B.Tech.

FOURTH SEMESTER EXAMINATION, 2006-07

THEORY OF AUTOMATA AND FORMAL LANGUAGES

Time : 3 Hours **Total Marks : 100**

Note : (i) Attempt all questions.

(ii) All questions carry equal marks.

Q. 1. Attempt any two parts of the following :

Q. 1. (a) (i) Find the transitive closure R^+ and reflexive and transitive closure R^* of the relation : 4

$$R = \{(1, 2), (2, 3), (3, 4), (5, 4)\}$$

Ans. $R = \{(1,2), (2,3), (3,4), (5,4)\}$

$R^2 = RoR = \{(1,2), (2,3), (3,4), (5,4)\} \text{ o } \{(1,2), (2,3), (3,4), (5,4)\} = \{(1,3), (2,4)\}$

$R^3 = RoR = \{(1,3), (2,4)\} \text{ o } \{(1,2), (2,3), (3,4), (5,4)\} = \{(1,4)\}$

$R^4 = R^3 oR = \{(1,4)\} \text{ o } \{(1,2), (2,3), (3,4), (5,4)\} = \phi$

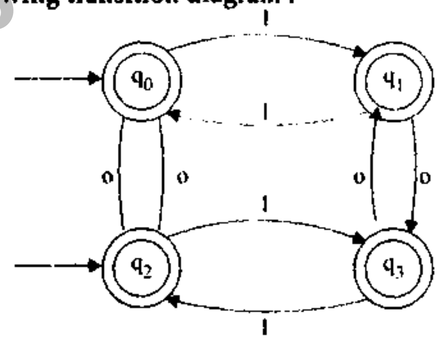
$R^5 = R^6 \ldots\ldots = \phi$

$\therefore R^+ = R \cup R^2 \cup R^3 = \{(1,2),(2,3),(3,4),(5,4),(1,3),(2,4),(1,4)\}$

$R^* = R^+ \cup \{(1,2), (2,2), (3,3), (4,4), (5,5)\}$

$= \{(1, 2), (2, 3), (3, 4), (5, 4), (1, 3), (2, 4), (1, 4), (1, 1), (2, 2), (3, 3), (4, 4), (5, 5)\}$

(ii) Consider the following transition diagram : 6



Test whether the string 110101 is accepted by the finite automata represented by above transition diagram. Show the entire sequence of states traversed.

Ans. For the string 110101, the sequence of states is as follows :

$q_0 \;^{1}\; q_1 \;^{1}\; q_0 \;^{0}\; q_2 \;^{1}\; q_3 \;^{0}\; q_1 \;^{1}\; q_0$ ©©7 ; 7 ⌈

q_0 is final state, therefore the string 110101 is accepted.

Q. 1. (b) Give DFA accepting the following languages over the alphabet {0, 1} : **10**

(i) The set of all strings with three consecutive zeros.

(ii) The set of all strings such that every block of 05 consecutive symbols contains at least two zeros.

Ans. (i) We have to make a DFA which accept all strings over {0, 1}, with three consecutive zeros.

If we write a regular expression for it, it may be of the following form :

(any combination of 0 and 1)* 000 (any combination of 0 and 1)*

(a) The desired DFA may be start with 0 or 1.

(b) Whenever there are three consecutive 0's then we reach from initial state to final state.

(c) If there is 1 after 0 then we will return tothe critical state.

(d) If after accepting 3 0's there are remaining frequency of 0's or is then we will be remain on the final state.

So, the desired DFA will be :

(ii) Let the input alphabets be {0, 1}. If we consider all combinations over {0, 1} of length o5, then there aer $2^5 = 32$ combinations are possible. But out of these 32, some may not contain at least two 0's. These strings may be :

1 1 1 1 1

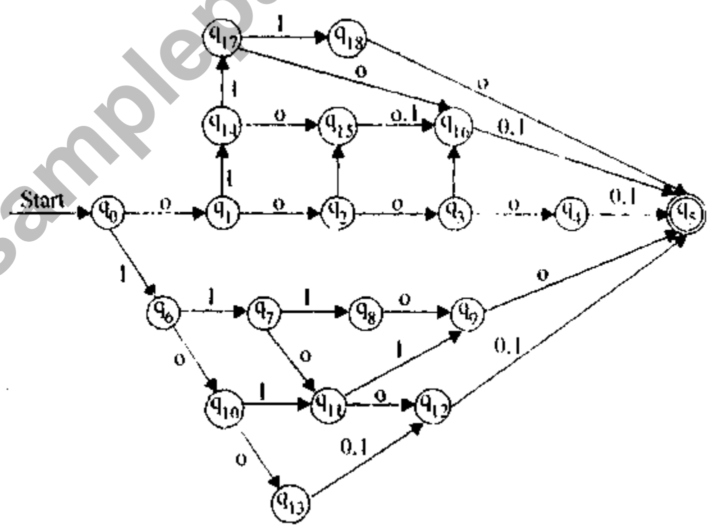
1 1 1 1 0

1 1 1 0 1

1 1 0 1 1

1 0 1 1 1

0 1 1 1 1



Now we have to show the transition graph which accept all strings of length 5, except the above listed strings. The DFA may be as shown in the figure.

