INTRODUCTION

* Theory of computation (THOC):-
  It is a mathematical study of computing machine and its capabilities.

  OR

  It is a study of formal languages (C, C++, Java etc.) and automata theory.

* Formal languages:-
  It is a collection of strings which are formed based on some condition
  eg $L = \{a^n b^n \mid n \geq 1\} = \{ab, aabb, \ldots\}$

SET THEORY

* Disjoint set and superset :-

Disjoint set:- Two sets are called disjoint if their intersection is the empty set.

Let $A = \{1,3,5,7,9\}$ and $B = \{2,4,6,8,10\}$
Because $A \cap B = \phi$, A & B are disjoint

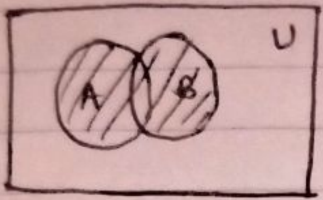
Superset:- A is a superset of B if set A has same elements as B, or more.

* Partitions of a set :-

↳ A collection of disjoint subsets of a given set. The union of the subsets must be equal to the entire original set.
↳ For eg. one possible partition of $\{1,2,3,4,5,6\}$ is $\{1,3\}$, $\{2\}$, $\{4,5,6\}$.

* Union:-

↳ Union of two given sets is the smallest set which contains all the elements of the both the sides.
↳ To find the union of two givens sets A & B is a set which consists of all the elements of A and all the elements of B such that no elements is repeated.
↳ The symbol for denoting union is sets is 'U'.
↳ For eg. Let set $A = \{2,4,5,6\}$ and set $B = \{4,6,7,$
↳ Taking every elements of both the sets A & B, without repeating any elements, we get a new set $= \{2,4,5,6,7,8\}$



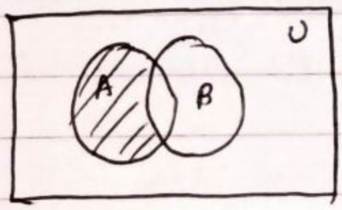
$A \cup B$ is shaded.

* Difference:-

↳ If A and B are two sets, then their difference is given by $A - B$ or $B - A$.
- If $A = \{2, 3, 4\}$ and $B = \{4, 5, 6\}$

↳ $A - B$ means elements of A which are not the elements of B.
i.e. in the above eg. $A - B = \{2, 3\}$

In general, $B - A = \{x : x \in B,$ and $x \notin A\}$
If A & B are disjoints sets, then $A - B = A$ &
$B - A = B$

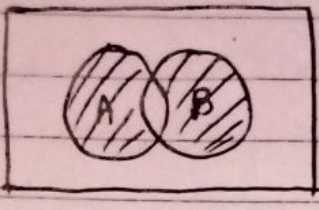


$A - B$ is shaded

* Symmetric Difference:-

↳ The symmetric difference using Venn Diagram of two subsets A & B is a subset of U, denoted by $A \Delta B$ and is defined by
↳ $A \Delta B = (A - B) \cup (B - A)$
↳ Let A & B are two sets. The symmetric difference of two sets A & B is the set $(A - B) \cup (B - A)$ and is denoted by $A \Delta B$.
↳ Thus, $A \Delta B = (A - B) \cup (B - A) = \{x : x \notin A \cap B\}$

Or $A \Delta B = \{x : [x \in A \text{ \& } x \notin B] \text{ or } [x \in B \text{ and } x \notin A]\}$



$A \Delta B$ is shaded.

* Intersection:-

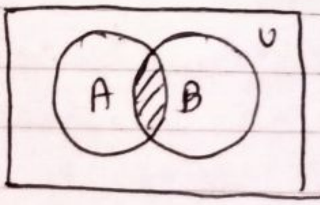
↳ Let $A = \{2, 3, 4, 5, 6\}$, $B = \{3, 5, 7, 8\}$

↳ In this two sets , the elements 3 & 5 are common the set containing these common elements i.e. $\{3, 5\}$ is the intersection of set A & B.

↳ The symbol used for the intersection of two sets is '$\cap$'.

↳ Therefore , symbolically , we write intersection of the two sets A & B is $A \cap B$ which means A intersection B.

↳ The intersection of two sets A & B is represented as $A \cap B = \{x : x \in A \text{ and } x \in B\}$



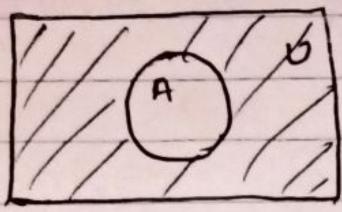
$A \cap B$ is shaded

* NOTE: The symmetric difference can also be expressed using the XOR operation $\oplus$ on the predicates describing the two sets in a set-builder notation.

DELUXE
PAGE NO.:
DATE:

* Complement:-

↳ The complement of set A, denoted by 'A', is the set of all elements in the universal set that are not in A. It is denoted by A'.



A' is shaded.

* Countably Infinite set:-

↳ A set is said to be finite if there is a one to one correspondence between the elements in the set and the elements in some set n, where n belongs to N; n is said to be the cardinality of set.

eg {a,b,c} {a,d,e}.

↳ A set is said to be countably infinite if there is one-one correspondence between the elements in the set & the elements in N.

eg. Set of all natural numbers : {0,1,2,3...}
Set of non-negative even integer - {0,2,4,6..}

* Uncountable infinite sets:-

↳ A set is said to be uncountably infinite if there is no one-one correspondence between the elements in the set and the elements in N.
↳ eg. set of real numbers betn 0 & 1.

