Basics of TOC Part - II

Foundation Course on Theory of Computation for GATE 2023

Sanchit Jain • Lesson 2 • Apr 7, 2022

Educator highlights

- Works at **Knowledge Gate**
- Studied at **Delhi Technological University**
- Have qualified gate Have experience of more than 8 years Have a YouTube channel with 5 lakh follower
- Lives in **Ghaziabad, Uttar Pradesh, India**
- Unacademy Educator since **19th July, 2019**
- **844,388 live minutes** taught in last 30 days
- Knows **Punjabi, Hinglish, Hindi and English**

Sanchit Jain

Follow

Legend in GATE & ESE

I am passionate for teaching computer science, having experience of more than 10 years. I teach all computer science subjects for GATE.

56M Watch mins | **2M** Watch mins (last 30 days) | **27K** Followers | **2K** Dedications

Months Wise Calendar for GATE-2023 (Unacademy Plus) (Sanchit Jain)

Month	Morning Class (6:30am – 8:00am)	Evening Class (7:00pm – 8:30pm)
Feb-2022	CN(16-Feb to 26-March) (54 Hours)	DM (2-Feb to 2-Mar) (50 Hours)
Mar-2022	**TOC (6-Apr to 26-May) (58 Hours)**	**DBMS (16-Mar to 2-May) (50 Hours)**
Apr-2022	Compiler (27-May to 15-June) (34 Hours)	DE (2-May to 1-June) (40 Hours)
May-2022	DS & Programming) (16-June to 05-July) (56 Hours)	COA (23-June to 20-July) (38 Hours)
Jun-2022	Algo (7-July to 5-Aug) (38 Hours)	OS (21-July to 24-Aug) (58 Hours)
Jul-2022		
Aug-2022	DM (11-Aug to 20-Sept) (50 Hours)	CN (25-Aug to 5-Oct) (54 Hours)
Sep-2022	DBMS (22-Sept to 29-Oct) (50 Hours)	TOC (6-Oct to 20-Oct) (58 Hours)
Oct-2022	DE (3-Nov to 23-Nov) (40 Hours)	Compiler (24-Oct to 20 Nov) (34 Hours)
Nov-2022	COA (24-Nov to 13-Dec) (38 Hours)	**DS & C** (22-Nov to 20-Dec) **(56 Hours)**
Dec-2022	OS (15-Dec to 31-Jan) (58 Hours)	Algo (21-Dec to 3-Feb) (38 Hours)
Jan-2023		
Feb-2023		

TOC Schedule For GATE-2023

Morning Batch					
Date	Day	Subject-1	Topic	Time	Duration
06-04-2022	Wednesday	TOC	Basics of TOC Part-1	6:30 am to 8:00 am	1.5
07-04-2022	Thursday	TOC	Basics of TOC Part-2	6:30 am to 8:00 am	1.5
08-04-2022	Friday	TOC	Deterministic Finite Automata Part-1	6:30 am to 8:00 am	1.5
11-04-2022	Monday	TOC	Doubt Clearing Session	6:30 am to 8:00 am	1.5
12-04-2022	Tuesday	TOC	Deterministic Finite Automata Part-2	6:30 am to 8:00 am	1.5
13-04-2022	Wednesday	TOC	Deterministic Finite Automata Part-3	6:30 am to 8:00 am	1.5
14-04-2022	Thursday	TOC	Deterministic Finite Automata Part-4	6:30 am to 8:00 am	1.5
15-04-2022	Friday	TOC	Doubt Clearing Session	6:30 am to 8:00 am	1.5
18-04-2022	Monday	TOC	Deterministic Finite Automata Part-5	6:30 am to 8:00 am	1.5
19-04-2022	Tuesday	TOC	Deterministic Finite Automata Part-6	6:30 am to 8:00 am	1.5
20-04-2022	Wednesday	TOC	Non-Deterministic Finite Automata Part-1	6:30 am to 8:00 am	1.5
21-04-2022	Thursday	TOC	Doubt Clearing Session	6:30 am to 8:00 am	1.5
22-04-2022	Friday	TOC	Non-Deterministic Finite Automata Part-2	6:30 am to 8:00 am	1.5
25-04-2022	Monday	TOC	Non-Deterministic Finite Automata Part-3	6:30 am to 8:00 am	1.5
26-04-2022	Tuesday	TOC	Non-Deterministic Finite Automata Part-4	6:30 am to 8:00 am	1.5
27-04-2022	Wednesday	TOC	Doubt Clearing Session	6:30 am to 8:00 am	1.5
28-04-2022	Thursday	TOC	Identification of Regular Language	6:30 am to 8:00 am	1.5
29-04-2022	Friday	TOC	Regular Expression Part-1	6:30 am to 8:00 am	1.5
02-05-2022	Monday	TOC	Regular Expression Part-2	6:30 am to 8:00 am	1.5
03-05-2022	Tuesday	TOC	Doubt Clearing Session	6:30 am to 8:00 am	1.5
04-05-2022	Wednesday	TOC	Regular Expression Part-3	6:30 am to 8:00 am	1.5
05-05-2022	Thursday	TOC	Regular Expression Part-4	6:30 am to 8:00 am	1.5
06-05-2022	Friday	TOC	Regular Grammar	6:30 am to 8:00 am	1.5
09-05-2022	Monday	TOC	Doubt Clearing Session	6:30 am to 8:00 am	1.5
10-05-2022	Tuesday	TOC	Moore and Mealy Machine	6:30 am to 8:00 am	1.5
11-05-2022	Wednesday	TOC	Puch Down Automata Part-1	6:30 am to 8:00 am	1.5
12-05-2022	Thursday	TOC	Puch Down Automata Part-2	6:30 am to 8:00 am	1.5
13-05-2022	Friday	TOC	Doubt Clearing Session	6:30 am to 8:00 am	1.5
16-05-2022	Monday	TOC	Context Free Grammar Part-1	6:30 am to 8:00 am	1.5
17-05-2022	Tuesday	TOC	Context Free Grammar Part-2	6:30 am to 8:00 am	1.5
18-05-2022	Wednesday	TOC	Turing Machine Part-1	6:30 am to 8:00 am	1.5
19-05-2022	Thursday	TOC	Doubt Clearing Session	6:30 am to 8:00 am	1.5
20-05-2022	Friday	TOC	Turing Machine Part-2	6:30 am to 8:00 am	1.5
23-05-2022	Monday	TOC	Turing Machine Part-3	6:30 am to 8:00 am	1.5
24-05-2022	Tuesday	TOC	Turing Machine Part-4	6:30 am to 8:00 am	1.5
25-05-2022	Wednesday	TOC	Doubt Clearing Session	6:30 am to 8:00 am	1.5
26-05-2022	Thursday	TOC	Complexity Theory	6:30 am to 8:00 am	1.5

Evening Batch

Date	Day	Subject-2	Topic	Time	Duration
16-03-2022	Wednesday	DBMS	Basics and Introduction	7:00 pm to 8:30 pm	1.5
17-03-2022	Thursday	DBMS	Transaction & Concurrency Control Part-1	7:00 pm to 8:30 pm	1.5
18-03-2022	Friday	DBMS	Transaction & Concurrency Control Part-2	7:00 pm to 8:30 pm	1.5
21-03-2022	Monday	DBMS	Doubt Clearing Session	7:00 pm to 8:30 pm	1.5
22-03-2022	Tuesday	DBMS	Transaction & Concurrency Control Part-3	7:00 pm to 8:30 pm	1.5
23-03-2022	Wednesday	DBMS	Transaction & Concurrency Control Part-4	7:00 pm to 8:30 pm	1.5
24-03-2022	Thursday	DBMS	Transaction & Concurrency Control Part-5	7:00 pm to 8:30 pm	1.5
25-03-2022	Friday	DBMS	Doubt Clearing Session	7:00 pm to 8:30 pm	1.5
28-03-2022	Monday	DBMS	E R Diagram Part - 1	7:00 pm to 8:30 pm	1.5
29-03-2022	Tuesday	DBMS	E R Diagram Part - 2	7:00 pm to 8:30 pm	1.5
30-03-2022	Wednesday	DBMS	E R Diagram Part - 3	7:00 pm to 8:30 pm	1.5
31-03-2022	Thursday	DBMS	Doubt Clearing Session	7:00 pm to 8:30 pm	1.5
01-04-2022	Friday	DBMS	Functional Dependency Part -1	7:00 pm to 8:30 pm	1.5
04-04-2022	Monday	DBMS	Functional Dependency Part -2	7:00 pm to 8:30 pm	1.5
05-04-2022	Tuesday	DBMS	Functional Dependency Part -3	7:00 pm to 8:30 pm	1.5
06-04-2022	Wednesday	DBMS	Doubt Clearing Session	7:00 pm to 8:30 pm	1.5
07-04-2022	Thursday	DBMS	Normalization Part -1	7:00 pm to 8:30 pm	1.5
08-04-2022	Friday	DBMS	Normalization Part -2	7:00 pm to 8:30 pm	1.5
11-04-2022	Monday	DBMS	Normalization Part -3	7:00 pm to 8:30 pm	1.5
12-04-2022	Tuesday	DBMS	Doubt Clearing Session	7:00 pm to 8:30 pm	1.5
13-04-2022	Wednesday	DBMS	Indexing and Physical Structure Part-1	7:00 pm to 8:30 pm	1.5
14-04-2022	Thursday	DBMS	Indexing and Physical Structure Part-2	7:00 pm to 8:30 pm	1.5
15-04-2022	Friday	DBMS	Indexing and Physical Structure Part-3	7:00 pm to 8:30 pm	1.5
18-04-2022	Monday	DBMS	Doubt Clearing Session	7:00 pm to 8:30 pm	1.5
19-04-2022	Tuesday	DBMS	Relational Algebra Part - 1	7:00 pm to 8:30 pm	1.5
20-04-2022	Wednesday	DBMS	Relational Algebra Part - 2	7:00 pm to 8:30 pm	1.5
21-04-2022	Thursday	DBMS	Relational Algebra Part - 3	7:00 pm to 8:30 pm	1.5
22-04-2022	Friday	DBMS	Doubt Clearing Session	7:00 pm to 8:30 pm	1.5
25-04-2022	Monday	DBMS	SQL Part -1	7:00 pm to 8:30 pm	1.5
26-04-2022	Tuesday	DBMS	SQL Part -2	7:00 pm to 8:30 pm	1.5
27-04-2022	Wednesday	DBMS	SQL Part -3	7:00 pm to 8:30 pm	1.5
28-04-2022	Thursday	DBMS	Doubt Clearing Session	7:00 pm to 8:30 pm	1.5
29-04-2022	Friday	DBMS	Relational Calculus Part-1	7:00 pm to 8:30 pm	1.5
02-05-2022	Monday	DBMS	Relational Calculus Part-2	7:00 pm to 8:30 pm	1.5

