1

DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
ONLINE SESSIONAL EXAMINATION

B. Tech (CE) Sem : 6th
Subject : TAFL

Roll No : CE-107
Signature : ssPatel

Date : 23/03/21
Time : 9:00 am to 10:15 am
Total Pages : 10

Q-1

a) Deterministic Push Down Automata :

A Push Down Automata is Deterministic if ~~it Satisfies both given Condi~~, $M(Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$

1) for each $q \in Q$, every $a \in \Sigma \cup \{\Lambda\}$, and every $X \in \Gamma$, the set $\delta(q, a, X)$ has at most one element

2) for each $q \in Q$, each $a \in \Sigma$ & every $X \in \Gamma$, the two sets $\delta(q, a, X)$ and $\delta(q, \Lambda, X)$ cannot both be nonempty.

So, it is deterministic if it follows this two

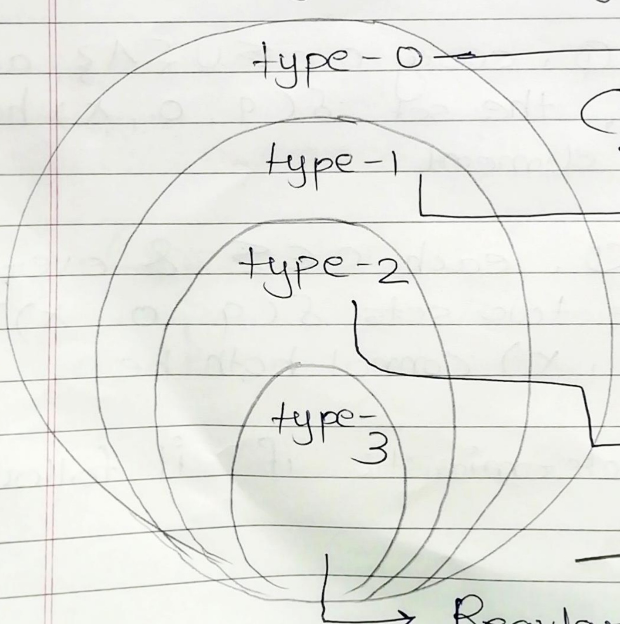
b) Pumping Lemma for CFLs:

⇒ if L is Context-free Language, Then there is integer n so that for every $u \in L$ with $|u| \geq n$, u can be written as $u = vwxyz$ for some strings v, w, x, y, z & Satisfies

1) $|wy| > 0$
2) $|wxy| \leq n$
3) $m \geq 0$, $vw^m x y^m z \in L$

⇒ So, it is technique to show that certain ~~Lanages~~ Languages are Context free or not.

c) Chomsky Hierarchy of Languages:



type-0 → Unristricted grammer (Recognized by Turing machine)

type-1 → Context sensitive grammer (Accepted by Linear Bound Automata)

type-2 → Context free grammer (Push Down Automata)

type-3 → Regular Grammer (Finite Automata)

3

d) Recursive Language:

A Language is Recursive if there exists a Turing machine that accepts every string of the Language & rejects every string that is not in the Language

★ Recursive Enumerable ~~Lanag~~ Language:

Language is Recursive if there exists a turing machine that accepts every string of the Language, and does not accept strings that are not in the Language.

e) Acceptance by Turing Machine:

for Turing machine $= (Q, \Sigma, T, q_0, \delta)$

Acceptance will be,

$(q_0, \Delta x) \vdash_T^* (ha, yqz)$

here, $q_0 \in Q$, $x \in \Sigma^*$

$y \in (T \cup \{\Delta\})^*$

$q \in (T \cup \{\Delta\})$

ha is haulting state.

$L(T) = \{ x \in \Sigma^* \mid x \text{ is accepted by } T \}$

$L(T)$ is Language accepted by Turing machine.

f) Turing's encoding fuction for input to Universal Turing machine

here $e(T)\,e(Z)$, input for Turing machine,

now suppose move is $\delta(P, A) = (q, a, R)$

$\cancel{e(m) = S(P)\, 1\, S(A)\, 1\, S(q)\, 1\, S(a)\, 1\, S(R)}$.

