

B.Tech.

Fourth Semester Examination 2009-10

Theory of Automata and Formal Language

Note : (i) This question paper contains Five questions.

(ii) Attempt all questions.

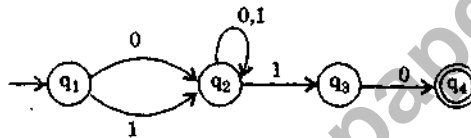
Q. 1. Attempt any four of the following :

(4 × 5 = 20)

Q. 1. (a) Draw a DFA that accepts set of all strings of 0 and 1 that end in the set two same symbols.

Ans. The set of all string ending in 00

$$= (0 + 1)^* 00$$



State/ Σ	0	1
$\rightarrow q_1$	q_2	q_1
q_2	q_1, q_3	q_2
q_3	q_4	—
$\widehat{q_4}$	—	—

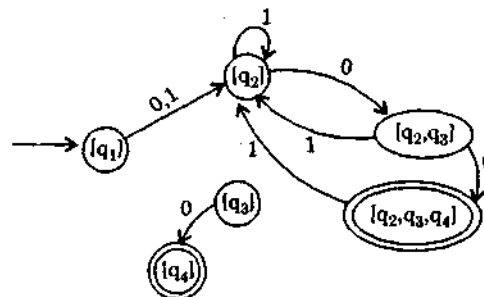
Now we can construct DFA for a above NFA

Transition table

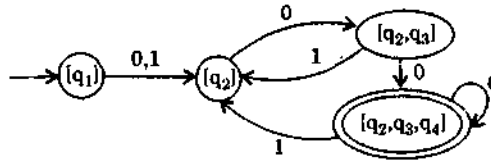
State/ Σ	0	1
$[q_1]$	$[q_2]$	$[q_2]$
$[q_2]$	$[q_2, q_3]$	$[q_2]$
$[q_3]$	$[q_4]$	—
$[q_2, q_3]$	$[q_2, q_3, q_4]$	$[q_2]$
$[q_2, q_3, q_4]$	$[q_2, q_3, q_4]$	$[q_2]$
$[q_4]$	—	—

New transition diagram for DFA

where $Q' = [q_1], [q_2], [q_3], [q_2, q_3], [q_4]$,



$[q_2, q_3, q_4]$



This is the final DFA which accepts $(0 + 1)^* 00$.

Q. 1. (b) Write down procedure for converting an NFA to its equivalent of A.

Ans. Equivalent DFA of the given NFA as

		1
$\rightarrow p$	$\{q, s\}$	$\{q\}$
$\{q, s\}$	$\{q, s, r\}$	$\{q, r\}$
$\{q, r, s\}$	$\{q, s, r\}$	$\{p\}$
$\{q, r, s\}$	$\{q, r, s\}$	$\{p\}$

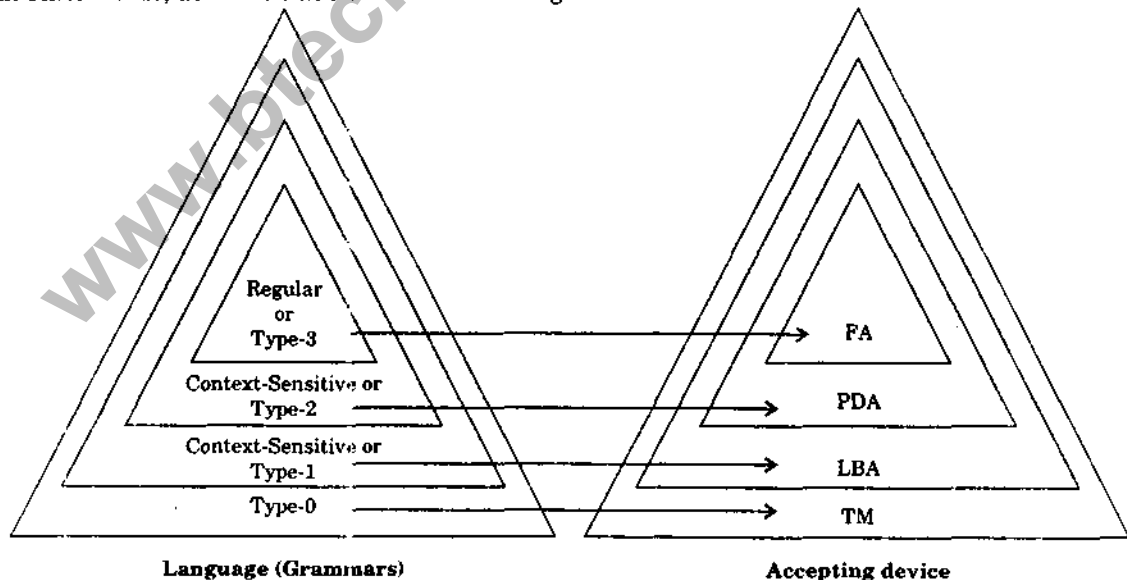
Q. 1. (c) Give the Chomsky Hierarchy of grammars specifically giving form of production rules in each class of grammar.

