USN

CMR INSTITUTE OF TECHNOLOGY, BENGALURU. ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 1 – Nov 2022

Sub:	Automata Theory and Computability					Sub Code:	18CS54	Branch:	CSE
Date:	5/11/2022	Duration:	90 mins	Max Marks:	50	Sem / Sec:	5 A,B,C		OBE

Answer any FIVE FULL Questions — MARKS | CO | RBT

1 (a) Define the following with examples : [04] CO1 L1

i) Alphabet ii) Language

Alphabet denoted by Σ is a finite set. The members of Σ are called symbols or characters.

Eg. English Alphabet Σ= {a, b, c,...,z}

Binary Alphabet Σ= {0,1}

Alphabet of digits Σ = {0,1,2,3,4,5,6,7,8,9}

Language: A language (finite/infinite) is a set of strings over a given alphabet, Σ. If there is more than one language, we will use ΣL to denote alphabets from which language L is formed.

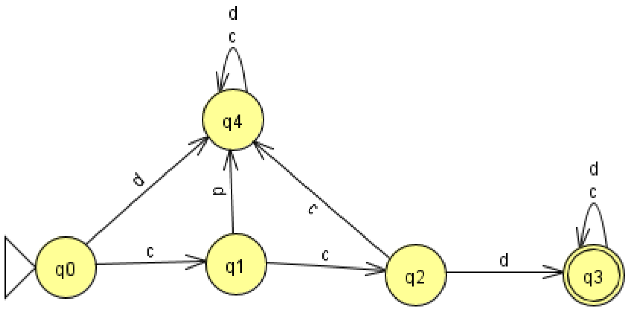
Eg.

L = {w|∈{0,1}*

:w begins and ends in a and |w| >=2}

Strings that belong to this language in lexicographic order are {aa,aaa, aba, aaaa, abaa, aaba,...}

(b) Design a DFSM for {w∈{c,d}*: w begins with ccd}. Write the definition. Show computation for w = ccddc and w=cdc and state whether it is an accepting or rejecting configuration. [06] CO1 L3



Definition

M= $(K, \Sigma, \delta, s, A)$

M= $(\{q_0, q_1, q_2, q_3, q_4\}, \{c, d\}, \delta, q_0, \{q_3\})$

Transition Table

δ	c	d
$\rightarrow q_0$	q_1	q_4
q_1	q_2	q_4
q_2	q_4	q_3
$* q_3$	q_3	q_3

Computation

$(q_0, ccddc) \vdash (q_1, cddc) \vdash (q_2, ddc)$
$\vdash (q_3, dc) \vdash (q_3, c) \vdash (q_3, \varepsilon)$

$(q_0, ccddc) *$
$\vdash (q_3, \varepsilon)$ *is an accepting configuration as* q_3
$\in A$ *of DFSM M*

$(q_0, cdc) \vdash (q_1, dc) \vdash (q_4, c) \vdash (q_4, \varepsilon)$

q_4	q_4	q_4

$(q_0, cdc) *$
$\vdash (q_4, \varepsilon)$ *is a rejecting configuration as* q_4
$\notin A$ *of DFSM M*

2 (a) Define the following with examples : 04] CO1 L1

i) String ii)Power of an alphabet

String: A finite Sequence, possibly empty, of symbols drawn from some alphabet Σ. Given any alphabet, the shortest string is ε. Σ* is the set of all possible strings over an alphabet Σ.

Example:

English Alphabet {a, b, c,...,z} Strings : {sat, laugh, happy}

Binary Alphabet {0,1} Strings: {011, 111, 1000, 0110}

ii) Power of an alphabet

the set of all strings can be expressed as a certain length from that alphabet by using exponential notation. The power of an alphabet is denoted by Σ^k and is the set of strings of length k.

