Fifth Semester B.E. Degree Examination, CBCS - Dec 2017 / Jan 2018

Automata Theory & Compatibility

Time: 3 hrs. Max. Marks: 80

Note : *Answer any FIVE full questions, selecting ONE full question from each module.*

Module - 1

1. a. Define the following terms with examples:
(i) Alphabet (ii) Power of an alphabet
(iii) Concatenation (iv) Languages **(04 Marks)**

Ans. i. **Alphabet :** A language consists of various symbol from which the words, statements etc, can be obtained. These symbols are called Alphabets. The symbol Σ denotes the set of alphabets of a language.

Ex : Σ = {a,b,......z, A,B,.......,Z,0.....9,#,C,),}....etc}

ii Power of an alphabet : If Σ is an alphabet, we can epress the set of all strings of a certain length from that alphabet by using the exponential notation.

Ex: Σ = {0,1} the

$\Sigma^1 = \{0,1\}, \Sigma^2 = \{00,01,10,11\}$

iii. Concatenation : The concatenation of two strings u and v is the string obtained by writing the letters of string u followed by the letters of string v.

$u = a_1a_2a_3.....a_n \quad v = b_1b_2b_3.....b_n$

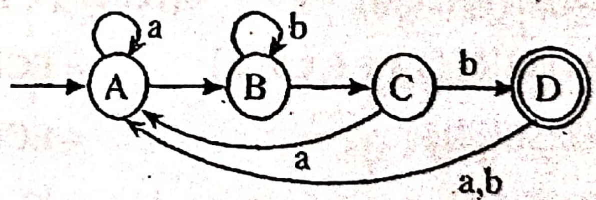
$uv = a_1a_2a_3.....a_nb_1b_2b_3.....b_n$

iv. Language : A language can be defined as a set of strings obtained from Σ* where Σ is set of alphabets of a particular language.

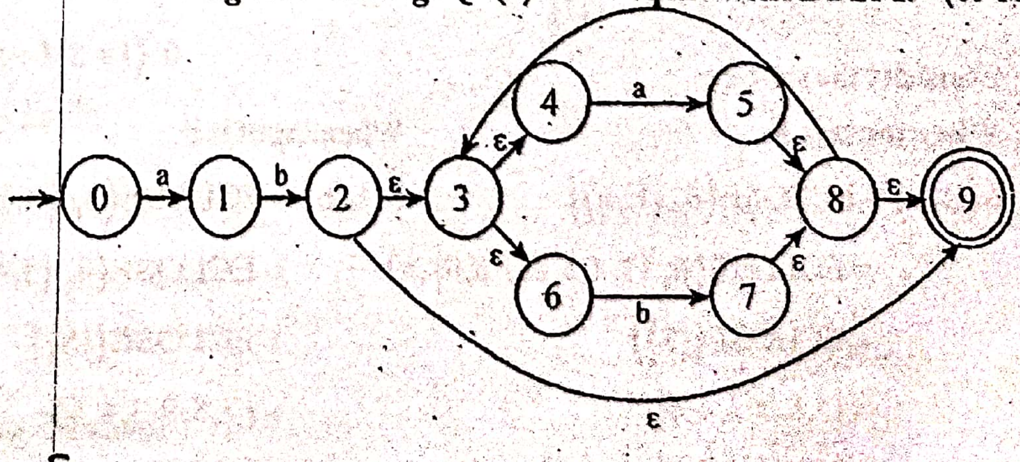
Ex : {ε,01,10,0011,1010,0101,0011,.....}

b. Draw a DFA to accept strings of a's and b's ending with 'bab'. **(03 Marks)**

Ans.



c. Convert the following NDFSM Fig. Ql (c) to its equivalent DFSM. **(09 Marks)**



Ans. Consider the state A :

When input is a :

$\delta(A,a) = ECLOSE(\delta_E(A,a))$

$= ECLOSE(\delta_E(0,a))$

$= \{1\} \to (B)$

When input is b :

