USN

Internal Assessment Test 1 – Nov 2021

Sub:	Automata Theory and Computability				Sub Code:	18CS54	Branch:	CSE
Date:	13/11/2021	Duration:	90 mins	Max Marks: 50	Sem / Sec:	5 A,B,C		OBE

<u>Answer any FIVE FULL Questions</u> — MARKS | CO | RBT

1 (a) Define the following with examples : [02] CO1 L1

i) String ii) Language

Ans: 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}

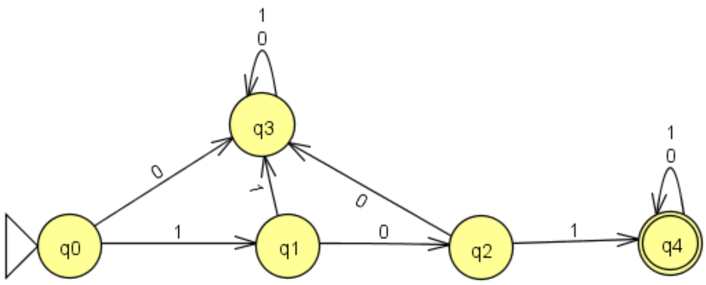
Language: A language (finite/infinite) is a set of strings over a given alphabet, Σ. If there is more than one language, we will use Σ_L to denote alphabets from which language L is formed.

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 L={w| w∈{0,1}*: w begins with 101}. Write the definition of DFSM. Show computation for w = 1010 and w=1100 and state whether it is an accepting or rejecting configuration using extended transition function. [08] CO1 L3



The DFSM is designed to only accept strings starting with 101. If the first alphabet is 0 or the second symbol is 1 or the third symbol is 0, it goes into a dead state. After having read 101, the DFSM accepts any combination of symbols and remains in the accepting state.

<u>Definition of DFSM M</u>

M = (k,Σ, δ, s,A) where

k= {q_0, q_1, q_2, q_3, q_4}

Σ = {0,1}

Δ = { ((q_0, 0), q_3), ((q_0, 1), q_1), ((q_1, 0), q_2), ((q_1, 1), q_3), ((q_2, 0), q_3), ((q_2, 1), q_4), ((q_3, 0), q_3), ((q_3, 1), q_3), ((q_4, 0), q_4), ((q_4, 1), q_4)}

s = q_0

Transition Table

δ_M	a	b
→ q_0	q_3	q_1
q_1	q_2	q_3
q_2	q_3	q_4
q_3	q_3	q_3
* q_4	q_4	q_4

$A = \{q_4\}$

Computation

$(q_0, \mathbf{1}010) \vdash (q_1, \mathbf{0}10) \vdash (q_2, \mathbf{1}0) \vdash (q_4, \mathbf{0}) \vdash (q_4, \varepsilon)$

Since $q_4 \in A_M$ is an accepting state after all the input symbols have been read, it is an accepting configuration and w=1010 is accepted by DFSM M.

$(q_0, \mathbf{1}100) \vdash (q_1, \mathbf{1}10) \vdash (q_3, \mathbf{1}0) \vdash (q_3, \mathbf{0}) \vdash (q_3, \varepsilon)$

Since $q_3 \notin A_M$ and is not an accepting state after all the input symbols have been read, it is a rejecting configuration and w=1100 is not accepted by DFSM M.

2 (a) Define the following with examples : [03] CO1 L1

i) Alphabet ii)Cardinality of a Language

ANS : 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}

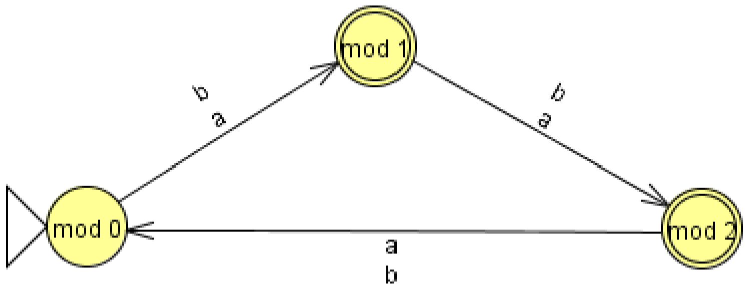
Alphabet of digits Σ = {0,1,2,3,4,5,6,7,8,9}

Cardinality of a language: It represents the number of strings a language contains. The cardinality of a language is at least 0 or countably infinite.