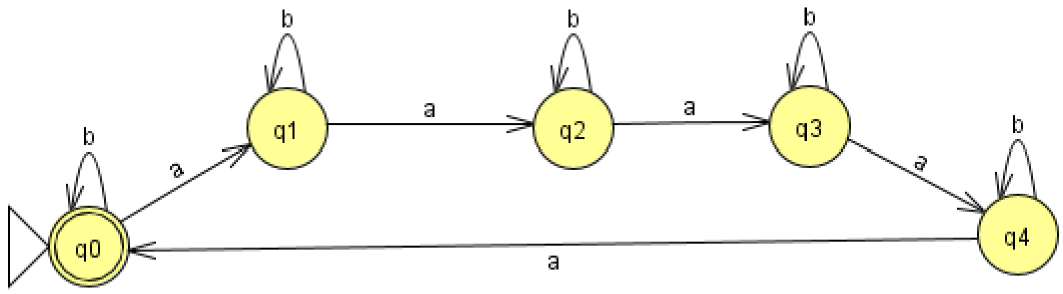
- Σ ={0,1}
- Σ1= {0,1} (21=2)
- Σ2= {00,01,10,11} (22=4)
- Σ3= {000,001,010,011,100,101,110,111} (23= 8)

The set of strings over an alphabet Σ is usually denoted by Σ*(Kleene closure)

For instance, Σ*= {0,1}*

={ ε,0,1,00,01,10,11,………}

(b) Design a DFSM for $\{w \in \{a,b\}^*: \#_a \bmod 5 = 0\}$. (Number of a's is divisible by 5) Write the definition. Show computation for w = aabaaa and w=abab and state whether it is an accepting or rejecting configuration. [06] CO1 L3



Definition

M= $(K, \Sigma, \delta, s, A)$

M= $(\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_0, \{q_0\})$

Transition Table

δ	**a**	**b**
$* \rightarrow q_0$	q_1	q_0
q_1	q_2	q_1

Computation

$(q_0, aabaaa) \vdash (q_1, abaaa) \vdash (q_2, baaa)$
$\vdash (q_2, aaa) \vdash (q_3, aa) \vdash (q_4, a) \vdash (q_0, \varepsilon)$

$(q_0, aabaaa) *$
$\vdash (q_0, \varepsilon)$ *is an accepting configuration as* q_a *A of DFSM M*

$(q_0, abab) \vdash (q_1, bab) \vdash (q_1, ab) \vdash (q_2, b) \vdash (q_2, \varepsilon)$

q_2	q_3	q_2
q_3	q_4	q_3
q_4	q_0	q_4

$(q_0, abab) *\vdash (q_2, \varepsilon)$ *is a rejecting configuration as* $q_4 \notin A$ *of DFSM M*

3 (a) Define Moore machine. Design Moore machine to output Y when there is a sequence 010 $\Sigma = \{0,1\}$ [05] CO1 L2

A Moore machine M is a seven-tuple$(K, \Sigma, O, \delta, D, s, A)$ where:

• K is a finite set of states,

•Σ is an input alphabet,

• O is an output alphabet,

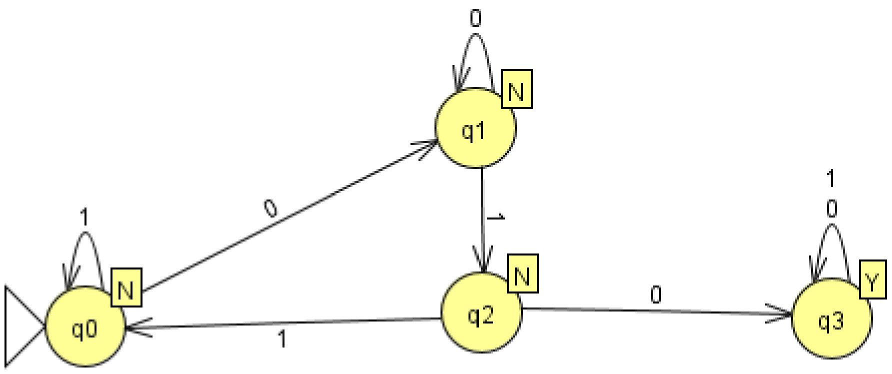
• $s \in K$ is the start state,

•$A \subseteq K$ is the set of accepting states (although for some applications this designation is not important),

• δ is the transition function. It is function from $(K \times \Sigma) to (K)$and

• D is the display or output function. It is a function from (K) to (O*).

A Moore machine M computes a function f(w) iff, when it reads the input string w, its output sequence is f(w).