Ans. Chomsky classified the grammars into four types in terms of production (types 0-3).

Type-0 : A type-0 grammar in any phrase structure grammar without any restriction. In a production of the form $\phi A \psi \rightarrow \phi \alpha \psi$ where A is a variable, ϕ is called the left context ψ the right context, and $\phi \alpha \psi$ the replacement string.

Type-1 : A grammar is called type-1 or context sensitive context department if all its production are type-1 production. A production if $a \neq \Lambda$. In type-1 productions, erasing of A is not permitted.

Type-2 : A type-2 production in a production of the form $A \rightarrow \alpha$, where $A \in V_N$ and $\alpha \in (V_N \cup \Sigma)^+$. In other words, the LHS has no left context or right context.



Type-3 : A grammar is called a type-3 or regular grammar if all its production or type-3 production. A production $S \rightarrow \Lambda$ is allowed in type-3 grammar, but in care S does not on the right hand side of any production.

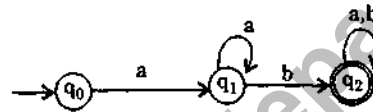
Chomsky Hierarchy : Chomsky hierarchy classified the grammars into four parts.

The given diagram shows hierarchy of grammar and their accepting device. The production of above grammar are as follows :

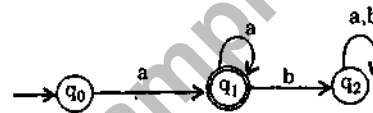
- (i) Regular $\Rightarrow A \rightarrow aB, A \rightarrow a, A, B \in V, a \in \Sigma$
- (ii) Context free $\Rightarrow A \rightarrow \alpha, A \in V, \alpha \in (V \cup \Sigma)^*$
- (iii) Context sensitive $\Rightarrow A \rightarrow \alpha, \alpha \in (V \cup \Sigma)^+, A \in V$
- (iv) Type-0 $\Rightarrow \alpha \rightarrow B, \alpha, B \in (V \cup \Sigma)^*$

Q. 1. (d) Given a DFA M. Suggest a procedure to draw DFA which accepts the complement of the language accepted by M.

Ans. If L is a regular set over Σ then $\Sigma^* - L$ is also regular over Σ . We construct another DFA M' ($Q, \Sigma, \delta, q_0, F'$) by defining $F' = Q - F$, i.e., M and M' differ only in their final states. A final state of M' is a nonfinal state of M and vice versa. The state diagram of M and M' are the same except for the final state.



Hence M' is



$W \in L(M')$ if and only if $\delta(q_0, W) \in F' = Q - F$

i.e., if $W \notin L$. This proves

$L(M') = \Sigma^* - L$

Q. 1. (e) Define equivalence relation and show that an equivalence relation partitions a set into a number of disjoint equivalence classes.

Ans. Assume that R is a relation on a set A in other word $R \subseteq A \times A$. As noted prev we write aRb instead of $(a, b) \in R$ to indicate that a is related to b via R .

1. R is reflexive if for every $a \in A$ aRa
2. R is symmetric if for every a and b in A if aRb then bRa .
3. R is transitive if for every a, b and c in A if aRb and bRc then aRc .
4. R is an equivalence relation on A if R is reflexive, symmetric and transitive suppose R is an

equivalence relation on A . For any element a of A , we denote by $[a]R$ or simply by $[a]$ the equivalence class containing a

$[a]R = \{x \in A / xRa\}$

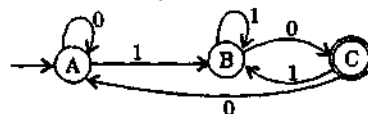
Note that the equivalence class containing a really does contain a .

For any partition C of a set A the relation R on A defined by

xRy if and only if x and y belong to the same element of C is an equivalence relation on A .