Eg. L = $\{a^n: n>=1\}$ is countably infinite as there is a one-to-one correspondence between natural numbers and the set of integers in the language

f(n) = {n if n>=1}

(b) Design a DFSM for L={w| w∈{a,b}*: |w| mod 3 ≠ 0}. Show computation for w = aaba and w=bab and state whether it is an accepting or rejecting configuration using extended transition function. [07] CO1 L3



The DFSM M is designed by 3 states, mod 0, mod 1and mod 2.

Mod 0 is the state with the length of w is divisible by 3 and remainder is 0. This is not an accepting state.

Mod1 is the state with length of w as 1,4, 7,… where dividing by 3 yields remainder 1.

Mod 2 is the state with length of w as 2,5,8,… where dividing by 3 yields remainder 2.

<u>Definition of DFSM M</u>

M = (k,Σ, δ, s,A) where

k= {mod0,mod1,mod2}

Σ = {a,b}

δ = { ((mod0, a), mod1), ((mod0, b), mod1), ((mod1, a), mod2), ((mod1, b), mod2), ((mod2, a), mod0), ((mod2, b), mod0),}

s = mod0

A = {mod1,mod2}

Transition Table

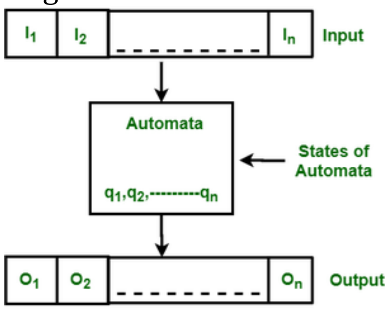
δ_M	a	b
→ *mod0*	mod1	mod1
* *mod1*	mod2	mod2
* *mod2*	mod0	mod0

(mod0, **a**aba) |— (mod1,**a**ba) |— (mod2,**b**a) |— (mod0,**a**) |— (mod1, ε)
Since mod1 ∈ A_M which is an accepting state after all the input symbols have been read, it is an accepting configuration and w=aaba is accepted by DFSM M.
(mod0, **b**ab) |— (mod1,**a**b) |— (mod2,**b**) |— (mod0, ε)
Since mod0 ∉ A_M which is not an accepting state after all the input symbols have been read, it is a rejecting configuration and w=bab is not accepted by DFSM M.

3 (a) What is a finite automaton? Explain the operation of FA with a neat basic block diagram. [03] CO1 L2

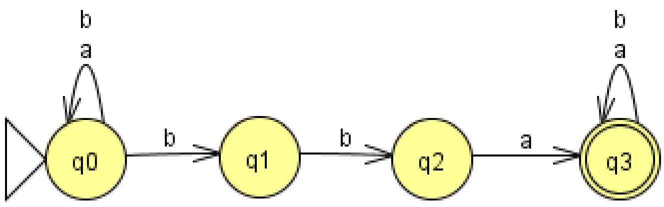
Finite Automaton is a computational device whose input is a string and whose output is one of two values that we can call Accept or Reject. It can be deterministic or non-deterministic. It is designed with a quintuple M = (k,Σ, δ, s, A) where

- k is the set of states.
- Σ is the input alphabet
- δ is the transition function in case of DFSM or Δ in case of NDFSM
- s is the start state
- A is the set of accepting states.



On reading an input symbol, a finite state machine may stay on a state or move to another state. If after all the input symbols in a string are read and the final state contains a state that belongs to A, then the srting is Accepted and the configuration is called the Accepting Configuration. Otherwise, we reject the string and it enters into a Rejecting configuration.