* Simple Results:-

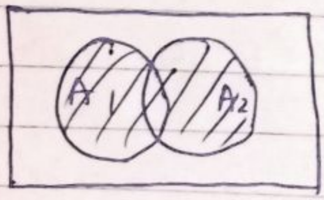
① $|P \cup Q| \leq |P| + |Q|$

② $|P \cap Q| \leq \min(|P|, |Q|)$

③ $|P \oplus Q| = |P| + |Q| - 2|P \cap Q|$

④ $|P - Q| \geq |P| - |Q|$

* Principle of Inclusion and exclusion :-



$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$

Que. Among a class of students, 60 plays football, 70 play badminton, 25 students play both football & badminton. How many students are there in a class who play some game?

→ $|F \cup B| = 60 + 70 - 25 = \boxed{105}$

DELUXE
PAGE NO.:
DATE:

Que. How many integers from 1 to 100 are multiples of 2 & 3?

$\rightarrow \left\lfloor \frac{100}{2} \right\rfloor + \left\lfloor \frac{100}{3} \right\rfloor - \left\lfloor \frac{100}{6} \right\rfloor$

$= 50 + 33 - 16$

$= \boxed{67}$

Que. Total $T = 30$

$R = 15$

$AC = 8$

$WW = 6$

$R \cap AC \cap WW = 3$

N = without any option.

$T = R + AC + WW - (R \cap AC + AC \cap WW + R \cap WW) + (AC \cap R \cap WW) + N$

$\therefore 30 = 15 + 8 + 6 - (x) + 3 + N$

$\therefore x = N + 2$

$\therefore x = 3(3) = 9$

$\therefore \boxed{N = 7}$

$\min(AC \cap R) = \min(R \cap WW) = \min(AC \cap NW) = 3$

Que. Let us determine the number of integers between 1 and 250 that are divisible by any of the integers 2, 3, 5, & 7.

→ $\boxed{193}$

$$\frac{250}{2} + \frac{250}{3} + \frac{250}{5} + \frac{250}{7} - \frac{250}{6} - \frac{250}{10} - \frac{250}{14} - \frac{250}{15} - \frac{250}{21}$$

$$- \frac{250}{35} + \frac{250}{30} + \frac{250}{42} + \frac{250}{70} + \frac{250}{105} - \frac{250}{210}$$

$= \boxed{193}$

* Laws in a set :-

① Idempotent law :-

$A \cup A = A$

$A \cap A = A$

② Associative laws:-

$(A \cup B) \cap C = A \cup (B \cup C)$, $(A \cap B) \cap C = A \cap (B \cap C)$

③ Commutative laws:- $A \cup B = B \cup A$, $A \cap B = B \cap A$

④ Distributive laws:-

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

⑤ Identity laws:-

$A \cup \phi = A$

$A \cap U = A$

$A \cup U = U$

$A \cap \phi = \phi$

⑥ Involution laws:-

$(A')' = A$

(4) complement laws:-

$A \cup A' = A$

$A \cap A' = \phi$

$U' = \phi$

$\phi' = U$

(5) Demorgan's laws:-

$(A \cup B)' = A' \cap B'$

$(A \cap B)' = A' \cup B'$

FUNCTIONS

* Functions:-

↳ A function f from a set A to a set B is an assignment of exactly one element of B to each element of A.

↳ We write, $f(a)=b$
if b is the unique element of B assigned by the function f to the element a of A.

↳ If f is a function from A to B, we write $f: A \rightarrow B$.

NOTE: Here, "$\rightarrow$" has nothing to do with if... then)

↳ If $f: A \rightarrow B$, we say that A is the domain of f and B is the codomain of f.

↳ If $f(a)=b$, we say that b is the image of a and a is the preimage of b.

↳ If range of $f: A \rightarrow B$ is the set of all images of all elements of A.

↳ We say that $f: A \rightarrow B$ maps A to B.

↳ Let us take a look at the function $f: P \rightarrow C$ with

P = {Vijay, Arvind, Praveen, Bhushan}
C = {GJ, Delhi, MP, AP}
f(Vijay) = GJ   f(Praveen) = MP
f(Arvind) = Delhi   f(Bhushan) = AP

Here, the range of f is c.

↳ Let us re-specify f as follows:

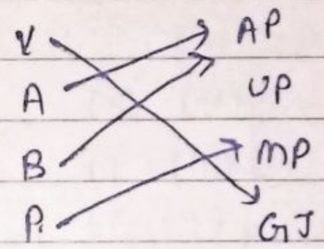
$f(V) = GJ$
$f(A) = AP$
$f(P) = MP$
$f(B) = AP$

Is f still a function? Yes.

Range = {GJ, AP, MP}

↳ Other ways to represent f:

x	f(x)
V	GJ
A	AP
P	MP
B	AP



↳ If the domain of our function f is large, it is convenient to specify f with a formula.

eg:

$f: R \rightarrow R$
$f(x) = 2x$

This leads to

$f(1) = 2$
$f(3) = 6$
$f(-3) = -6$
...

↳ We already know that the range of a function $f: A \to B$ is the set of all images of elements $a \in A$.

↳ If we only regard a subset $S \subseteq A$, the set of all images of elements $s \in S$ is called the image of S.