Q. 1. (f) Give the format of production rules for left and right linear grammars and show that they are equivalent.

Ans. Let $L = \{0, 1\}^* (10)$ the set of all string over $\{0, 1\}$ that end in 10



$A \Rightarrow 1B \Rightarrow 11B \Rightarrow 110C \Rightarrow 1100A \Rightarrow 11000A$
 $\Rightarrow 110001B \Rightarrow 1100010C \Rightarrow 11000101B$

$\Rightarrow 110001010C$

$A \rightarrow 1B \quad C \rightarrow 0A$

$B \rightarrow 1B \quad A \rightarrow 0A$

$B \rightarrow 0C \quad C \rightarrow 1B$

These include every production of form

$P \rightarrow aQ$ where $P \xrightarrow{a} Q$

is a transition in fA .

Q. 2. Attempt any four of the following :

(4 × 5 = 20)

Q. 2. (a) Write a regular expression for the language containing all strings of 0 and 1 that begin with 1 and contain even number of 0. Convert this regular expression into equivalent NFA without ϵ -moves.

Ans. $L = \{00, 01, 10, 11\}^*$

regular expression

$1(00 + 01 + 10 + 11)^*$

or

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

Q. 2. (b) Define a regular expression and write various operators used in constructing a regular expression. Also give their precedence.

Ans. (i) $\phi + R = R$

(ii) $\phi R = R\phi = \phi$

(iii) $\wedge R = R\wedge = R$

(iv) $\wedge^* = \wedge$ and $q^* = \wedge$

(v) $R + R = R$

(vi) $R^* R^* = R^*$

(vii) $RR^* = R^*R$

(viii) $(R^*)^* = R^*$

(ix) $\wedge + RR^* = R^* = \wedge + R^*R$

(x) $(PQ)^* P = P(QP)^*$

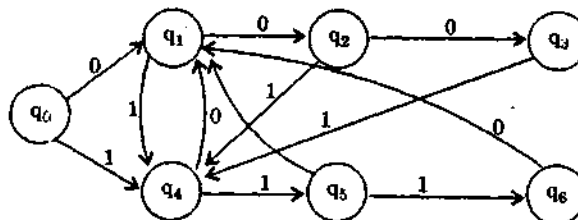
(xi) $(P + Q)^* = (P^*Q^*)^* = (P^* + Q^*)^*$

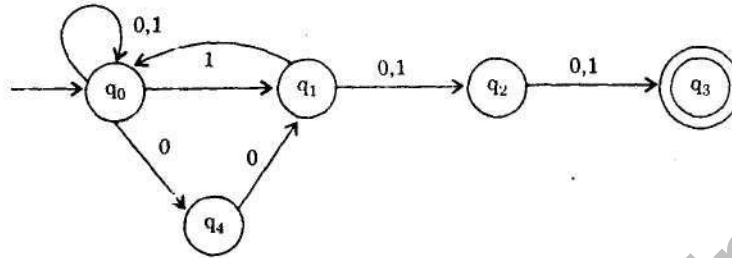
(xii) $(P + Q)R = PR + QR$ and

$R(P + Q) = RP + RQ$

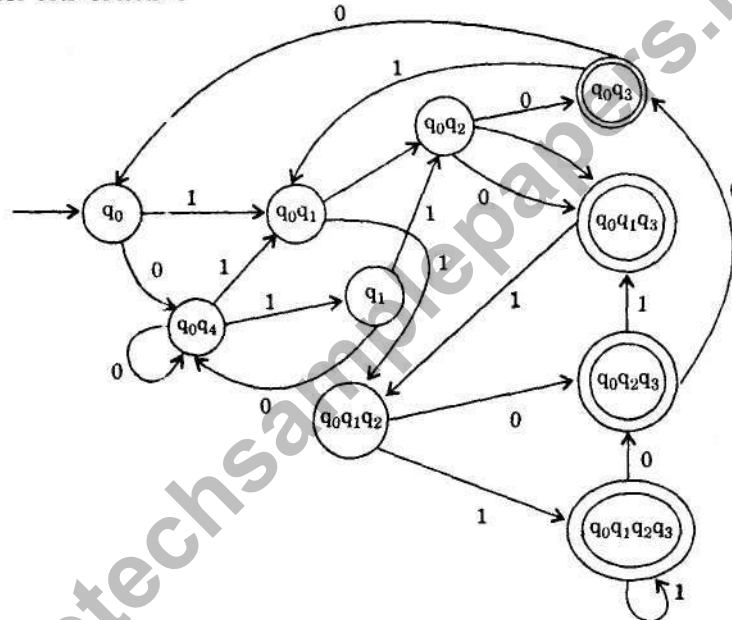
Q. 2. (c) Draw an NFA that accepts all strings of 0 and 1 containing 111 as substrings and hence convert this NFA to DFA.