(b) Design an NDFSM for L={w| w∈{a,b}*: w contains the substring bba}. Write the definition of NDFSM. Show computation for w = abba and w = aab and state whether it is an accepting or rejecting configuration using extended transition function. [07] CO1 L3



The NDFSM stays in state q_0 on reading any a or b and guesses a substring bba by also moving to q_1 on b. After going to q_2 on another b, it moves to the final state q_3. Once it reads a substring bba, it accepts any sequence of symbols following it on state q_3.
M = (k,Σ, Δ, s,A) where
k= $\{q_0, q_1, q_2, q_3\}$
Σ = {a,b}

$\Delta = \{ ((q_0,a),\{q_0\}),\ ((q_0,b),\{q_0,q_1\}),\ ((q_1,b),q_2),((q_2,a),q_3),$
$((q_3,a),q_3),((q_3,b),q_3)\}$
s=q_0
A = $\{q_3\}$

Transition Table

δ_M	a	b
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_1	Φ	q_2
q_2	q_3	Φ
$*q_3$	q_3	q_3

Computation:

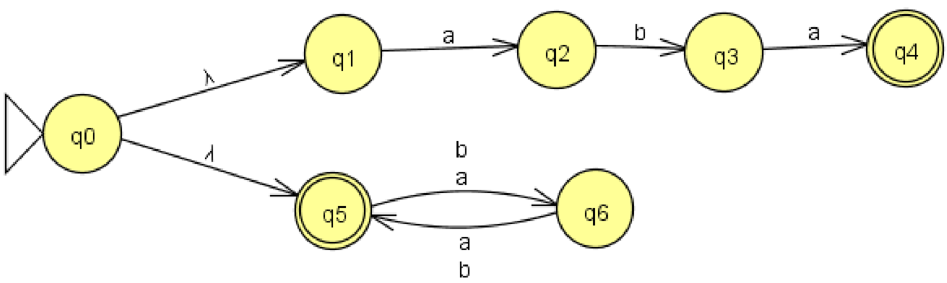
(q_0, abba) |— ({q_0},bba) |— ({q_0, q_1},ba) |— ({q_0, q_1, q_2},a) |— ({q_0, q_3}, ε)

Since {q_0, q_3}∩A≠Φ, (q_0, abba) |—*({q_0, q_3}, ε) is an accepting configuration and the string w = abba is accepted by NDFSM M.
(q_0, aab) |— ({q_0}, ab)|— ({q_0},b) |— ({q_0, q_1}, ε)
Since {q_0, q_1}∩A=Φ, (q_0, abba) |— *({q_0, q_3}, ε) is a rejecting configuration and the string w =aab is not accepted by NDFSM M.

4 (a) Differentiate DFSM and NDFSM. [04] CO2 L3

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
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
Behaved deterministically	Guesses the next step
ε - transitions are not allowed	ε- transitions are allowed

(b) Design an NDFSM for L = {w| w∈{a,b}*: w= aba or |w| is even. Give definition and explain how the NDFSM was designed [06] CO2 L3



M = (k,Σ, Δ, s,A) where
k= $\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$

$\Sigma = \{a,b\}$
$\Delta = \{ (q_0, \varepsilon), \{q_1, q_5\}), \; ((q_1, a), q_2), ((q_2, b), q_3), ((q_3, a), q_4),$
$((q_5, a), q_6), ((q_5, b), q_6), ((q_6, a), q_5), ((q_6, b), q_5)\}$
s=q_0
A = $\{q_4, q_5\}$

Transition Table

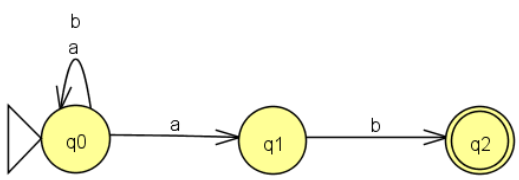
Δ_M	a	b	eps(q)
$\rightarrow q_0$	Φ	Φ	$\{q_0, q_1, q_5\}$
q_1	q_2	Φ	
q_2	Φ	q_3	
q_3	q_4	Φ	
$* q_4$	Φ	Φ	
$* q_5$	q_6	q_6	
q_6	q_5	q_5	

First an FSM for w= aba was designed by using 3 states.
Another FSM was designed using two states to accept |w| is even.
A new start state was introduced and ε – transition was used to connect to the start state of both the FSM's.
The states were renamed such that there is no two states with the same name in the new FSM.