↳ We denote the image of S by $f(S)$:
↳ $f(S) = \{ f(s) \mid s \in S \}$

↳ Let us look at the following well-known funⁿ:-

$f(V) = GJ$
$f(A) = AP$
$f(P) = MP$
$f(B) = AP$

Image of $S = \{V, A\}$? = $f(S) = \{GJ, AP\}$
Image of $S = \{A, B\}$ = $f(S) = \{AP\}$.

* Properties of Functions:-

↳ A function $f: A \to B$ is said to be one-to-one (or injective), if and only if

• $\forall x, y \in A \ (f(x) = f(y) \to x = y)$

↳ In other words; f is one-to-one if & only if it does not map two distinct elements of A onto the same element of B.

DELUXE
PAGE NO.:
DATE:

eg.

① $f(V) = GJ$
$f(A) = AP$
$f(P) = MP$
$f(B) = AP$
Is one to one?

NO, A & B are mapped onto the same element of the image.

② $g(V) = GJ$
$g(A) = UP$
$g(P) = MP$
$g(B) = AP$
Is one to one?

Yes, each element is assigned to a unique element of the image.

- How can we prove that a function f is one-to-one?

↳ whenever you want to prove something, first take a look at the relevant definition(s):

↳ $\forall x, y \in A \ (f(x) = f(y) \rightarrow x = y)$

eg. $f: R \rightarrow R$
$f(x) = x^2$

↳ Disproof by counterexample:
- $f(3) = f(-3)$, but $3 \neq -3$, so f is not one to one.

↳ and yet another eg:
$f: R \rightarrow R$
$f(x) = 3x$
- one-to-one : $\forall x, y \in A \ (f(x) = f(y) \rightarrow x = y)$

To show $f(x) \neq f(y)$ whenever $x \neq y$ (indirect proof)

$x \neq y$
$\Leftrightarrow 3x \neq 3y$
$\Leftrightarrow f(x) \neq f(y)$
So if $x \neq y$, then $f(x) \neq f(y)$. that is f is one-to-one.

↳ A fun^n $f: A \rightarrow B$ with $A, B \subseteq \mathbb{R}$ is called strictly increasing, if
$\forall x, y \in A \ (x < y \rightarrow f(x) < f(y))$,
and strictly decreasing if.
$\forall x, y \in A \ (x < y \rightarrow f(x) > f(y))$.

↳ Obviously, a function that is either strictly increasing or strictly decreasing is one-to-one.

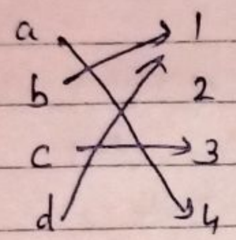
- A function $f: A \rightarrow B$ is called onto or surjective, if and only if for every element $b \in B$ there is an element $a \in A$ with $f(a) = b$.

- In other words, f is onto if and only if its range is its entire codomain.

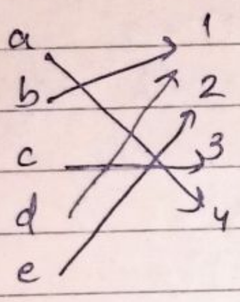
- A function $f: A \rightarrow B$ is a one-to one correspondence or a bijection, if and only if it is both one-to one and onto.

- Obviously, if f is a bijection and A & B are finite sets, then $|A| = |B|$.

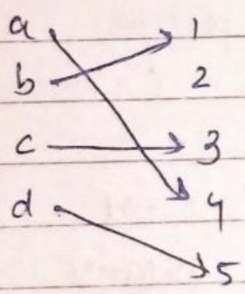
eg. In the following eg. we use the arrow representation to illustrate functions $f: A \to B$

① 

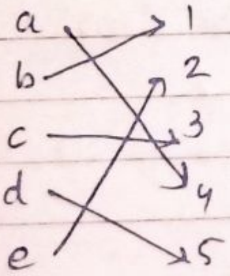
I (injective) = No
S (surjective) = No
B (bijective) = No

② 

I = No
S = Yes
B = No

③ 

I = Yes
S = No
B = No

④ 

I = Yes
S = Yes
B = Yes

* Inversion :-

↳ An Intersecting property of bijection is that they have an Inverse function

↳ the Inverse function of the bijection $f: A \rightarrow B$ is the function $f^{-1}: B \rightarrow A$ with
- $f^{-1}(b) = a$ whenever $f(a) = b$.

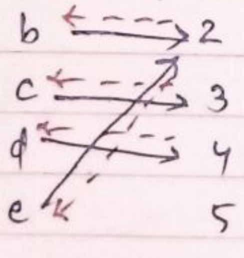
eg

$f(a) = 1$	Inverse fun^n f^{-1}
$f(b) = 2$	$f^{-1}(1) = a$
$f(c) = 3$	$f^{-1}(2) = b$
$f(d) = 4$	$f^{-1}(3) = c$
$f(e) = 5$	$f^{-1}(4) = d$
Clearly, f is bijective.	$f^{-1}(5) = e$

Inversion is only possible for bijection (= inversible fun^n)

$f \longrightarrow$
f^{-1} - - - >

NOTES: $f^{-1}: C \rightarrow P$ is no function, because it is not defined for all elements of C & assigns two images to the pre-image 2.



* Composition:-

↳ The composition of two funⁿ $g: A \to B$ &
$f: B \to C$, denoted by $f \circ g$, is defined by

- $(f \circ g)(a) = f(g(a))$

↳ This means that
- First, function g is applied to element $a \in A$, mapping it onto an element of B.

- then, function f is applied to this element of B, mapping it onto an element of C.