DBMS Schedule For GATE-2023

GATE & ESE | Plus | All courses | All languages | CS & IT

HINDI CS & IT

Complete Course on DBMS for GATE

Lesson 5 • Today, 7:00 PM

Sanchit Jain

HINDI CS & IT

Droppers Course on Operating System

Lesson 25 • Mar 23, 2022 1:30 PM

Sanchit Jain

HINDI CS & IT

Complete Course on Computer Networks for GATE

Lesson 29 • Today, 7:30 AM

Sanchit Jain

HINDI CS & IT

Orientation Course for GATE 2023 & 2024

Ended on Feb 11, 2022 • 1 lessons

Sanchit Jain

HINDI CS & IT

Foundation Course on Discrete Mathematics for GATE

Ended on Mar 8, 2022 • 25 lessons

Sanchit Jain

HINGLISH CS & IT

Foundation Course on Compiler for GATE 2022

Ended on Feb 3, 2022 • 17 lessons

Sanchit Jain

HINGLISH CS & IT

Foundation Course on Theory of Computation for GATE 2022

Ended on Feb 3, 2022 • 21 lessons

Sanchit Jain

HINGLISH CS & IT

Complete Course on Operating System for GATE

Ended on Jan 11, 2022 • 28 lessons

Sanchit Jain

HINGLISH ELECTRICAL ENGINEERING

Complete Course on Computer Architecture for GATE

Ended on Dec 13, 2021 • 19 lessons

Sanchit Jain

HINGLISH CS & IT

Complete Course on Computer Networks for GATE

Ended on Jan 7, 2022 • 27 lessons

Sanchit Jain

HINDI CS & IT

Complete Course on Digital Electronics for GATE

Ended on Nov 23, 2021 • 18 lessons

Sanchit Jain

HINDI CS & IT

Complete Course on Algorithms for GATE

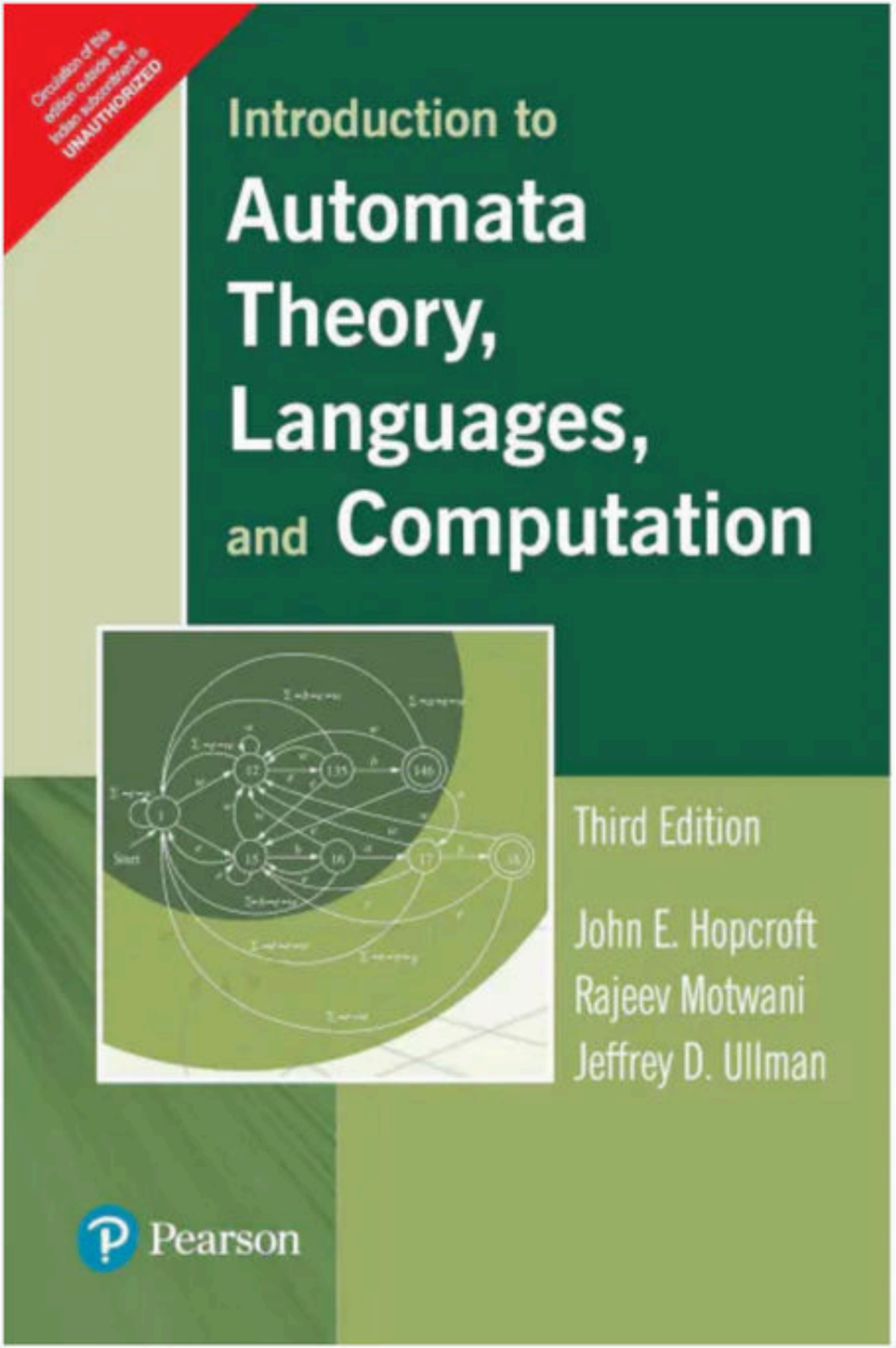
Ended on Nov 9, 2021 • 19 lessons

Sanchit Jain

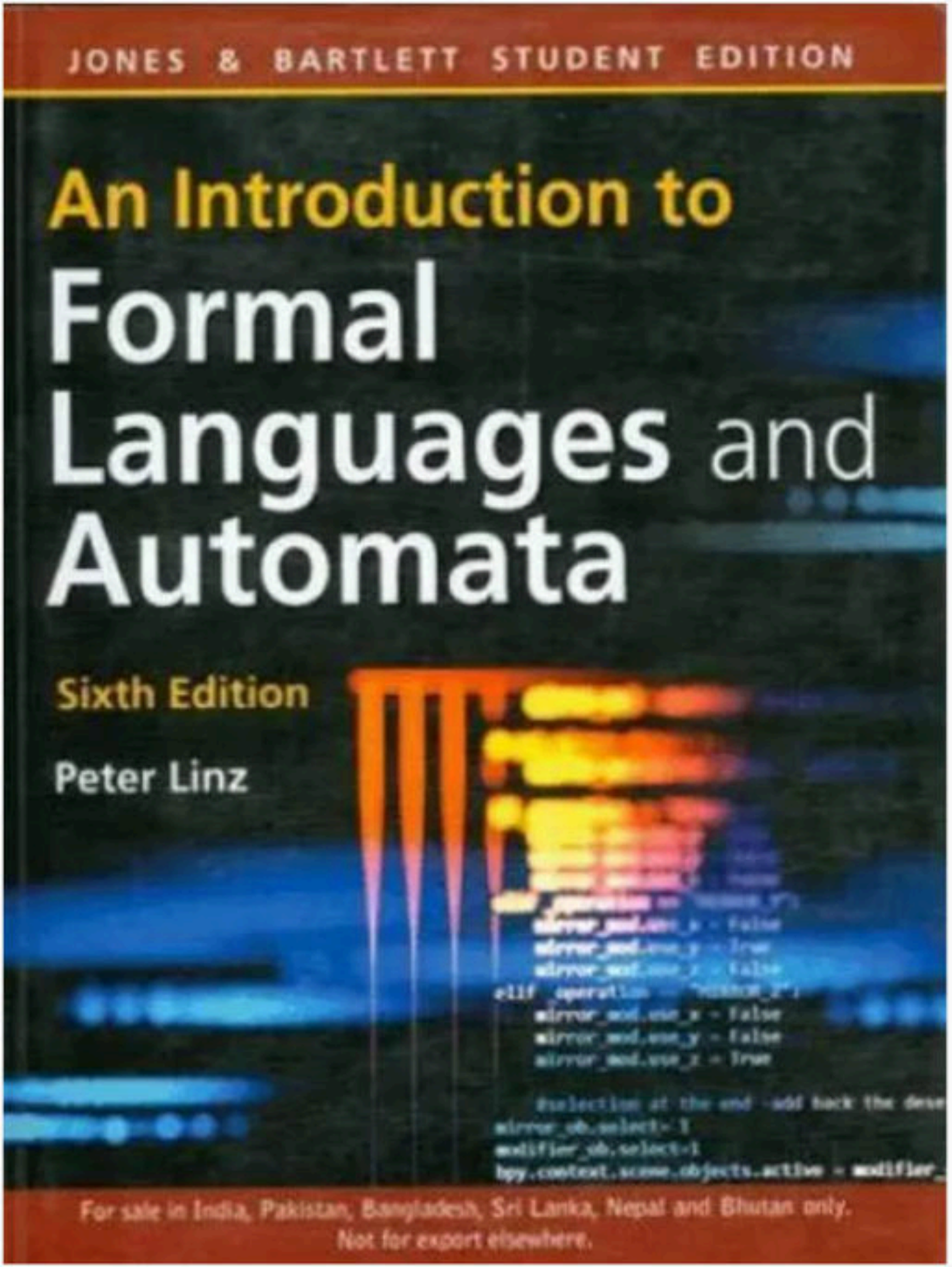
THESIS
PARTIAL
THEOREM
EXPRESSION
SOLVABLE
ALGORITHM
MATHEMATICIANS
FINITE
THEORY
CONCEPTUALLY
FORMALISM
AUTOMATA
THEORY
OF
COMPUTATION
REWRITING
TERM
MACHINE
DEFINING
MINIMALISTIC
GRAMMARS
THEORETICAL
PROGRAMMING
MATHEMATICS
STEP
LEXITY
CALCULUS
LOGIC
ASYMPTOTIC
LINEARLY
EFFICIENTLY
TERMINATES
FUNCTION
EQUIVALENT
ABSTRACT
SOLVENATURAL
RECURSIVE
SUBCLASS
HALTING
INPUT
SEQUENCE
AUTOMATON
SCIENTISTS
IDEALIZATION
INFINITE
FORMULATE
MODEL
MEMORY
REGISTER
ABSTRACTION
PARADOXES
SYMBOL
COMPUTES
SCIENCE
IMPORTANT
LAMBDA
COMPUTATIONAL
INSTRUCTIONS
LANGUAGE
REDUCIBLE
TAPE
COMPUTATION
SIZE
MU
FORMAL
DEDUCED
ARTICLE
PERFORM
REPRESENTATION

Theory of Computation

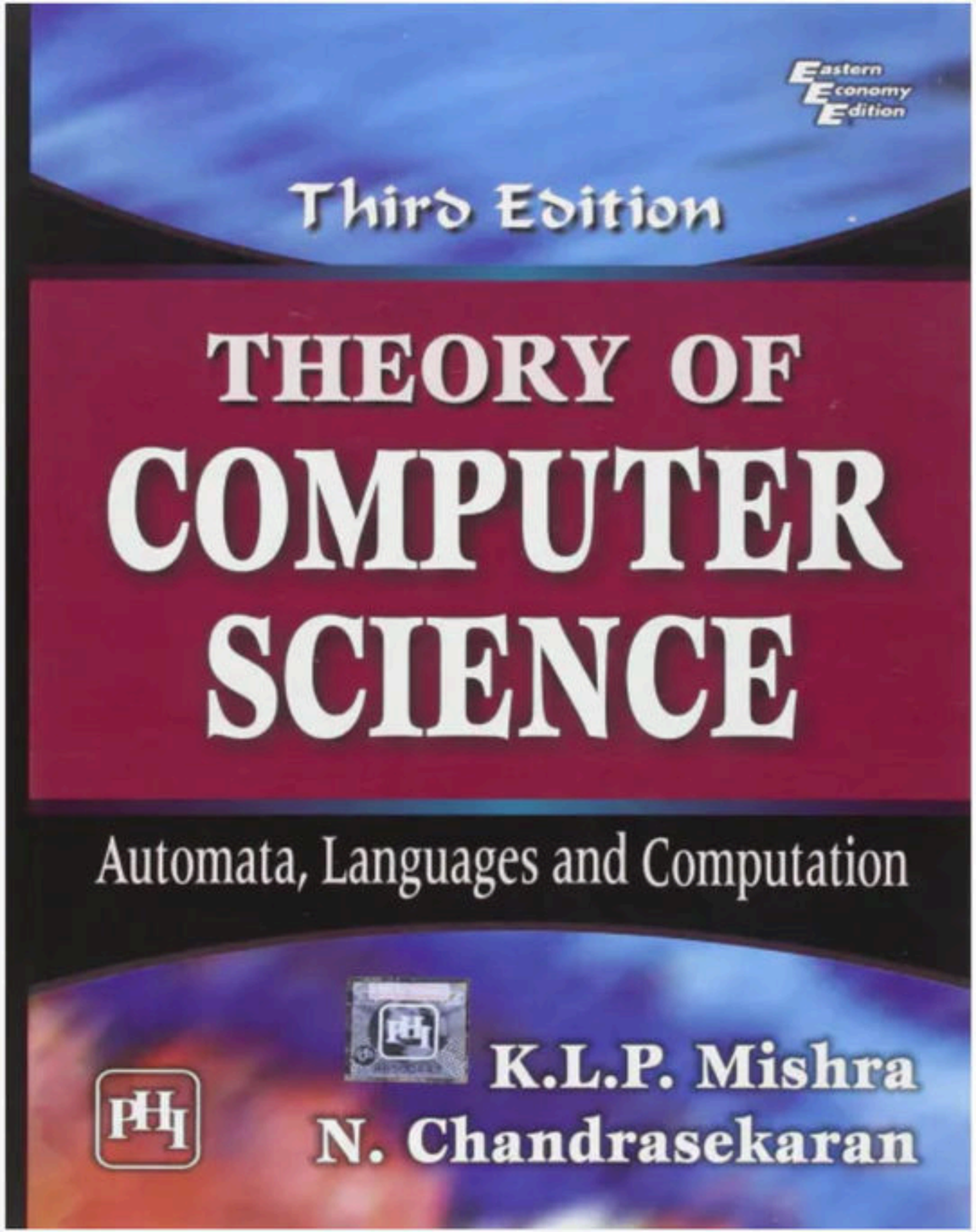
- Core subjects for CS/IT Students
- In GATE 8-10 Marks out of 100 Marks, and 6-7 questions on an average
- In NET 18-20 Marks out of 200 marks and 7-8 questions
- Both parts are important Theory and Numerical
- Needs less time, good scoring
- Not Applied in Industry



- **Book Name**
  - Introduction to Automata Theory, Languages, and Computation
- **Writers**
  - John E. Hopcroft
  - Rajeev Motwani
  - Jeffrey D. Ullman
- **Publisher** – Person
- **Edition** – 3th



- **Book Name**
  - An Introduction to Formal Languages and Automata
- **Writers** -
  - Peter Linz
- **Publisher** – Jones & Bartlett
- **Edition** – 6th



- **Book Name**
  - Theory of Computer Science: Automata, Languages and Computation
- **Writers** -
  - K.L.P. Mishra
- **Publisher** – PHI
- **Edition** – 3th

Syllabus

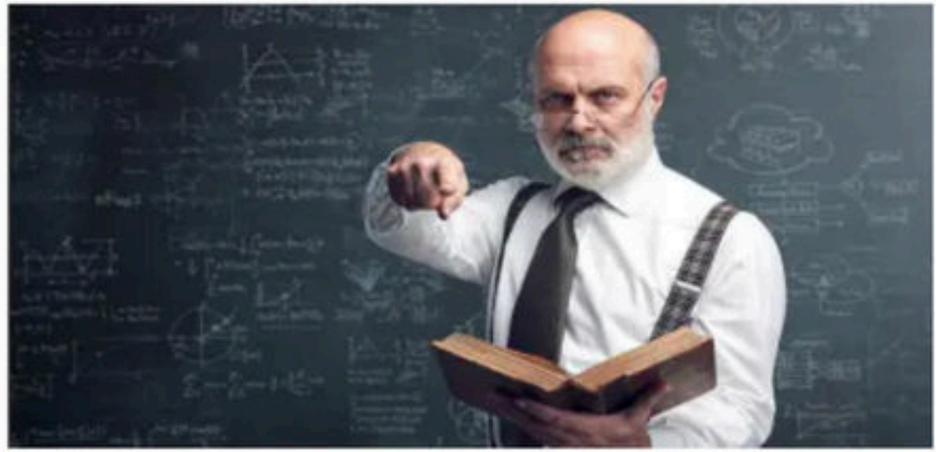
- Regular expressions and finite automata
- Context-free grammars and push-down automata
- Regular and context - free languages
- Pumping lemma
- Turing machines and undecidability

What you can expect from me

- Will take care of theory and numerical both
- Will give more weightage to the topics that are asked more frequently in GATE
- Will not emphasize more than required, on a topic
- Will provide PDF of related books

What i expect from you

- There is no hurry, feel free to ask questions any time through out the class, but first listen
- Please revise the entire lecture before and after the class
- Be regular, Consistency is most important
- More you practice, more clarity you will get
- If we follow all the above specified points Success in guaranteed



Break

INTRODUCTION TO THEORY OF COMPUTATIONS

- As word suggests ‘TOC’ is the study of 'mathematical' machines or systems called automata.
- It is the study of computational problems that can and cannot be solved using these machines, i.e. what is the extent to which a problem is solvable on a computer.
- Theory of computation can be considered as the study of all kinds of computational model in the field of computer science and it also considers how efficiently the problem can be solved (but not is depth).

PROBLEM

- Now a day's machines (digital, analog, mechanical) play a very important role in the development of human, we need some mechanism (language) to communicate with the machines.

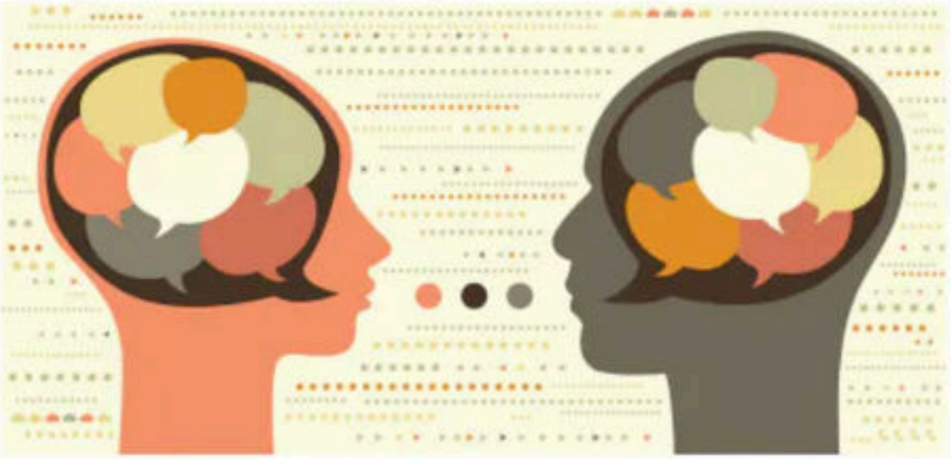
SOLUTION

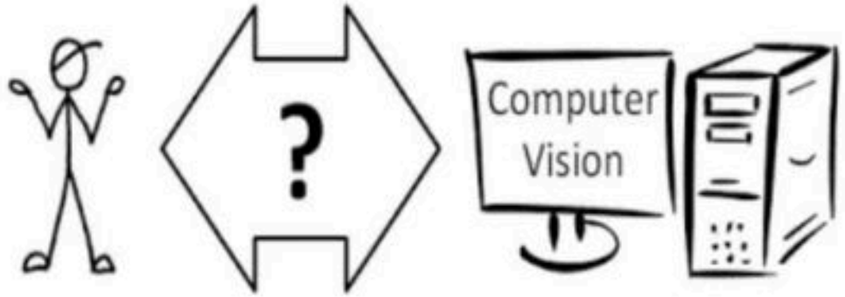
- We need a language for communication with machines.

- But we do not require natural languages to communicate with the machines, as natural languages are very complex and machine interaction require very fewer complex languages compare to natural languages.

- Languages can be of two types formal languages and informal languages, here in this subject we will only discuss formal languages.

- Dictionary defines the term informally as 'a system suitable for the expression of certain ideas, facts or concepts including a set of symbols for their manipulation'.





Break

MATHEMATICL DEFINATION OF LANGUAGE

- **SYMBOL**- Symbols are the basic building blocks, which can be any character/token. (cow, sheep, 🔵, white flag, ✸ , 🛇 , ✚ etc.) (in English we called them as letters).

- **ALPHABET**- An alphabet is a finite non empty set of symbols, (every language has its own alphabet). here in toc, we use symbol Σ for depicting alphabet. e.g. Σ = {0,1}. for English Σ = {a, b, c, …., z} (in English also alphabet is a set of letters, thought in general we called them as alphabet).

- **STRING** - It is a sequence of symbols (which are the member of set alphabet). E.g. Σ = {a, b} String- aabb, aa, b, so on. (in English we called them as words).

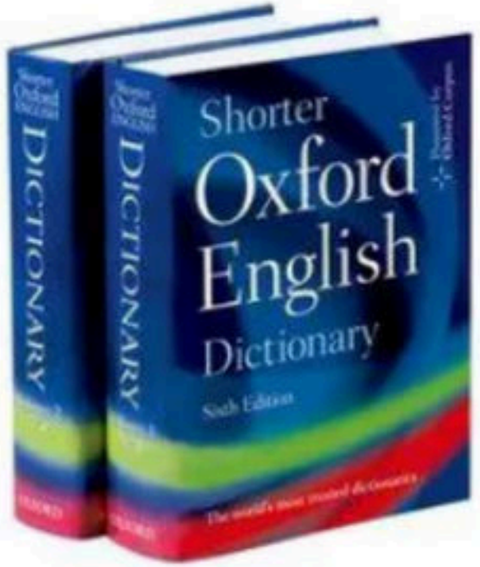
- **LANGUAGE** - A language is defined as a set of strings. (in natural language (set of words(predefined) and grammar) we apply this model from words to sentence).