5 (a) Explain various functions on languages. [03] CO1 L2

Length of a string s, is denoted by |s|
$|\varepsilon| = 0$
$|aba| = 3$
Number of symbols in a given string, s is represented by $\#_c(s)$
$\#_a(ababb) = 2$, the number of a's in string ababb is 2
Concatenation of two strings s and t, is represented using s||t or st which is formed by appending t to s.
Eg. x=good, y=bye
xy=goodbye
ε is the identity for concatenation where ε s = s ε = s
Replication : For each string w and each natural number, i, w^i is defined as
$w^0 = \varepsilon$
a^3 = aaa
$(bye)^2$=byebye
String Reversal : w^R is the reverse of string w.
If |w|=0, $w^R = w = \varepsilon$
If |w|>0, $\exists a \in \sum(\exists u \in \sum^* (w = ua))$, i.e. last character of w is a, $w^R = au^R$

(b) Convert the following NDFSM to an equivalent DFSM and write its definition. Show steps. [07] CO2 L3



Transition Table

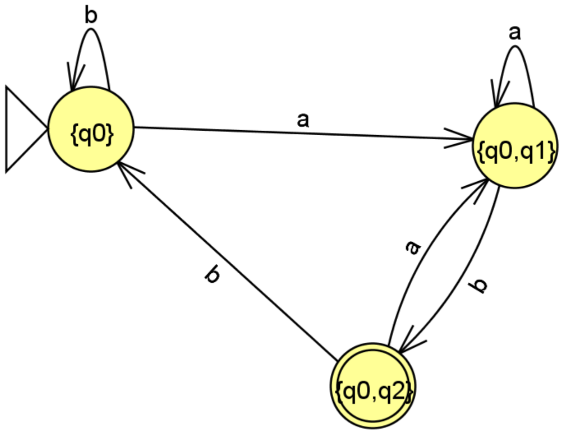
Δ_M	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0
q_1	Φ	q_2
$* \; q_2$	Φ	Φ

Transition Table

$\Delta_{M'}$	a	b	
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0	$\{q_0, q_1\}$ *so compute transitions for them*
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$	$(\{q_0, q_1\}, \text{a}) = \{q_0, a\} \cup \{q_1, a) = \{q_0, q_1\} \cup \Phi = \{q_0, q_1\}$ $(\{q_0, q_1\}, \text{b}) = \{q_0, b\} \cup \{q_1, b) = \{q_0, q_1\} \cup \Phi = \{q_0, q_2\}$ $\{q_0, q_1\}$ is a new state, so compute transitions
$* \{q_0, q_2\}$	$\{q_0, q_1\}$	q_0	$(\{q_0, q_2\}, \text{a}) = \{q_0, a\} \cup \{q_2, a) = \{q_0, q_1\} \cup \Phi = \{q_0, q_1\}$ $(\{q_0, q_2\}, \text{b}) = \{q_0, b\} \cup \{q_2, b) = \{q_0\} \cup \Phi = \{q_0\}$ No new states, hence, stop computing.

The power set of the given NDFSM gives $2^3 = 8$ states. We start with the ε-closure of the start state. For each new state created, the transitions on a and b is computed. We stop when no new state is generated.

The accepting states in the DFSM is $\{q_0, q_2\}$ because $\{q_0, q_2\} \cap A\ of\ ndfsm \neq \emptyset$ where A of given ndfsm = $\{q_2\}$
Transition diagram for DFSM:



The definition of the DFSM is as follows.
M' = (k',Σ, δ, s',A') where
K'= $\{\{q_0\}, \{q_0, q_1\}, \{q_0, q_2\}\}$

$\Sigma = \{a,b\}$

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

$s' = \{q_0\}$

$A' = \{ \{q_0, q_2\} \}$

6 (a) How to calculate the epsilon closure, ε-closure, eps(q) of a state q? [03] CO2 L2

eps(q) or ε-closure are the set of states reachable from q following 0 or more ε-transitions.