- Therefore, the composite function maps from A to C.

eg. $f(x) = 7x - 4$, $g(x) = 3x$
$f: R \to R$, $g: R \to R$

- $(f \circ g)(5) = f(g(5)) = f(15) = 105 - 4 = 101$

- $(f \circ g)(x) = f(g(x)) = f(3x) = 21x - 4$.

- Composition of a function and its Inverse:-

- $(f^{-1} \circ f)(x) = f^{-1}(f(x)) = x$

- The composition of a funⁿ and its Inverse is the Identity function $I(x) = x$.

* Graphs:-

↳ The graph of a function $f: A \to B$ is the set of ordered pairs $\{(a,b) \mid a \in A \text{ \& } f(a) = b\}$.

↳ The graph is a subset of $A \times B$ that can be used to visualise f in a two-dimensional coordinate system.

* Floor and ceiling functions:-

↳ The floor and ceiling functions map the real number onto the integers $(R \to Z)$

- The floor function assigns to $r \in R$ the largest $z \in Z$ with $z \leq r$, denoted by $\lfloor r \rfloor$.

eg $\lfloor 2.3 \rfloor = 2$, $\lfloor 2 \rfloor = 2$, $\lfloor 0.5 \rfloor = 0$, $\lfloor -3.5 \rfloor = -4$

- The ceiling function assigns to $r \in R$ the smallest $z \in Z$ with $z \geq r$, denoted by $\lceil r \rceil$

eg $\lceil 2.3 \rceil = 3$, $\lceil 2 \rceil = 2$, $\lceil 0.5 \rceil = 1$, $\lceil -3.5 \rceil = -3$

RELATIONS

* Relations:

↳ If we want to describe a relationship between elements of two sets A and B, we can use ordered pairs with their first element taken from A and their second element taken from B.

↳ Since this is a relation between two sets, it is called a binary relation.

Definition:- Let A and B be sets. A binary relation from A to B is a subset of $A \times B$.

In other words, for a binary relation R we have $R \subseteq A \times B$. We use the notation aRb to denote that $(a,b) \in R$ and $a \not{R} b$ to denote that $(a,b) \notin R$.

- when (a,b) belongs to R, a is said to be related to b by R.

eg. Let P be a set of people, C be a set of cars and D be the relation describing which person drives which cars(s).

$P = \{$ Carl, Suzanne, Peter, Carla $\}$
$C = \{$ Mercedes, BMW, tricycle $\}$
$D = \{$ (Carl, merdedes), (Suzanne, Mercedes), (Suzanne, BMW), (Peter, tricycle) $\}$.

→ This means that Carl drives a Mercedes, Suzanne drives Mercedes and BMW, Peter drives a tricycle and Carla does not drive any of these vehicles.

* Function as Relations:-

↳ You might remember that a function f from a set A to a set B assigns a unique element of B to each element of A.

↳ The graph of f is the set of ordered pairs (a, b) such that $b = f(a)$.

↳ Since the graph of f is a subset of $A \times B$, it is a relation from A to B.

↳ Moreover, for each element a of A, there is exactly one ordered pair in the graph that has a as its first element.

↳ Conversely, if R is a relation from A to B such that every element in A is the first element of exactly one ordered pair of R, then a function can be defined with R as its graph.

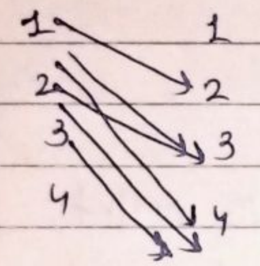
↳ This is done by assigning to an element $a \in A$ the unique element $b \in B$ such that $(a, b) \in R$.

* Relations on a set:

A relation on the set A is a relation from A to A.
In other words, a relation on the set A is a subset of $A \times A$.

eg. Let $A = \{1, 2, 3, 4\}$. Which ordered pairs are in the relation $R = \{(a,b) \mid a < b\}$?

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



R	1	2	3	4
1		X	X	X
2			X	X
3				X
4				

Que How many different relations can we define on a set A with n elements?

↪ A relation on a set A is a subset of $A \times A$.
How many elements are in $A \times A$?
- There are n^2 elements in $A \times A$.

- The no. of subsets that we can form out of a set with m elements is 2^m.
Therefore, 2^{n^2} subsets can be formed out of $A \times A$

DELUXE
PAGE NO.:
DATE:

* Properties of Relations:-

① Reflexive:-
A relation R on a set A is called reflexive if $(a,a) \in R$ for every element $a \in A$.

eg. Are the following relations on $\{1,2,3,4\}$ reflexive.

$R = \{(1,1), (1,2), (2,3), (3,3), (4,4)\}$ No
$R = \{(1,1), (2,2), (2,3), (3,3), (4,4)\}$ Yes
$R = \{(1,1), (2,2), (3,3)\}$ No.

→ no of RR = $2^{n(n-1)}$

② Irreflexive :-
A relation on a set A is called irreflexive if $(a,a) \notin R$ for every element $a \in A$.
→ no of IRR = 2^{n^2-n}

③ Symmetric:-
A relation R on a set A is called symmetric if $(b,a) \in R$ holds when $(a,b) \in R$.
eg $\{(1,2), (2,1), (3,4), (4,3)\}$

→ no of SR = $2^{n(n+1)/2}$

④ Antisymmetric:-
Relation R is antisymmetric if either $(x,y) \notin R$ or $(y,x) \notin R$ whenever x not equal to y.
↳ A relation R is said to be antisymmetric on a set A if xRy and yRx hold when $x = y$

eg. $\{(1,2),(2,3)\}$ Antisymmetric
$\{(1,2),(2,3),(3,3)\}$ Antisymmetric
$\{(1,2)(2,1),(3,3)\}$ Not antisymmetric

→ no of ANS = $2^n \cdot 3^{n\cdot(n-1)/2}$

⑤ Asymmetric:-
A relation R on a set A is called asymmetric if no $(b,a) \in R$ when $(a,b) \in R$.
eg $\{(1,2),(2,3)\}$
→ no of AS = $3^{n^2-N/2}$

Que. Are the following relations on $\{1,2,3,4\}$ symmetric, antisymmetric or asymmetric?

