TARUSI MITTAL
1901CS65

CS-303

HOMEWORK #2

Que 1:- Which of the following languages are not context free.

(a) $L = \{a^n b^j : n = j^2\}$

$\Rightarrow$ We will prove the statements whether they are context free or not by Pumping Lemma.

To go by pumping lemma to prove that a language is context free then lets:

$s = uvxyz$ and $s \in L \Rightarrow |vy| > 0$ and $|vxy| \leq P$

$P =$ pumping length.

$\Rightarrow uv^ixy^iz \in L$, $i \geq 0$

Now, for the string $L = \{a^n b^j\}$, if we consider the pumping length to be 'l'

and the string ~~to~~ be $\Rightarrow a^{l^2} b^l$

Let $uvxyz = a^{l^2} b^l$, Also, $|v| = |x| = |z| = 0$

$\therefore uy = a^{l^2} b^l$

Taking $u = a^{l^2}$, $y = b^l$

So, now if L is CFL then $uv^ixy^iz \in L$ $i \geq 0$

Taking $i = 0$ $\rightarrow uv^ixy^iz = uy^0 = a^{l^2}$

$\Rightarrow a^{l^2} \notin L$

$\therefore$ By Contradiction.

$\Rightarrow L = \{a^n b^j\}$ is not context free Language

(b) $L = \{ w \in \{a, b, c\}^* : n_a(w) = n_b(w) \leq n_c(w) \}$

$\Rightarrow$ To prove L is not CFL we can take any string. For simplicity we will take $a^n b^n c^m$, $(m \geq n)$.

Taking the pumping length to be l $\Rightarrow$ $n = l$ & $m = l+1$

$s = a^l b^l c^{l+1}$

$s = uvxyz$

Let $u = a^l$, $|v| = |x| = 0$, $y = b^l$, $z = c^{l+1}$

$\therefore uv^i xy^i z \in L'$

Lets take $i = 9$

$uv^9 xy^9 z = uy^9 z = a^l b^{l^9} c^{l+1}$

$\Rightarrow a^l b^{l^9} c^{l+1} \notin L'$

Hence by contradiction, as $a^n b^n c^m$ is not a CFL

$\therefore L = \{ w \in \{a, b, c\}^* \; n_a(w) = n_b(w) \leq n_c(w) \}$ is not CFL.

(c) $L = \{ w \in \{a, b, c\}^* : \dfrac{n_a(w)}{n_b(w)} = n_c(w) \}$.

Taking the string to be $s = a^{nm} b^m c^n$; $s \in L$

Taking the pumping length to be l

$\Rightarrow n = m = l$.

$\Rightarrow uvxyz = a^{l^2} b^l c^l$

Let $|v| = |x| = 0$; $u = a^{l^2}$, $y = b^l$, $z = c^l$

$\therefore uv^i xy^i z \in L$ for $i \geq 0$

Taking $i = 2$

$\Rightarrow uv^i xy^i z = a^{l^2} (b)^{2l} c^l$

$\Rightarrow \dfrac{n_a(w)}{n_b(w)} = \dfrac{l^2}{2l} = \dfrac{l}{2}$

$n_c(w) = l \neq \dfrac{l}{2}$

$\Rightarrow a^{l^2} b^{2l} c^l \notin L \Rightarrow a^{l^2} b^{2l} c^l$ is not a CFL by contradiction

$\Rightarrow L$ is not a CFL.

(d) $L = \{a^n b^j : n \geq (j-1)^3\}$

$\Rightarrow$ Taking the pumping length to be 'l'

let $j = l$, $n = l^3 \geq (l-1)^3$.

$s = a^{l^3} b^l$

$s = uvxyz \Rightarrow uvxyz = a^{l^3} b^l$

Taking $u = a^{l^3}$

$|v| = 0$ $\quad |y| = b^l$

$|x| = 0$ $\quad |z| = 0$

We know, $uv^i xy^i z \in L$ if L is CFL $\forall\, i \geq 0$

Taking $i = l^3$

$s' = uv^3 xy^3 z = a^{l^3} (b)^{3l}$

$\Rightarrow n = l^3$, $j = 3l$

if $n \geq (j-1)^3$ then $s' \in L$.

$(j-1)^3 = (3l-1)^3 = 27l^3 - 18l - 1$

Now

$n \geq (g-1)^3$

$l^3 \geq 27l^3 - 18l - 1$

$0 \geq 26l^3 - 18l - 1$

But we know for the equation

$26x^2 - 18x - 1 \leq 0$, there are no, positive integral solutions.