M= $(\{q_0, q_1, q_2, q_3\}, \{0,1\}, \{Y, N\}, \delta, q_0, \{q_0\}, \{q3\})$

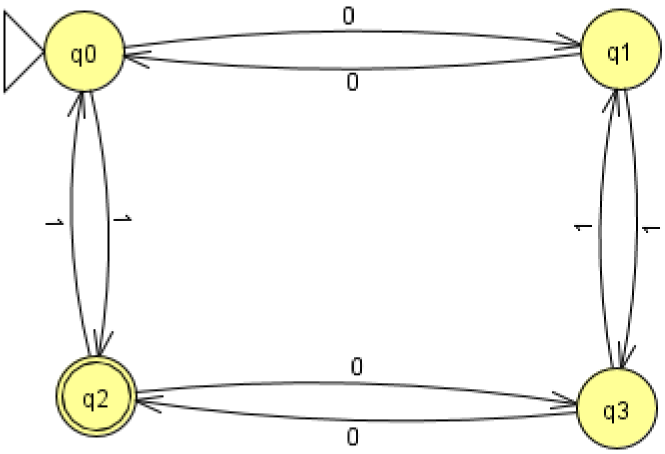
δ	**0**	**1**
$*\longrightarrow q_0/N$	q_1	q_0
q_1/N	q_1	q_2
q_2/N	q_3	q_0
q_3/Y	q_3	q_3

(b) Design DFSM to $\{w \in \{0,1\}^*$: w contains even number of 0s and odd number of 1s$\}$. [05] CO1 L3

Definition

M= $(K, \Sigma, \delta, s, A)$

M= $(\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_2\})$



q0 – even number of 0s, even number of 1s
q1 – odd number of 0s, even number of 1s
q2 – even number of 0s, odd number of 1s
q3 – odd number of 0s, odd number of 1s

δ	0	1
$\rightarrow q_0$	q_1	q_2
q_1	q_0	q_3
$* q_2$	q_3	q_0
q_3	q_2	q_1

$$(q_0, 010) \vdash (q_1, 10) \vdash (q_3, 0) \vdash (q_2, \varepsilon)$$
$$(q_0, 010) *\vdash (q_2, \varepsilon) \textit{ is a accepting configuration as } q_2 \in A \textit{ of DFSM } M$$

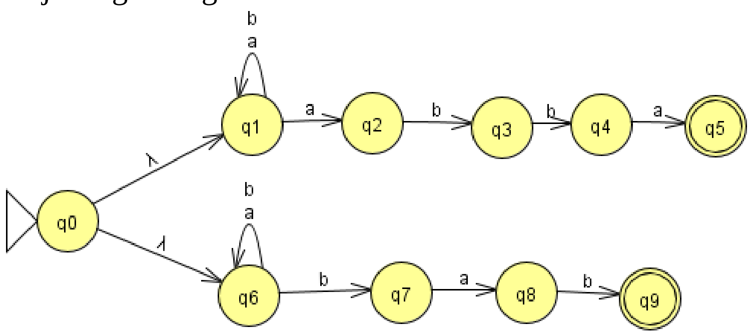
$$(q_0, 00) \vdash (q_1, 0) \vdash (q_0, \varepsilon)$$
$$(q_0, 00) *\vdash (q_0, \varepsilon) \textit{ is a rejecting configuration as } q_0 \notin A \textit{ of DFSM } M$$

4 (a) Write difference between NDFSM and DFSM [05] CO1 L2

DFSM	NDFSM
Uses a transition function, δ that maps a state to another state based on the input symbol read. maps k x input symbol to k Where k is a state	Uses a transition relation Δ which is a finite subset of (k×(Σ ∪ {ε}))×k
On each input symbol there is exactly one transition	There may or may not be a transition on a input symbol. There may be more than one transition on an input symbol (competing moves)
There is only one configuration for an input string	There may be more than one configuration for an input string
After reading a string, if the final state is an accepting state, then the string is accepted	After reading a string, if one of the states in the final configuration is accepting state, the string is accepted by the machine
Difficult to construct	Easy to construct
Behaves deterministically	Guesses the next step