Q. 1. (c) Find the equivalence parition and corresponding reduced machine in standard form, for the following machine : **10**

PS	NS, Z	
	X = 0	X = 1
A	F, 0	B, 1
B	G, 0	A, 1
C	B, 0	C, 1
D	C, 0	B, 1

E	D, 0	A, 1
F	E, 1	F, 1
G	E, 1	G, 1

where PS = Present State, NS = Next State, Z = Output, X = I/P

Ans. In the given question, the maching generates an output with every (state, input) combination. So the given machine is Mealy machine.

Now we have to convert it into corresponding Moore machine.

The give Mealy machine M_1 is : $M_1 = \{Q, \Sigma, \Delta, \delta, \lambda, s\}$

Where $Q = \{A, B, C, D, E, F, G\}$

$\Sigma = \{X, Y\}$

$\Delta = \{0, 1\}$

δ = Shown in table

λ = Shown in table

$s = A$

State	X	Output	Y	Output
A	F	0	B	1
B	G	0	A	1
C	B	0	C	1
D	C	0	B	1
E	D	0	A	1
F	E	1	F	1
G	F	1	G	1

Let $M_2 = (Q^1, \Sigma, \Delta^1, \delta^1, \lambda^1, s^1)$ be the equivalent Moore machine

where (1) $Q^1 \subseteq \{[A,0], [A,1], [B,0], [B,1], [C,0], [C,1], [D,0], [D,1], [E,0], [E,1], [F,0], [F,1], [G,0], [G,1]\}$

Since $Q^1 \subseteq Q \times \Delta$

(2) $\Sigma = \{X, Y\}$

(3) $\Delta^1 = \{0, 1\}$

(4) Let starting state $s^1 = [A, 0]$ where A is the initial state and 0, is the output symbol of Mealy machine,

(5) δ^1 is defined as follows :

For initial state [A, 0] :

$$\delta^1([A,0], X) = [\delta(A, X), \lambda(A, X) = [F, 0])$$

$$\delta^1([A,0], Y) = [\delta(A, Y), \lambda(A, Y)] = [B, 1]$$

For state [F, 0] :

$$\delta^1([F,0], X) = [\delta(F, X), \lambda(F, X)] = [E, 1]$$
$$\delta^1([F,0], Y) = [\delta(F, Y), \lambda(F, Y)] = [F, 1]$$

For state [B, 1] :

$$\delta^1([B,1], X) = [\delta(B, X), \lambda(B, X)] = [G, 0]$$
$$\delta^1([B,1], Y) = [\delta(B, Y), \lambda(B, Y)] = [A, 1]$$

For state [E, 1] :

$$\delta^1([E,1], X) = [\delta(E, X), \lambda(E, X)] = [D, 0]$$
$$\delta^1([E,1], Y) = [\delta(E, Y), \lambda(E, Y)] = [A, 1]$$

For state [F, 1] :

$$\delta^1([F,1], X) = [\delta(F, X), \lambda(F, X)] = [E, 1]$$
$$\delta^1([F,1], Y) = [\delta(F, Y), \lambda(F, Y)] = [F, 1]$$

For state [G , 0] :

$$\delta^1([G,0], X) = [\delta(G, X), \lambda(G, X)] = [E, 1]$$
$$\delta^1([G,0], Y) = [\delta(G, Y), \lambda(G, Y)] = [G, 1]$$

For state [A, 1] :

$$\delta^1([A,1], X) = [\delta(A, X), \lambda(A, A)] = [F, 0]$$
$$\delta^1([A,1], Y) = [\delta(A, Y), \lambda(A, Y)] = [B, 1]$$

For state [D, 0] :

$$\delta^1([D,0], X) = [\delta(D, X), \lambda(D, X)] = [C, 0]$$
$$\delta^1([D,0], Y) = [\delta(D, Y), \lambda(D, Y)] = [B, 1]$$

For state [G, 1] :

$$\delta^1([G,1], X) = [\delta(G, X), \lambda(G, X)] = [E, 1]$$
$$\delta^1([G,1], Y) = [\delta(G, Y), \lambda(G_1, Y)] = [G, 1]$$

For state [C, 0] :

$$\delta^1([C,0], X) = [\delta(C, X), \lambda(C, X)] = [B, 0]$$
$$\delta^1([C,0], Y) = [\delta(C, Y), \lambda(C, X)] = [C, 1]$$

For state [B, 0] :

$$\delta^1([B,0], X) = [\delta(B, X), \lambda(B, X)] = [G, 0]$$
$$\delta^1([B,0], Y) = [\delta(B, Y), \lambda(B, Y)] = [A, 1]$$

For state [C, 1] :

$$\delta^1([C,1], X) = [\delta(C, X), \lambda(C, X)] = [B, 0]$$
$$\delta^1([C,1], Y) = [\delta(C, Y), \lambda(C, Y)] = [C, 1]$$

So, the transition table is – and the equivalent Moore machine is :

State	Input		Output
	X	Y	
[A, 0]	[F, 0]	[B, 1]	0
[F, 0]	[E, 1]	[F, 1]	0
[B, 1]	[G, 0]	[A, 1]	1
[E, 1]	[D, 0]	[A, 1]	1
[F, 1]	[E, 1]	[F, 1]	1

{G, 0}	{E, 1}	{G, 1}	0
{A, 1}	{F, 0}	{B, 1}	1
{D, 0}	{C, 0}	{B, 1}	0
{G, 1}	{E, 1}	{G, 1}	1
{C, 0}	{B, 0}	{C, 1}	0
{B, 0}	{G, 0}	{A, 1}	0
{C, 1}	{B, 0}	{A, 1}	1