$\Rightarrow s' \notin L$

$\Rightarrow$ ~~[illegible]~~

By contradiction ; L is not context free.

(e) $L = \{a^{n!} : n \geq 0\}$

$\Rightarrow$ Taking the pumping length to be l

Taking $s = a^{l!}$

$s = uv^i xy^i z$, $i \geq 0$

$a^{l!} = uv^i xy^i z$

$|vy| = k$; $1 \leq k \leq p$.

$s' = a^{l!+k}$.

$\therefore$ if $a^{l!+k} \in$ to L

$\because (l+1)! - l! = l(l)! > l \geq k$

$\therefore (l+1)! - l! > k$

$(l+1)! > l! + k$

$\therefore (l+1)! > l! + k > l!$

Hence $(l! + k)$ is not a factorial

$\Rightarrow a^{l!+k} \notin L$

Hence by contradiction ; L is not context free

Que 2 :- (i) **Intersection**

CFL is not closed under intersection.

If $L_1 = a^n b^n c^m$ $\{m, n \geq 0\}$
$L_2 = a^m b^n c^n$

Both of them separately are CFL's but now taking the intersection $L_1 \cap L_2$ which comes out to be $a^n b^n c^n$, $n \geq 0$

Now, we know $a^n b^n c^n$ is not a CFL

Therefor CFL's are not closed under intersection.

Complementation

CFL's are not closed under complementation:

~~Proof~~ → we will prove this by contradiction

Taking 2 CFL's L_1 & L_2 now ⇒ $\bar{L_1}$ & $\bar{L_2}$ are also CFL

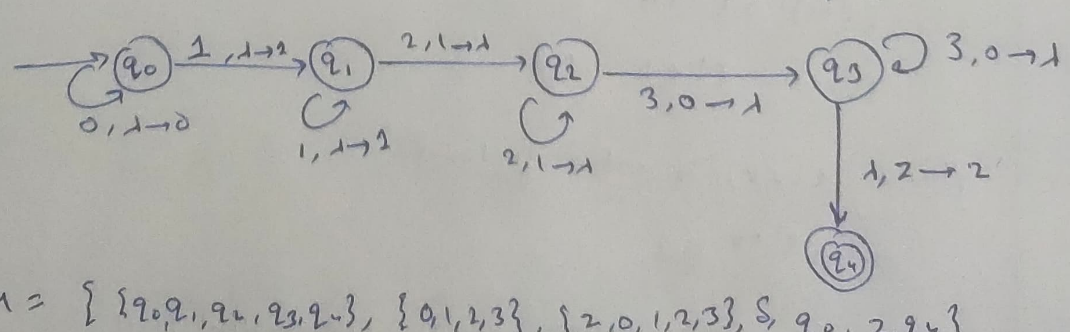
Now $\overline{\bar{L_1} \cup \bar{L_2}} = L_1 \cap L_2$ (De Morgan's Law)

But we proved in the first example that CFL are not closed under intersection.

⇒ Our assumption is wrong
⇒ Contradiction
⇒ CFL's are not closed under complementation.

(ii) Construct PDA for $L = \{0^n 1^m 2^m 3^n \mid n \geq 1, m \geq 1\}$

→



$M = \{\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1, 2, 3\}, \{Z, 0, 1, 2, 3\}, S, q_0, Z, q_4\}$

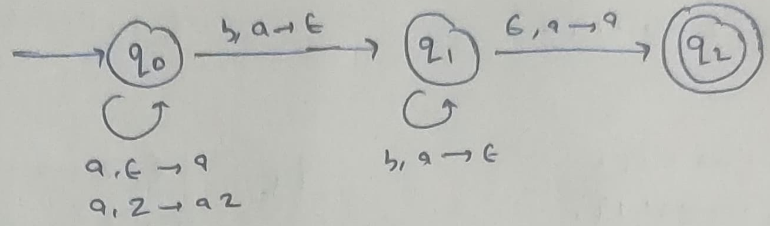
(iii) $S \rightarrow XY$

$X \rightarrow aX \mid a$

$Y \rightarrow aYb \mid \epsilon$

Now by the above CFG, the language generated is:

$L = a^n b^m$; $n > m$, $n > 0$.