$\delta(A,b) = ECLOSE(\delta_E(A,b))$

$= ECLOSe(\delta_E(0,b))$

$= \{\phi\}$

Consider the state B:

When input is a:

$\delta(B,a) = ECLOSE(\delta_E(B,a))$

$= ECLOSE(\delta_E(1,a))$

$= \phi$

When input is b

$\delta(B,b) = ECLOSE(\delta_E(B,b))$

$= ECLOSE(\delta_E(1,b))$

$= ECLOSE(\{2\})$

$= \{2,3,4,6,9\} \to (6)$

Consider the state C :

When input is a :

$\delta(c,a) = ECLOSE(\delta_E(c.a))$

$= ECLOSE(\delta_E\{2.3.4.6,9\}.a)$

$= ECLOSE(5)$

$= \{5,8,9,3,4,6\}$

$= \{3,4,5,6.8,9\} \to (D)$

When input is b :

$\delta(A,b) = ECLOSE(\delta E(A,b))$

$= ECLOS(\delta_E\{2,3.4,6,9\},b)$

$= ECLOSE(\{7\})$

$= \{7,8,9,3,4,6\}$

$= \{3.4,6,7,8,9\} \to (E)$

Consider the state D:

When input is a:

$\delta(D,a) = ECLOSE(\delta_E(D,a))$

$= ECLOSE(\delta_E\{3,4,5.6,8,9\},a)$

$= ECLOSE(\{7\})$

$= \{7,8,9,3,4,6\}$

$= \{3,4,6,5,8,9\} \to (D)$

When input is b

$\delta(B,b) = ECLOSE(\delta_E(B,b))$

$= ECLOSE(\delta E\{3,4,6,8,5,9\},b)$

$= ECLOSE(7,8,9,3,4,6)$

$= \{3,4,6,7,8,9\} \to (E)$

Consider the state E:

When input is a:

$\delta(E,a) = ECLOSE(\delta_E(E,a))$

$= ECLOSE(\delta_E\{3.4,5,6,7,8,9\},a)$

$= ECLOSE(\{S\})$

$= \{5,8,9,3.4,6\}$

$= \{3.4,5,6,8,9\} \to (D)$

When input is b

$\delta(E,b) = ECLOSE(\delta_E(B,b))$

$= ECLOSE(\delta_E\{3,4,6,7,8,5,9\},b)$

$= ECLOSE(\{S\})$

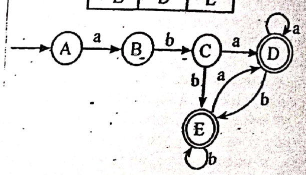
$= (7,8,9,3,4,6)$

$= \{3.4,6,7,8.9\} \to (E)$

Since no new state, will stop

Since no new state, will stop

δ	a	b
A	B	ϕ
B	ϕ	C
*C	D	E
*D	D	E
*E	D	E

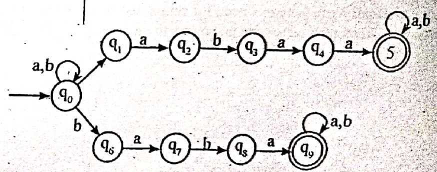


OR

2. a. Draw a DFSM to accept the language,

$1 = \{\omega \in \{a, b\}^* : \forall x, y \in \{a, b\}^* ((\omega = x\ abbaay) \vee (\omega = x\ babay))\}$ (03 Marks)

Ans.



b. Define distinguishable and indistinguishable states. Minimize the following DFSM,

S	0	1
A	B	A
B	A	C
C	D	B
*D	D	A
E	D	F
F	G	E
G	F	G
H	G	D

(i) **Draw the table of distinguishable and indistinguishable state for the automata.**
(ii) **Construct minimum state equivalent of automata.** (09 Marks)

Ans. Refer Q.no.4(a) of MQP - 2.

c. **Write differences between DFA, NFA and e -NFA.** (04 Marks)

Ans:

DFA	NFA	ε - NFA
1. The DFA is S-tuple $M = (Q,\Sigma,\delta,q_0,F)$ where Q is finite states Σ is set of input $\delta : Q \times \Sigma$ to Q q_0 is the start state FCQ is set of final state	An NFA is 5 - tuple $(\Sigma ; Q, \delta, q_0, F) = M$ Where Q is firite states Σ is set of input $\delta : Q \times \Sigma$ to 2^Q q_0 is the sturt state FCQ is final state	Am ε - NFA is 5 - tuple $M = (Q,\Sigma,\delta,q_0,F)$ where Q is finite states Σ is set of input $\delta : Q \times (\Sigma \cup Q)$ to Z^Q q_0 is th start state $F \leq Q$ is final state
2. There can be zero or one transition from a state on an input symbol	There can be zero, one or more transitions from a state on input symbol	There can be zero , one or more transition from state with or without any input symbol
3. More number of transition	Less number of transition	Relatively more transition when compared with NFA
4. Difficult to construct	Easy to construct	Easy to construct using regular expression..

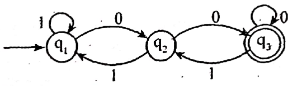
Module-2

3. a. **Consider the DFA shown below:**

State	0	1
$\rightarrow q_1$	q_2	q_1
q_2	q_3	q_1
$*q_3$	q_3	q_2

Obtain the regular expressions $R_{ij}^{(0)}$, $R_{ij}^{(1)}$ and simplify the regular expressions as much as possible. (09 Marks)

Ans.



Basic when k = 0

$R_{11}^{(0)} = \varepsilon + 1$

$R_{12}^{(0)} = 0$

$R_{13}^{(0)} = \phi$

$R_{21}^{(0)} = 1$

$R_{22}^{(0)} = \phi + \varepsilon = \varepsilon$

$R_{23}^{(0)} = 0$

$R_{31}^{(0)} = \phi$

$R_{32}^{(0)} = 1$

$R_{33}^{(0)} = \varepsilon + 0$

Induction: $R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)}\left[R_{kk}^{(k-1)}\right]^* R_{kj}^{(k-1)}$

When k = 1

$R_{11}^{(1)} = R_{11}^{(0)} + R_{11}^{(0)}\left[R_{11}^{(0)}\right]^* R_{11}^{(0)}$

$= (\varepsilon+1)(\varepsilon+1)(\varepsilon+1)^*(\varepsilon+1)$

$= (\varepsilon+1) + (\varepsilon+1)1^*(\varepsilon+1)$

$= (\varepsilon+1) + 1^*$

$= 1^*$

$R_{12}^{(1)} = R_{12}^{(0)} + R_{11}^{(0)}\left[R_{11}^{(0)}\right]^* R_{12}^{(0)}$

$= 0 + (\varepsilon+1)(\varepsilon+1)^* 0$

$= 0 + 1^*0$

$= 1^*0$

$R_{13}^{(1)} = R_{13}^{(0)} + R_{11}^{(0)}\left[R_{11}^{(0)}\right]^* R_{13}^{(0)}$

$= \phi + (\varepsilon+1)^*(\varepsilon+1)\phi$

$= \phi$

$R_{21}^{(1)} = R_{21}^{(0)} + R_{21}^{(0)}\left[R_{11}^{(0)}\right]^* R_{11}^{(0)}$

$= 1 + 1(\varepsilon+1)^*(\varepsilon+1)$

$= 1 + 11^* = 11^*$

$R_{22}^{(1)} = R_{22}^{(0)} + R_{21}^{(0)}\left[R_{11}^{(0)}\right]^* R_{12}^{(0)}$

$= \varepsilon + 1(\varepsilon+1)^* 0$

$= \varepsilon + 11^*0$

$R_{23}^{(1)} = R_{23}^{(0)} + R_{21}^{(0)}\left[R_{11}^{(0)}\right]^* R_{13}^{(0)}$