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

② $R = \{1,1)\}$ sym. & antisym

③ $R = \{(1,3),(3,2),(2,1)\}$ antisym & asym

④ $R = \{(4,4),(3,3),(1,4)\}$ antisym

⑥ Transitive:-
A relation R on a set A is called transitive if whenever $(a,b) \in R$ and $(b,c) \in R$, then $(a,c) \in R$ for $a,b,c \in A$.

Que. Are the following relations on $\{1,2,3,4\}$ transitive?

DELUXE
PAGE NO.:
DATE:

① $R = \{(1,1), (1,2), (2,2), (2,1), (3,3)\}$ Yes
② $R = \{(1,3), (3,2), (2,1)\}$ No.
③ $R = \{(2,4), (4,3), (2,3), (4,1)\}$ No.

* Combining Relations :-

↳ Relations are sets and therefore, we can apply the usual set operations to them.

↳ If we have two relations R_1 & R_2, and both of them are from a set A to a set B, then we can combine them to $R_1 \cup R_2$, $R_1 \cap R_2$, or $R_1 - R_2$.

↳ In each case, the result will be another relation from A to B.

* n-ary Relations:-

↳ In order to study an interesting application of relations, namely databases, we first need to generalize the concept of binary relations to n-ary relations.

- Let $A, A_2, \ldots A_n$ be sets. An n-ary relations on these sets is a subset of $A_1 \times A_2 \times \ldots \times A_n$. The sets $A_1, A_2 \ldots A_n$ are called the domains of the relation, and n is called its degree.

eg. Let $R = \{(a,b,c) \mid a = 2b \wedge b = 2c$ with $a,b,c \in N\}$

what is degree of R? 3, so its elements are triples

what are its domains?
Its domain are all equal to the set of integers.

eg Is (2, 4, 8) in R? No
Is (4, 2, 1) in R? Yes.

* Representing Relations :-

→ we already know different ways of representing relations.
1. zero-one matrices
2. directed graph.

↪ If R is a relation from $A = \{a_1, a_2 \ldots a_n\}$ to $B = \{b_1, b_2 \ldots b_n\}$ then R can be represented by the zero-one matrix $M_R = [m_{ij}]$ with
$m_{ij} = 1$ if $(a_i, b_j) \in R$, and
$m_{ij} = 0$ if $(a_i, b_j) \notin R$.

NOTE: For creating this matrix we first need to list the elements in A & B in a particular, but arbitrary order.

DELUXE
PAGE NO.:
DATE:

eg. How can we represent the relation $R = \{(2,1), (3,1), (3,2)\}$ as a zero-one matrix?

→ $M_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$

↳ They are square matrices.

Que What do we know about matrices representing reflexive relations?

→ All the elements on the diagonal of such matrices M_{ref} must be 1s

$$M_{ref} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

Que. What do we know about the matrices representing symmetric relations?

→ These matrices are symmetric, that is $M_R = (M_R)^T$

$$M_R = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \qquad M_R = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Sym matrix
Sym relation

non sym relation
non sym matrix

* Representing Relations :-

- The Boolean operations join and meet can be used to determine the matrices representing the union and the intersection of two relations, respectively.

- To obtain the join of two zero-one matrices, we apply the Boolean "or" function to all corresponding elements in the matrices

- To obtain the meet of two zero-one matrices, we apply the Boolean "and" function to all corresponding elements in the matrices

Eg. Let the relations R & S be represented by the matrices

$$M_R = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad M_S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

What are the matrices representing $R \cup S$ & $R \cap S$?

→ These matrices are given by

$$M_{R \cup S} = M_R \vee M_S = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$M_{R \cap S} = M_R \wedge M_S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

DELUXE
PAGE NO.:
DATE:

* Representing Relations Using Matrices :-

Do you remember the boolean product of two zero-one matrices ?

Let $A = [a_{ij}]$ be an $m \times K$ zero-one matrix and $B = [b_{ij}]$ be a $K \times n$ zero-one matrix.

Then the Boolean product of A and B, denoted by $A \odot B$, is the $m \times n$ matrix with (i, j)th entry $[c_{ij}]$, where

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \cdots \vee (a_{iK} \wedge b_{Kj})$$

$c_{ij} = 1$ if and only if at least one of the terms $(a_{in} \wedge b_{nj}) = 1$ for some n; otherwise $c_{ij} = 0$.

Eg. Find the matrix representing R^2, where the matrix representing R is given by.

$$M_R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

→ The matrix for R^2 is given by

$$M_{R^2} = M_R^{[2]} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

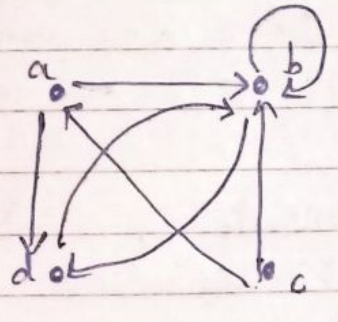
* Representing Relations Using Digraphs:-

↳ A directed graph, or digraph, consists of a set V of vertices (or nodes) together with a set E of ordered pairs of elements of V called edges (or arcs)

↳ The vertex a is called the initial vertex of the edge (a,b), and the vertex b is called the terminal vertex of this edge.