Ans. The NFA is





The DFA after conversion is



Q. 2. (d) Design a Mealy Machine that scans sequence of inputs of 0 and 1 and generates output 'A' if the input string terminates in 00, output 'B' if the string terminates in 11, and output 'C' otherwise.

Ans. Mealy Machine : If the string ends on 11 or 00 O/P = Yes

If the string not ends on otherwise O/P = No

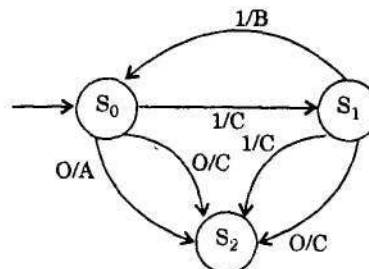
Set of states $S \{S_0, S_1, S_2\}$

Initial state $S = S_0$

Input alphabet $E \{0, 1\}$

Output alphabet $Y \{A, B, C\}$

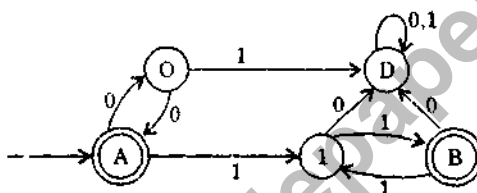
Input words can be any sequence of ones and zeros.



State	Input	New State	Output
S_0	0	S_2	C
S_0	1	S_1	C
S_1	0	S_2	C
S_1	1	S_0	B
S_2	0	S_0	A
S_2	1	S_1	C

Q. 2. (e) Draw a DFA that accepts the language $L = \{a^{2n} \mid n \geq 1\}$.

Ans. $\{00\}^* \{11\}^*$



Q. 2. (f) Write the statement of Pumping Lemma and using it prove that the language $\{a^n b^n c^n \mid n \geq 1\}$ is not regular.

Ans. 1. Let us suppose L is regular and we get a contradiction suppose finite automator has n states that accepts long L .

2. Let $z = 0^n 1^n$ then $|z| = 2n > n$. By pumping Lemma we write $z = uvw$ with $|uv| \leq n$ and $|v| \neq 0$.

3. Now we have to find l so that $uv^l \in L$ for getting a contradiction.

The string V can be in any one of the three possible to

1. The string V is constructed by using only 0's it means $v = 0^k$ for some $k \geq 1$.

2. The string v is constructed by using only 1's it means $v = 1^l$ for some $l \geq 1$.

3. The string v is constructed by using both symbol 0's and 1's. Then the string v will be of the form $v = 0^m 1^p$ for some $m \geq 1$ and $p \geq 1$.

In case (1) we take $l = 0$. As $uvw = 0^n 1^n$

$uv = 0^{n-k} 1^n$ as $k \geq 1$, $n-k \neq n$. Therefore $uv \notin L$.

In case II we take $l = 0$ $uvw = 0^n 1^{n-l}$, $n \neq n-l$ $uvw \notin L$

In case III if we take $l = 2$

As we have

$uvw = 0^{n-m} 0^m 1^p 0^m 1^p 1^{n-p}$

Now uv^2w will be

$$0^{n-m} \underbrace{0^m 1^p 0^m 1^p}_{uv^2w} 1^{n-p}$$

As the string $0^{n-m} 0^m 1^p 0^m 1^p 1^{n-p}$ can not be expressed in the form $0^n 1^n$ $\therefore uv^2w \notin L$.

Hence the long L is not regular.

Q. 3. Attempt any two of the following :

(2 × 10 = 20)

Q. 3. (a) Consider the grammar with following production rules :

$S \rightarrow AB \mid AC$

$A \rightarrow aA \mid bAa \mid a$

$B \rightarrow bbA \mid aB \mid AB$

$C \rightarrow aCa \mid aD$

$D \rightarrow aD \mid bC$

Convert the above grammar into Greibach Normal Form.