⇒ here,

$e(T) = S(q)\, 1\, e(m_1)\, 1\, e(m_2)\, 1 \ldots e(m_k)\, 1$

$m_1, m_2 \ldots m_k$ are distinct moves of T arranged in arbitary order

⇒ $e(Z) = 1S(Z_1)\, 1\, S(Z_2) \ldots 1S(Z_k)$

This $e(Z)$ is for one to many mapping.

here $m_1, m_2, \ldots m_k$ → moves for TM

$\cancel{e(Z) = 1\, S(Z_1)}$

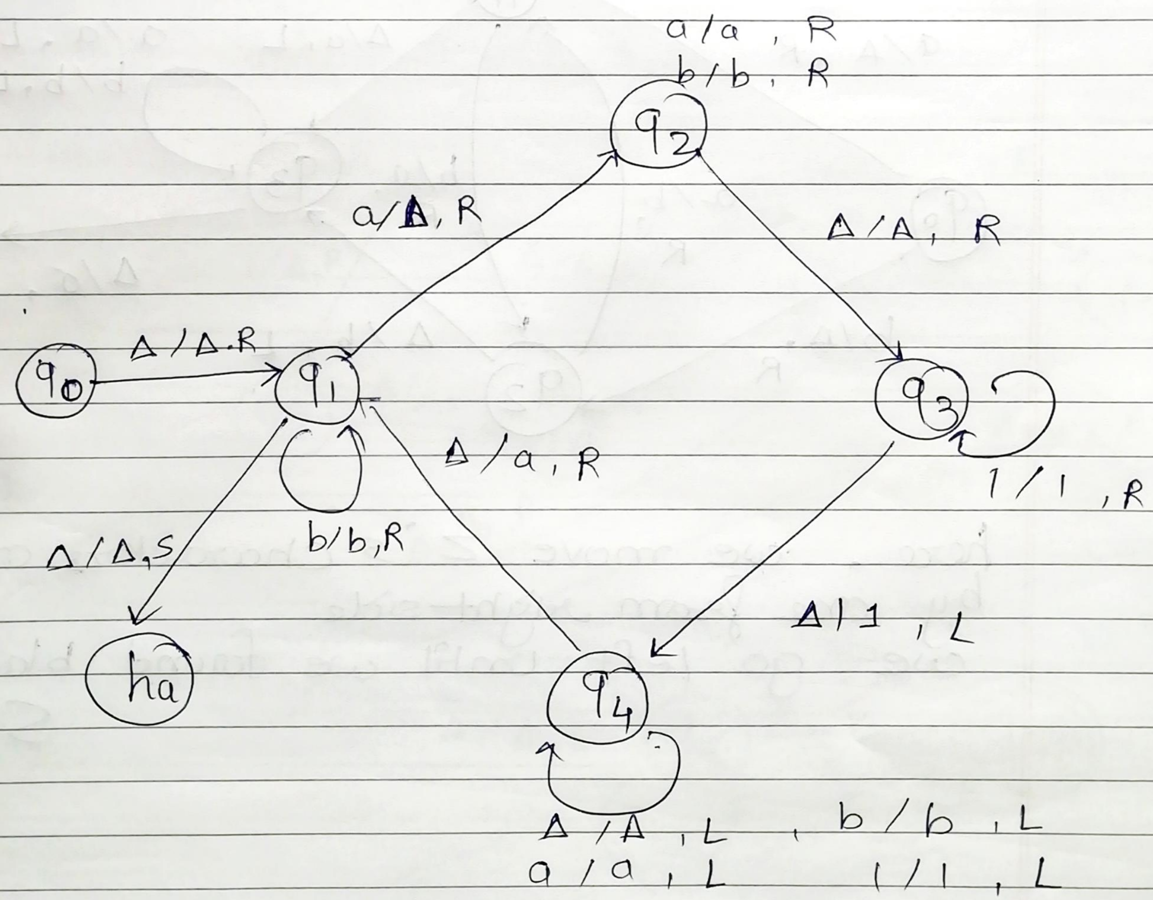
& $Z = Z_1 Z_2 Z_3 \ldots Z_k \quad Z_i \in S$

Q-2

a) TM that calculates number of a's in a given string.