$= 0 + 1(\varepsilon+1)^*\phi$

$= 0$

$R_{31}^{(1)} = R_{31}^{(0)} + R_{31}^{(0)}\left[R_{11}^{(0)}\right]^* R_{11}^{(0)}$

$= \phi + \phi(\varepsilon+1)^*(\varepsilon+1)$

$= \phi$

$R_{32}^{(1)} = R_{32}^{(0)} + R_{31}^{(0)}\left[R_{11}^{(0)}\right]^* R_{12}^{(0)}$

$= 1 + \phi(\varepsilon+1)^* 0$

$= 1$

$R_{33}^{(1)} = R_{33}^{(0)} + R_{31}^{(0)}\left[R_{11}^{(0)}\right]^* R_{13}^{(0)}$

$= (\varepsilon+0) + \phi(\varepsilon+0)^*\phi$

$= (\varepsilon+0)$

When k=2

$$R_{11}^{(2)} = R_{11}^{(1)} + R_{12}^{(1)}\left[R_{22}^{(1)}\right]^{*} R_{21}^{(1)}$$

$$= 1^{*} + 1^{*}0(\varepsilon + 11^{*}0)^{*}11^{*}$$

$$= 1^{*} + 1^{*}0(11^{*}0)^{*}11^{*}$$

$$R_{12}^{(2)} = R_{12}^{(1)} + R_{12}^{(1)}\left[R_{22}^{(1)}\right]^{*} R_{22}^{(1)}$$

$$= 1^{*}0 + 1^{*}0(\varepsilon + 11^{*}0)^{*}(\varepsilon + 11^{*}0)$$

$$= 1^{*}0 + 1^{*}0(11^{*}0)^{*}(\varepsilon + 11^{*}0)$$

$$R_{13}^{(2)} = R_{13}^{(1)} + R_{12}^{(1)}\left[R_{22}^{(1)}\right]^{*} R_{23}^{(1)}$$

$$= \phi + 1^{*}0(\varepsilon + 11^{*}0)^{*}0$$

$$= (0 + \varepsilon) + 1(11^{*}0)^{*}0$$

$$R_{21}^{(2)} = R_{21}^{(1)} + R_{22}^{(1)}\left[R_{22}^{(1)}\right]^{*} R_{21}^{(1)}$$

$$= 11^{*} + (\varepsilon + 11^{*}0)(\varepsilon + 11^{*}0)^{*}11^{*}$$

$$= 11^{*} + (\varepsilon + 11^{*}0)(11^{*}0)11^{*}$$

$$R_{22}^{(2)} = R_{22}^{(1)} + R_{22}^{(1)}\left[R_{22}^{(1)}\right]^{*} R_{22}^{(1)}$$

$$= (\varepsilon + 11^{*}0) + (\varepsilon + 11^{*}0)(\varepsilon + 11^{*}0)^{*}(\varepsilon + 11^{*}0)$$

$$= (\varepsilon + 11^{*}0) + (\varepsilon + 11^{*}0)(11^{*}0)(\varepsilon + 11^{*}0)$$

$$R_{31}^{(2)} = R_{31}^{(1)} + R_{32}^{(1)}\left[R_{22}^{(1)}\right]^{*} R_{21}^{(1)}$$

$$= \phi + 1(\varepsilon + 11^{*}0)^{*}11^{*}$$

$$= 1(11^{*}0)^{*}11^{*}$$

$$R_{32}^{(2)} = R_{32}^{(1)} + R_{32}^{(1)}\left[R_{22}^{(1)}\right]^{*} R_{22}^{(1)}$$

$$= 1 + 1(\varepsilon + 11^{*}0)^{*}(\varepsilon + 11^{*}0)$$

$$= 1 + 1(11^{*}0)^{*}(\varepsilon + 11^{*}0)$$

$$R_{33}^{(2)} = R_{33}^{(1)} + R_{32}^{(1)}\left[R_{22}^{(1)}\right]^{*} R_{23}^{(1)}$$