Ans. Given production of CFG (V_N, Σ, P, S) where production P are as follows :

$S \rightarrow AB \mid AC$

$A \rightarrow aAb \mid bAa \mid a$

$B \rightarrow bbA \mid aaB \mid AB$

$C \rightarrow abCa \mid aDb$

$D \rightarrow bD \mid aC$

Construction of P' : Now we can change the production of A as : $A \rightarrow aab \mid baa \mid a$

Now, we can change the production of B as :

$B \rightarrow bba \mid aabb \mid aB$

$B \rightarrow bba \mid aabba \mid a bb A.$

$B \rightarrow bba \mid aabba \mid a bb aa' \mid abbaa \mid aa$

Now, we can make any change in variable C and D and it is also not terminated so it will not include in V_N .

Hence our final CFG is as follows :

$P \rightarrow$

$S \rightarrow AB \mid AC$

$A \rightarrow aab \mid baa \mid a$

$B \rightarrow bba \mid aabba \mid abbaa \mid abbaa \mid aa.$

Q. 3. (b) By drawing two rightmost or two leftmost derivations for a suitably selected string, show that the following grammar is an ambiguous grammar

$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid D$

$D \rightarrow a \mid b \mid c \mid d$

Also convert the grammar into un-ambiguous grammar.

Ans. For $E \rightarrow E + E$

New production $E \rightarrow E + T \mid T$

for $E \rightarrow E * E$

New production $E \rightarrow E * f \mid f$

$E \rightarrow a \mid b$ remain same because they do not show any ambiguity.

Hence the constructed un-ambiguous grammar (G') equivalent to G is

$G' = (\{E, T, f\}, \{a, b, c, d\}, P, E)$

$P = \{E \rightarrow E + T \mid T, E \rightarrow E * f \mid f, E \rightarrow a \mid b\}$

Q. 3. (c) Use CYK algorithm to check whether string $a + b * c - d$ is in $L(G)$ of the un-ambiguous grammar obtained in (b) above.

Ans. The given string is $-W = a + b * c - d$

the grammar is :

$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid E \mid D$

$D \rightarrow a \mid b \mid c \mid d$

Let E be S_1 then the grammar will be

$S_1 \rightarrow S_1 + S_1 \mid S_1 * S_1 \mid S_1 - S_1 \mid S_1 / S_1 \mid a \mid b \mid c \mid d$

For CYK algorithm, the grammar must be in CNF.

A grammar will be in CNF if it follows the following rules :

Variable \rightarrow Terminator

or

Variable \rightarrow Variable₁ Variable₂

A grammar will be in CNF only if it is a simplified grammar so, first of all we will make the given grammar simplified.

$$\Rightarrow E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid a \mid b \mid c \mid d$$

$$\Rightarrow E \rightarrow E + T \mid E - R \mid E * S \mid E / P \mid a \mid b \mid c \mid d$$

$$T \rightarrow$$

$$\vdots \vdots \vdots \vdots$$

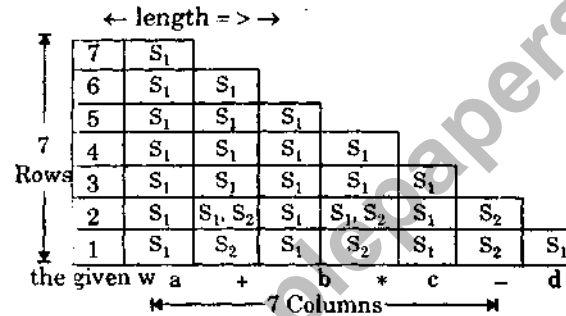
The GNF grammar will be:

$$\text{Let } S_2 \text{ be } S_2 \rightarrow + \mid * \mid = \mid /$$

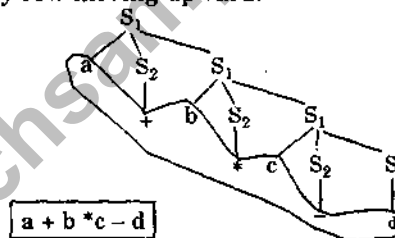
$$\text{then } S_1 \rightarrow aS_2S_1 \mid bS_2S_1 \mid cS_2S_1 \mid dS_2S_1$$

$$S_1 \rightarrow a \mid b \mid c \mid d, S_2 \rightarrow + \mid - \mid * \mid / \quad W$$

Now CYK: We have to check $W = a * b - c / d$



We will fill the table row by row moving upward.