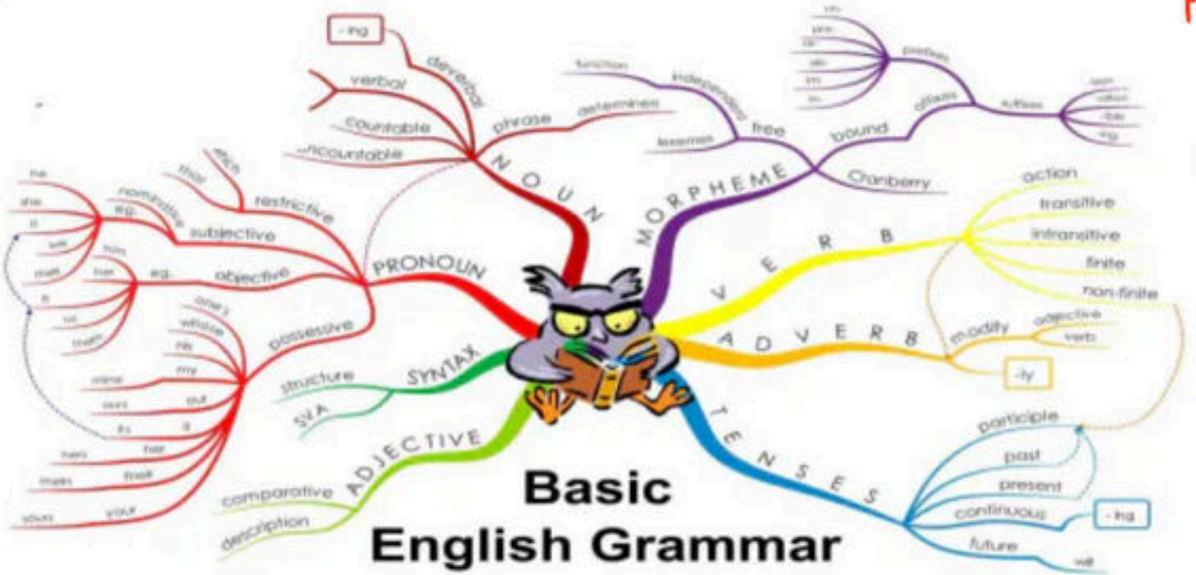
- In the next level we consider programs as a string and programming constructs/tokens like int, floats as letters/symbols.

METHODS TO DEFINE LANGUAGE

- In natural language we define the list of words in a dictionary because they are finite and predefined, but we cannot list all the sentence which can be formed using these words as they are infinite.

A → S





- So, we have a mechanism called grammar/rules using which we can decide which sentence is valid and which is invalid.

- Similarly, in our system we have finite number of symbols/letters but using those letters we can generate infinite strings/words.

- So, we may have languages that have infinite number of words, so it is not possible for us to list them, we have to use some framework, which can somehow represent the same language.
  - There are mainly two methods to represent a language
  - by a grammar that generates a language [RG generate RL]
  - by a machine that accepts a language [FA accept RL language]

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

$L = \{ab, aa\}$

$G \longrightarrow L \longrightarrow M$

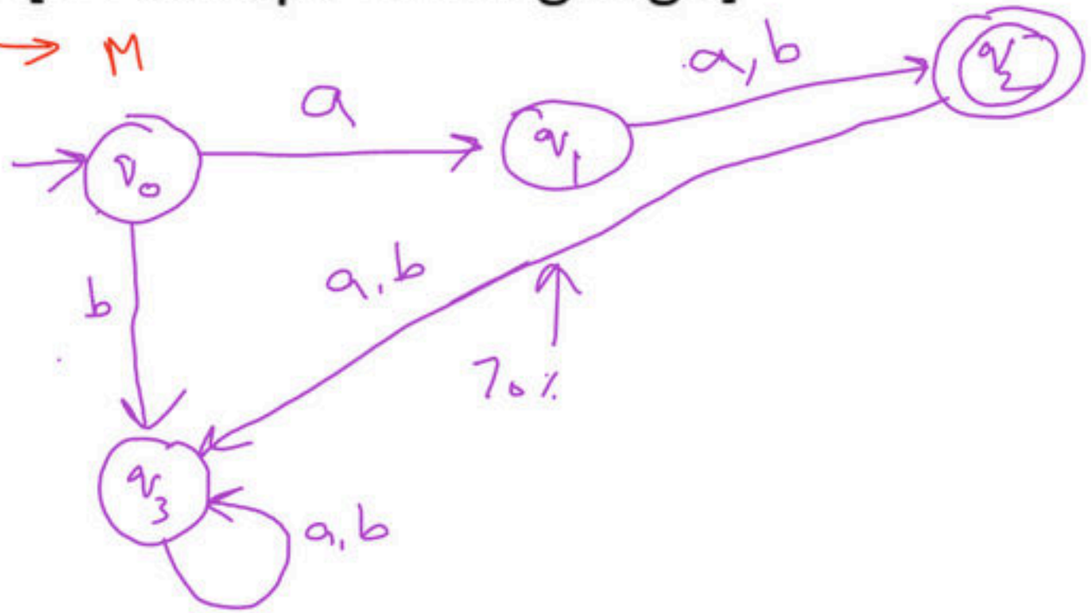
$S \rightarrow AC$

$C \rightarrow B / A$

$A \rightarrow a$

$B \rightarrow b$

$\longrightarrow L \longrightarrow$



Break

SOME BASIC OPERATIONS ON STRINGS

- So, before we proceed further let's do a little home work on string, which will help us throughout the subject.

- **Length of a string** - It is defined as number symbol in the string. Denoted like |W|, e.g. length of string |00110| = 5. |w|

- |aaba| = 4

- |010| = 3

Concatenation of string- Let x and y be two strings, then concatenation is defined as the string formed by making a copy of string x followed by a copy of string y. (NOTE- It's not commutative)

E.g. w = ab, x= ba

$w_1 \cdot w_2 \neq w_2 \cdot w_1$

$n_1 + n_2$

$\times$

wx = abba

xw = ~~baab~~

$w = w_1 w_2 w_3 \ldots\ldots w_m$

$x = x_1 x_2 x_3 \ldots\ldots x_n$

wx = $w_1 w_2 \cdots w_m \cdot x_1 x_2 \cdots x_n$

$|wx| = |w| + |x|$

Reverse of a string – If there is a string w then reverse of a string a denoted by w^r it is just the same string but written in reverse order.

$w = abbb$

$W = w_1 w_2 w_3 \ldots\ldots w_n$

$w^R = bbba$

$w^r = w_n w_{n-1} \ldots\ldots w_3 w_2 w_1$

$|w| \boxed{=} |w^r|$

Empty/Null String- The string with zero occurrence of symbols. It is denoted by ∈, |∈|=0. If there is a string w, then w^n stands for the strings obtained by repeating w, n times.

$w^3 = www$

$w^2 = ww$

$w^1 = w$

$w^0 = \in$

$w\in = \in w = w$

$\in / \lambda / \Lambda$

$|\in| = 0$

$w^2 = w.w$

$2^2 = 2 \times 2$

$2^1 = 2 \times 2$

$2^0 = 2$

$2^0 = 1$

$a^n = \underbrace{a \times a \times a \cdots a}_{n}$

$A \cup \phi = A$

$a \times 0 = a$

$a + 0 = a$

$M \cdot I = M$

$A = \{\ \} / \phi$

$Z/2$ $Z/6$ $Z/3$

$\in$

0

0

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

$\Sigma^2 = \{a, b\} \cdot \{a, b\}$

$= \{aa, ab, ba, bb\}$

Substring- Any string of consecutive symbols in some string 'w' can be collectively said as a substring. E.g. w= abab its substrings can be ab, a, ba…etc.

S	u	b	s	t	r	i	n	g

1. ~~utg~~
2. ~~sbr~~
3. ~~rts~~
4. str
5. sub
6. ∈

Break

Q Consider a sting ‘GATE’ find the total number of substring possible?

Boy

Q Consider a sting 'GATE' find the total number of substring possible?

	Substring of length 0	Substring of length 1	Substring of length 2	Substring of length 3	Substring of length 4
Number of sting possible	1	4	3	2	1
SUBSTRINGS	~~ε~~	G A T E	GA AT TE	GAT ATE	~~GATE~~

11

$|w| = n$

$\frac{n(n+1)}{2} + 1$

Q Consider a sting 'GATE' find the total number of substring possible?

	Substring of length 0	Substring of length 1	Substring of length 2	Substring of length 3	Substring of length 4
Number of sting possible	1	4	3	2	1
SUBSTRINGS	∈	G, A, T, E	GA, AT, TE	GAT, ATE	GATE

Q Consider a sting 'GGGE' find the total number of substring possible?

	Substring of length 0	Substring of length 1	Substring of length 2	Substring of length 3	Substring of length 4
Number of sting possible	1	2	2	2	1
SUBSTRINGS	∈	G, E	GG GE	GGG GGE	GGGE

= 8

- If a string has ‘n’ distinct symbols then total number of different sub string will be [n(n+1)/2] +1

- If w is any string than empty string ∈ and the string w itself is called a trivial substring and the remaining of the other are the non-trivial sub string[[n(n+1)/2]-1]

$$\left[\frac{n(n+1)}{2}+1\right]-2$$

$$\frac{n(n+1)}{2}-1$$

Break

Q Consider a sting 'GATE' find the total number of prefix and suffix possible?

Prefix(GATE) = { ε, G, GA, GAT, GATE }

$|w| = n \longrightarrow \underline{n+1}$

Suffix(GATE) = { ε, E, TE, ATE, GATE }

going

- If there is a string of length n then no of prefix or suffix will be n+1

Break

Q if $\Sigma = \{a, b\}$ then, find the following?

$\Sigma^0 = \{\epsilon\}$

$\Sigma^1 = \{a, b\}$

$\Sigma^2 = \{aa, ab, ba, bb\}$

$\Sigma^3 = \{aaa, aab, \ldots, bbb\}$

$\Sigma^k = \{w \mid |w| = k\}$

$\Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots \cup \Sigma^k \cup \ldots \Sigma^\infty$

$= \{w \mid |w| \leq k\}$

Σ^*

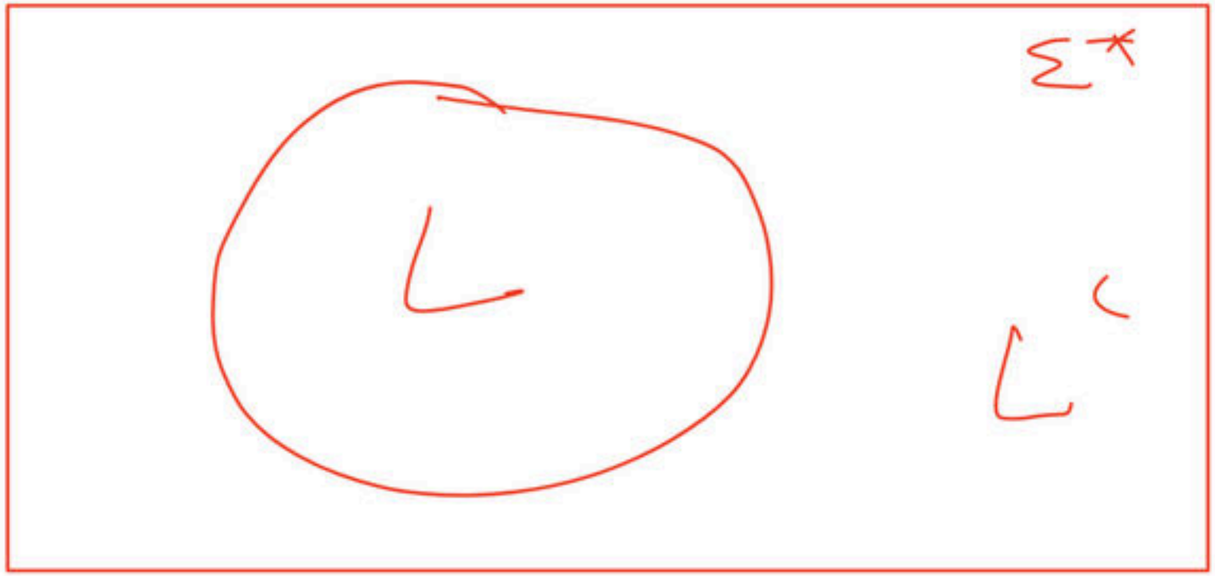
- $\sum^K$ is the set of all the strings from the alphabet $\sum$ of length exactly K.

- $\sum^k = \{W \mid |W| = K\}$ (using the symbols from the alphabet $\sum$)

Kleene closure- If ∑ is a set of symbols, then we use ∑* to denote the set of strings obtained by concatenating zero or more symbols from ∑ of any length, in general any string of any length which can have only symbols specified in ∑.

- $\Sigma^* = \bigcup_{i=0}^{i=\infty}\{w \mid |w| = i\}$ (using the symbols from the alphabet ∑)

$L \subseteq \Sigma^*$



<u>Positive closure</u> – If $\sum$ is a set of symbols, then we use $\sum^+$ to denote the set of strings obtained by concatenating one or more symbols from $\sum$ of any length, in general any string of any length which can have only symbols specified in $\sum$ (except $\in$).

- $\sum^+ = \bigcup_{i=1}^{i=\infty}\{w \mid |w| = i\}$ (using the symbols from the alphabet $\sum$)

Break

LANGUAGES

- Since languages are sets, the union, intersection and difference of two languages are immediately defined.

- The complement of a language is defined with respect of $\sum^*$
  - $L^c =$

- The reverse of a language is the set of all the strings after reversal

  $L^r =$

- The concatenation of two languages L_1 and L_2 is the set of all the strings obtained by concatenating any elements of L_1 and L_2.
  - L_1L_2 =

- Let L^n is defined as L is concatenated with itself n times.
  - L^n =

  - L^* =

Q Given the language L = {ab, aa, baa}, which of the following strings are in L*? **(GATE-2012) (1 Marks)**

1) abaabaaabaa

2) aaaabaaaa

3) baaaaabaaaab

4) baaaaabaa

(A) 1, 2 and 3
(B) 2, 3 and 4
(C) 1, 2 and 4
(D) 1, 3 and 4

Q The number of substrings that can be formed from string given by 'adefbghnmp' is? **(NET-DEC-2018)**

a) 10 **b)** 45 **c)** 55 **d)** 56

Q In a string of length n, how many proper prefixes can be generated

a) 2^n **b)** n **c)** n (n + 1)/2 **d)** n – 1

Q The number of substrings (of all lengths inclusive) that can be formed from a character string of length n is

a) n!

b) n^2

c) $\frac{n(n-1)}{2}$ -1

d) $\frac{n(n+1)}{2}$ +1

Q Suppose L_1 = {10, 1} and L_2= {011, 11}. How many elements are there in L = $L_1 L_2$

a) 4 **b)** 3 **c)** 2 **d)** None of these

Q Let x_n denote the number of binary strings of length n that contain no consecutive 0^s, which of the following recurrences does x_n satisfy?

a) $x_n = 2x_{n-1}$

b) $x_n = x_{\left[\frac{n}{2}\right]} + 1$

c) $x_n = x_{\left[\frac{n}{2}\right]} + n$

d) $x_n = x_{n-1} + x_{n-2}$

Q The value of x_5 is?

a) 5

b) 7

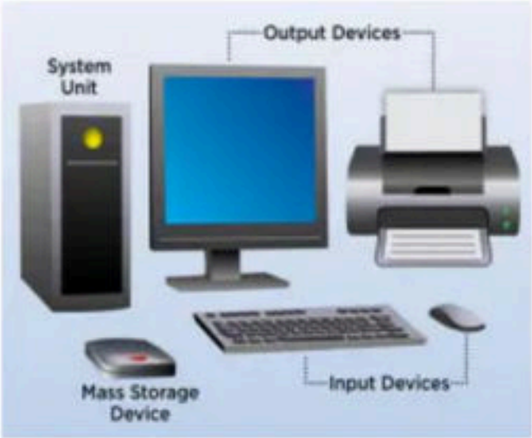
c) 13

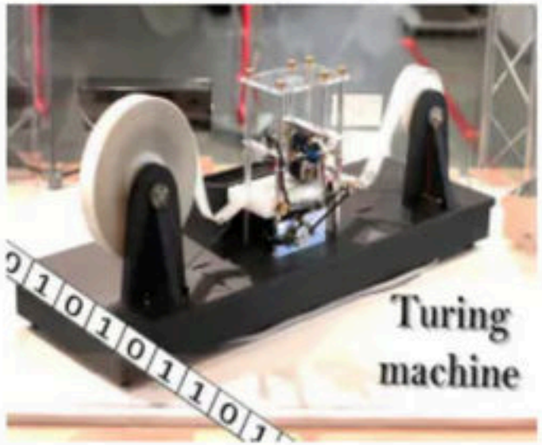
d) 16

Automaton

- An automaton is defined as a system where energy, materials and information are transformed, transmitted and used for performing some functions without direct participation of man.

- Example: automatic machine tools, automatic packing machines, and automatic photo printing machines.





FINITE AUTOMATA

- A **Finite automaton** is a model that has a finite set of states (represented in the figure by circles) and its control moves from one state to another state in response to external inputs (represented by arrows).

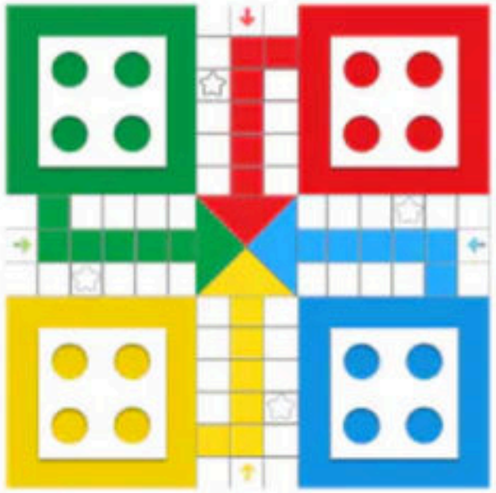
- Finite automata can be broadly classified into two types-

1. **Finite automata without output**
    1. Deterministic finite automata.
    2. Non deterministic finite automata.
    3. Non deterministic finite automata with $\in$

2. **Finite automata with output**
    1. Moore machine
    2. Mealy machine

- In general, this type of automata is characterized by machine having no temporary storage, as it is severely limited in its capacity to remember things during the computation.