ε - transitions are not allowed	ε- transitions are allowed

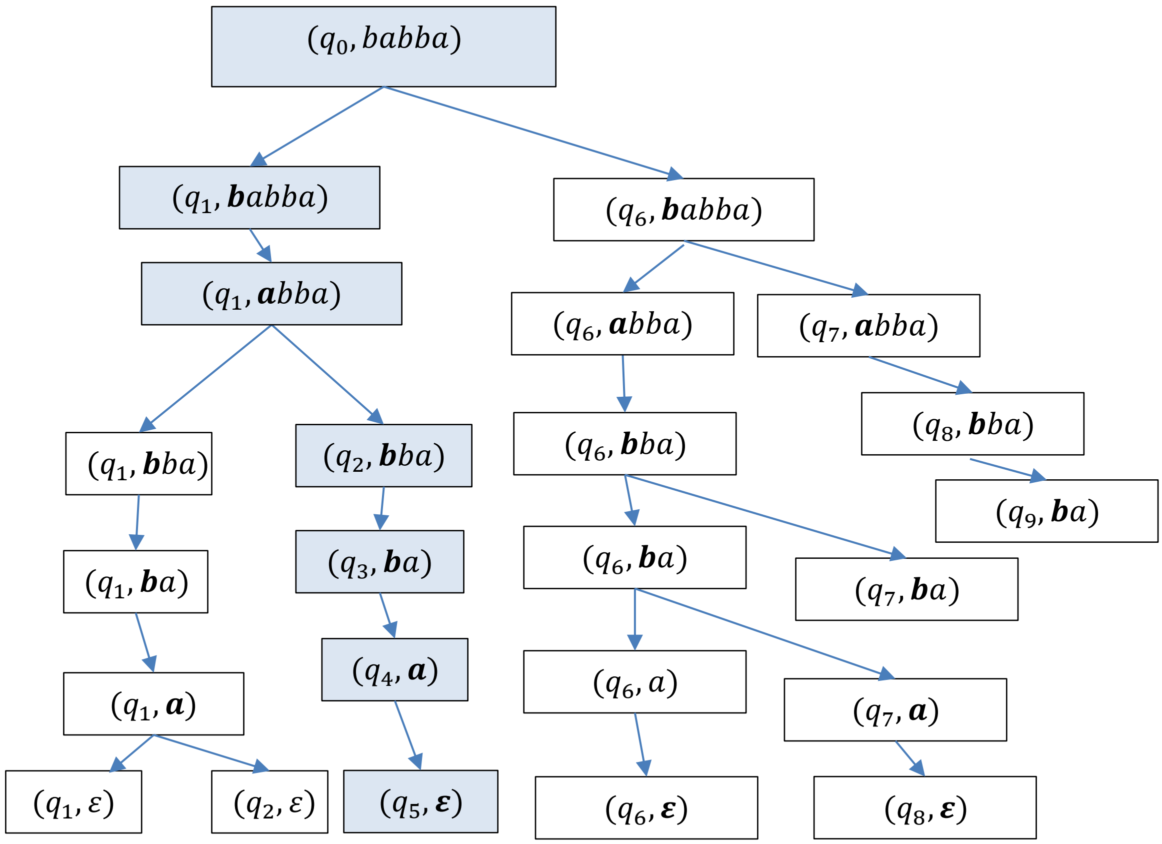
(b) Design an NDFSM for {w ∈ {a,b}*: w ends with abba or w ends with bab}. Write the definition. Show computation for w =babba and w = abab and state whether it is an accepting or rejecting configuration.. [05] CO1 L3



