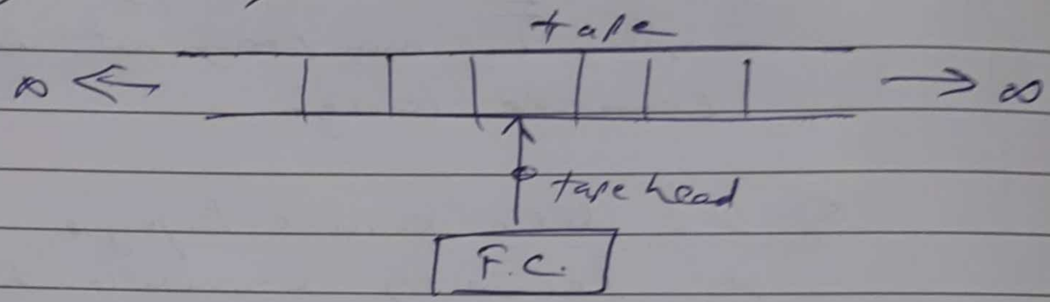
If L_6 & L_7 are Recursively Enumerable then

$L_6 - L_7$ may / may not be Recursively Enumerable.

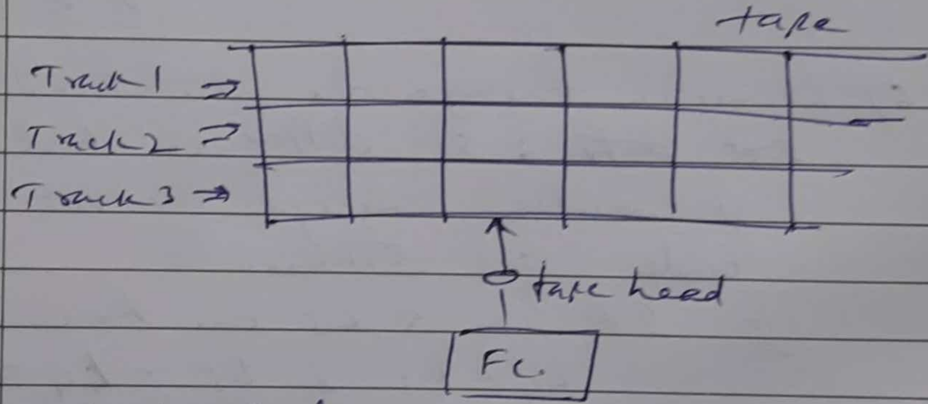
Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE
	6b	Variants of Turing Machine

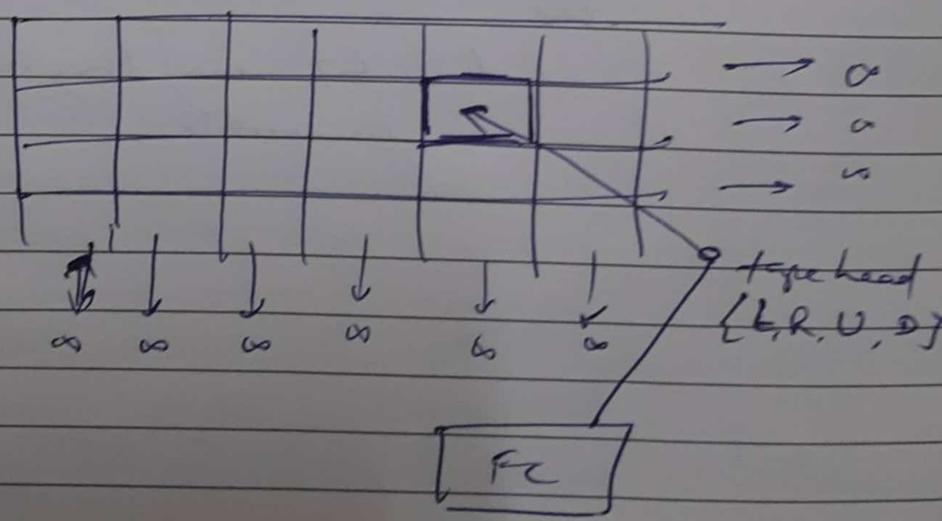
a) Two-way infinite model of TM:



b) Multi track TM



c) Multidimensional TM.

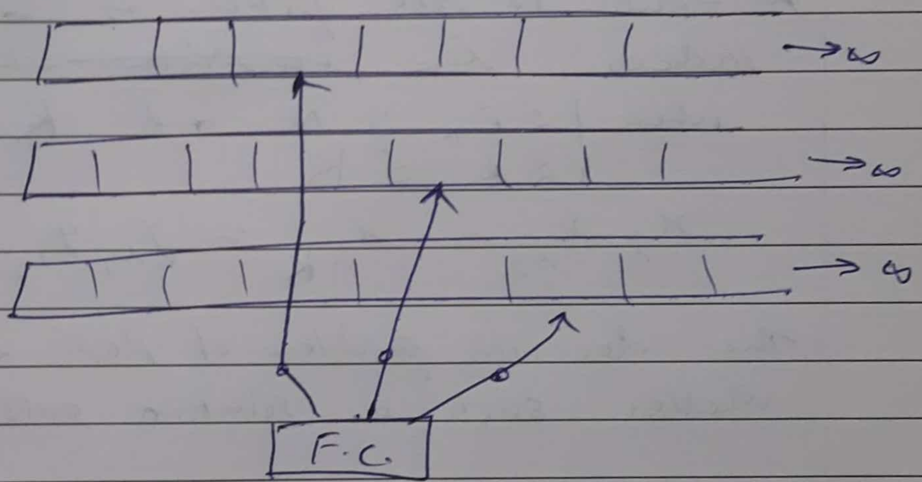


Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

d) Multi head TM.

e) Multi tape TM



f) Non Deterministic TM: The one in which δ maps $Q \times \Sigma$ to a possible set of $Q \times \Gamma \times \{L, R, S\}$.

As per Church's Hypothesis, all variants of TM are as powerful as Basic Model of TM.

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

~~6)~~

6 c) Definition of PCP:

Let Σ be any alphabet set of atleast 2 symbols. Let X and Y be two finite lists of same size (ie same number of strings over Σ) such as

$$X = [x_1, x_2, -- x_N]$$
$$Y = [y_1, y_2, -- y_N]$$

A solution to the problem is a sequence of indices i_K, ~~where 1 ≤ K ≤ where ≤ i_K~~ when $1 \leq i_K \leq N$ and $K \geq 1$ such that $1 \leq k \leq K$

$$x_{i_1} x_{i_2} \cdots x_{i_K} = y_{i_1} y_{i_2} \cdots y_{i_K}$$

The decision problem of PCP is to decide whether such a solution exists or not.

Given 2 lists

X_list = [b, a, ca, abc]
Y_list = [ca, ab, a, c]

Index sequence	X_list	Y_list
2	a	ab
2 1	ab	abca
2 1 3	abca	abcaa
2 1 3 2	abcaa	abcaaab
2 1 3 2 4	abcaaabc	abcaaabc

Total Marks of Question no.		Examiner	
		Moderator	
		Re-Assessor	

Space for Marks	Question No.	START WRITING HERE

Since $x_2 x_1 x_3 x_2 x_4 = y_2 y_1 y_3 y_2 y_4 =$

abcaaabc

we conclude that the given PCP has a solution is "2 1 3 2 4".