Q. 2. Attempt any two questions :

Q. 2. (a) Construct DFA equivalent to the NFA : **10**

({p, q, r, s}, {0, 1}, δ, p, {s}), where δ is given by

	0	1
p	**p, q**	**p**
q	**r**	**r**
r	**s**	**–**
s	**s**	**s**

Ans. The given DFA M_1 is ({p, q, r, s}, {0, 1}, δ, p, {s}) and δ is

	0	1
p	p, q	p
q	r	r
r	s	–
s	s	s

Let the equivalent DFA be $M_2 = (Q^1, \Sigma^1, \delta^1, s^1, F^1)$

where Q^1 = set of all states in machine M_2

Σ^1 = input alphabet set for machine $M_2 = \Sigma = \{0, 1\}$

δ^1 = transition function for machine M_2 defined as : $\delta(Q^1, \Sigma^1) \rightarrow Q^1$

s^1 = start state for machine M_2 = start state of $M_1 = p$

F^1 – set of final states = will contain all those set of new states, which contain final state of M_1

The construct DFA, we have to find out the new transition function δ^1.

We will have to provide input on all new states in the transition table until we did not get repeated states.

So, start from initinal state

$\delta(p, 0) = \{p, q\}$ (new state)

$\delta(p, 1) = \{p\}$ (repeated state)

$\delta(\{p, q\}, 0) = (\delta(p, 0) \cup \delta(q, 0)) = (p, q) \cup (r) = (p, q, r)$ **(new state)**

$\delta(\{p, q\}, 1) = (\delta(p, 1) \cup \delta(q, 1)) = \{p\} \cup \{r\} = (p, r)$ **(new state)**

$\delta(\{p, q, r\}, 0) = (\delta(p, 0) \cup \delta(q, 0) \cup \delta(r, 0)) = (\{p, q\} \cup \{r\} \cup \{s\}) = \{p, q, r, s\}$ **(new state)**

$\delta(\{p, q, r\}, 1) = (\delta(p, 1), \cup \delta(q, 1)\, \delta(r, 1)) = \{p\} \cup \{r\} \cup \phi = (p, r)$ **(repeated state)**

$\delta(\{p, r\}, 0) = (\delta(p, 0), \cup \delta(r, 0)) = (\{p, q\} \cup \{s\}) = (p, q, s)$ **(new state)**

$\delta(\{p, r\}, 1) = (\delta(p, 1) \cup \delta(r, 1)) = (\{p\} \cup \phi) = (p)$ **(repeated state)**

$\delta(\{p, q, r, s\}, 0) = (\delta(p, 0), \cup \delta(q, 0) \cup \delta(r, 0) \cup \delta(s, 0)) = (\{p, q\} \cup \{r\} \cup \{s\} \cup \{s\})$ (p, q, r, s)
repeated state

$\delta(\{p, q, r, s\}\ 1) = (\delta(p, 1) \cup \delta(q, 1) \cup \delta(r, 1) \cup \delta(s, 1) = (\{p\} \cup \{r\} \cup \phi \cup \{s\}) = (p, r, s)$ **new state**

$\delta(\{p, q, r\}, 0) = (\delta(p, 0) \cup \delta(q, 0) \cup \delta(s, 0) = (\{p, q\} \cup \{r\} \cup \{s\}) = (p, q, r, s)$ **repeated state**

$\delta(\{p, q, s\}, 1) = (\delta(p, 1) \cup \delta(q, 1) \cup \delta(s, 1)) = (\{p\} \cup \{r\} \cup \{s\} = (p, r, s)$ **repeated state**

$\delta(\{p, r, s\}, 0) = (\delta(p, 0) \cup \delta(r, 0) \cup \delta(s, 0)) = (\{p, q\} \cup \{s\} \cup \{s\} = (p, q, s)$ **repeated state**

$\delta(\{p, r, s\}, 1) = (\delta(p, 1) \cup \delta(r, 1) \cup \delta(s, 1)) = (\{p\} \cup \{\phi\} \cup \{s\}) = (p, s)$ **(new state)**

$\delta(\{p, s\}\ 0) = (\delta(p, 0) \cup \delta(s, 0)) = (\{p\ q\} \cup \{s\}) = (p, q, s)$ **(repeated state)**

$\delta(\{p, s\}, 1) = (\delta(p, 1) \cup \delta(s, 1)) = (\{p\} \cup \{s\}) = (p, s)$ **repeated state**

Now no new state remian to explose so the desired DFA is : $M_2 = \{Q^1, \Sigma^1, \delta^1, s^1, F^1\}$
where $Q^1 = [(p)(p, q), (p, r), (p, s)(p, q, r), (p, q, s)(p, r, s)(p, q, r, s)]$

$\Sigma^1 = \{0, 1\}, s^1 = \{p\}, F^1 = \{(p, s), (p, q, s)(p, r, s), (p, q, r, s)\}$

and transition function δ^1 is defined by the table below :

	0	1
p	p, q	p
p, q	p, q, r	p, r
p, r	p, q, r, s	p
p, q, r	p, q, r, s	p, r
p, q, s	p, q, s	p, r, s
p, r, s	p, q, s	p, s
p, q, r, s	p, q, r, s	p, r, s
p, s	p, q, s	p, s