$$= 1^{*}0(11^{*}0)^{*}0$$

Final RE can be calculated as

$$R_{13}^{(3)} = R_{13}^{(2)} + R_{13}^{(2)}\left[R_{33}^{(2)}\right]^{*} R_{33}^{(2)}$$

$$= 1^{*}0(11^{*}0)^{*}0 + 1^{*}0(11^{*}0)^{*}0\left[(0+\varepsilon) + 1(11^{*}0)^{*}0\right]^{*}(0+\varepsilon) + 1(11^{*}0)^{*}0$$

b. Give Regular expressions for the following languages on /_= {a,b,c}
(i) all strings containing exactly one a
(ii) all strings containing no more than 3 a's.
(iii) all strings that contain at least one occurance of each symbol in V.
(03 Marks)

Ans.

(i) $R\,E = (b+c)^{*}a(b+c)^{*}$

(ii) $R\,E = (b+c)^{*}(\sigma+a)(b+c)^{*}(\varepsilon+a)(b+c)^{*}$

(iii) $(a+b+c)^{*}$

3. c. Let L be the language accepted by the following finite state machine. (04 Marks)

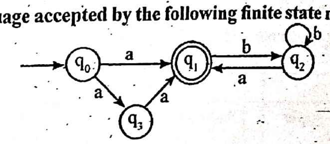


Fig. Q3 (c;

Indicate for each of the following regular expressions, whether it correctly describes L:
(i) (a U ba) bb* a
(ii) (ε U b) a (bb* a)*
(iii) ba U ab*a
(iv) (a U ba) (bb*a)*

Ans. i. NO
ii. YES
iii. NO
iv. YES

OR

4. a. Prove that the following language in not regular :
$L = \{0^n 1^n \mid \eta > 0\}$ **(05 Marks)**

Ans. **Step 1 :** Let L i.e, regular and η be the number of states
$x = 0^n b^n$

Step 2 : Since $|x| = 2\eta > \eta$ we can split x into uvw such that $|uv| \le \eta$ and $|V| \ge 1$ as

$$x = \overbrace{\underbrace{aaaaaa}_{u}\ \underbrace{a}_{V}}^{n}\ \underbrace{\overbrace{bbbbbbbb}^{n}}_{w}$$

Step 3 : According to pumping lemma $uv^i\omega \in L$ for $i = 0,1,2.......$
When $i = 0$ 'V' doesn't exist so $L = \{0^n 1^n \mid n > 0\}$ is not regular

b. If L_1 and L_2 are regular languages then prove that $L_1 U L_2$, $L_1 . L_2$ and L_1^* are regular languages. **(05 Marks)**

Ans. Refer Q.no.3(b) of MQP - 2.

c. Is the following grammar is ambiguous? (06 Marks)
S → i C ts|ict ses|a
C → b
(06 Marks)

Ans. Refer Q.no.5(b) of MQP - 2.

Module-3

5. a. Define Grammar, Derivation, Sentential forms and give one example for each. **(03 Marks)**

Ans. A grammer G is 4 tuple or quadruple. G = (V,T,P,S) where 'V' is variable , T is terminals, P is production and 'S' is start symbol.

Ex: $S \to \varepsilon$, $S \to aS$

$A \Rightarrow \alpha \beta r$, the process of obtaining strings of terminal s and / or non - terminals from the start symbol by applying some or all productions is called derivation.

$E \Rightarrow E + E$, $E \Rightarrow id + E$, $E \Rightarrow id + id$

Let G = (V,T,P,S) be a grammar. The string ω obtained from the grammar G such that $S \Rightarrow w$ is called sentence of grammar G. Here, w is the string of terminals.

b. What is CNF? Obtain the following grammar in CNF **(09 Marks)**

$S \to ASB \mid \varepsilon$

$A \to aAS \mid a$

$B \to SbS \mid A \mid bb$

Ans. Let G = (V,T,P,S) be a CFG. The grammr G is said to be in CNF if all productions are of the form.