for ~~a~~ first step, input string, $\underline{\Delta}\, x$

one O/P will be : $\Delta\, x\, \underline{\Delta}\, 1111 ---$

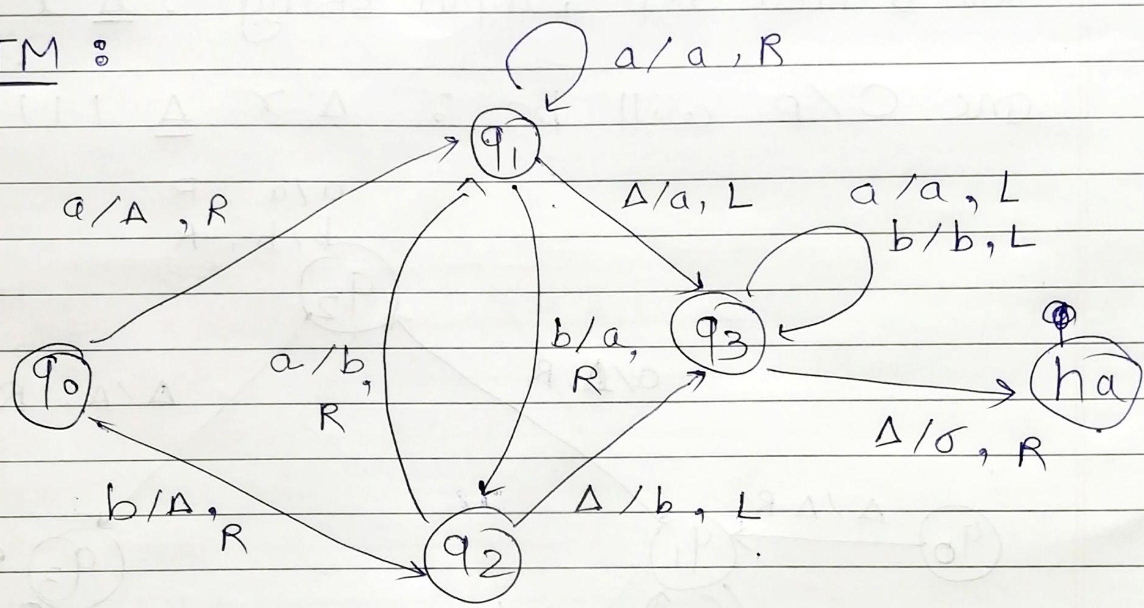


⑥

C) Insert Turing Machine, which changes ~~that~~ the tape contents from yz to yσz.

$y \in (\Sigma \cup \{\Delta\})^*$, $\sigma \in (\Sigma \cup \{\Delta\})$ and $z \in \Sigma^*$

TM :



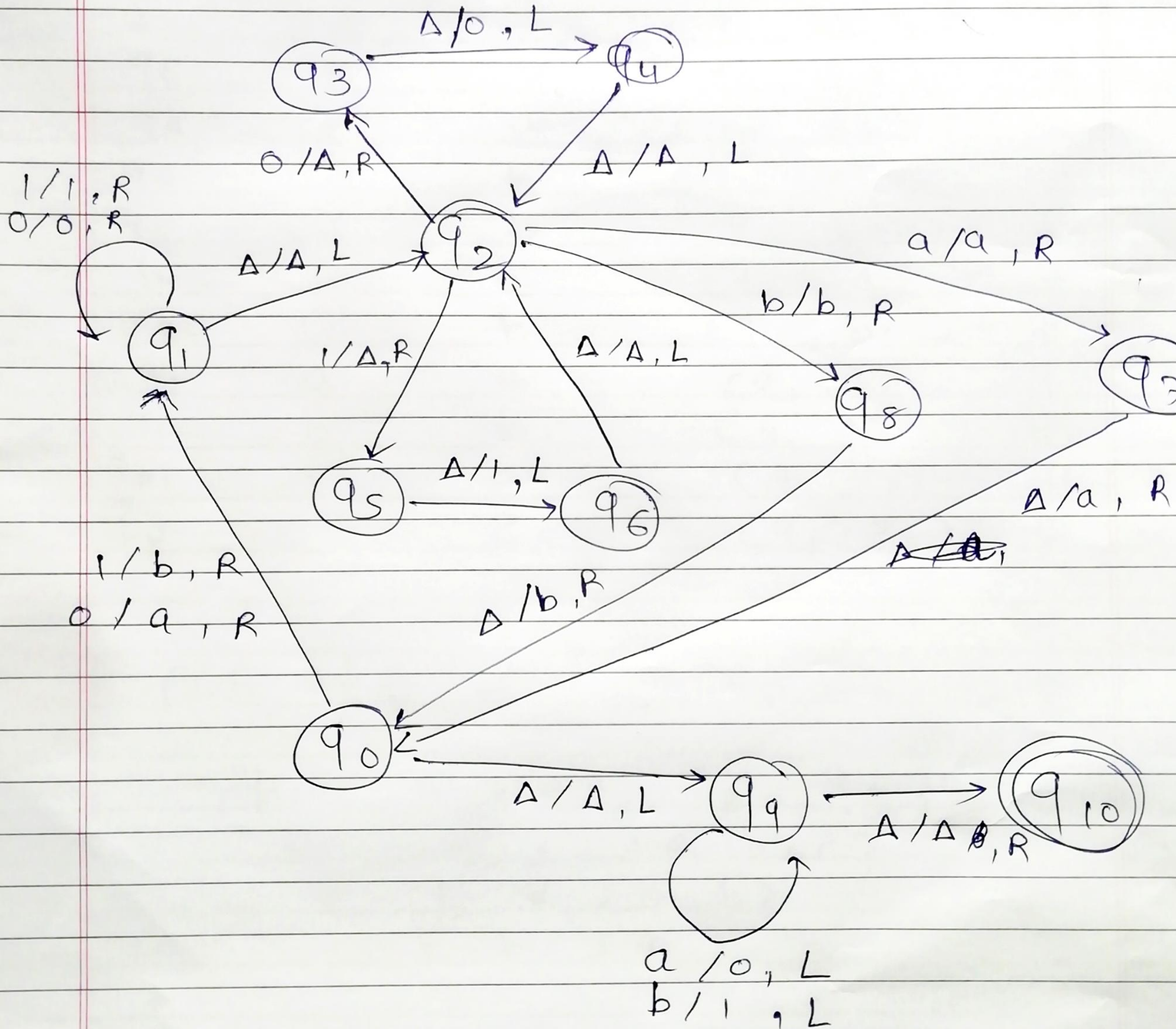
here, we move z's characters one by one from right side
we go left until we found blank spot.