Q. 2. (b) Construct NFA for (a / b)$^+$ and derive DFA through subset construction algorithm **10**

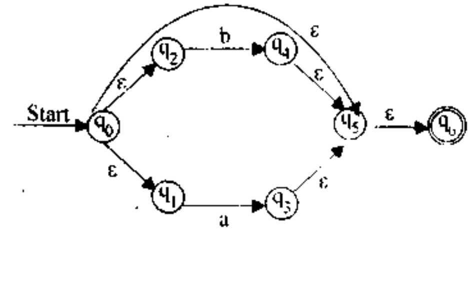
Ans. The given expression is $(a / b)^+$ the DFA for this statement will be :

Let the equivalent DFA be $M = (Q, \Sigma, \delta, s_0, F)$

Now $s_0 = \varepsilon$ clouser (starting state NFA) $= \varepsilon$ clouser (q_0)

= (states rechable from q_0 on input ε, including q_0)

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



Now $\delta(s_0, a) = \varepsilon$ clouser of $(\cup \delta(s_0, a))$

$= \varepsilon$ coluser of $(\delta(q_0, q_1, q_2), a) = \varepsilon$ coluser of

$(\delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a))$

$= \varepsilon$ closer of $(\phi \cup \phi_3 \cup \phi) = \varepsilon$ closer of $\{q_3\} = \{q_3, q_5, q_6, q_0, q_1, q_2\} = A$

$\delta(s_0, b) = \varepsilon$ coluser $(\cup \delta(s_0, b))$

$= \varepsilon$ clouser $(\delta(q_0, q_1, q_2), b) = \varepsilon$ coluser $(\delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_2, b))$

$= \varepsilon$ clouser $(\phi \cup \phi \cup q_4) = \varepsilon$ clouser $(q_4) = \{q_4, q_5, q_6, q_0, q_1, q_2\} = B$

Now we have two next states A and B so we have to define the possible transitions from these states and we continue the process until no new state remain to be considered for defining its transition.

$\delta(A, a) = \varepsilon$ clouser $(\cup \delta(A, a)) = \varepsilon$ clouser $(\cup \delta(q_0, q_1, q_2, q_3, q_5, q_6), a)$

$= \varepsilon$ clouser $(\delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a) \cup \delta(q_3, a) \cup \delta(q_5, a) \cup \delta(q_6, a))$

$= \varepsilon$ clouser $(\phi \cup q_3 \cup \phi \cup \phi \cup \phi \cup \phi) = \varepsilon$ clouser $(q_3) = A$

$\delta(A, b) = \varepsilon$ clouser $(\cup \delta(A, b)) = \varepsilon$ clouser $(\cup \delta \{q_0, q_1, q_2, q_3, q_5, q_6\}, b)$

$= \varepsilon$ clouser $(\delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_2, b) \cup \delta(q_3, b) \cup \delta(q_5, b) \cup \delta(q_6, b))$

$= \varepsilon$ clouser $(\phi \cup \phi \cup q_4 \cup \phi \cup \phi \cup \phi) = \varepsilon$ clouser $(q_4) = B$

$\delta(B, a) = \varepsilon$ clouser $(\cup \delta(B, a)) = \varepsilon$ clouser $(\cup \delta(q_0, q_1, q_2, q_4, q_5, q_6), a)$

$= \varepsilon$ clouser $(\delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a) \cup \delta(q_4, a) \cup \delta(q_5, a) \cup \delta(q_6, a))$

$= \varepsilon$ clouser $(\phi \cup \phi \cup q_3 \cup \phi \cup \phi \cup \phi) = \varepsilon$ clouser $(q_3) = A$

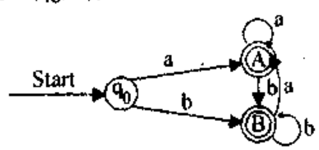
$\delta(B, b) = \varepsilon$ clouser $(\cup \delta(B, b)) = \varepsilon$ clouser $(\cup \delta \{\{q_0, q_1, q_2, q_4, q_5, q_6\} b)$

$= \varepsilon$ clouser $(\delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_2, b) \cup \delta(q_4, b) \cup \delta(q_5, b) \cup \delta(q_6, b))$

$= \varepsilon$ clouser $(\phi \cup \phi \cup q_4 \cup \phi \cup \phi \cup \phi) = \varepsilon$ clouser $= (q_4) = B$

So the DFA $M = (\{s_0, A, B\}, \{a, b\}, \delta \{s_0\}, \{A, B\})$

Transition Function 'δ'



	a	b
s_0	A	B
A	A	B
B	A	B

Q. 2. (c) Prove or disprove the following for regular expressions r, s and t **10**

(i) $(r + s)^* = r^* + s^*$

(ii) $s(rs + s)^* r = rr^* s(rr^* s)^*$

Ans. (i) $(r + s)^* = r^* + s^*$

L.H.S R.H.S

If LHS = RHS then it is true

If LHS ≠ RHS then it is false

The LHS of the question is of the form $(r + s)^*$, i.e., Kleen clouser of either r or s or combination of r and s. So by this representation we will get the following words :

$= \{\wedge, r, s, rr, rs, sr, ss, rrr, sss, ...\}$

= {all strings over r and s, in which either r or s may be first and either of them may be the last alphabet of the string}

The RHS of the question is of the form $(r^* + s^*)$. Means and frequency of r union with any frequency of s.