Q. 4. Attempt any two of the following :

(2 × 10 = 20)

Q. 4. (a) Write transition rules for a PDA corresponding to the following Context Free Language :

$$L = \{w c w^R \mid w \text{ is in } (0 + 1)^* \text{ and } w^R \text{ represents reverse } W\}$$

Also obtain context Free Grammar for this PDA.

$$\text{Ans. } M = (\{q_0, q_1, q_2\}, \{0, 1, \Delta\}, \{0, 1, z_0\}, \delta_1, q_0, z_0 \{q_2\})$$

$$\delta(q_0, 0, z_0) = (q_0, 0z_0) \quad \delta(q_0, 2, 0) = (q_1, 0)$$

$$\delta(q_0, 1, z_0) = (q_0, 1z_0) \quad \delta(q_0, 2, 1) = (q_1, 1)$$

$$\delta(q_0, 0, 0) = (q_0, 00) \quad \delta(q_0, 2, z_0) = (q_1 z_0)$$

$$\delta(q_0, 1, 1) = (q_0, 11) \quad \delta(q_1, 0, 0) = (q_1, \pi)$$

$$\delta(q_0, 1, 0) = (q_0, 10) \quad \delta(q_1, 1, 1) = (q_1, \pi)$$

$$\delta(q_0, 0, 1) = (q_0, 0, 1) \quad \delta(q_1, \pi, z_0) = (q_0, z)$$

Q. 4. (b) Prove that the PDA that accepts strings through empty stack and final state mechanisms are equivalent.

$$\text{Ans. } \{a^m b^m c^n \mid m, n \geq 1\}$$

$$\delta(q_0, a_1, z_0) = (q_0, a z_0)$$

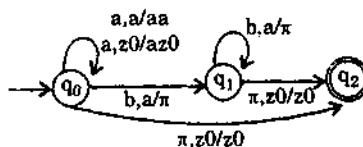
$$\delta(q_0, a, a) = (q_0, a a)$$

$$\begin{aligned} \delta(q_0, b, a) &= (q_1, \epsilon) \\ \delta(q, b, a) &= (q, \epsilon) \\ \delta(q_1, c, z_0) &= (q_1, z_0) \\ \delta(q_0, \epsilon, z_0) &= (q_f, \epsilon) \end{aligned}$$

Q. 4. (c) Draw a PDA for the CFG given below :

$$S \rightarrow a | b | \epsilon$$

Ans.

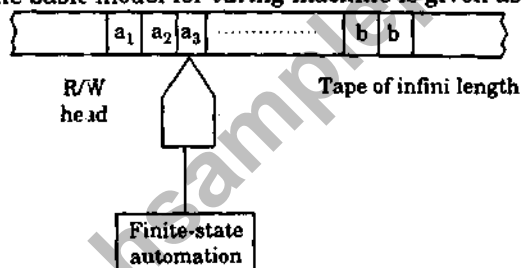


Q. 5. Attempt any four of the following :

(4 × 5 = 20)

Q. 5. (a) How do you define Instantaneous Description for a Turing Machine ?

Ans. Turing Machine : The Turing machine can be thought of as a finite state automaton connected to a R/W head. The basic model for Turing machine is given as follows :



In one move the machine examines the present symbol under R/w head on the tape and the present state of an automation to determine.

- (i) a new symbol to be written on the tape.
- (ii) motion of the R/W head either left (L) or right (R).
- (iii) the next state.
- (iv) whether to halt or not

A Turing machine M is a 7-tuple

$$i.e., (Q, \Sigma, T, \delta, q_0, b, F)$$

where

Q = is a finite non-empty set of states

T = is a finite non-empty set of tape symbols

$b \in T$ = is blank

Σ = is a set of input symbols and is a subset of T

δ = is transition function from $Q \times T$ to $Q \times T \times (L, R)$

q_0 = is initial state

F = is a set of final states.

Q. 5. (b) Design a Turing Machine that accepts the language $L = \{ww^R \mid w \text{ is in } (0+1)^*\}$ and w^R represents reverse w .

Ans. Turing machine for the language $ww^R \mid w$ is any string of 0's and 1's w^R is the reverse of string w . We have the following steps for processing ww^R .

(a) The Turing machine M scans the first symbol of the input tape (0 and 1), erase it and change state (q_1 or q_2).

- (b) M scans the remaining part without changing the tape symbol until it encounters b .
 (c) The R/w head moves to the left. If the rightmost symbol tallies with the leftmost symbol, the right most symbol is erased, other wise M halts.
 (d) The R/w head moves to the left until b is encountered.