- Only finite amount of information can be in the control unit by placing the unit into a specific state but since the number of states is finite, a finite automaton can only deal with situation in which the information to be stored at any time is strictly bounded.

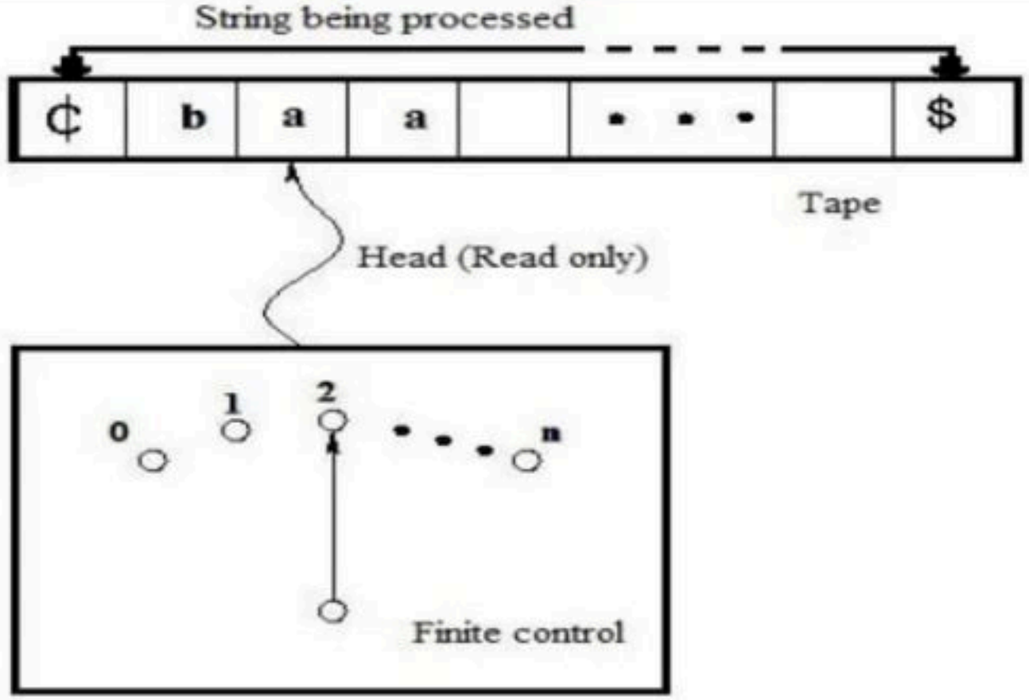


DETERMINISTIC FINITE AUTOMATA

A **deterministic finite automaton (DFA)** is defined by 5-tuple (Q,Σ,δ,S,F) where:

- Q is a finite and non-empty set of states
- Σ is a finite non-empty set of finite input alphabet
- δ is a transition function, ($\delta: Q \times \Sigma \rightarrow Q$)
- S is **initial state** (always one) ($S \in Q$)
- F is a set of final states ($F \subseteq Q$) ($0<=|F|<=N$, where n is the number of states)

- The various components of the block diagram are explained as follows:
- **Input tape**: The input tape is divided into squares, each square containing a single symbol from the input alphabet ∑. The end squares of the tape contain the end marker ¢ at the left end and the end marker $ at the right end. The absence of end markers indicates that the tape is of infinite length. The left-to-right sequence of symbols between the two end markers is the input string to be processed.



Finite Automaton

- **Reading head (R-head)**: The head examines only one square at a time and will move one square to the right.

- **Finite control:** is the inference engine take care of transition

NOTE-

- Produces a unique computation (or run) of the automaton for each input string.
- Deterministic refers to the uniqueness of the computation.
- DFAs are useful for doing lexical analysis (Spell check) in compiler design.

Break

Representation

TRANSITION STATE DIAGRAM- (Graphical, easy to understand, easy to design) A transition graph or a transition system is a finite directed labelled graph in which each circle represents a state and the directed edges indicate the transition of one state to another state. The initial state is represented by a circle with an arrow pointing towards it, the final state by two concentric circles.

TRANSITION TABLE- it is a two-dimensional table where number of columns is equal to number of input alphabets and number of rows is equal to number of states.

δ		Σ	
		a	b
Q	q_0		
	q_1		

TRANSITION ID- $\delta\ \{q_i, a) = q_j$, this means that on a state q_i take an input symbol a machine will make a transition q_j

ACCEPTANCE BY DFA

- Let 'w' be any string designed from the alphabet Σ, corresponding to w, if there exist a transition for which it starts at the initial state and ends in any One of the final states, then the string 'w' is said to be accepted by the finite automata. $\delta^*(q_0, w) = q_f$ for some $q_f \in F$.

- Mathematically, it can be represented as: - $L(M) = \{w \in \Sigma^* \mid \delta^*(S, w) \in F\}$

Q Consider the following table of an FA

IF the final state is q_4, the which of the following strings will be accepted?

1. aaaaa

2. aabbaabbbbb

3. bbabababbb

δ	a	b
start	q_1	q_0
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_2
q_3	q_4	q_3
q_4	q_4	q_4

a) 1 and 2

b) 2 and 3

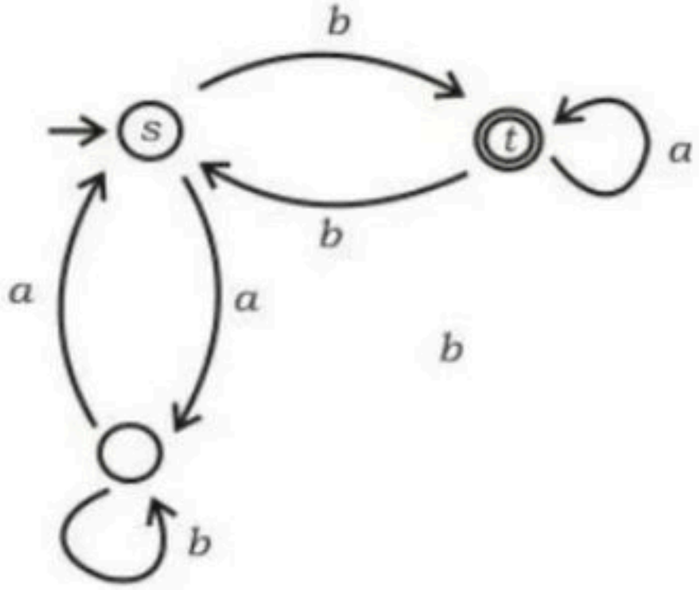
c) 3 and 1

d) all of the above

Q In the automaton below, s is the start state and t are the only final state. **(GATE-2006) (2 Marks)**

Consider the strings u = ababab, v = bab, and w = aabb.

Which of the following statements is true?



(A) The automaton accepts u and v but not w

(B) The automaton accepts each of u, v, and w

(C) The automaton rejects each of u, v, and w

(D) The automaton accepts u but rejects v and w

Break

DFA Designing

- A language a said to be regular language if it can be accepted by a DFA.

- The best knowledge about DFA can be only understood by designing a number of DFA, by doing so we will first understand the process of DFA designing and secondly, we will understand Regular Language.

Q Design a minimal DFA that accepts a language ‘L’, where L={a} over the alphabet ∑={a}.

Q design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, where every accepted string 'w' starts with substring s

i) s = b

ii) s = ab

iii) s = abb

Conclusion

- if $w = sx$, $|s| = m$, then no of states in the DFA is $m+2$
- Dead state is required
- Loop on final state

Q design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$. where every accepted string 'w' ends with substring 's'.

i) s = ab

ii) s = aa

iii) s = bab

Conclusion

- if $w = xs$, $|s| = m$, then no of states in the DFA is $m+1$
- No need of dead state
- Can not stay of final state for ever

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$. where every accepted string 'w' contains sub string s.

i) abb

ii) aba

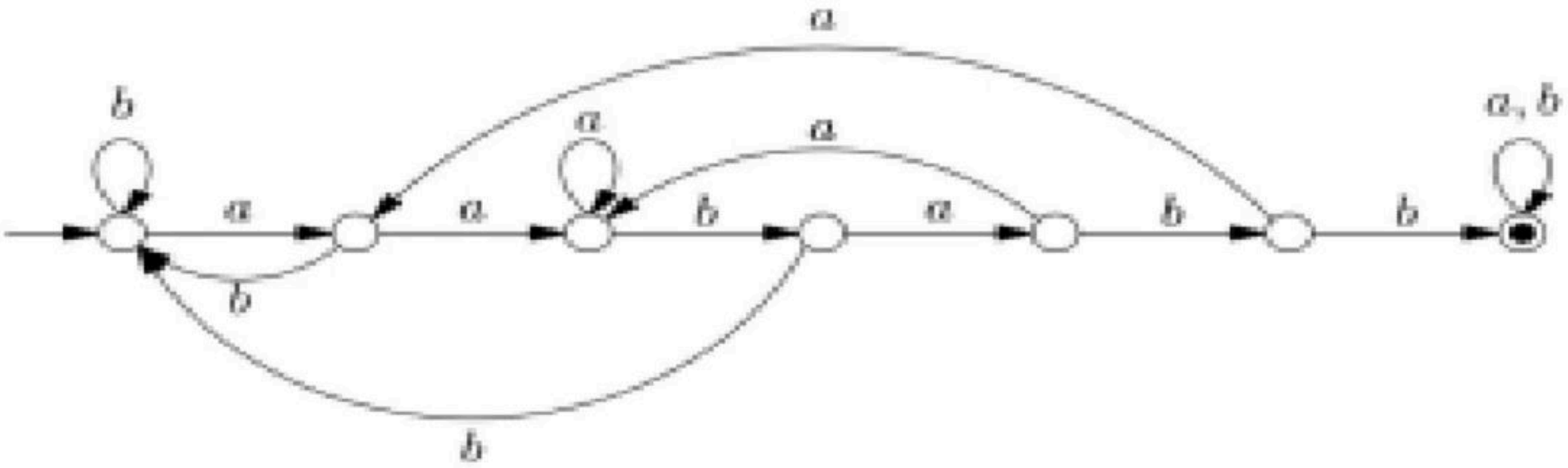
Conclusion

- if w = xsx, |s| = m, then no of states in the DFA is m+1
- No need of dead state
- Can loop on final state

Q No of states in a minimal dfa that accepts all the strings of a, b such that every string must have ‘aabab’ as sub string

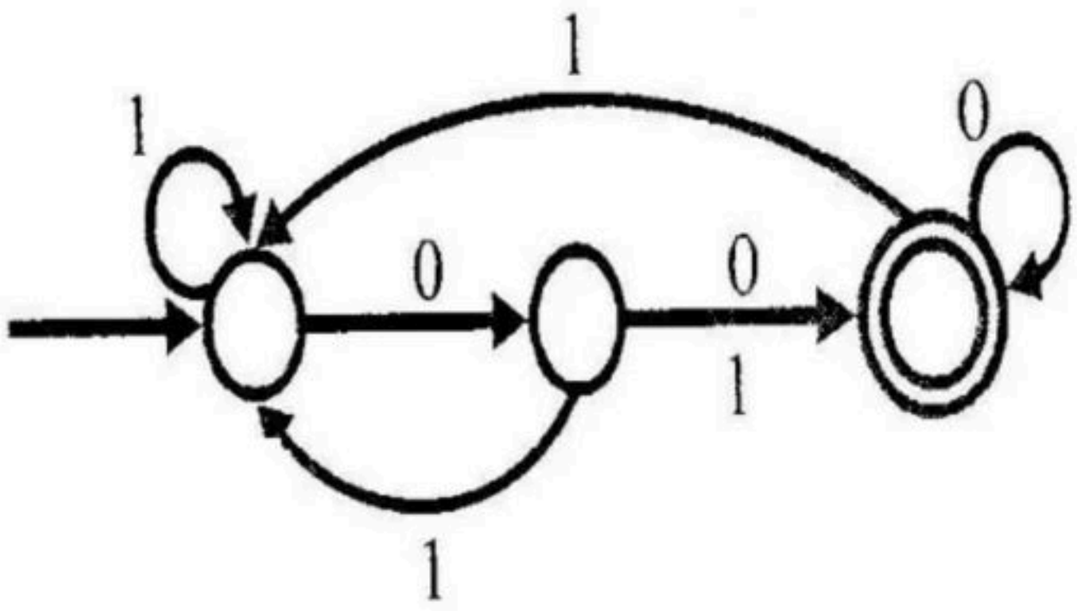
a) 5 **b)** 6 **c)** 7 **d)** none

Q Consider the following Deterministic Finite Automata **(GATE-2015) (2 Marks)**
Which of the following is true?



(A) It accepts all strings with prefix as “aababb”
(B) It accepts all strings with substring as “aababb”
(C) It accepts all strings with suffix as “aababb”
(D) None of the above

Q The following DFA accepts the set of all strings over {0, 1} that **(GATE-2009) (1 Marks)**



a) Begin either with 0 or 1

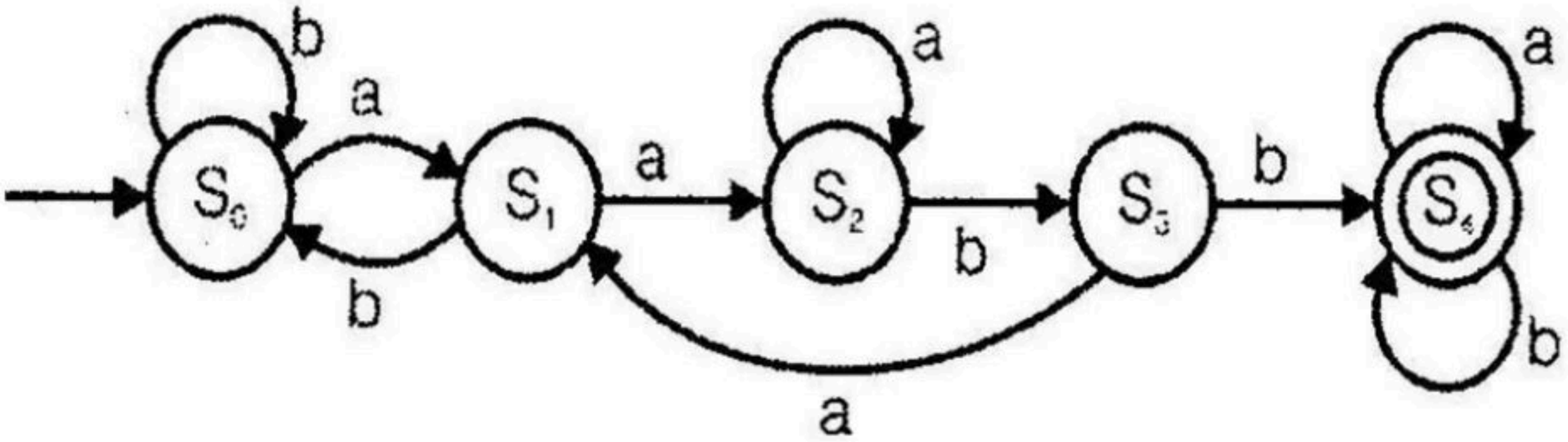
b) End with 0

c) End with 00

d) Contain the substring 00

Q Consider the following machine M

What is the language L(M) accepted by this machine?



a) L(M) = {Set of all words starting with aabb}

b) L(M) = {Set of all words having aabb as a sub word}

c) L(M) = {Set of all words ending with aabb}

d) L(M) = {Set of all words with exactly one occurrence of aabb}

Break

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$ such that every accepted string start and end with a.

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$ such that every accepted string start and end with same symbol.

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$ such that every accepted string start and end with different symbol.