∴ $r^* + s^* = \{\wedge, r, rr, rrr, rrrr, ...\} \cup \{\wedge, s, ss, sss, ssss, ...\}$

$= \{\wedge, r, rr, rrr, rrrr, ... s, ss, sss, ssss, ...\}$

= {all strings over r or all strings over s}

∵ The string in LHS are not equal to the string in R.H.S. so the statement :

$$(r + s)^* = r^* + s^*$$

is false

(ii) : $(rs + s)^* r = rr^* s (rr^* s)^*$

LHS – $s (rs + s)^* r$ will represent all the strings of the form s ($\wedge$ **or** rs **or** s **or** srs **or** rss **or** ...) r

= (sr **or** $srsr$ **or** ssr **or** $ssrsr$ **or** $srssr$ ...)

RHS – $rr^* s (rr^* s)^*$ will represent all the strings of the form

r ($\wedge$ **or** r **or** rr **or** rrr) s ($\wedge$ or rs or $rrrs$) = (rs, rr, ...)

We can see that the strings represented by LHS will must start with 's' and all the strings represented by RHS must be start with 'r'. So the statement

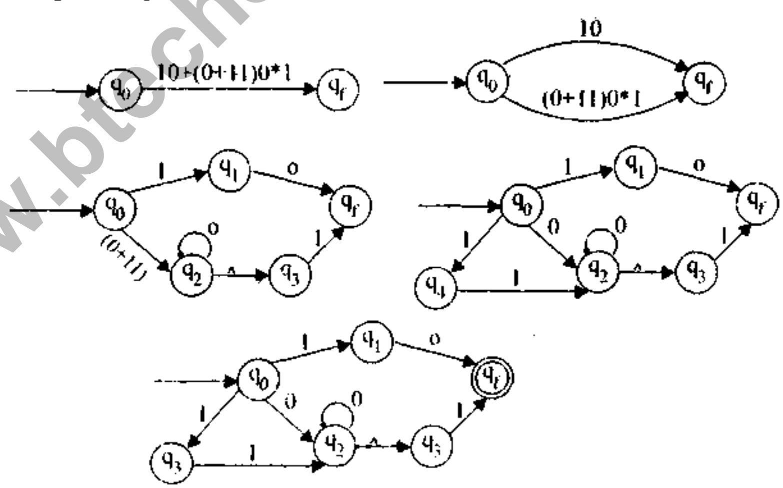
$$s (rs + s)^* r = rr^* s (rr^* s)^*$$

is **false**

Q. 3. Attempt any four questions :

Q. 3. (a) Construct finite automa equivalent to following regular expression : 10 + (0 + 11) 0* **5**

Ans. FA for regular expression 10 + (0 + 11) 0* 1



Q. 3. (b) Write regular expression for the following language over the alphabet {0, 1} : "The set of all strings not containing 101 as a substring". **5**

Ans. Regular expression for

"The set of all string not containing 101 as a substring is"

(0*1*00)*0*1*

Q. 3. (c) Explain the procedure to convent a Moore machine into its corresponding Mealy machine, with the help of an example. **5**

Ans. Procedure to convert a Moore Machine into its correspondin Mealy Machine.

Taking the example as follows

PS	NS		Output
	a = 0	a = 1 state	
→ q_0	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

This is a Moore Machine now for every input symbol we form the pair consisting of the next state and the corresponding output and reconstruct the table for Mealy Machine.

PS	NS			Output
	a = 0 state	Output	a = 1	
→ q_0	q_3	0	q_1	0
q_1	q_1	1	q_2	1
q_2	q_2	0	q_3	0
q_3	q_3	0	q_0	0

Q. 3. (d) Find parse tree for the expression abbcde considering the productions : **5**

S → a Ac Be

A → Ab

A → b

B → d

Ans. Given $S \longrightarrow aAcBe,\ A \longrightarrow Ab,\ A \longrightarrow b,\ B \longrightarrow d$

For the expression abbcde

The productions are

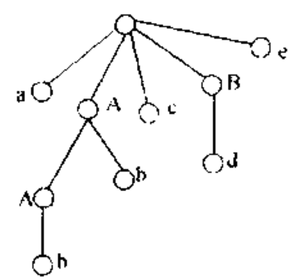
S ⇒ aAcBe

⇒ aAbcBe

⇒ abbcBe

⇒ abbcde

The parse tree is



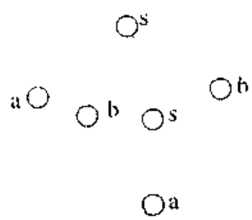
Q. 3. (e) What is ambiguous grammar ? Explain with example. **5**

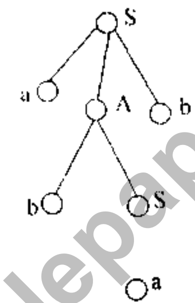
Ans. Ambiguous grammar : A terminal string $W \varepsilon L(G)$ is ambiguous if these exist two or more derivation trees for w and a context free grammar G is ambiguous if these exist some string $w \varepsilon L(G)$ which is ambiguous.

Taking the grammar

$$S \longrightarrow a|abSb|aAb$$

$$A \longrightarrow bS|aAAb$$





For the string abba, we have two differnt derivation as :

$S \Rightarrow abSb$ $\qquad$ $S \Rightarrow aAb$

$\Rightarrow abab$ $\qquad$ $\Rightarrow abSb$

$\Rightarrow abab$

Hence this grammar is ambiguous.