NOTE: We can use arrows to display graphs.

Eg. Display the diagraph with $V = \{a, b, c, d\}$
$E = \{(a,b), (a,d), (b,b), (b,d), (c,a), (c,b), (d,b)\}$.



An edge of the form (b,b) is called a loop.

↳ Obviously, we can represent any relation R on a set A by the diagraph with A as its vertices and all pairs $(a,b) \in R$ as its edges.

↳ Vice versa, any diagraph with vertices V and edges E can be represented by a relation on V containing all the pairs in E.

DELUXE
PAGE NO.:
DATE:

↳ This one-to-one correspondence between relations and diagraphs means that any statement about relations also applies to diagraphs, and vice versa.

✱ Equivalence Relations:-

↳ Equivalence relations are used to relate objects that are similar in some way.

↳ A relation on a set A is called an equivalence relation if it is reflexive, symmetric, and transitive.

↳ Two elements that are related by an equivalence relation R are called equivalent.

↳ Since R is symmetric, a is equivalent to b whenever b is equivalent to a.

↳ Since R is reflexive, every element is equivalent to itself.

↳ Since R is transitive, if a & b are equivalent and b and c are equivalent, then a and c are equivalent.

Obviously, these three properties are necessary for a reasonable definition of equivalence.

DELUXE
PAGE NO.
DATE:

Eg. Suppose that R is the relation on the set of strings that consist of English letters such that aRb if and only if $l(a) = l(b)$, where $l(x)$ is the length of the string x. Is R an equivalence relation?

→ R is reflexive, because $l(a) = l(a)$ and therefore aRa for any string a.

→ R is symmetric, because if $l(a) = l(b)$, then $l(b) = l(a)$. so if aRb then bRa.

→ R is transitive, because if $l(a) = l(b)$ & $l(b) = l(c)$ then $l(a) = l(c)$, so aRb, and bRc implies aRc.

∴ R is equivalence relation.

* Equivalence classes :-

↳ Let R be an equivalence relation on a set A. The set of all elements that are related to an element a of A is called the equivalence class of a.

↳ The equivalence class of a with respect to R is denoted by $[a]_R$.

↳ When only one relation is under consideration we will delete the subscript R and write $[a]$ for this equivalence class.

↳ If $b \in [a]_R$, b is called a representative of this equivalence class.

Que. In the previous example (strings of identical length), what is the equivalence class of the word mouse, denoted by [mouse]?

→ [mouse] is the set of all english words containing five letter.

→ For eg. 'horse' would be a representative of this equivalence class.

Theorem:- Let R be an equivalence relation on a set A. The following statements are equivalent:

: aRb

: $[a] = [b]$

: $[a] \cap [b] \neq \phi$

↳ A partition of a set S is a collection of disjoint nonempty subsets of S that have S as their union. In other words, the collection of subsets A_i, $i \in I$, forms a partition of S if and only if

(i) $A_i \neq \phi$ for $i \in I$

• $A_i \cap A_j = \phi$, if $i \neq j$

• $\cup_{i \in I} A_i = S$

Que. Let S be the set {u, m, b, r, o, c, k, s}
Do the following collections of sets partition S?

① {{m, o, c, k}, {r, u, b, s}} yes

② {{c, o, m, b}, {u, s}, {r}} No (k is missing)

③ {{b, r, o, c, k}, {m, u, s, t}} no (t is not in S)

④ {{u, m, b, r, o, c, k, s}} yes

⑤ {{b, o, o, k}, {r, u, m}, {c, s}} yes
{b, o, o, k}
= {b, o, k}

⑥ {{u, m, b}, {r, o, c, k, s}, ϕ} no
(ϕ is not allowed).

Theorem:- Let R be an equivalence relation on a set S. Then the equivalence classes of R form a partition of S. Conversely, given a partition $\{A_i \mid i \in I\}$ of the set S, there is an equivalence relation R that has the sets A_i, $i \in I$, as its equivalence classes.

Que. Let us assume that Frank, Suzanne and George live in Boston, Stephanie and Max live in Lubeck, and Jennifer lives in Sydney.

→ Let R be the equivalence relation $\{(a,b) \mid a$ and b live in the same city$\}$ on the set $P = \{$Frank, Suzanne, George, Stephanie, Max, Jennifer$\}$.

$R = \{(F,F), (F,S), (F,G), (S,F), (S,S), (S,G), (G,F), (G,S), (G,G), (ST,M), (M,ST), (ST,ST), (M,M), (J,J)\}$

↳ Then the equivalence classes of R are:
$\{\{F, S, G\}, \{ST, M\}, \{J\}\}$

This is a partition of P.

↳ The equivalence classes of any equivalence relation R defined on a set S constitute a partition of S, because every element in S is assigned to exactly one of the equivalence classes.

Que. Let R be the relation $\{(a,b) \mid a \equiv b \pmod{3}\}$ on the set of integers. Is R an equivalence relation?

↳ Yes, R is equivalence rel^n.

What are the equivalence classes of R?