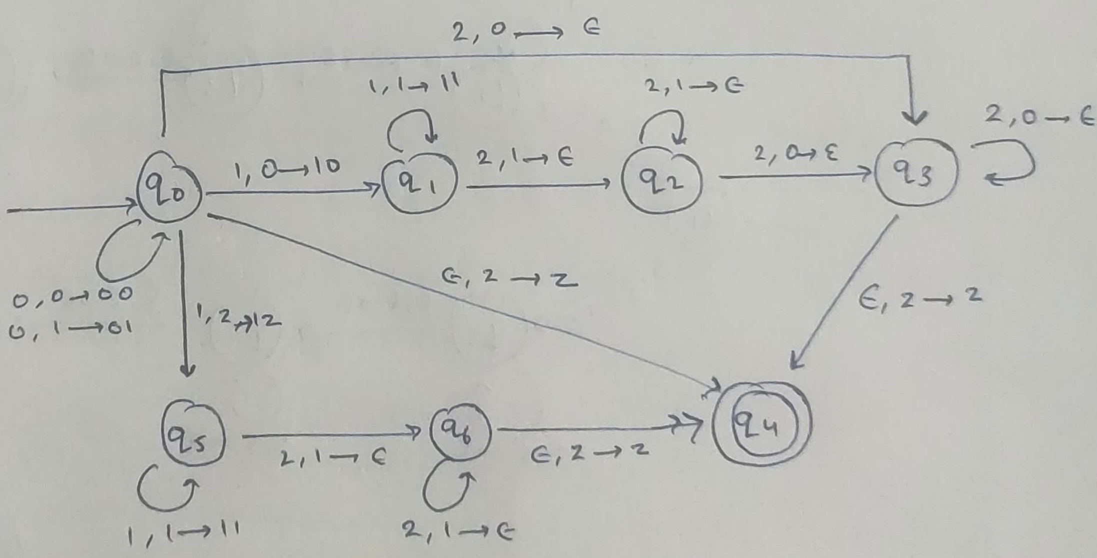


$M = \{\{q_0 q_1 q_2\}, \{a b\}, \{Z, a\}, S, q_0, Z, q_2\}$

Que 3: $\{a^i b^j c^{i+j} \mid i, j \geq 0\}$

(i) Taking

$a = 0$, $b = 1$, $c = 2$



The required PDA

$M = \{\{q_0, q_1 q_2 q_3 q_4 q_5\}, \{0, 1, 2\} \{Z, 0, 1\}, S, q_0, Z, q_4\}$

$\delta \rightarrow$

q_0 $(0, z \rightarrow 0z)\, q_0$
$(0, 0 \rightarrow 00)\, q_0$
$(1, 0 \rightarrow 10)\, q_1$
$(1, z \rightarrow 1z)\, q_5$
$(2, 0 \rightarrow \epsilon)\, q_3$
$(\epsilon, z \rightarrow z)\, q_4$

q_1 $(2, 1 \rightarrow \epsilon)\, q_2$
$(1, 1 \rightarrow 11)\, q_1$

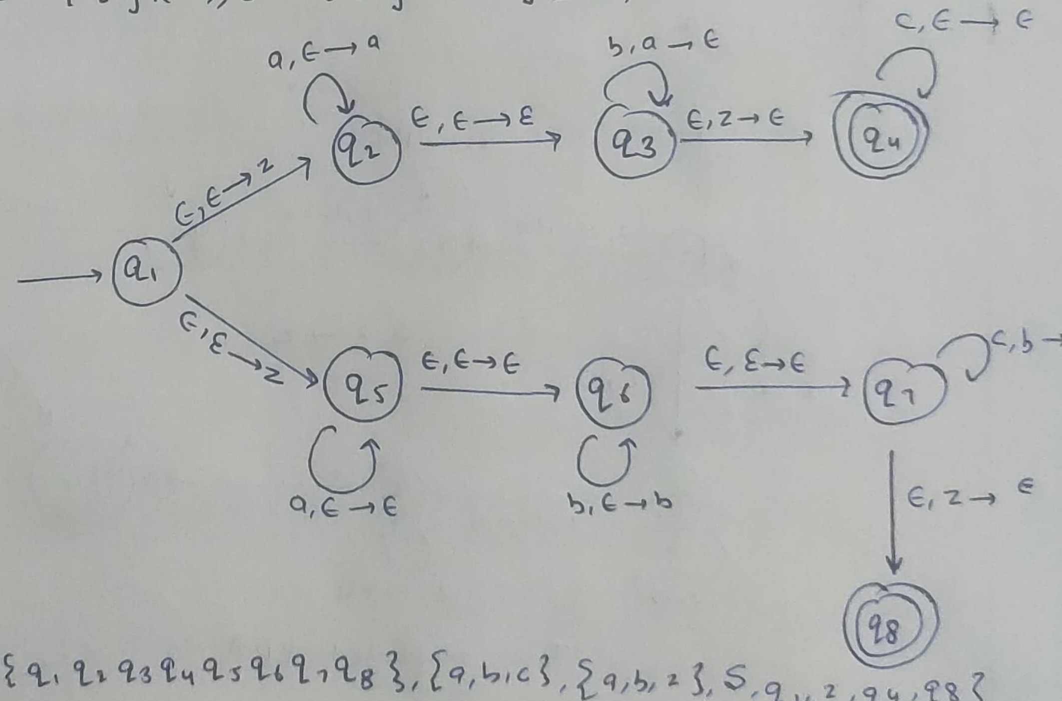
q_2 $(2, 0 \rightarrow \epsilon)\, q_3$
$(2, 1 \rightarrow \epsilon)\, q_2$