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$ such that every accepted string w, is like w=SX.

i) s= aa/bb

ii) s=aaa/bbb

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$ such that every accepted string w, is like w=XS.

i) s= aa/bb

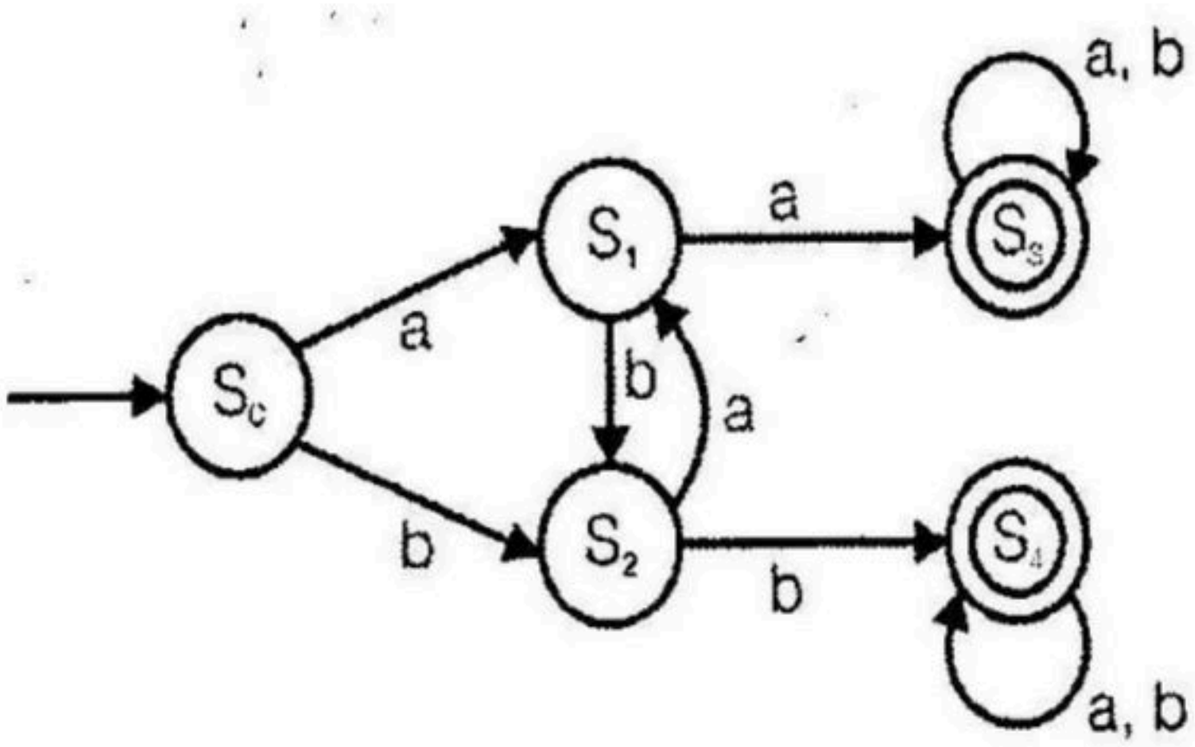
ii) s=aaa/bbb

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$ such that every accepted string w, is like w=XSX.

i) s= aa/bb

ii) s=aaa/bbb

Q Consider the machine M shown below L(M)?



a) L(M) = {words starting with aa or bb}

b) L(M) = {words ending with aa or bb}

c) L(M) = {words containing aa or bb as a sub word}

d) None of these

Q Let L be the set of all binary strings who's last two symbols are the same. The number of states in the minimum state deterministic finite 0 state automation accepting L is **(GATE-1998) (1 Marks)**

a) 2 **b)** 5 **c)** 8 **d)** 3

Consider the transition diagram of an DFA as given below:

Q Which should be the final state(s) of the DFA if it should accept strings starting with 'a' and ending with 'b'?

a) q_0 **b)** q_1 **c)** q_0, q_1 **d)** q_3

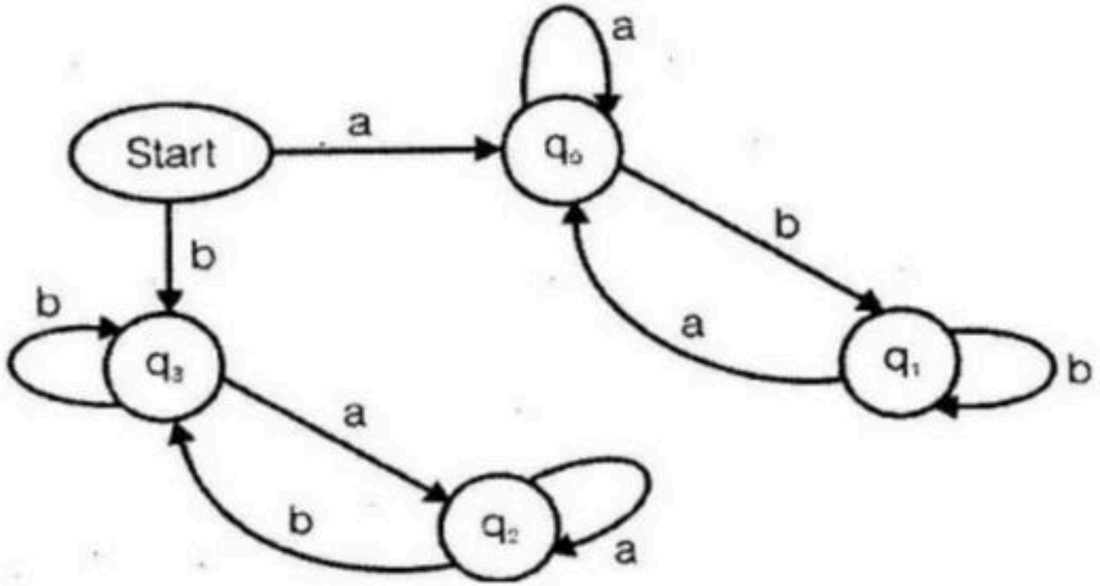
Q Which of the following strings will be accepted if q_0 and q_1 are accepting states?

1. ababab **2.** Babaaa **3.** aaaba

a) 1,2 **b)** 2,3 **c)** 1,3 **d)** None of these

Q Which of the following represents the set of accepting states if the language to be accepted contains strings having the same starting and ending symbols?

a) $\{q_0\}$ **b)** $\{q_0, q_3\}$ **c)** $\{q_3\}$ **d)** $\{q_3, q_1\}$



Break

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string 'w' accepted must be like

i) $|w| = 3$

ii) $|w| <= 3$

iii) $|w| >= 3$

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string accepted must contain exactly two a's, $|w|_a=2$

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string accepted must contain at least two a's, $|w|_a >= 2$

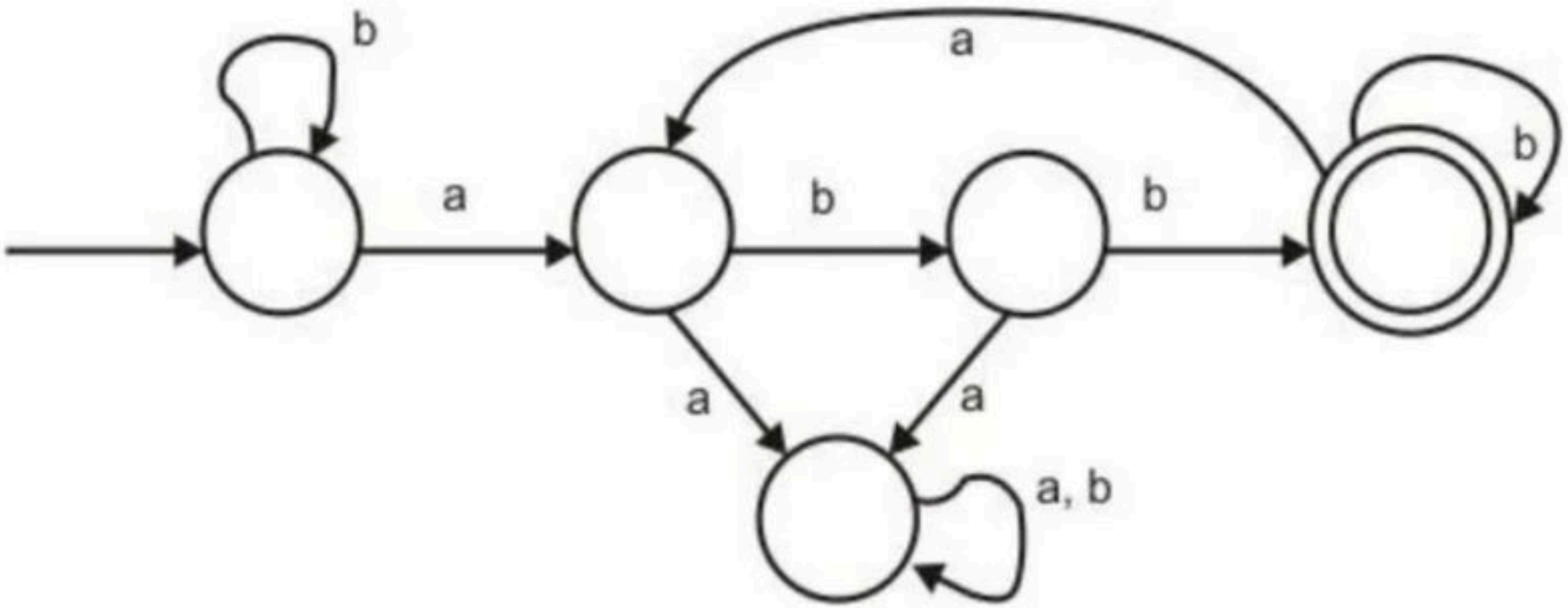
Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string accepted must contain at most two a's, $|w|_a <= 2$

Q The minimum possible number of states of a deterministic finite automaton that accepts a regular language L is_____**(GATE-2017) (2 Marks)**

$L = \{w_1 a w_2 \mid w_1, w_2 \in \{a, b\}^*, |w_1| = 2, |w_2| >= 3\}$

Q Consider the machine M **(GATE-2005) (2 Marks)**

The language recognized by M is :



(A) {w ∈ {a, b}* / every a in w is followed by exactly two b's}

(B) {w ∈ {a, b}* every a in w is followed by at least two b'}

(C) {w ∈ {a, b}* w contains the substring 'abb'}

(D) {w ∈ {a, b}* w does not contain 'aa' as a substring}

Break

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string 'w' accepted must be like

i) $|w| = 0 \pmod 3$

ii) $|w| = 1 \pmod 4$

Conclusion

No of states will be n, if the equation is on format r (mod n)

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string accepted must contain

i) number of a is, $|w|_a = 0 (\text{mod } 2)$

ii) number of b is, $|w|_b = 1 (\text{mod } 2)$

iii) number of a is $|w|_a = 2 (\text{mod } 3)$

iv) number of a is $|w|_a = 1 (\text{mod } 4)$

Q How many minimum states are required in a DFA to find whether a given binary string has odd number of 0's, there can be any number of 1's. **(GATE-2015) (1 Marks)**

(A) 1 **(B)** 2 **(C)** 3 **(D)** 4

Q The smallest finite automation which accepts the language {x | length of x is divisible by 3} has **(GATE-2002) (2 Marks)**

(A) 2 states **(B)** 3 states **(C)** 4 states **(D)** 5 states

Q Let $\sum = \{0, 1\}$, then an automaton A accepting only those words from $\sum$ having an odd number of 1's requires _____ states including the start state

a) 2 **b)** 3 **c)** 4 **d)** 5

Q The smallest finite automaton which accepts the language {x | length of x is divisible by 500} has

a) 200 states **b)** 300 states **c)** 400 states **d)** 500 states

Break

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string accepted must contain odd occurrence of the substring 'ab'.

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string accepted must contain even occurrence of the substring 'baa'.

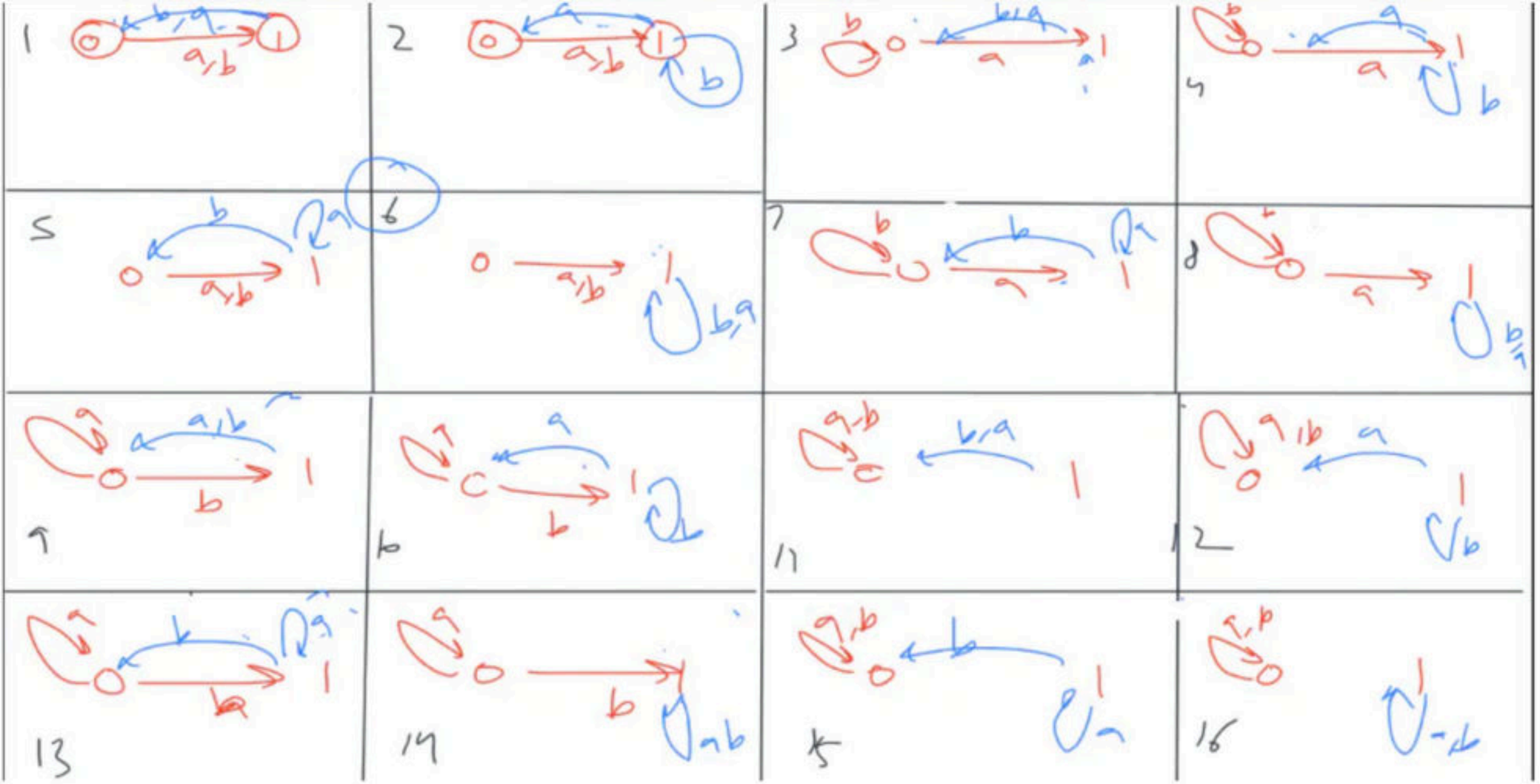
Q how many DFA can be designed, with a fixed initial state, over an alphabet $\sum = \{a, b\}$, and the number of states be two.

Q There are exactly _____ different finite automata with three states x, y and z over the alphabet {a, b} where x is always the start state. **(NET-DEC-2015)**

a) 64 **b)** 56 **c)** 1024 **d)** 5832

Conclusion

- If no of states is $|Q| = n$, and no of input alphabet $|\sum| = m$
- Then no of states will be $n^{n*m} * 2^n$


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Q how many DFA can be constructed over the alphabet, such that no of states $|Q| = 2$ and size of input alphabet is $|\sum| = 2$, that accepts empty language(Φ)?

Q how many DFA can be constructed over the alphabet, such that no of states $|Q| = 2$ and size of input alphabet is $|\sum| = 2$, that accepts universal language ($\sum^*$)?

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$ such that for every accepted string 2nd from left end is always b.

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$ such that for every accepted string 4th from left end is always a.

Conclusion

- No of states will be n+2, if position is n

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$ such that for every accepted string 2nd from right end is always b.

Q The no of final states in a minimal DFA, that accept all string over a, b such that 2nd symbol from right end is always b.

a) 1 **b)** 2 **c)** 4 **d)** none

Conclusion

- No of states will be 2^n, if position is n

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string 'w' which is accepted starts with 'a' & length is divisible by 3, i.e. 0(mod 3)?

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string 'w' which is accepted starts with 'ab' & length is divisible by 2(mod 4)?

Conclusion

- No of states will be n, n = first condition + second condition -1

Q No of states in a minimal DFA that accept all string over a, b such that every string starts with ‘abb’ and length is concurrent to 4 (mod 7) is _____

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string 'w' which is accepted starts with 'abbb' & length is divisible by 3(mod 15)?

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string accepted must contain be like,
no of a = 0(mod 2) || no of b = 0(mod 2) ?

cross product method(one more approach , where concentrate on even and odd logic)

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string accepted must contain be like, no of a = 0(mod 2) && no of b = 0(mod 2)?

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b, c\}$, such that every string accepted must contain be like, a = 0(mod 2) && no of b = 0(mod 3) && no of c = 0(mod 5)?