$\{\{\ldots -6, -3, 0, 3, 6 \ldots\},$

$\{\ldots -5, -2, 1, 4, 7 \ldots\},$

$\{\ldots -4, -1, 2, 5, 8 \ldots\}\}.$

DELUXE
PAGE NO.:
DATE:

MATHEMATICAL INDUCTION

* Steps of MI:-

① Basic of Induction:-
② Induction step:

Que) Show that

$$1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6} \quad n \geq 1$$

① Basis of Induction, for $n=1$

$$1^2 = \frac{1(2)(3)}{6} = 1$$

② Induction step. Assume that

$$1^2 + 2^2 + \cdots + k^2 = \frac{k(k+1)(2k+1)}{6}$$

We have

$$1^2 + 2^2 + \cdots + k^2 + (k+1)^2 = \frac{k(k+1)(2k+1)}{6} + (k+1)^2$$

$$= (k+1)\left[\frac{k(2k+1)}{6} + k+1\right]$$

$$= \frac{(k+1)(2k^2 + 7k + 6)}{6}$$

$$= \frac{(k+1)(k+2)(2k+3)}{6}$$

$$= \frac{(k+1)[k+1+1](2(k+1)+1)}{6}$$

DELUXE
PAGE NO.:
DATE:

Que2. Prove by Induction that for $n \geq 1$

$$1.2 + 2.3 + \dots + n(n+1) = \frac{n(n+1)(n+2)}{3}$$

(1) Basic of Induction For $n=1$

$$1(2) = \frac{1(2)(3)}{3} = 2$$

(2) Induction step :-

assume that.

$$1.2 + 2.3 + \dots + k(k+1) = \frac{k(k+1)(k+2)}{3}$$

$$1.2 + 2.3 + \dots + k(k+1) + (k+1)(k+2) =$$
$$\frac{k(k+1)(k+2)}{3} + (k+1)(k+2)$$

$$= \frac{(k+1)(k+2)(k+3)}{3}$$

$$= \frac{(k+1)(k+1+1)(k+2+1)}{3}$$

Que3. Prove $1 + 2 + \dots + n = \frac{n(n+1)}{2}$

Que4. Prove $1^3 + 2^3 + \dots n^3 = \frac{n^2(k+1)^2}{4}$

Que5. Prove $n^3 + 2n$ is divisibly by 3.

Que6. Prove $n! > 2^n$ for $n \geq 4$.

Que 7 Show that any integer composed of 3^n identical digits is divisible by 3^n.
For eg 222 and 777 are divisible by 3, 222 222 222 by 9).

① Basis of Induction - for $n = 1$

$$3^n = 3^1 = 3$$

② Induction step:-

Let x be an integer composed of 3^{k+1} identical digits. We note that x can be written as

$$x = y \times z$$

where y is an integer composed of 3^k identical digits and

$$z = 10^{2 \cdot 3^k} + 10^{3^k} + 1 = 1\underbrace{000000\ldots0}_{3^k - 1\ 0s}1\underbrace{000000\ldots0}_{3^k - 1\ 0s}1$$

Since we assume that y is divisible by 3^k, and z is clearly divisible by 3, we conclude that x is divisible by 3^{k+1}.

FINITE AUTOMAATA

* Basics of automata :-

① Alphabets :- It is any finite non-empty set of symbols is known as alphabets

eg $\{1, 2, 3 \ldots\}$ ×

$\{A, B, C\}$ ✓

② Strings :- Finite sequence of symbols over given alphabets.

eg $\Sigma = \{0, 1\} = \{0, 1, 01, 10\}$

$\Sigma = \{a, b\} = \{a, b, aa, bb, ab, ba\}$

NOTE :- ϵ is a valid string of length of 0.

③ Prefix of string :- Sequence of leading symbols over a given string.

eg TOC

$\{\epsilon, T, TO, TOC\}$

NOTE :- Number of prefix of string = $n+1$.

④ Suffix of string :- Sequence of trailing symbols over a given string.

eg TOC

$\{\epsilon, C, OC, TOC\}$

NOTE :- Number of suffix of string = $n+1$

DELUXE
PAGE NO.:
DATE:

⑤ Substring :- Any consecutive sequence of symbols over a given string.

eg TOC
Length of 0 = ϵ
" of 1 = 3 substr = T, O, C
" of 2 = 2 substr = TO, OC
" of 3 = 1 substr = TOC

NOTE :- number of substring = $\frac{(n+1)n}{2} + 1$

Que. How many K length string are there with string of length n?
→ $n-K+1$.

⑥ Language :- Any set of strings over a given alphabets is called language.
eg {0, 1, 01, 10, 101, ...} infinite language
{0, 10, 01} finite language
L = {ϵ} or {ϕ} finite language

Que symbol = {0, 1}
{0, 1, 10, 012, 2} It is not a language.

⑦ Grammar :- Set of rules used to described strings of a languages.
↳ It is represented using G = {V, T, P, S}
where V = set of variables
T = set of terminals

P - Production symbol
S - starting symbol.

Que. $L = \{a^n b^n \mid n > 1\}$
$G: S \rightarrow aSb \mid ab$

For terminating aaabbb
else for continuing aaaSbbb

→ S is recursive.

$V = \{a, b\}$
$T = \{a, b\}$
$P = \{aSb\}$
$S = \{a\}$

* Finite Automata:-

↳ It is a mathematical model, which contains finite no of steps, having single starting state and transition define between them.