q_3 $(\epsilon, z \rightarrow z)\, q_4$
$(2, 0 \rightarrow \epsilon)\, q_3$

q_5 $(1, 1 \rightarrow 11)\, q_5$
$(2, 1 \rightarrow \epsilon)\, q_6$

q_6 $(\epsilon, z \rightarrow z)\, q_4$
$(2, 1 \rightarrow \epsilon)\, q_6$

(b) $\{a^i b^j c^k \mid i, j, k \geq 0 \quad i = j \text{ or } j = k\}$



$M = \{\{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}, \{a, b, c\}, \{a, b, z\}, \delta, q_1, z, q_4, q_8\}$

$\delta \rightarrow$

At q_1 there is a deterministic branch.
If string is $a^i b^i c^k$ $(i = j)$ then PDA goes from $q_1 \rightarrow q_2$. If string is $a^i b^j c^j$ $(j = k)$, PDA goes from $q_1 \rightarrow q_5$

q_2 $(\epsilon, \epsilon \rightarrow \epsilon)\, q_3$
$(a, \epsilon \rightarrow a)\, q_2$

q_5 $(a, \epsilon \rightarrow \epsilon)\, q_5$
$(\epsilon, \epsilon \rightarrow \epsilon)\, q_6$

q_3 $(b, a \rightarrow \epsilon)\, q_3$
$(\epsilon, z \rightarrow \epsilon)\, q_4$

q_6 $(b, \epsilon \rightarrow b)\, q_6$
$(\epsilon, \epsilon \rightarrow \epsilon)\, q_7$

q_4 $(c, \epsilon \rightarrow \epsilon)\, q_4$

q_7 $(c, b \rightarrow \epsilon)\, q_7$
$(\epsilon, z \rightarrow \epsilon)\, q_8$

Que 4:-

$M = \{\{q_0 q_1\}, \{[,]\}, \{z, [\}, \delta, q_0, z, q_1\}$

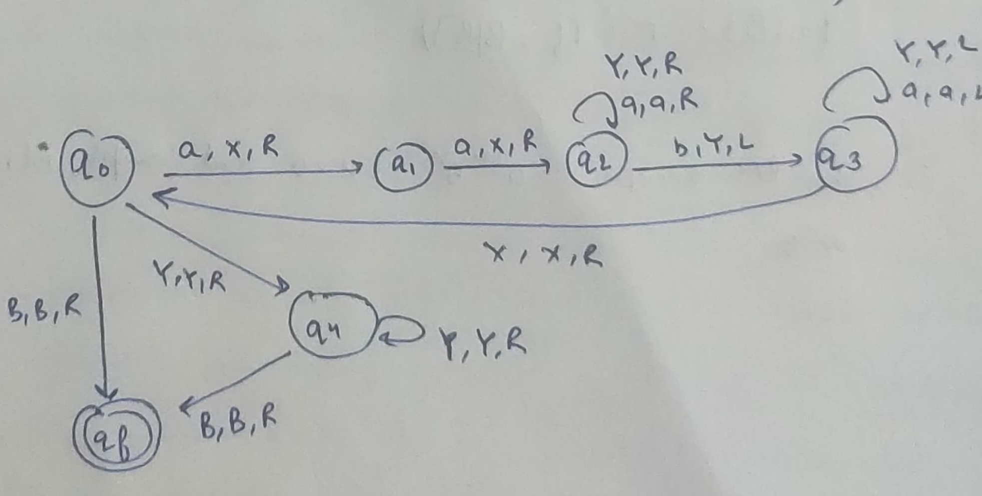
$\delta \Rightarrow$

q_0 $([, [\rightarrow [[)\, q_0$
$([, z \rightarrow [z)\, q_0$
$(], [\rightarrow \epsilon)\, q_0$
$(\epsilon, z \rightarrow z)\, q_1$