Conclusion

- if, n(mod m) && p(mod r), then the no of states will be n*m, if n & m are relatively prime to each other

Break

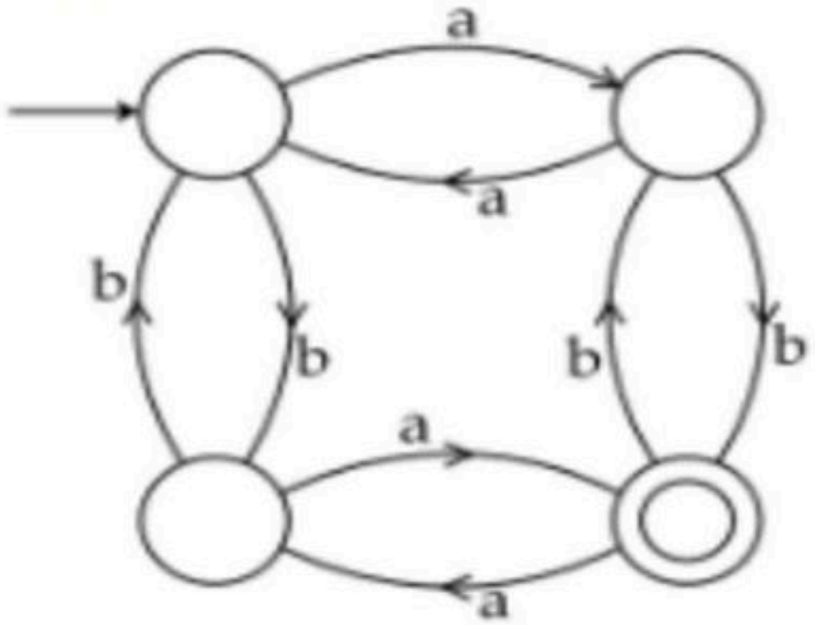
Q The finite state machine given in figure below recognizes: **(NET-JULY-2018)**

(a) any string of odd number of a's

(b) any string of odd number of b's

(c) any string of even number of a's and odd number of b's

(d) any string of odd number of a's and odd number of b's



Q The transition function for the language L = {w | n_a (w) and n_b(w) are both odd} is given by:

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_2$

$\delta(q_1, a) = q_0$

$\delta(q_1, b) = q_3$

$\delta(q_2, a) = q_3$

$\delta(q_2, b) = q_0$

$\delta(q_3, a) = q_2$

$\delta(q_3, b) = q_1$

The initial and final states of the automata are: **(NET-JUNE-2015)**

A) q_0 and q_0 respectively

b) q_0 and q_1 respectively

c) q_0 and q_2 respectively

d) q_0 and q_3 respectively

Q A minimum state deterministic finite automaton accepting the language L={w | w ε {0,1} *, number of 0^s and 1^s in w are divisible by 3 and 5, respectively} has **(GATE-2007) (2 Marks)**

(A) 15 states **(B)** 11 states **(C)** 10 states **(D)** 9 states

Q Consider a DFA over $\sum = \{a, b\}$ accepting all strings which have number of a's divisible by 6 and number of b's divisible by 8. What is the minimum number of states that the DFA will have? **(GATE-2001) (2 Marks)**

(A) 8 **(B)** 14 **(C)** 15 **(D)** 48

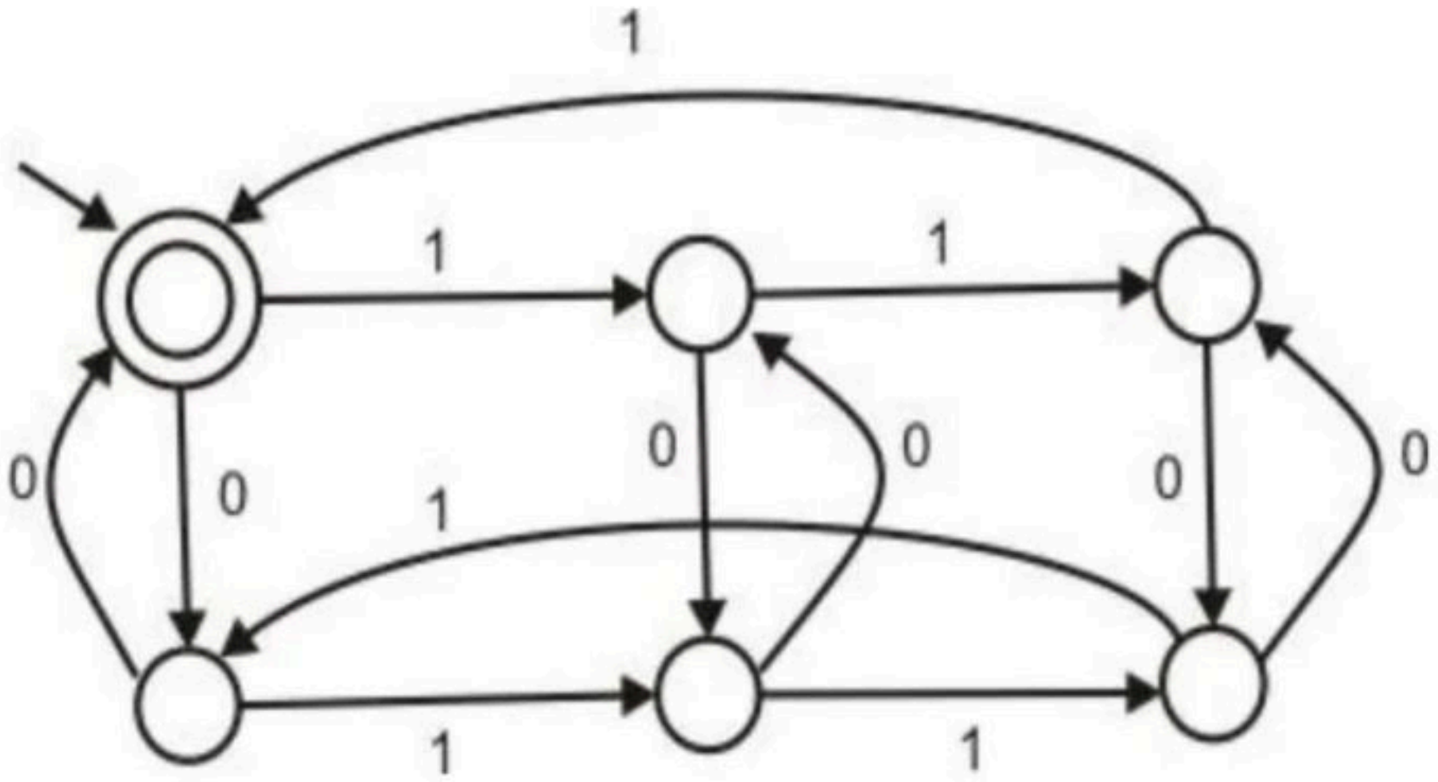
Q The following finite state machine accepts all those binary strings in which the number of l's and 0's is respectively. **(GATE-2004) (2 Marks)**

(A) divisible by 3 and 2

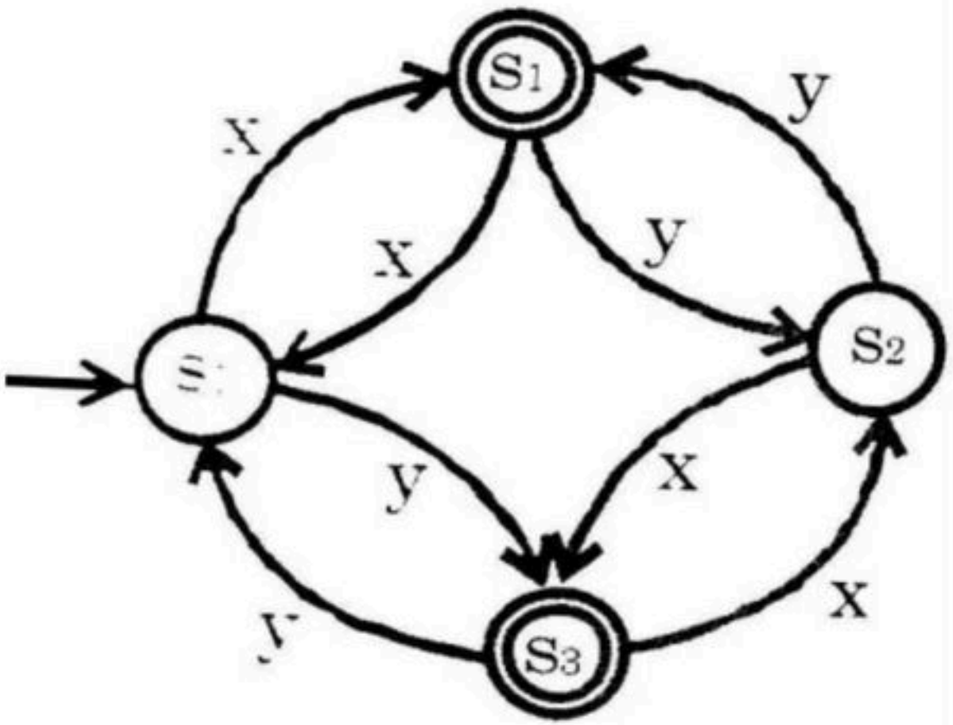
(B) odd and even

(C) even and odd

(D) divisible by 2 and 3



Q Consider the following DFA in which s_0 is the start and s_1, s_3 are the final sates. What language does this DFA recognize? **(GATE-2007) (2 Marks)**



a) All strings of x and y

b) All strings of x and y and even number of x and even number of y or odd number or x and odd number of y

c) All strings of x and y which have equal number of x and y

d) All strings of x and y with either even number of x and odd number of y or odd number of x and even number of y.

Q Given below are two finite state automata (→ indicates the start state and F indicates a final state) Which of the following represents the product automaton Z×Y? **(GATE-2008) (2 Marks)**

Y:

	a	b
→ 1	1	2
2 (F)	2	1

Z:

	a	b
→ 1	2	2
2 (F)	1	1

(A)

	a	b
→ P	S	R
Q	R	S
R (F)	Q	P
S	Q	P

(B)

	a	b
→ P	S	Q
Q	R	S
R (F)	Q	P
S	P	Q

(C)

	a	b
→ P	Q	S
Q	R	S
R (F)	Q	P
S	Q	P

(D)

	a	b
→ P	S	Q
Q	S	R
R (F)	Q	P
S	Q	P

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{0,1\}$, such that every string 'w' which is accepted has a decimal equivalent

i) 0(mod 3)

ii) 2(mod 4)

iii) 0(mod 5)

iv) 2(mod 6)

v) 3(mod 8)

Conclusion

on a format of m (mod n)

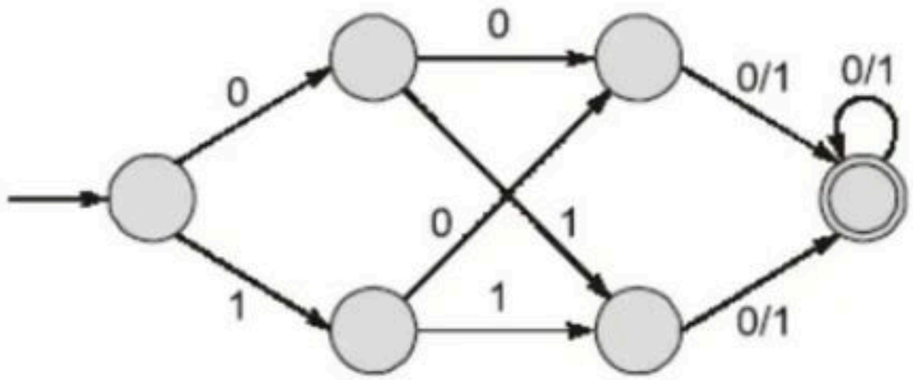
1) if n is odd, number of states will be n

2) if n is even and $n=2^m$, then number of states will be m+1

3) if n is even and $n! =2^m$, then for number of states no direct formula

Break

Q.10 Consider the following deterministic finite automation (DFA). **(GATE-2021)**



The number of strings of length 8 accepted by the above automation is _____.

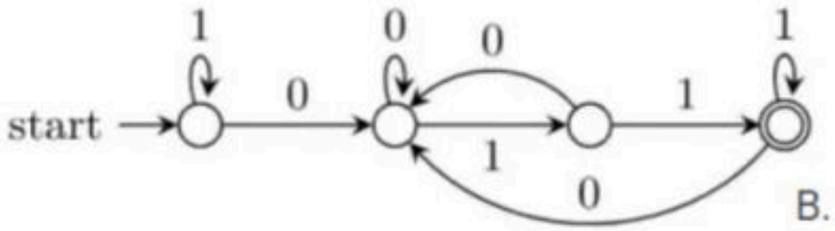
Consider the following language:

(Gate-2021) (1 Marks)

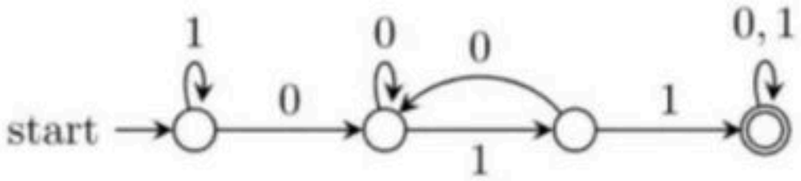
$$L = \{w \in \{0,1\}^* \mid w \text{ ends with the substring } 011\}$$

Which one of the following deterministic finite automata accepts L?

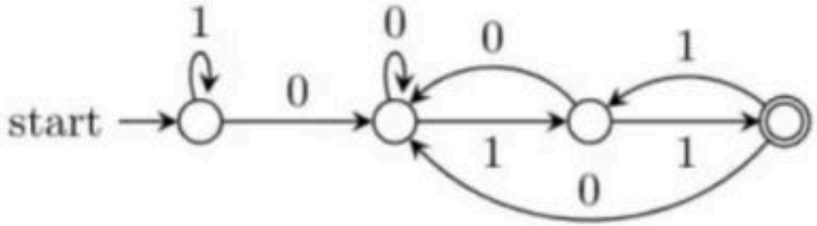
A.



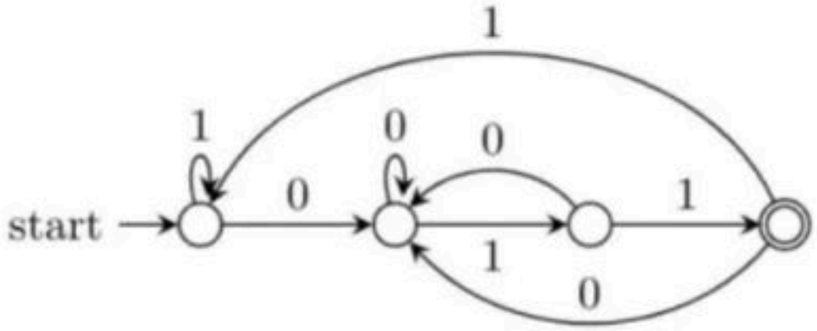
B.



C.



D.



Let $L \subseteq \{0,1\}^*$ be an arbitrary regular language accepted by a minimal DFA with k states. Which one of the following languages must necessarily be accepted by a minimal DFA with k states?

(GATE-2021)

A. $L - \{01\}$
B. $L \cup \{01\}$
C. $\{0,1\}^* - L$
D. $L \cdot L$

Break

COMPLIMENT OF DFA

Q Design a minimal DFA that accepts all strings over the alphabet $\sum = \{a, b\}$, such that every string accepted must not contain a substring aaa?