Q. 3. (f) Consider the grammar ({S, A, B}, {a, b}, P, S) that has the productions : **5**

S → bA/aB

A → bAA/aS/a

B → aBB/bS/b

Find an equivalent grammar in CNF.

Ans. As these are no un it productions or null productions, we proceed to step (2)

Step 2 : Let $G_1 = (V_N^1, \{a, b\}, P_1, S)$

Where V_N^1 and P_1 are constructed as follows :

(i) $A \longrightarrow a$, $B \longrightarrow b$ are addet to P_1

(ii) $S \longrightarrow bA / aB$ give rist to $S \longrightarrow C_b A / C_a aB$, $C_a \longrightarrow a$, $C_b \longrightarrow b$

$A \longrightarrow bAA / aS$ give rise to $A \longrightarrow C_b AA / C_a S$

$B \longrightarrow aBB / bS$ give rise to $B \longrightarrow C_a BB / C_b S$

$\therefore \quad V_N^1 = \{S, A, B, C_a, C_b\}$

Step 3. P_1 consists of

$$A \longrightarrow a, B \longrightarrow b, C_a \longrightarrow a, C_b \longrightarrow b, S \longrightarrow C_b A / C_a B$$

$$A \longrightarrow C_b AA / C_a S, B \longrightarrow C_a BB / C_b S$$

Let $G_2 = (V_N^{11}, \{a, b\}, P_2, S)$

where V_N^{11} and P_2 are constructed as :

$$A \longrightarrow C_b AA \text{ is replaced by } A \longrightarrow C_b C_1, C_1 \longrightarrow AA$$

$$B \longrightarrow C_a BB \text{ is replaced by } B \longrightarrow C_a C_2, C_2 \longrightarrow BB$$

The remaining productions in P_1 are added to P_2.

$\therefore$ $G_2 = (\{S, A, B, C_a, C_b, C_1, C_2\}, \{a, b\}, P_2, S)$

where P_2 consist of

$$S \longrightarrow C_b A / C_a B, A \longrightarrow C_b C_1 / C_a S / a, B \longrightarrow C_a C_2 / C_b S / b$$

$$C_1 \longrightarrow AA, C_2 \longrightarrow BB, C_a \longrightarrow a, C_b \longrightarrow b$$

Hence G_2 is in CNF.

Q. 4. Attempt any two questions :

Q. 4. (a) Define concept and working of a PDA. 10

Ans. PDA : A PDA is a 7 triple i.e.,

$$(Q, \Sigma, T, S, V_0, Z_0, F)$$

Where Q = is a finite non-empty set of states

Σ = is a finite non-empty set of input symbols

T = is a finite non-empty set of push down symbols

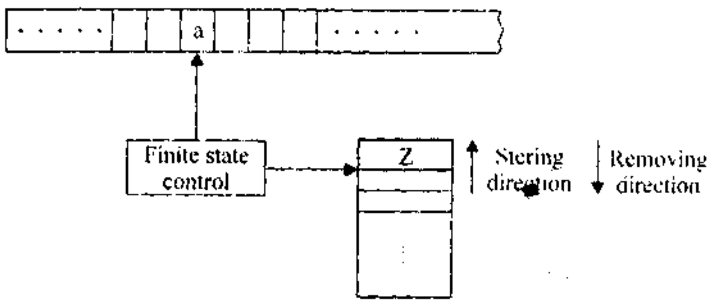
δ = is a transition functino from $Q\ X(\Sigma \cup \{\wedge\})XT$ to the set of finite subset of QXT^*.

q_0 = is caled the initial state

Z_0 = is a special symbol called initial symbol on PDS

F = is a set of final states

The model of a PDA is shown in figure.



Pushdown Store (PDS)

It has a read-only input tape, input alphabet, finite state control, a set of final states and an initial state. It has a stack called PDS. It is a read-write pushdown store as we add elements to PDS and removing elements from PDS. The FA is in seme state and on reading an input symbol moves to a new state. The PDA is also in some state and on reading an input symbol and the topmost symbol in PDS, moves to a new state and writes a strong of symbols in PDS.

Q. 4. (b) Construct a PDa equivalent to the following grammar : 10

S → aAA

A → aS/bS/a

Ans. Given $S \longrightarrow aAA$

$$A \longrightarrow aS / bS / a$$

The PDA equivalent to the above grammar is

$$A = (\{q\}, \{a, b\}, \{S, A, a, b\}, \delta, q, S, \phi)$$

where δ is defined by the following rules

$$\delta(q, \wedge, s) = \{(q, aAA)\}$$

$$\delta(q, \wedge, A) = \{(q, aS), (q, bS), (q, a)\}$$

$$\delta(q, a, a) = \{(q, \wedge)\}$$

$$\delta(q, b, b) = \{(q, \wedge)\}$$

Q. 4. (c) Construct a PDA accepting the language **10**

$\{a^i b^j c^k / i \neq j \text{ or } j \neq k\}$

Ans. We have to construct a PDA which accept the language defined by :

$$L \{a^i b^j c^k \mid i \neq j \text{ or } j \neq k\}$$

Step 1. Let we first defined the context free grammer, for the language L.