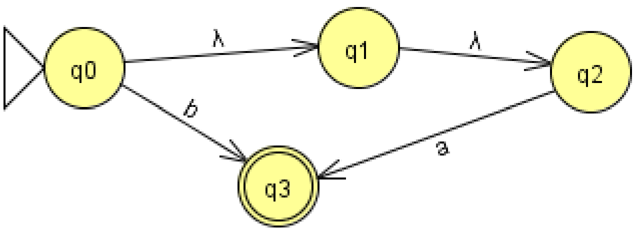
$eps(q) = \{p \in k: (q,w) \vdash^* (p,w)\}$

where eps(q) is the closure of {q} under the relation {(p,r): there is a transition (p, ε, r)∈ Δ}

eps(q:state)=

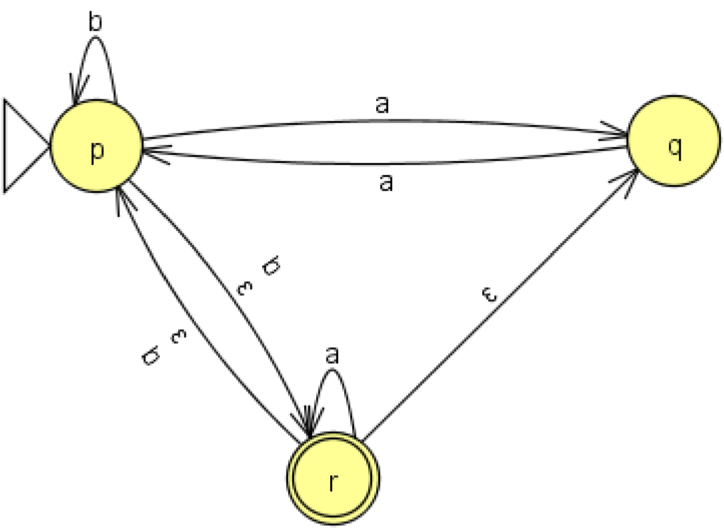
1. result = {q}
2. While there exists some p∈result and q∉result, and some transition, (p, ε, r)∈Δ do : Insert r into result.
3. Return result

For example,



eps(q0) = {q0, q1, q2}

(b) Convert the following ε-NDFSM to an equivalent DFSM and write it's definition. [07] CO2 L3



Transition Table

Δ_M	a	b	Eps(q)
$\rightarrow p$	{q}	{p,r}	{p,q,r}
q	{p}	Φ	{q}
$* \ r$	{r}	{p}	{p,q,r}

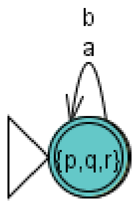
M = (k,Σ, Δ, s,A) where

k= {p,q,r}

Σ = {a,b}

Transition Table

$\Delta_{M'}$	a	b	
$\rightarrow$ {p,q,r}	$\{p,q,r\}$	{p,q,r}	$(\{p,q,r\},a) = (p,a) \cup (q,a) \cup (r,a)$ $= eps(q) \cup eps(p) \cup eps(r)$ $= \{q\} \cup \{p,q,r\} \cup \{p,q,r\}$ $= \{p,q,r\}$ $(\{p,q,r\},b) = (p,b) \cup (q,b) \cup (r,b)$ $= eps(p) \cup eps(r) \cup eps(p)$ $= \{p,q,r\} \cup \{p,q,r\} \cup \{p,q,r\}$ $= \{p,q,r\}$



The definition of the DFSM is as follows.
M' = (k',Σ, δ, s',A') where
K'= $\{p,q,r\}$}
Σ = {a,b}
δ = { (({p,q,r}, a), $\{p,q,r\}$), (({p,q,r}, b), $\{p,q,r\}$)}
s'={{p,q,r}}
A' = { {p,q,r}}

7 (a) What is a regular expression? What are the rules for forming regular expressions? [05] CO 1 L1

A regular expression contains 2 kinds of symbols.

- A set of symbols to which we attach particular meanings when they occur in regular expressions, Φ, ∪, ε, (.), * and +
- An alphabet Σ which contains the symbols that regular expressions will match against.