Note

- The finite automata which is obtained by interchanging final & non-final states is known as complement of the finite automata.
- $L(FA) \cup L(FA^C) = \sum^*$ $L(FA) \cap L(FA^C) = \Phi$
- $L(FA)$ subset of $\sum^*$ and $L(FA)^C$ subset of $\sum^*$
- No of states in FA = no of states in complement of FA
- No of final states in a FA and it's Complement FA may have different final states
- Complement always exist for DFA (as it is a complete system)

Q Deterministic automata of a language over alphabets {0, 1}, which does not contain 3 consecutive 0's. Minimum how many states 'S' DFA will have and how many of them will final states, F?

a) ISI = 5 and IFI = 1

b) ISI = 5 and IFI = 4

c) ISI = 4 and IFI = 1

d) ISI = 4 and IFI = 3

Q construct the DFA for the following languages:

i) $L = \{a^m b^n \mid m, n >= 0\}$

ii) $L = \{a^m b^n c^p \mid m, n, p >= 0\}$

Conclusion

- If type is $L = \{a^m b^n \mid m >= 0, n >= 0\}$, then no of states is no of alphabet +1

Q construct the DFA for the following languages:

i) $L = \{a^m b^n \mid m >= 0, n >= 1\}$

ii) $L = \{a^m b^n \mid m >= 0, n >= 2\}$

Conclusion

- If type is $L = \{a^m b^n \mid m >= 0, n >= j\}$, then no of states is j+2

Q construct the DFA for the following languages:

i) $L = \{a^m b^n \mid m >= 1, n >= 0\}$

ii) $L = \{a^m b^n \mid m >= 2, n >= 0\}$

Conclusion

- If type is $L = \{a^m b^n \mid m >= i, n >= 0\}$, then no of states is i+3

Q construct the DFA for the following languages:

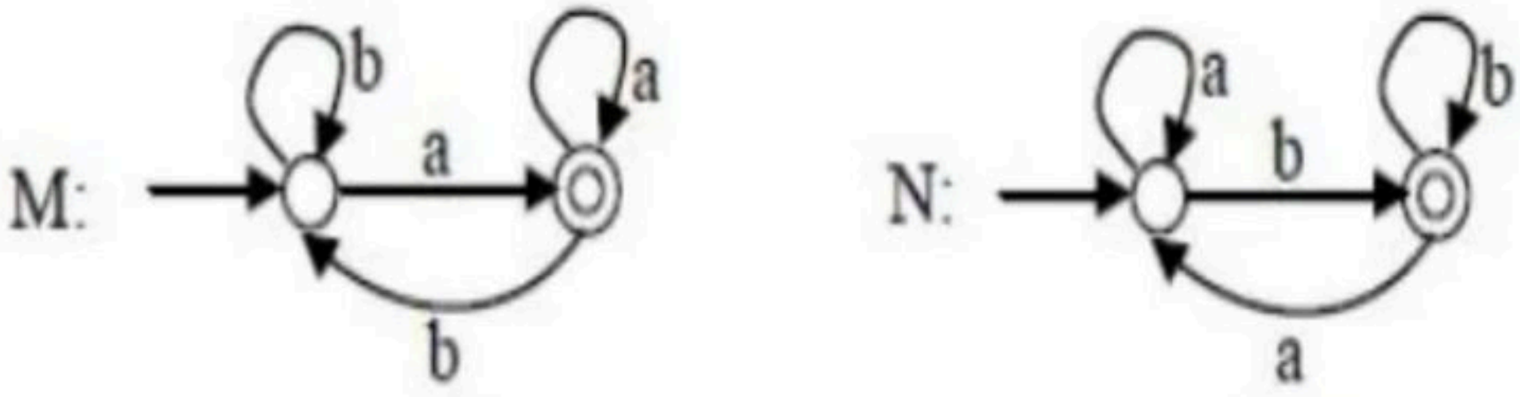
$L = \{a^m b^n \mid m >= 1, n >= 2\}$

Conclusion

- If type is $L = \{a^m b^n \mid m >= i, n >= j\}$, then no of states is $i+j+2$

Q Consider the DFAs M and N given above. The number of states in a minimal DFA that accepts the language L(M) ∩ L(N) is _____. **(GATE-2015) (2 Marks)**

(A) 0

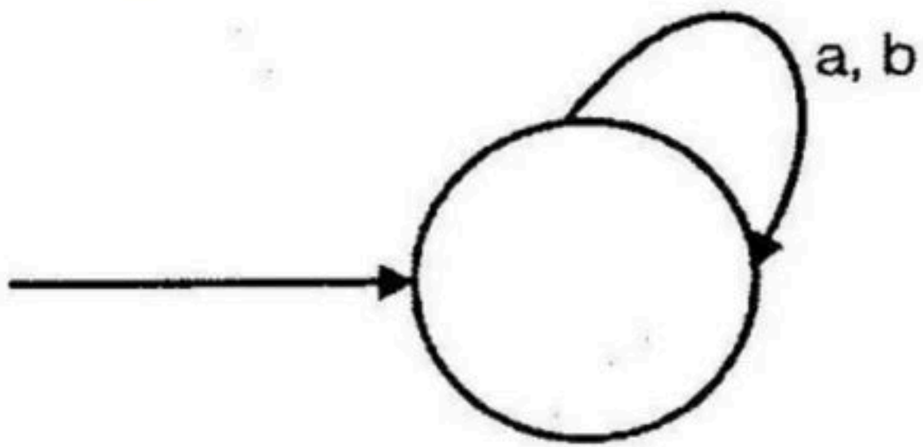


(B) 1

(C) 2

(D) 3

Q The FSM over an alphabet {a, b} shown in the figure accepts



a) all stings

b) no stings

c) ϵ - alone

d) None of these

Q Definition of a language L with alphabet {*a*} is given as following.

$L = \{a^{nk} \mid k>0$, and n is a positive integer constant$\}$

What is the minimum number of states needed in DFA to recognize L? **(NET-DEC-2018) (2 Marks) (GATE-2011)**

(A) k+1 **(B)** n+1 **(C)** $2^{(n+1)}$ **(D)** $2^{(k+1)}$

Q Which of the following set can be recognized by a deterministic Finite state Automation? **(GATE–1998) (2 Marks)**

a) The numbers 1,2,4, 2^n, written in unary

b) The set of binary string in which the number of zeroes is the same as the number of ones.

c) The numbers 1,2,4,8, 2^n, written in binary

d) The set { 1, 101, 11011, 1110111,-------}

Q Minimal deterministic finite automaton for the language $L = \{0^n \mid n \geq 0, n \neq 4\}$ will have: **(NET-JUNE-2015)**

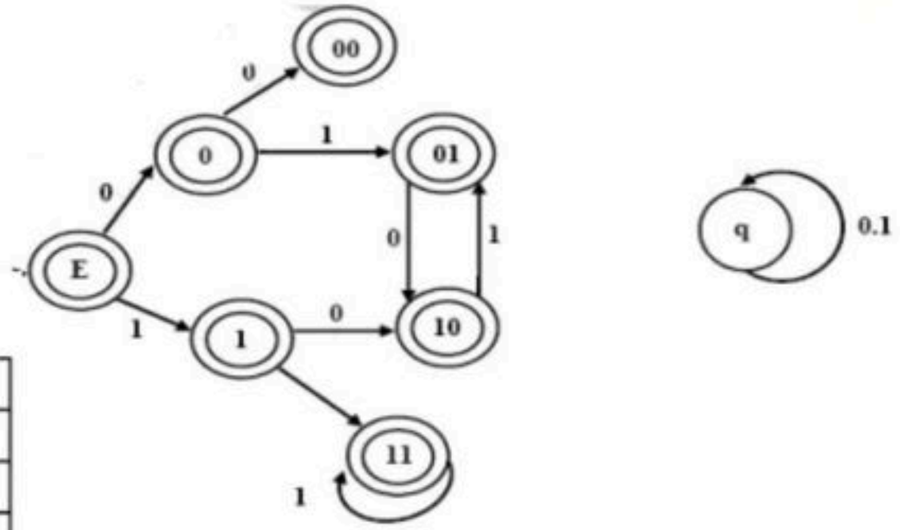
a) 1 final state among 5 states

b) 4 final states among 5 states

c) 1 final state among 6 states

d) 5 final states among 6 states

Q Consider the set of strings on {0,1} in which, every substring of 3 symbols has at most two zeros. For example, 001110 and 011001 are in the language, but 100010 is not. All strings of length less than 3 are also in the language. A partially completed DFA that accepts this language is shown below. **(GATE-2012) (2 Marks)**



The missing arcs in the DFA are

(A)

	00	01	10	11	q
00	1	0			
01				1	
10	0				
11			0		

(B)

	00	01	10	11	q
00		0			1
01		1			
10				0	
11		0			

(C)

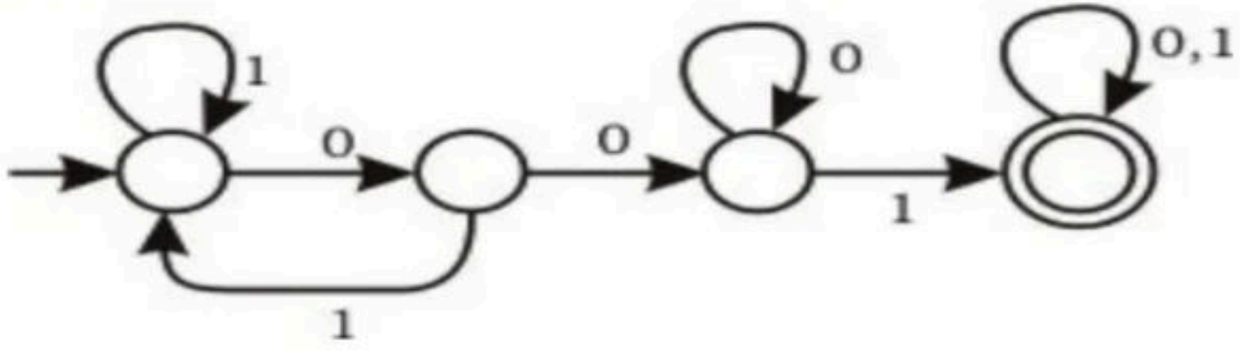
	00	01	10	11	q
00		1			0
01		1			
10			0		
11		0			

(D)

	00	01	10	11	q
00		1			0
01				1	
10	0				
11			0		

Q Consider the following deterministic finite state automaton M. Let S denote the set of seven-bit binary strings in which the first, the fourth, and the last bits are 1. The number of strings in S that are accepted by M is **(GATE-2003) (2 Marks)**

(A) 1 **(B)** 5 **(C)** 7 **(D)** 8



Break

MINIMIZATION OF FINITE AUTOMATA

- The process of elimination of states whose presence or absence doesn't affect the language accepting capability of deterministic Finite Automata is called minimization of automata and the result is minimal deterministic finite automata or commonly called as minimal finite automata as MFA.

- **NOTE- MFA is always unique for a language.**

- It is sometimes difficult to design a minimal DFA directly so, a better approach is to first design the DFA and then minimize it.

- Based on **productivity**, the states of DFA can be mainly classified in two types-

- **PRODUCTIVE STATES**- State is said to be productive, if it adds any accepting power to the machine that is its presence and absence effect the language accepting capability of the machine.

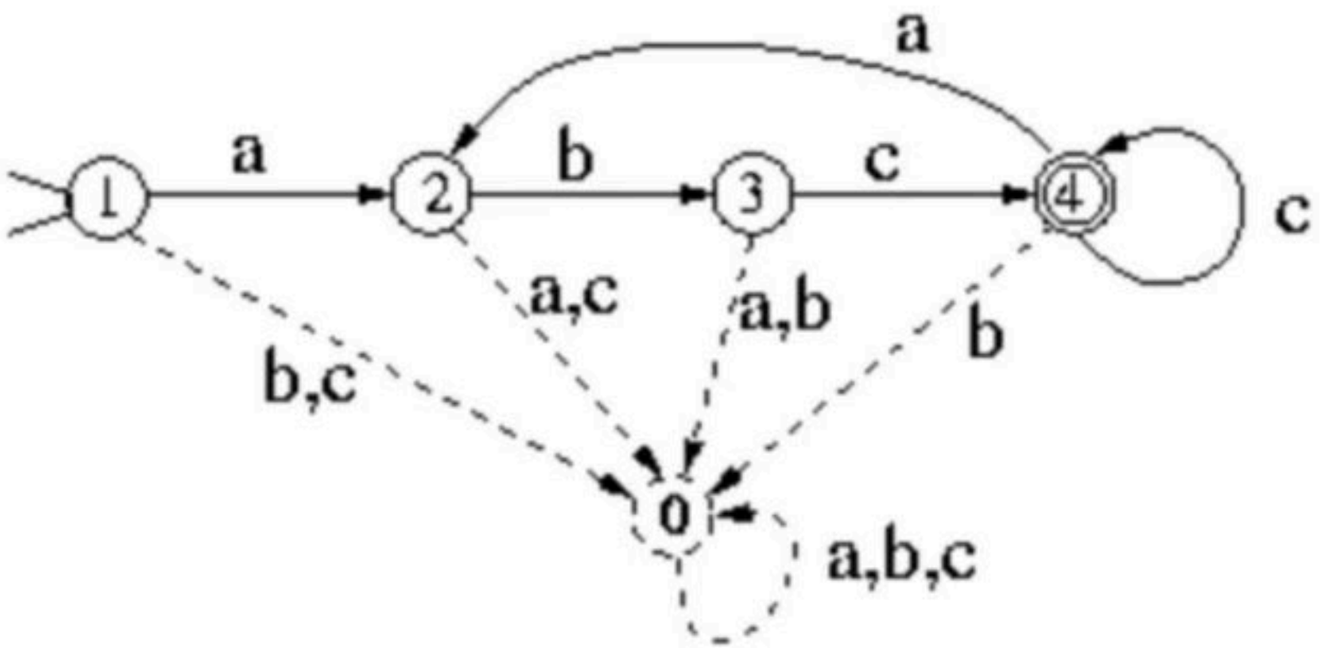
- **NON- PRODUCTIVE STATES**- These states don't any add anything to the language accepting power to the machine.

NON- PRODUCTIVE STATES- These states don't add anything to the language accepting power to the machine. They can further be divided into three types-

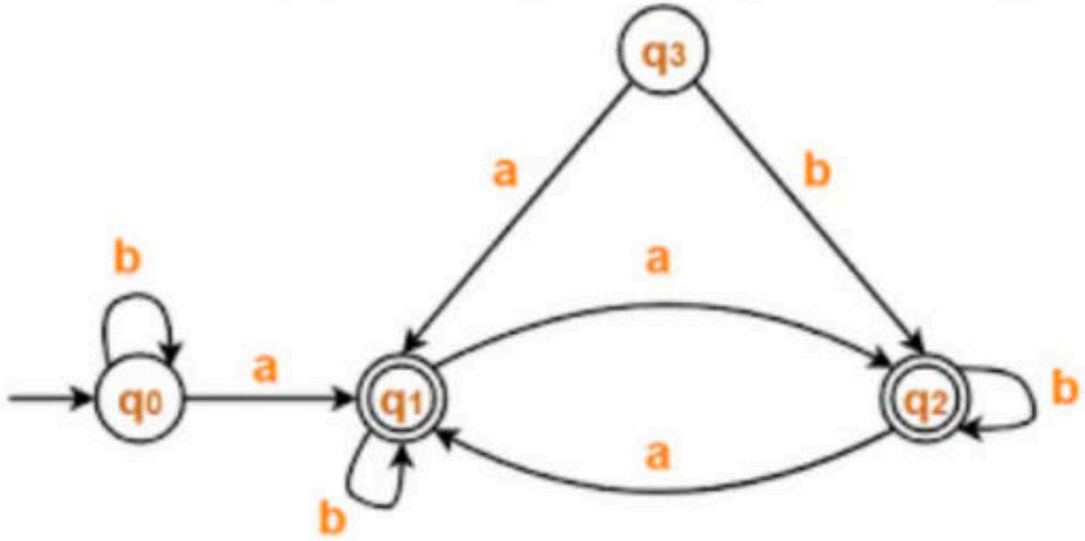
- Dead State
- Unreachable Sate
- Equal State

- **<u>Dead State</u>**- It is basically created to make the system complete, can be defined as a state from which there is no transition possible to the final state.

- In a DFA there can be more than one dead state but logically always one dead state is sufficient to complete the functionality.



- **<u>Unreachable Sate</u>**- It is that state which cannot be reached starting from initial state by parsing any input string.

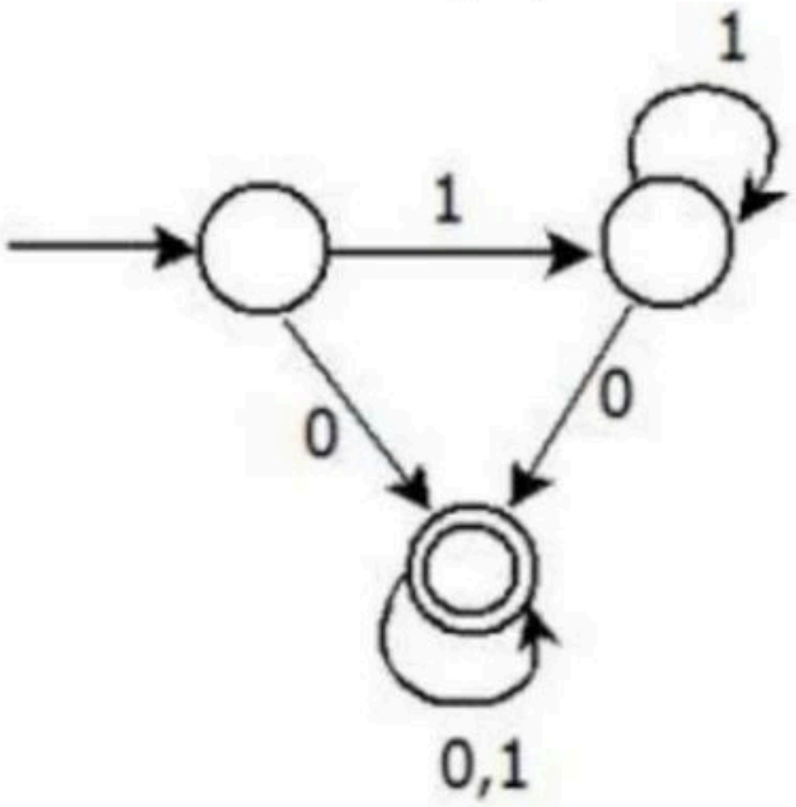


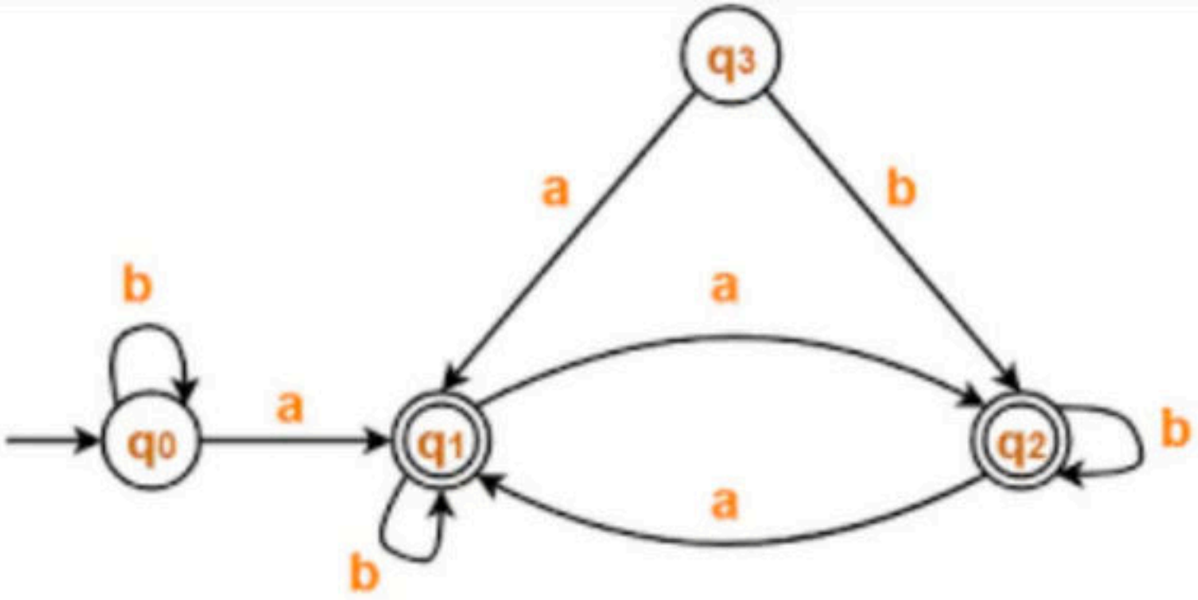
- **<u>Equal State</u>**-These are those states that behave in same manner on each and every input string. That is for any string w where $w \in \Sigma^*$ either both of the states will go to final state or both will go to non-final state. (remember the example of an equal state DFA).