The string in question: [[][]][]

$(q_0, [[][]][], z) \vdash (q_0, [][]][], [z) \vdash (q_0,][]][], [[z)$
$\vdash (q_0, []][], [z) \vdash (q_0,]][], [[z) \vdash (q_0,][], [z) \vdash$
$(q_0, [], z) \vdash (q_0,], [z) \vdash (q_0, \epsilon, z) \vdash (q, \epsilon, z)$

Ques:-



The Turing Machine accepts the language

$L = \{(a^{2n} b^n ; n \geq 0) \cup (\epsilon)\}$

1. Showing transition for $n=1$

(i) is aab

$(q_0, \boxed{a}ab) \vdash (q_1, X\boxed{a}, b) \vdash (q_2, XX, \boxed{b}) \vdash (q_3, X\boxed{X}Y)$

$\vdash (q_0, XX\boxed{Y}) \vdash (q_4, XXY\boxed{B}) \vdash (q_f, XXYB\boxed{B})$

And this string is accepted.

(ii) Same we can show for. $n=2$ aaaabb

$(q_0, \boxed{a}aaabb) \vdash (q_1, X\boxed{a}aabb) \vdash (q_2, XX\boxed{a}abb) \vdash (q_2, XXa\boxed{a}bb)$

$\vdash (q_2, XXaa\boxed{b}b) \vdash (q_3, XXa\boxed{a}Yb) \vdash (q_3, X\boxed{X}aaYb)$

$\vdash (q_0 XX\boxed{a}aYb) \vdash (q_1, XXX\boxed{a}Yb) \vdash (q_2, XXXX\boxed{Y}, b) \vdash$

$(q_2, XXXXY\boxed{b}) \vdash (q_3, XXXX\boxed{Y}Y) \vdash (q_3, XXX\boxed{X}YY) \vdash$

$(q_0, XXXX\boxed{Y}Y) \vdash (q_4, XXXXY\boxed{Y}) \vdash (q_4, XXXXYY\boxed{B}) \vdash (q_f, XXXXYYB\boxed{B})$

And this string is also accepted.

(iii) Now checking for blank strings

Transition for ϵ strings

$(q_0, \boxed{B}) \vdash (q_f, B\boxed{B})$

$\therefore$ The TM accepts $a^{2n}b^n$, $n>0$ and empty blank strings only.

Que 6:-

(a) Every language accepted by a multi tape TM is recursively enumerable.

⇒ Suppose languge L is accepted by a k-Tape TM "M".
we will simulate M with a one Tape TM "N" whose tape we will assume as having 2k tracks.
Half those tracks hold the tapes of M and other half of the tracks each hold only a single marker that indicates where the head of the corresponding tape of M is currently located.

Assuming k = 2, The second and the fourth track hold the contents of the first and second tapes of M, track 1 holds the position of the head of tape 1 and track 3 holds the position of the second head tape

control storage

Track 1	- -				X			. . .
Track 2	- -	A1	A2	--	Ai	--		- - -
Track 2	- - -						X	- -
Track 4	- -	B1	B2	--	Bi	--	Bj	- -

To simulate a move of M, N's head must visit must visit the k head markers, So that N not get lost, it must remember how many head markers are to its left all the times; that count is stored as a component of N's finite control.
After visiting each head marker and storing the scanned symbol in the component of its finite control, N knows what tape symbols are being scanned by each of M's heads.
N also knows the state of M, which it stores in N's own finite control. Thus, N knows what move M will make.

N now revisits each of the head markers on its tape, changes the symbol in the track representing the corresponding ~~states of~~ tapes of M, and moves the head markers left or right.