Transition Table

Present state	Input symbol		
	0	1	n
$\rightarrow q_0$	bRq_1	bRq_2	bRq_7
q_1	$0Rq_1$	$1Rq_1$	bLq_3
q_2	$0Rq_1$	$1Rq_2$	bLq_4
q_3	$0Lq_5$	-	-
q_4	-	bLq_5	-
q_5	$0Lq_5$	$1Lq_5$	bRq_0
q_6	$0Lq_6$	$1Lq_6$	bRq_0
$\textcircled{q_7}$	-	-	-

Q. 5. (c) Write a Turing machine that performs proper subtraction of two integer numbers.

Ans. Assume that the input tape has $0^m 10^n$ where $M - n$ is required. We have the following steps :

- (a) the leftmost 0 is replaced by b and R/W head moves to the right.
 (b) the R/W head replaces the first 0 after 1 by 1 and moves to the left. On reaching the blank at the left end the cycle is repeated.
 (c) Once the 0's to the left of 1's are exhausted M replaces all 0's and 1's by b 's $a - b$ is the number of 0's left over in the input tape and equal to 0.
 (d) Once the 0's to the right of 1's are exhausted, n 0's have been changed to 1's by one 0 and n b's. The number of 0's left over gives the values of $a - b$ the transition table is given as follows :

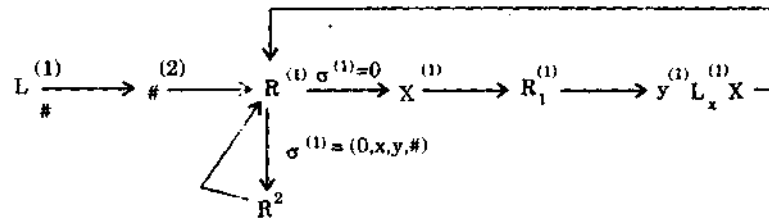
Present state	Input symbol		
	0	1	b
$\rightarrow q_0$	bRq_1	bRq_5	-
q_1	$0Rq_1$	$1Rq_2$	-
q_2	$1Lq_3$	$1Rq_2$	bLq_4
q_3	$0Lq_3$	$1Lq_3$	bRq_0
q_4	$0Lq_4$	bLq_4	$0Rq_6$
q_5	bRq_5	bRq_5	bRq_6
q_6	-	-	-

Q. 5. (d) What is Post Correspondence Problem (PCP) ? Discuss it with one example.

Ans. The post correspondence problem over an alphabet Σ belongs to a class of Yes/No problems and is stated as follows :

consider the two lists $x = (x_1, \dots, x_n)$, $y = (y_1, \dots, y_n)$ of nonempty strings over an alphabet $\Sigma = \{0, 1\}$. The PCP is to determine whether or not there exist $i_1 \dots i_m$ where $1 \leq i_j \leq n$. Such that

$$x_{i_1} \dots x_{i_m} = y_{i_1} \dots y_{i_m}$$



This one will store on the second tape.

Q. 5. (f) Differentiate with suitable examples between recursive and recursively enumerable languages.

Ans. Suppose that $T_1 = (Q_1, \Sigma, \Gamma_1, q_1, \delta_1)$ and $T_2 = (Q_2, \Sigma, \Gamma_2, q_2, \delta_2)$ are two Turing machines accepting L and L' respectively is complement of L .

We construct a two-tape TM $\rightarrow T$ to recognize L .

We include in our set of states the pairs in $Q_1 \times Q_2$.

The two-tape machine $T = (Q, \Sigma, \Gamma, q_0, \delta)$ begins by copying the input string x onto tape 2 and inserting the marker # at the beginning of each tape. The simultaneous simulation on T_1 on tape 1 and tape 2 is accomplished by allowing every possible move

$$\delta((P_1, P_2), (a_1, a_2)) = ((q_1, q_2), (b_1, b_2), (D_1, D_2))$$

where for both values of i

$$\delta_i(p_i, a_i) = (q_i, b_i, D_i)$$

Here we know in advance that on any input x , precisely one of the machines T_1 and T_2 will halt.

Therefore it is sufficient to modify T as follows :

(i) When T_1 or T_2 halts, T erases tape 1 and

(ii) leaves the appropriate output before halting-1 or 0 depending on whether the machine that halted was the one accepting L or the one accepting L' .