$A \to BC$ or $A \to a$

Eliminate ε - production

0v	nv	Production
ϕ	$S \to \varepsilon$	$S \to \varepsilon$
S	S	–

V = {A,B} are nullable variables

Production	Resulting production (p')
$S \to ASB$	$S \to AB$
$A \to aAS$	$A \to aA \mid a$
$B \to SbS$	$B \to SbS \mid bS \mid Sb \mid b \mid A \mid bb$

Given Production	Action	
$S \to AB$	Already in CNF	$S \to AB$
$A \to aA$	Replace a by A_0 $A_0 \to A$	$A \to A_0A$ $A_0 \to a$ $A \to a$
$B \to SbS \mid bS \mid Sb \mid A$	Replace b by B_0 $B_0 \to b$	$B \to SB_0S \mid B_0S \mid SB_0 \mid a$

Replace B_0S with B_1

$B \to SB_1$

$B_1 \to B_0S$

Sunstar Exam Scanner

$G' = (V', T, P', S)$ 'S' is start symbol

$V = (S, A, B, B_0, B_1, A_0)$

P = {
$S \to AB$
$A \to A_0A$
$A_0 \to a$
$A \to a$
$B \to SB_1 \mid B_0S \mid SB_0 \mid a \mid$
$B_0 \to b$
$B_1 \to B_0S$
}

c. Let G be the grammar,

$S \to aB \mid bA$

$A \to a \mid aS \mid bAA$

$B \to b \mid bS \mid aBB$

For the string qaabbabbba find a

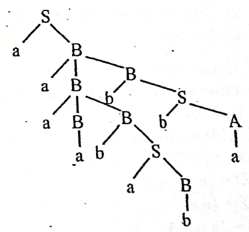
(i) Left most derivation.

(ii) Right most derivation.

(iii) Parse tree. **(04 Marks)**

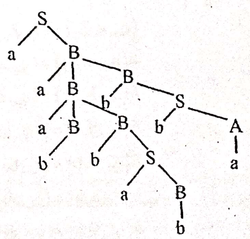
Ans. i. Let most derivation

$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aaaBBB$
$\Rightarrow aaaabbSB$
$\Rightarrow aaaabbaBB$
$\Rightarrow aaaabbabB$
$\Rightarrow aaaabbabbS$
$\Rightarrow aaaabbabbbA$
$\Rightarrow aaaabbabbba$



ii. Right most derivation

$S \Rightarrow aB$
$\Rightarrow aaBB$
$\Rightarrow aaaBbS$
$\Rightarrow aaaBbbA$
$\Rightarrow aaaBbba$
$\Rightarrow aaaaBBbba$
$\Rightarrow aaaaBbSbba$
$\Rightarrow aaaBbaBbba$
$\Rightarrow aaaaBbabbba$
$\Rightarrow aaaabbabbba$



Sunstar Exam Scanner

OR

6. a. Explain the following terms:
(i) Pushdown automata (PDA).
(ii) Languages of a PDA.
(iii) Instantaneous description of a PDA. **(03 Marks)**

Ans. i. Pushdown Automata (PDA) : A PDA is a seven tuple
$M = (Q, \Sigma, |, \delta, q_0, Z_0, F)$
Q is set of finite states
Σ is set of input alphabets
| is set of stack alphabets
δ is transition $Q \times (\Sigma \cup \varepsilon) \times | - Q \times |^*$
$q_0 \varepsilon Q$ is start state
$Z_0 \varepsilon |$ is initial symbol on stack
$F \leq Q$ is set of final state

ii. Language of PDA : The language LCM accepted by a final state is defined as
$L(M) = \{w \mid (q_0, w, Z_0) \mid {}^*(P, \varepsilon, \alpha)\}$

iii. Instantaneous description : Let $M = (Q, \Sigma, |, \delta, q_0, Z_0, F)$ be a PDA. An ID (instantaneous description) is defined as 3-tuple or a triple (q, w, α)

b. Construct a PDA to accept the language $L = \{ww^R \mid w\varepsilon\{a,b\}^*\}$. Draw the graphical representation of this PDA. Show the moves made by this PDA for the string aabbaa. **(10 Marks)**