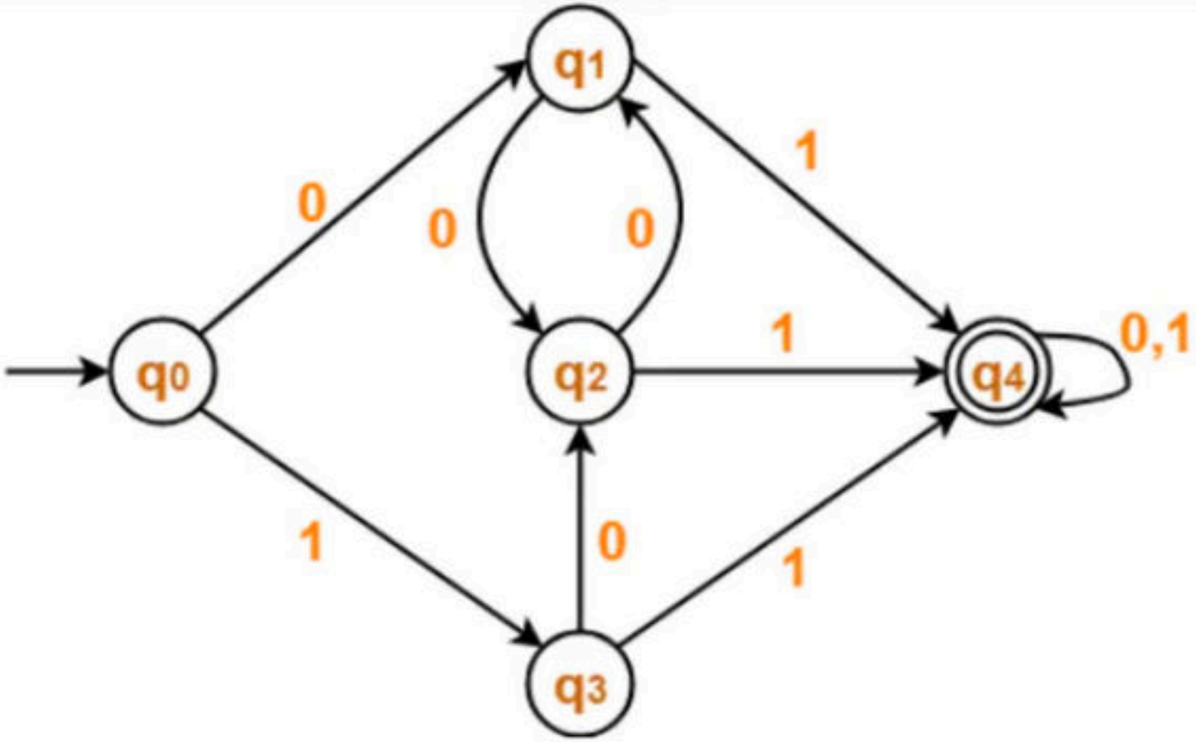
- More formally, two states q_1 and q_2 are equivalent (denoted by $q_1 \cong q_2$) if both $\delta(q_1, x)$ and $\delta(q_2, x)$ are final states or both of them are non-final states for all $x \in \sum^*$. If q_1 and q_2 are k-equivalent for all $k \geq 0$, then they are *k*-equivalent.

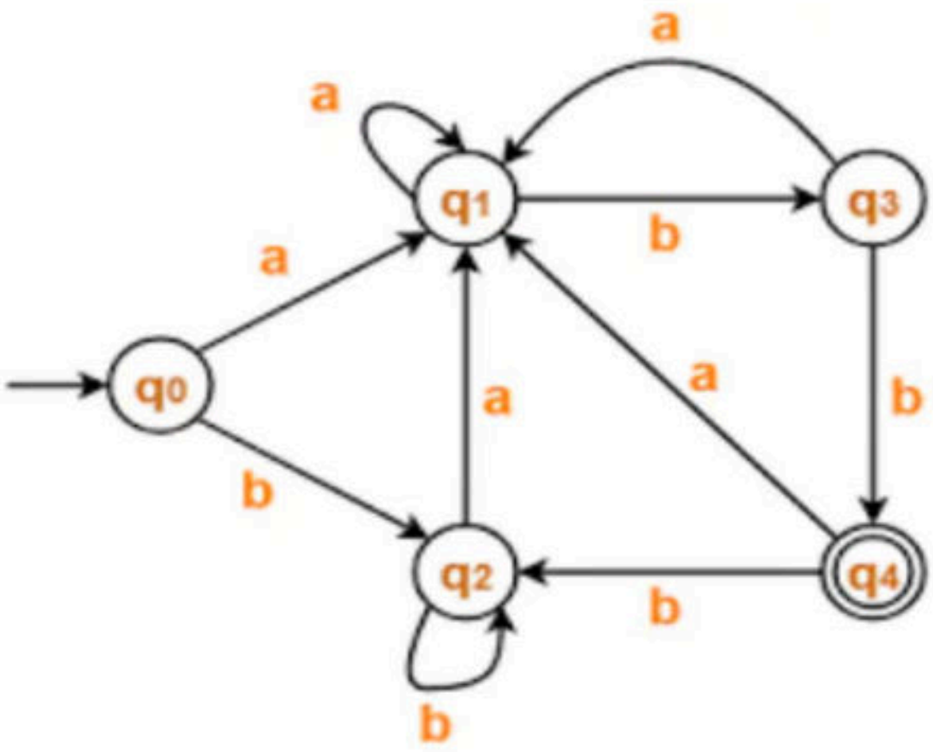
Procedure of Minimization

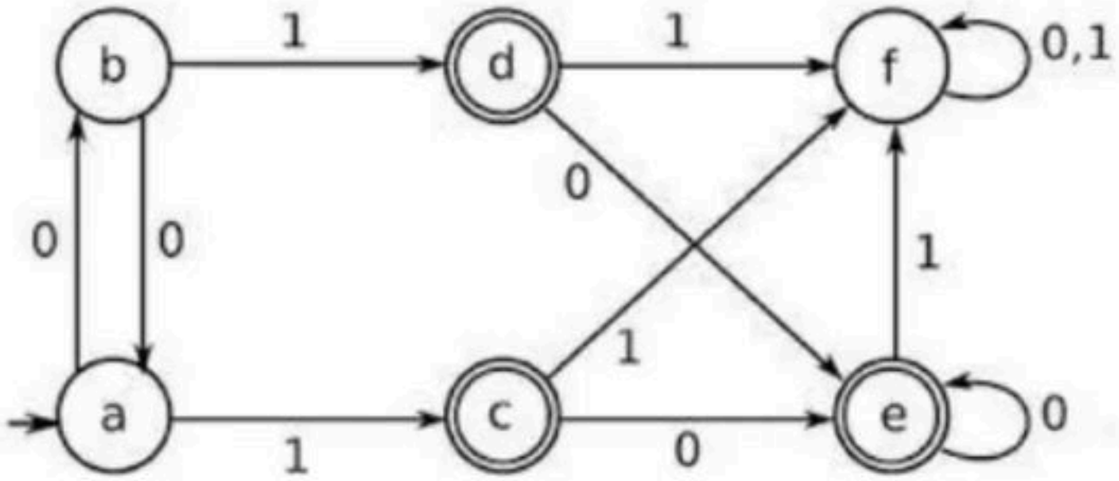
- For this first of all, group all the non-final states in one set and all final states in another set.
- Now, on both the sets, individually check, whether any of the underlying elements (states) of that particular set are behaving in the same way, that is are they having same transition(to same set) on each input alphabet.
- if the answer is yes, then these two states are equal, otherwise not.



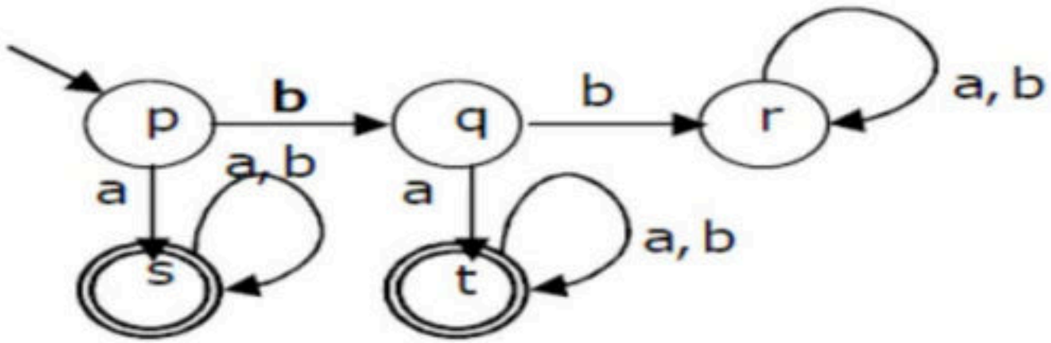

q3
a
b
b
a
a
q0
q1
q2
b
a
b


q1
q0
q2
q4
q3
0
0
0
1
1
0,1
1
0
1


a
a
q1
q3
b
a
q0
a
a
b
b
q2
q4
b
b

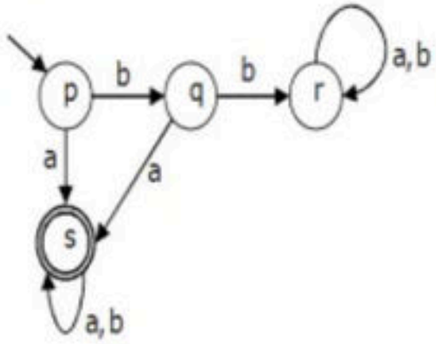

b
d
f
a
c
e
1
1
0,1
0
0
0
1
1
1
0
0

Q A deterministic finite automation (DFA)D with alphabet {a, b} is given below **(GATE-2011) (2 Marks)**

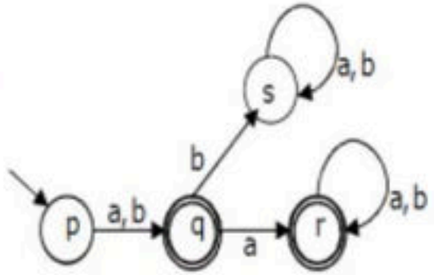


Which of the following finite state machines is a valid minimal DFA which accepts the same language as D?

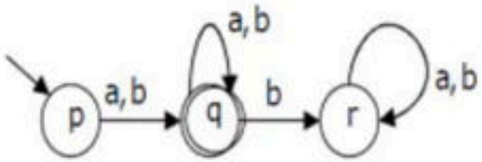
(A)



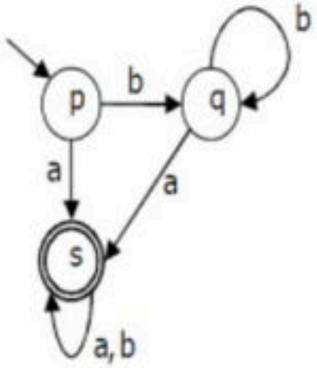
(B)



(C)



(D)



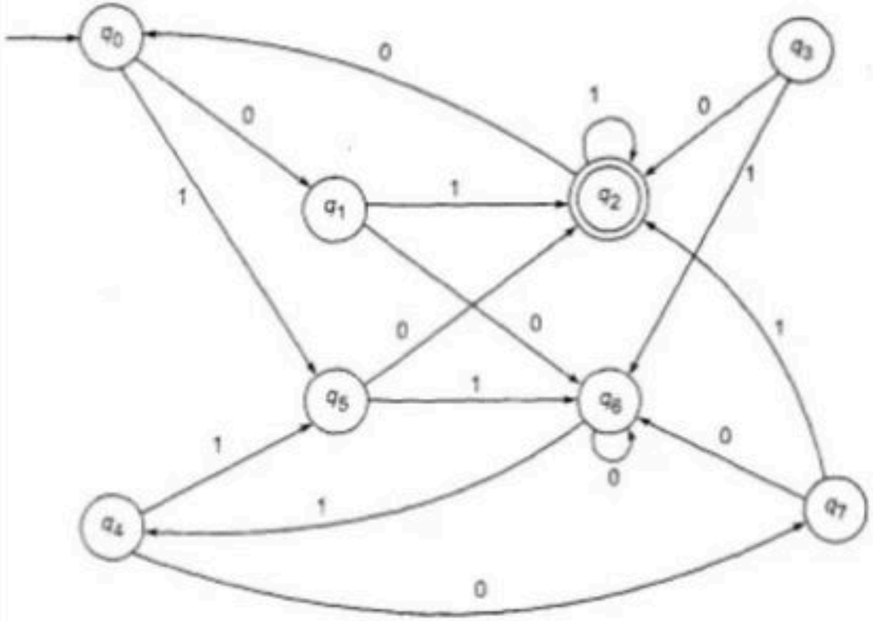
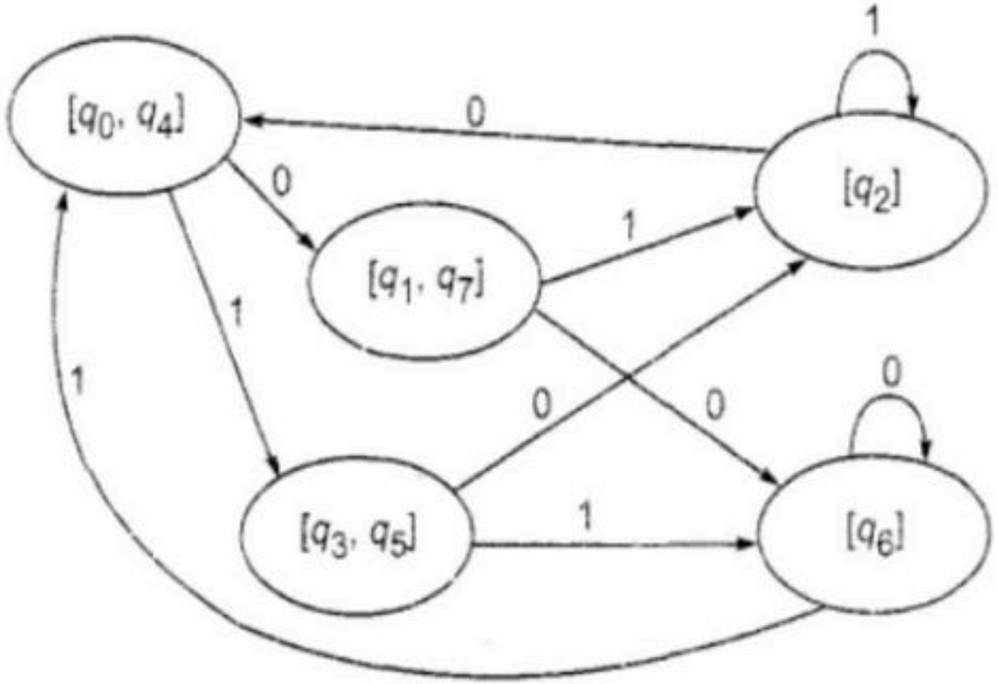
Q IF a DFA is represented by the following transition table, then how many states does the corresponding minimal DFA contains?

a) 2

b) 3

c) 4

d) 5

State	Input 0	Input 1
(start) → A	B	C
B	B	D
C	B	C
D	B	E
(final) → E	B	C

Q Find weather the Conversion of the DFA below to MDFA is correct or not?