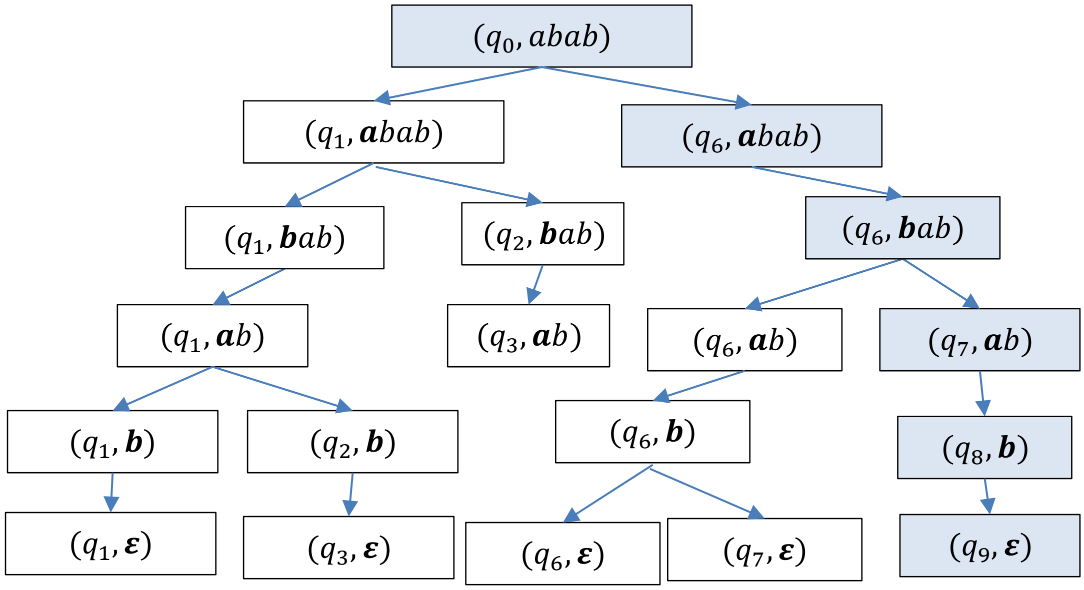
M= $(\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}, \{a, b\}, \Delta, q_0, \{q_5, q_9\})$

Δ	**eps(q)**	**a**	**b**
$\longrightarrow q_0$		$-$	$-$
q_1		$\{q_1, q_2\}$	$\{q_1\}$
q_2		$-$	q_3
q_3		$-$	q_4
q_4		q_5	$-$
q_5		$-$	$-$
q_6		q_6	$\{q_6, q_7\}$
q_7		q_8	$-$
q_8		$-$	q_9
q_9		$-$	$-$

Computation



$(q_0, \boldsymbol{babba}) \vdash * (q_5, \boldsymbol{\varepsilon})$ is an accepting configuration and hence abba is accepted by NDFSM M/
$(\{q_0, q_1, q_6\}, \boldsymbol{b}abba) \vdash (\{q_1, q_6, q_7\}, \boldsymbol{a}bba) \vdash (\{q_1, q_2, q_6, q_8\}, \boldsymbol{b}ba) \vdash$
$(\{q_1, q_{3,}, q_6, q_7, q_9\}, \boldsymbol{b}a) \vdash (\{q_1, q_{4,}, q_6, q_7\}, \boldsymbol{a}) \vdash (\{q_1, q_2, q_{5,}, q_6, q_8\}, \varepsilon)$.
Since q_5 is an accepting state the computation is accepting

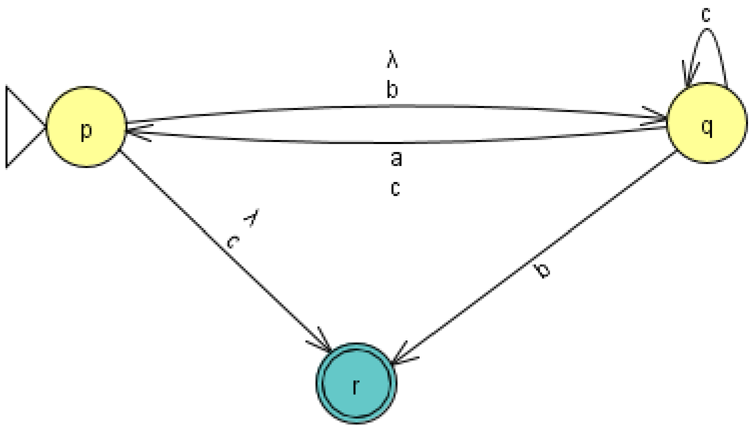


$(q_0, \boldsymbol{abab}) \vdash* (q_9, \boldsymbol{\varepsilon})$ is an accepting configuration as $q_9 \in A$ and hence bab is accepted by NDFSM M
$(\{q_0, q_1, q_6\}, \boldsymbol{a}bab) \vdash (\{q_1, q_2, q_6\}, \boldsymbol{b}ab) \vdash (\{q_1, q_3, q_6, q_7\}, \boldsymbol{a}b) \vdash (\{q_1, q_2, q_6, q_8\}, \boldsymbol{b}) \vdash (\{q_1, q_3, q_6, q_7, q_9\}, \varepsilon)$. Since q_9 is an accepting state, the computation is accepting