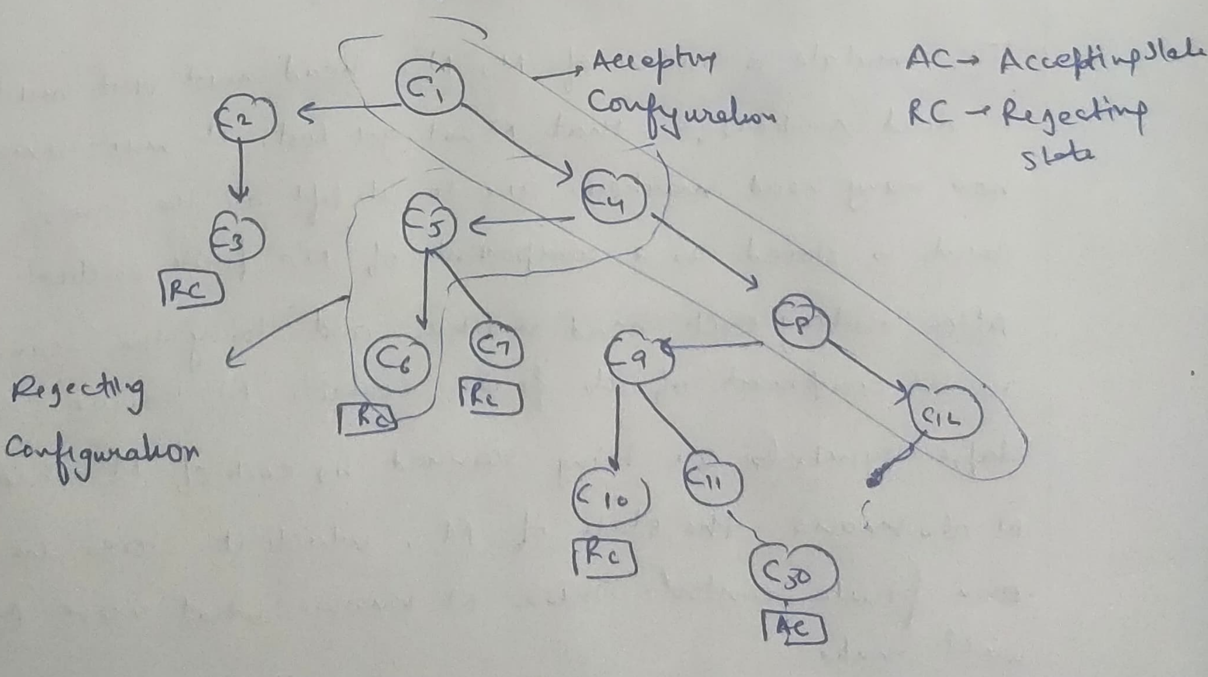
Finally, N changes the states the state of M as recorded in its own finite control. At this point, N has simulated one move of M.

We select N's accepting state as all those states that record M's state as 1 of the accepting state. Thus whenever M accepts, N also accepts and vice-versa

Hence Proved, that language accepted by a multitape TM is recursively enumerable.

(b) If M1 is non deterministic TM, then there is a deterministic TM M_2 such that $L(M1) = L(M2)$

⇒ We will try all the branches possible of some NTM until accepting or halt configuration is reached at the end.



Now to construct a Deterministic TM, we follow the steps as:

1) We will do a breadth first Search Manner to process the configuration.

2) From the tape that is given we maintain a queue from it.

3) We get next configuration from the head of the queue, we will then erase that configuration and then we will push the resulting configuration.

4) Pushing of the resulting configuration takes place at the back of the queue.

5) As we do each copy, we will simulate ~~all~~ the non deterministic TM moves.

   ⇒ We will determine the value of each cell

   ⇒ If the value is valid we will update the tape.

   ⇒ If not we will go back

   ⇒ We abort if transition represent reject

6) We will keep doing this until the accepting state is reached.

Hence proved;

Non deterministic TM can be converted into Determenistic TM.

Que 7:

(a) Now, we have a non deterministic TM, so there we can have a branch that is non deterministic at initial state to either move left or right, entering one of 2 different states on either side of initial state.

State	B	\$
q	$\{(q_1, B, R), (q_2, B, L)\}$	$(p, \$, L)$
q_1	(q_1, B, R)	$(p, \$, R)$
q_2	(q_2, B, L)	$(p, \$, L)$
p		

Each of this state can proceed in its own direction and will keep moving unless they encounter \$ which will make them enter p.

The pointer has to move off the \$ entering another state and then move back to \$ to enter into state p while moving backwards.

(b) Now considering that the TM were deterministic then this implies that we cannot have 2 branches. Here we can go one step left or right. and we will have to use right and left markers to keep a track.

Now we can go, one step left and mark a X, then 2 steps right and mark a Y, then 3 steps left and mark a X and then 4 steps right to mark a Y. We will keep doing this until we find a \$.

We can enter the state p deterministically after we find our symbol \$.