Let the language $KL = L_1, \cup L_2$ where $L_1 = \{a^i b^j c^k \mid i \neq j\}$

and $L_2 = \{a^i b^j c^k \mid j \neq k\}$

Let the grammers $G_1 = \{U_{n_1}, \{a, b, c\}, P_1, S_1\}$

and $G_2 = \{V_{n2}, \{a, b, c\}, P_2, S_2\}$

For language L_1 we have two conditions :

(i) $i > j$ and (ii) $j < k$

For language L_2 we have two conditions

(i) $j > k$ and (ii) $j < k$

Set of production rules for P_1 are as follows :

$$S_1 \longrightarrow A_1B_1, A_1 \longrightarrow C_1D_1 / E_1D_1, C_1 \longrightarrow aC_1 / a$$

$$D_1 \longrightarrow aD_1b / \varepsilon, E_1 \longrightarrow bE_1 / b$$

$$B_1 \longrightarrow cB_1 / \varepsilon$$

Set of production rules for P_2 are as follows :

$$S_2 \longrightarrow A_2B_2, A_2 \longrightarrow aA_2 / \varepsilon, B_2 \longrightarrow C_2D_2 / E_2D_2$$

$$C_2 \longrightarrow bC_2 / b, D_2 \longrightarrow bD_2c / \varepsilon, E_2 \longrightarrow cE_2 / c$$

Now let the desired grammer $G = \{V_{n1}, \{a, b, c\} P, S\}$ where P has the following production rules– $S \longrightarrow S_1 / S_2$, where S_1 and S_2 are the starting symbols of grammer G_1 and G_2 and all the production rule of G_1 and G_2 are also included in G.

Step 2. Now the PDA for the language (based on the production rules in the Grammer G) are as follows :

Let the corresponding PDA be

$M = (\{q\}, \{a, b, c\}, \{a, b, c, S, A_1, A_2, B_1, B_2, C_1, C_2, D_1, D_2, E_1, E_2\}, \delta, q, S, \phi)$

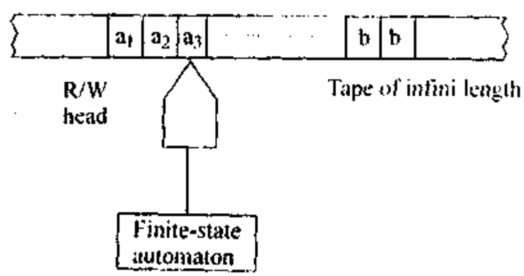
where transition function δ is defined as :

	Production Rules	Corresponding Transition
1.	$S \longrightarrow A_1 B_1$	$\delta(q, \varepsilon, S) = \{q, A, B_1\}$
2.	$S \longrightarrow A_2 B_2$	$\delta(q, \varepsilon, S) = \{q, A_2, B_2\}$
3.	$A_1 \longrightarrow C_1 D_1$	$\delta(q, \varepsilon, A_1) = \{q, C_1, D_1\}$
4.	$A_1 \longrightarrow E_1 D_1$	$\delta(q, \varepsilon, A_1) = \{q, E_1 D_1\}$
5.	$C_1 \longrightarrow aC_1$	$\delta(q, \varepsilon, C_1) = \{q, aC_1\}$
6.	$C_1 \longrightarrow a$	$\delta(q, \varepsilon, C_1) = \{q, a\}$
7.	$D_1 \longrightarrow aD_1 b$	$\delta(q, \varepsilon, D_1) = \{q, a, D_1 b\}$
8.	$D_1 \longrightarrow \varepsilon$	$\delta(q, \varepsilon, D_1) = \{q, \varepsilon\}$
9.	$E_1 \longrightarrow bE_1$	$\delta(q, \varepsilon, E_1) = \{q, bE_1\}$
10.	$E_1 \longrightarrow b$	$\delta(q, \varepsilon, E_1) = \{q, b\}$
11.	$B_1 \longrightarrow cB_1$	$\delta(q, \varepsilon, B_1) = \{q, CB_1\}$
12.	$B_1 \longrightarrow \varepsilon$	$\delta(q, \varepsilon, B_1) = \{q, \varepsilon\}$
13.	$A_2 \longrightarrow aA_2$	$\delta(q, \varepsilon, A_2) = \{q, aA_2\}$
14.	$A_2 \longrightarrow \varepsilon$	$\delta(q, \varepsilon, A_2) = \{q, \varepsilon\}$
15.	$B_2 \longrightarrow C_2 D_2$	$\delta(q, \varepsilon, B_2) = \{q, C_2 D_2\}$
16.	$B_2 \longrightarrow E_2 D_2$	$\delta(q, \varepsilon, B_2) = \{q, E_2\ D_2\}$
17.	$C_2 \longrightarrow bC_2$	$\delta(q, \varepsilon, C_2) = \{q, bC_2\}$
18.	$C_2 \longrightarrow b$	$\delta(q, \varepsilon, C_2) = \{q, b\}$
19.	$D_2 \longrightarrow bD_2 C$	$\delta(q, \varepsilon, D_2) = \{q, bD_2 C\}$
20.	$D_2 \longrightarrow \varepsilon$	$\delta(q, \varepsilon, D_2) = \{q, \varepsilon\}$
21.	$E_2 \longrightarrow CE_2$	$\delta(q, \varepsilon, E_2) = \{q, CE_2\}$
22.	$E_2 \longrightarrow C$	$\delta(q, \varepsilon, E_2) = \{q, C\}$

Q. 5. Attempt any four questions :

Q. 5. (a) Define the basic model of a Turing machine. 5

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 automaton to determine.