Q - 2

b)

~~TM for~~

initial configuration (q, Δx)



Date
Page

Q-3

a> Prove,

There are ~~CF~~ CFLs L1 & L2 so that $L_1 \cap L2$ is not CFL.

here given that

L_1 & L_2 are CFLs

So, suppose that,

~~$L_1 = \{ a^i b^j c$~~

$$L_1 = \{ a^x, b^y, c^z \mid x = y \}$$

$$L_2 = \{ a^x, b^y, c^z \mid y = z \}$$

now for $L_1 \cap L2$.

suppose. $L_3 = L_1 \cap L_2$

$$L_3 = \{ a^x b^y c^z \mid x = y \} \cap \{ a^x b^y c^z \mid y = z \}$$

$$= \{ a^x b^y c^z \mid x = y \ \& \ y = z \}$$

here, here context free grammer for L_3 is not possible

as, in L_3 we have condition like $x = y$ & $y = z$.

So, here construction of PDA with comparing x to y & y to z at same time is not possible

& , ~~for~~ L_1 & L_2 we can make PDA & can write context free ~~gma~~ grammer

now here $L_3 = L_1 \cap L_2$

So, it proves that, if L_1 & L_2 are CFLs then $L_1 \cap L_2$ is not CFL

b) PDA for Language $S \longrightarrow aSa \mid bSb \mid c$

~~states for~~

here for $S \rightarrow aSa \mid bSb \mid c$

S	S	S	S
aSa	aSa	bSb	bSb
abSba	aaSaa	baSab	bbSbb
abcba	aaCaa	baCab	bbcbb

a	a	b	b
Z_0	Z_0	Z_0	Z_0

~~States~~

~~Sat~~

State	i/p	Tos	Move	Rule No.
q_0	a	Z_0	(q_0, aZ_0)	1
q_0	b	Z_0	(q_0, bZ_0)	2
q_0	a	a	(q_0, aa)	3
q_0	b	a	(q_0, ba)	4
q_0	b	b	(q_0, bb)	5
q_0	a	b	(q_0, ab)	6
q_0	c	a	(q_1, a)	7
q_0	c	b	(q_1, b)	8
q_1	a	a	$(q_1, \wedge)$	9
q_1	b	b	$(q_1, \wedge)$	10
q_1	$\wedge$	Z_0	(q_2, Z_0)	11
q_0	c	Z_0	(q_1, Z_0)	12

$$PDA = \{Q, \Sigma, \Gamma, \delta, q_0 Z_0, F\}$$

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

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

$\Gamma = \{a, b\}$

$F = \{q_2\}$

X———— end ————X