Ans. $L(M) = \{ww^R \mid w\varepsilon\{a,b\}^*\}$
$M = (Q, \Sigma, |, \delta, q_0, Z_0, F)$
$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$| = \{a, b, Z_0\}$
δ : {

$$(q_0, \varepsilon, Z_0) = (q_1, Z_0)$$
$$\delta(q_0, a, Z_0) = (q_0, aZ_0)$$
$$\delta(q_0, b, Z_0) = (q_0, bZ_0)$$
$$\delta(q_0, b, a) = \{(q_0, aa), (q, \varepsilon)\}$$
$$\delta(q_0, a, b) = (q_0, ba)$$
$$\delta(q_0, a, b) = (q_0, ab)$$
$$\delta(q_0, a, b) = (q_0, ab)$$
$$\delta(q_0, b, b) = \{(q_0, bb), (q, \varepsilon)\}$$
$$\delta(q_1, a, a) = (q_1, \varepsilon)$$
$$\delta(q_1, b, b) = (q_1, \varepsilon)$$
$$\delta(q_1, \varepsilon, Z_0) = (q_0, Z_0)$$

}

$q_0 \in Q$ is start state
$Z_0 \in P$ is initial stack symbol
$F = \{q_2\}$ is final state

$(q_0, aabbaa, z_0)$
↓
$(q_0, abbaa, az_0) \longrightarrow (q_0, aabbaa, z_0) \longrightarrow (q_2, aabbaa, z_0)$
↓
$(q_0, bbaa, aaz_0) \longrightarrow (q_1, bbaa, z_0)$ ⟶ [illegible]
↓
$(q_0, baa, baaz_0) \longrightarrow (q_0, aa, bbaaz_0)$ ⟶ [illegible]
↓
(q_1, aa, aaz_0)
↓
(q_1, a, az_0)
↓
$(q_1, \varepsilon, z_0) \rightarrow (q_2, \varepsilon, z_0)$

c. Convert the following CFG to PDA
$S \rightarrow aABB \mid aAA$
$A \rightarrow aBB \mid a$
$B \rightarrow bBB \mid A$
$C \rightarrow a$ (03 Marks)

Ans. $Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$\Gamma = \{S, A, B, C_1, Z_0\}$
δ : {

$$\delta(q_0, \varepsilon, Z_0) = (q_1, SZ_0)$$
$$\delta(q_1, a, S) = (q_1, ABB)$$
$$\delta(q_1, a, S) = (q_1, AA)$$
$$\delta(q_1, a, A) = (q_1, BB)$$
$$\delta(q_1, a, A) = (q_1, \varepsilon)$$
$$\delta(q_1, b, B) = (q_1, BB)$$
$$\delta(q_1, a, B) = (q_1, BB)$$
$$\delta(q_1, a, B) = (q_1, \varepsilon)$$
$$\delta(q_1, a, c) = (q_1, \varepsilon)$$
$$\delta(q_1, \varepsilon, Z_0) = (q_f, Z_0)$$

}

$q_0 \in Q$ is start state
$Z_0 \in T$ is stack symbol
$F = \{q_f\}$ is final state

Module-4

7. a If L_1 and L_2 are context free languages then prove that $L_1 \cup L_2$, $L_1 L_2$ and L_1^*, are context free languages. (04 Marks)

Ans.