5 Write the procedure for eps(q) [03] CO2 L2

(a) eps(q) or ε-closure are the set of states reachable from q following 0 or more ε-transitions.
eps(q) = {p∈k: (q,w) |—*(p,w)
where eps(q) is the closure of {q} under the relation {(p,r): there is a transition (p, ε, r)∈ Δ}
eps(q:state)=
1. result = {q}
2. While there exists some p∈result and q∉result, and some transition, (p, ε,r)∈Δ do :
   Insert r into result.
3. Return result

(b) Convert the following NDFSM to an equivalent DFSM and write its definition. Show steps.Σ = {a,b,c}, λ represents ε-transitions [07] CO2 L3



Δ	a	b	c	eps(q)

$\longrightarrow p$	-	q	r	{p,q,r}
q	p	r	{p,q}	-
$*r$	-	-	-	{r}

NdfsmToDFSM

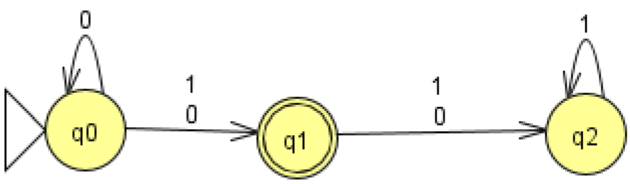
active_states	**a**	**b**	**c**
$\longrightarrow * \{p,q,r\}$	$eps(p) = \{p,q,r\}$	$eps(q) \cup eps(r) = \{q,r\}$	$eps(r) \cup eps(p) \cup eps(q) = \{p,q,r\}$
$* \{q,r\}$	$eps(p) = \{p,q,r\}$	$eps(r) = \{r\}$	$eps(r) \cup eps(p) \cup eps(q) = \{p,q,r\}$
$* \{r\}$	$\emptyset$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

6 (a) Write procedure for NDFSM to DFSM. [05] CO2 L2

ndfsmtodfsm(*M*: NDFSM) =

1. For each state q in K do:
    Compute *eps*(q).
2. s' = *eps*(s)
3. Compute δ':
  3.1. *active-states* = $\{s'\}$.
3.2. $\delta' = \phi$
3.3. While there exists some element Q of *active-states* for which δ' has not yet been computed do:
    For each character c in Σ do:
        new-state = ϕ.
        For each state q in Q do:
            For each state p such that $(q, c, p) \in \Delta$ do:
                new-state = *new-state* ∪ *eps*(p).
        Add the transition (Q, c, *new-state*) to δ'
If *new-state* ∉ *active-states* then insert it into *active-states*.
4. K' = *active-states*.
5. $A' = \{Q \in K' : Q \cap A \neq \Phi\}$

(b) Convert the following NDFSM to an equivalent DFSM and write it's definition. $\Sigma = \{0,1\}$. (Note that it is not required to calculate eps(q) as there are no ε-transitions) [05] CO2 L3



Δ	**0**	**1**
$\longrightarrow q_0$	$\{q_0, q_1\}$	$\{q_1\}$
$*q_1$	q_2	q_1
q_2	-	q_2

ndfsmToDfsm simulate

active_states	**0**	**1**
$\longrightarrow q_0$	$\{q_0, q_1\}$	$\{q_1\}$
$*\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$
$*\{q_1\}$	$\{q_2\}$	$\{q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$
$\{q_2\}$	$\emptyset$	$\{q_2\}$
$*\{q_1, q_2\}$	$\{q_2\}$	$\{q_2\}$
$\emptyset$	$\emptyset$	$\emptyset$

$\delta(\{q_0, q_1\},0) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \{q_2\} = \{q_0, q_1, q_2\}$

The DFSM M'

$\boldsymbol{\delta'}$	**0**	**1**
$\longrightarrow A$	B	C
$*B$	D	B
$*C$	E	E
$*D$	D	F
E	G	E
$*F$	E	E
G	G	G