(i) a new symbol to be written ont he tape

(ii) motion of the R/W head either left (L) or right (R)

(iii) the next state

(iv) whether to half 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

$b \varepsilon T$ - 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) Explain the techniques for Turing machines construction. 5

Ans. Construction of Turing Machine : The basic guidelines for designing a turing machine are :

(i) The fundamental objective in scanning a symbol by R/W head is to know what to do in the future.

(ii) The machine must remember the past symbols scanned. The turing machine can remember this by going to the next unique state.

(iii) The number of states must be minimised this can be achieved by changing the states only when there is a change in the wirtten symbol or when there is a change in the movement of R/W head.

Q. 5. (c) Explain Church's thesis. 5

Ans. The assumption that the notation of computable function can be identified with the class of partial recursive functions is known as **Church's Thesis.** This thesis states that all possible models of computation, if they are sufficiently broad, must be equivalent. It also implies that there is an inherent limiation in this and that there are functions that cannot be expressed in any way to give an explicit method for their computation. The claim is of course very closely related to Turing's thesis and the combined notion is sometimes called Church-Turing thesis. It provides a general principle for algorithmic computation and, while not provable, gives strong evidence that no more powerful model can be found.

Q. 5. (d) Design Turing machine to compute the function 5

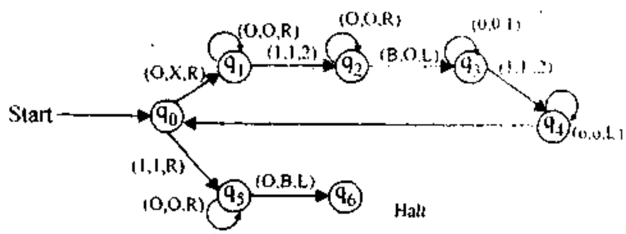
$f(n) = n^2$

Ans. The given function is

$$f(n) = n^2 \text{ or } f(n) = n * n$$

Means we have to find the multiplication of one number with itself.

Since the meaning of multiplication of two numbers x and y i.e., $(x * y)$ is the successive addition of x, y times. This is one of the computation logic which to determine the multiplcation of two numbers.



Initially take cells contains (n + 1) 0's corresponding to string x followed by a symbol 1 (breaker) and next to its symbol blanks. Now our task is to copy this block of (n + 1) 0's, n times.

Thus turing machine starts the computation over the string n and it copy the block according to the state diagram shown follows :

Q. 5. (e) Design Turing machine to recognize the language : 5

"Test set of strings with an equal no. of 0's and 1's."

Ans. Let the Turing Machine's take has infinite ends at both sides. So we will clearly take the advantage of infinite two ends of the input take. Let the input can be placed as follows :

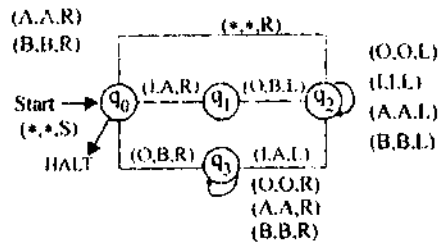
..	×	0	1	1	0	×	

We will have a simple logic as if we get 1 we will replace it by A and move right in search of 0, if we get 0 we will replace it by B and move left. We will move to the left most end (i.e., *) and again repeat the same process of marking 1 by A and 0 by B. We may have string starting with 0 or 1, so from initial state only we will have two separate paths shown for 1 and 0. Remember we have to check both the ends, to check wether the string is completely scanned or not. let us first understand the logic and then show the design of Turing Machine.

0110 Convert O in to B
↑
B110 Convert 1 into A and move towards left
↑
BA10 Skip B and move left
↑
BA10 Keep * as it is and move right
↑
BA10 Move right
↑
BA10 Move right
↑
BA10 Convert 1 to A and move right
↑
BAA0 Convert 0 to B and move left
↑
BAAAB Move left
↑
BAAB Move left
↑
BAAB Move left
↑
BAAB If * comes to right by ignoring all A's and B's
↑
BAAB Move right
↑
BAAB Move right
↑
BAAB Move right
↑
BAAB Move right
↑

BAAB   * is reached goto HALT state (*,*, R)
↑

Turing Machine



Q. 5. (f) Give recursive definitions for : n + m. **5**

Ans. The meaning of this question is that we are given a function $f(n, m) = n + m$ for every $n, m \in N$ (set of netural numbers).

To solve this question we have to make/design a Turing machine for $f(n, m)$.

We therefore wish to turn the situation

01 101 10 (n, m)

to the situation

01 11 10 (n, m)

(i.e., we wish to remove the 0 in middle). This can be achived if the Turing Machine designed to move towards the right, keeping is as is; stop when it encounters the digit 0, replacing it with a digit 1; move towards the left and replace the leftmost 1 by 0, move one position to the right; and then move on to the halting state. This can be achived by the table below :

	0	1
0	1L1	1R0
1	0R2	1L1
2		0R3

The same result can be achieved if the Turing Machine is designed to change the first 1 to 0, move to the right, keeping is as is, change the first 0 it encounters to 1, move to the left to position itself at the left most 1 remaining; and then move to the halting state. This can be achived by the table below :

	0	1
0		0R1
1	1L2	1L1
2	0R3	1L2