(i) $G_1 = (V_1, T_1, P_1, S_1)$

$G_2 = (V_2, T_2, P_2, S_2)$

$G_3 = (V_1 \cup V_2 \cup S_3, T_1 \cup T_2, P_3, S_3)$

S_3 is a start state G_3 and $S_3 \in (V_1 \cup V_2)$

$P_3 = P_1 \cup P_2 \cup \{S_3 \to S_1 / S_2\}$

$L_3 = L_1 \cup L_2$

(ii) $G_4 = (V_1 \cup V_2 \cup S_4, T_1 \cup T_2, P_4, S_4)$

S_4 is a start symbol for the grammar G_4 and $S_4 \in (V_1 \cup V_2)$

$P_4 = P_1 \cup P_2 \cup \{S_4 \to S_1 S_2\}$

$\therefore L_3 = L_1 . L_2$

(iii) $G_5 = (V, US_5, T_1, P_5, S_5)$

S_5 is a the start symbol of Grammar G_5

$P_5 = P_1 \cup \{S_5 \to S_1 S_5 | \varepsilon\}$

$TL_5 = L_5^*$

b. **Give a decision procedure to answer each of the following questions:**
(i) Given a regular expression a and a PDA M, the language accepted by M a subset of the language generated by a?
(ii) Given a context-free Grammar G and two strings Si and S2, does G generate S_1S_2?
(iii) Given a context free Grammar G, does G generate any even length strings.
(iv) Given a Regular Grammar G, is L(G) context-free? (12 Marks)

Ans. i. Observe that this is true if $\perp(M) \cap L(\alpha) = \phi$. So the following procedure answers the question :

1. From α, build a PDA M* so that L(M*) = L(α)
2. From M and M*, build a PDA M** that accepts L(M) ∩ L(M*)
3. If L(M**) is empty, return true else return false.

ii. 1. Convert G to chomsky normal forms2. Try all derivations in G of length up to $2|S_1S_2|$. If any of them generates S_1S_2 return True, else return false

iii. 1. Use CFG to PDA topolown (G) to build a PDA P hta taccepts L(G).
2. Build an FSM E that accepts all even length strings over the alphabet Σ_G.
3. Use insert PDA and FSM(P,E) to build a PDA P* that accepts L(G) ∩ L(E).
4. Return decioleCFLempty(P*)

iv. i. Return True (Since every regular language is context free)

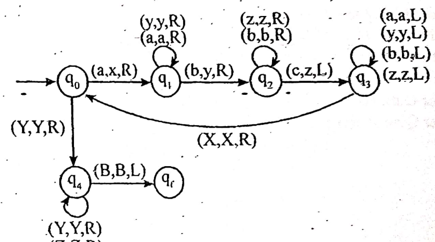
OR

8. a. **Explain with neat diagram, the working of a Turing Machine model. (05 Marks)**

Ans. Refer Q.no.9(a) of MQP - 2.

b. **Design a Turing machine to accept the language L = {$a^n b^n c^n$ | n >= 1}. Draw the transition diagram. Show the moves made by this turing machine for the string aabbcc.** (11 Marks)

Ans.



aabbcc
xabbcc
xabbcc
xaybcc
xaybcc
xaybzc
xxybzc
xxyyzc
xxyyzz

Module-5

9 Write short notes on:
a. **Multi-tape turning machine.**
b. **Non-deterministic turning machine.**
c. **Linear Bounded automata.** (16 Marks)

Ans. a. Refer Q.no. 9(b) of MQP - 1.

b. **Non - deterministic turning machine :** In a non - deterministic turning machine, for every state and symbol, there are a group of actions the TM can have. So here the transitions are not deterministic. The computation of a non - deterministic turning machine is a tree of configurations that can be reached from the start configuration. An input is accepted if there is at least one node of the tree which is an accept configuration, otherwise it is not accepted. If all branches of the computational tree hact on all inputs, the non - deterministic turning machine is called a decideg and if for some input, all branches are rejected, the input is also rejected.

c. Linear bounded automata : Refer Q.no. 10(b) of MQP - 1.

OR

10. Write short notes on:

a. Undecidable languages.

b. Halting problem of turning machine.

c. The post correspondence problem. **(16 Marks)**

Ans. a. Refer Q.no.9(b) of MQP - 2

b. Refer Q.no.10(a) of MQP - 1

c. Refer Q.no.10b(i) of MQP - 2.