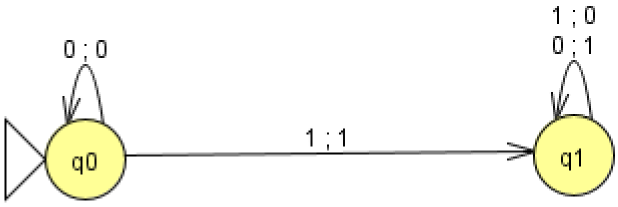
$$M' = (K', \Sigma, \delta', s', A')$$
$$M' = (\{A, B, C, D, E, F, G\}, \{0,1\}, \delta', A, \{B, C, D, F\}$$

7 (a) Define Mealy machine. Design a Mealy machine to find 2s complement of a binary number. Show the output string for the input 11100 [05] CO 1 L1

A Mealy machine M is a six-tuple $(K, \Sigma, O, \delta, s, A)$ where:

- K is a finite set of states,
- Σ is an input alphabet,
- O is an output alphabet,
- $s \in K$ is the start state,
- $A \subseteq K$ is the set of accepting states, and
- δ is the transition function. It is function from $(K \times \Sigma) to (K \times O^*)$

A Mealy machine M computes a function f(w) iff, when it reads the input string w, its output sequence is f(w).



M= $(\{q_0, q_1\}, \{0,1\}, \{0,1\}, \delta, q_0, \{q_0\})$

δ	0	1
$* \longrightarrow q_0$	$q_0/0$	$q_1/0$
q_1	$q_1/1$	$q_1/1$

Output string for 11100 -
11100
Starting from LSB – 00100

(b) Design a DFSM which will accept decimal numbers divisible by 4. Show the acceptanc eof the input 1124. [05] CO 1 L2

δ	0	1	2	3	4	5	6	7	8	9
* ⟶Q0	Q0	Q1	Q2	Q3	Q0	Q1	Q2	Q3	Q0	Q1
Q1	Q2	Q3	Q0	Q1	Q2	Q3	Q0	Q1	Q2	Q3
Q2	Q0	Q1	Q2	Q3	Q0	Q1	Q2	Q3	Q0	Q1
Q3	Q2	Q3	Q0	Q1	Q2	Q3	Q0	Q1	Q2	Q3

$$\boldsymbol{\delta(q0, 1) = q1}$$

$$\boldsymbol{\delta(q0, 11) = \delta(q1, 1) = q3}$$

$$\boldsymbol{\delta(q0, 112) = \delta(q3, 2) = q0}$$
$$\boldsymbol{\delta(q0, 1124) = \delta(q0, 4) = q0}$$

It is accepted.

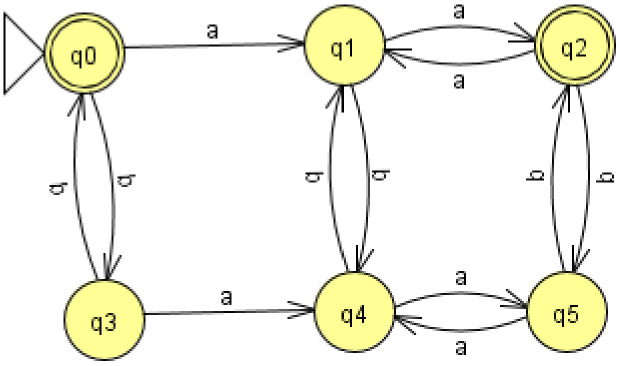
Q8 option available only for Section A&C

8 (a) What is meant by indistinguishable states, i.e. when q≡p. What is meant by distinguishable states? [03] CO2 L2

$\equiv p$ are indistiguishable iff for all strings $w \in \Sigma*$ either w drives M to an accepting state from both q and p or it drives M to a rejecting state from both q and p.

q and p are distinguishable if for all strings, $w \in \Sigma*$ w drives M to an accepting state from q and a non-accepting state from p or vice versa.