A regular expression is a string that can be formed according to the following rules.

1. Φ is a regular expression
2. ε is a regular expression
3. Every element in Σ is a regular expression
4. Given two regular expressions, α and ß, αß is also a regular expression
5. Given two regular expressions, α and ß, α ∪ ß is also a regular expression
6. Given a regular expression, α, α* is also a regular expression

7. Given a regular expression, α, α^+ is also a regular expression
8. Given a regular expression, α, (α) is also a regular expression

(b) Write regular expression for [05] CO 1 L3

(i) $\{w \in \{0,1\}^* :$ w are natural numbers in binary encoding that are powers of 4 with no leading zeros. $\{1,100,10000,\ldots\}$

Ans : $1\,(00)^*$

(ii) $\{w \in \{0,1\}^*$: w has 101 as a substring.}

Ans : $(0 \cup 1)^* 101 (0 \cup 1)^*$

(iii) $\{w \in \{a,b\}^*$: w doesn't end with ba}

$(a \cup b)^*(aa \cup bb \cup ab)$

(iv) $\{0^n 1^m : n+m \text{ is even}\}$

Ans : $(00)^*(11)^* \cup (\,0\,(00)^*\,1\,(11)^*)$

(v) $\{a^n b^m : n<=2 \text{ and } m>=3\}$

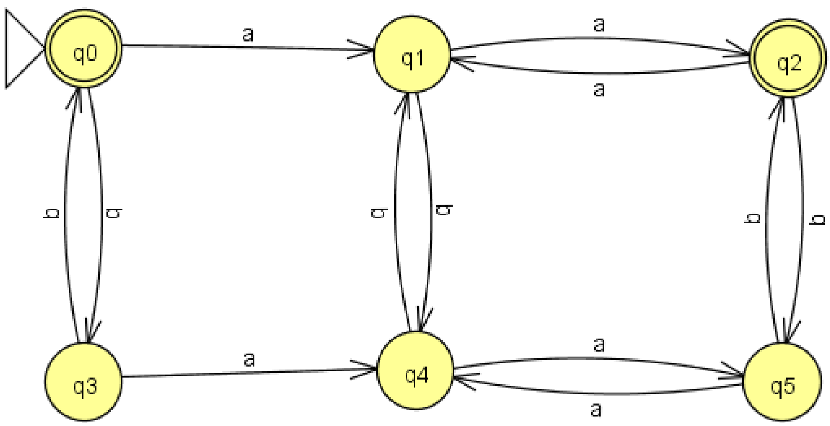
Ans : $bbbb^* \cup abbbb^* \cup aabbbb^*$

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

q≡*p* are indistiguishable iff for all strings *w*∈Σ∗ 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*∈Σ∗ w drives M to an accepting state from q and a non-accepting state from p or vice versa.

(b) Let M be the following DFSM. Find a DFSM with minimal states. Show the steps. Define the minimal DFSM. [07] CO2 L3



First, we divide into accepting and non accepting classes.

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

Check if splitting is required

[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]

Transition Table

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

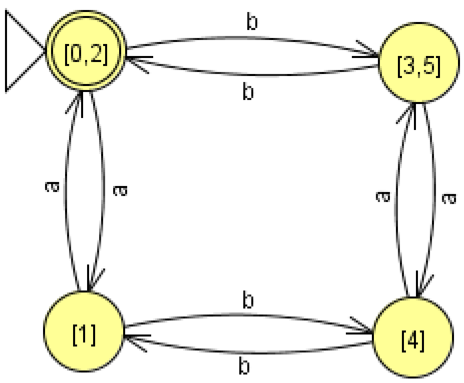
[3,5]	**a**	**b**	
[3]	[4]	[0,2]	*No splitting required as both 3 and 5 drive a to non-accepting class [4] and b to accepting class [0,2]*
[5]	[4]	[0,2]	

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]
A' = {[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]



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 model, with an emphasis on semantic precision and conciseness.	**3,4**	2	3	2	2	-	2	-	-	-	-	-	-	2	2	3	3
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				