(b) Minimize the following DFSM : [07] CO2 L3



First, we divide into accepting and non accepting classes.Classes = [0,2], [1,3,4,5]

[0,2]	a	b	
[0]	[1,3,4,5]	[1,3,4,5]	*No splitting required as both 0 and 2 drive a and b to the same non-accepting class*
[2]	[1,3,4,5]	[1,3,4,5]	

[1,3,4,5]	a	b	
[1]	[0,2]	[1,3,4,5]	*Splitting required, [1],[3,5],[4]*
[3]	[1,3,4,5]	[0,2]	
[4]	[1,3,4,5]	[1,3,4,5]	
[5]	[1,3,4,5]	[0,2]	

Classes = [0,2],[1],[3,5],[4]

[0,2]	**a**	**b**	
[0]	[1]	[3,5]	*No splitting required as both 0 and 2 drivea to non-accepting class [1] and b to non-accepting class [3,5]*
[2]	[1]	[3,5]	

[3,5]	**a**	**b**	
[3]	[4]	[0,2]	*No splitting required as both 3 and 5 drivea to non-accepting class [4] and b to accepting class [0,2]*
[5]	[4]	[0,2]	

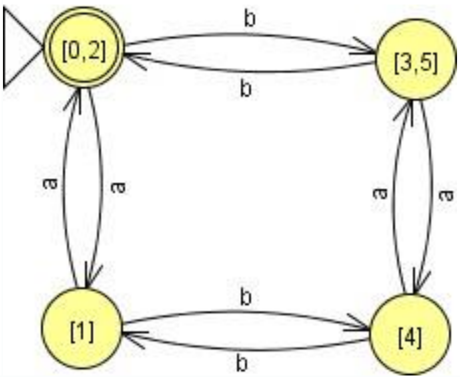
Classes = [0,2],[1],[3,5],[4]

The definition of the DFSM is as follows.
M’ = (k’,Σ, δ, s’,A’) where
K’= {[0,2],[1],[3,5],[4]}
Σ = {a,b}
δ = { (([0,2], a),[1]), (([0,2], b), [3,5]),
(([1], a), [0,2]), (([1], b), [4]),
(([3,5], a), [4]), (([3,5], b), [0,2]),
(([4], a), [3,5]), (([4], b), [1])
}
s’=[0,2]

Transition Table

$\delta_{M'}$	**a**	**b**
→*[0,2]	[1]	[3,5]
[1]	[0,2]	[4]
[3,5]	[4]	[0,2]
[4]	[3,5]	[1]

$A' = \{[0,2]\}$



CO PO Mapping

Course Outcomes		Modules covered	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	Acquire fundamental understanding of the core concepts in automata theory and Theory of Computation	**1,2,3,4,5**	2	3	-	-	-	2	-	-	-	-	-	-	-	3		3
CO2	Learn how to translate between different models of Computation (e.g., Deterministic and Non-deterministic and Software models).	**1,2**	2	3	2	2	2	2	-	-	-	-	-	-	-	3	3	3
CO3	Design Grammars and Automata (recognizers) for different language classes and become knowledgeable about restricted models of Computation (Regular, Context Free) and their relative powers.	**2,3**	2	3	2	2	2	2	-	-	-	-	-	-	2	-	3	-
CO4	Develop skills in formal reasoning and reduction of a problem to a formal	**3,4**	2	3	2	2	-	2	-	-	-	-	-	-	2	2	3	3

	model, with an emphasis on semantic precision and conciseness.																	
CO5	Classify a problem with respect to different models of Computation	**5**	2	3	2	2	-	3	-	-	-	-	-	-	3	3	3	3

COGNITIVE LEVEL	REVISED BLOOMS TAXONOMY KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO)				CORRELATION LEVELS	
PO1	Engineering knowledge	PO7	Environment and sustainability	0	No Correlation
PO2	Problem analysis	PO8	Ethics	1	Slight/Low
PO3	Design/development of solutions	PO9	Individual and team work	2	Moderate/ Medium
PO4	Conduct investigations of complex problems	PO10	Communication	3	Substantial/ High
PO5	Modern tool usage	PO11	Project management and finance		
PO6	The Engineer and society	PO12	Life-long learning		
PSO1	Develop applications using different stacks of web and programming technologies				
PSO2	Design and develop secure, parallel, distributed, networked, and digital systems				
PSO3	Apply software engineering methods to design, develop, test and manage software systems.				
PSO4	Develop intelligent applications for business and industry				