Type of finite automata:-

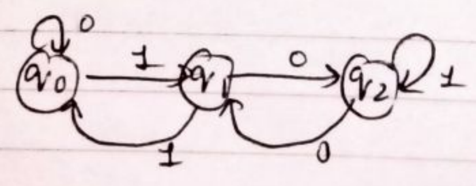
① Language Recognizers
  ↳ DFA
  ↳ NFA
  ↳ ϵ NFA

② O/P generator
  ↳ moore machine
  ↳ mealey machine

* ① Determinstic Finite Automata:-

↳ It is a finite automata in which from each and every state on every I/P symbol exactly one transition should exist.

×



✓

DFA = $\{Q, \Sigma, \delta, q_0, F\}$

Here
Q = Infinite no. states
Σ = Input alphabet symbols
δ = Transition funⁿ
q_0 = Initial state (only one)
F = State of final states (can be multiple) (including zero state)

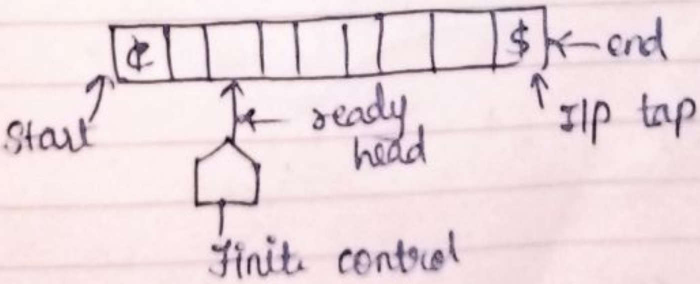
* Transition function is of two types:-

① Direct Transition funⁿ :- $\delta : Q \times \Sigma \rightarrow Q$
② Indirect Transition funⁿ :- $\delta : Q \times \Sigma^* \rightarrow Q$

$\Sigma^* \rightarrow$ Including initial state

* Acceptance Machine :-

↳ By reading complete string from left to right while the end of the string, DFA enters into final state, then string is accepted otherwise not.

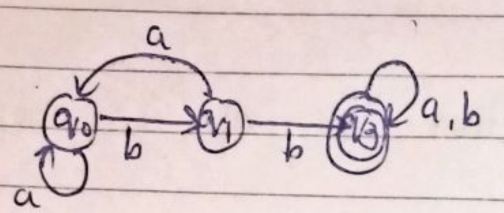


DELUXE
PAGE NO.:
DATE:

NOTE: ① Every DFA will accept only one language
② All the string of the language should be accepted by given DFA.

③ Given DFA should not accept any invalid string.

Que 1.

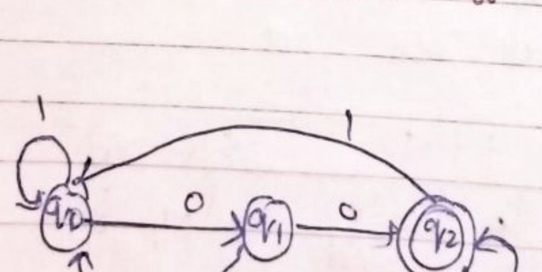


Represent which type of language it is?

× (A) string starting with bb (aabb contradiction)
× (B) string ending with bb (bbaa contradicts)
× (C) String contains atleast two b. (abab)
✓ (D) NONE

this string contains atleast two b but it is not accepted by the language and do not reach at final state.

Gate 2010.
Que 2.



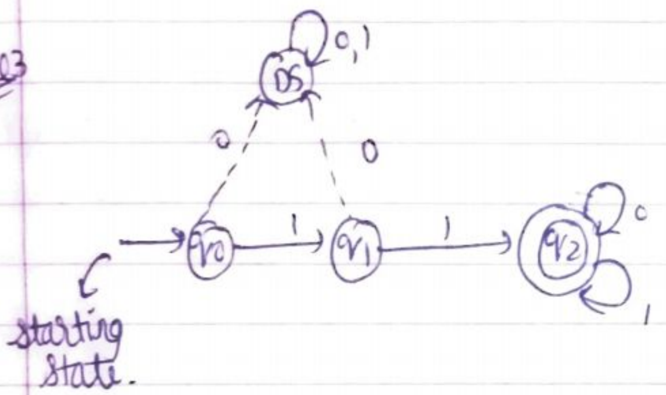
Which option is correct for representing the language?

× (A) starting with 0 (can start with 1)
× (B) ~~starting~~ ending with 0 (string "0" it does not end at final state.)
× (C) substring with 00 (001 contradicts)
✓ (D) NONE

→ accept ending with 00

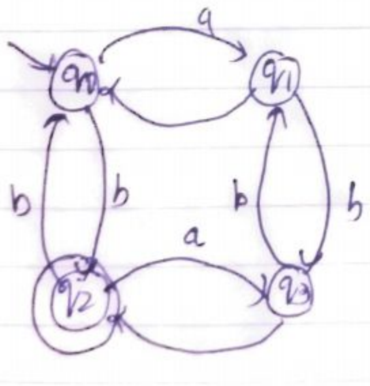
* **Dead state** :- If on any state, transition is not possible, then we can represent that transition on dead state

Que3



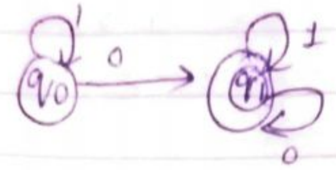
✓ (A) Starting with 11
× (B) ending with 11 (011 contradicts)
× (C) Substring with 11 (011 contradicts)
× (D) NONE

Que



× (A) even no of a & b
✓ (B) even no of a & odd no of b
{ b, bbb, bbbbb }
× (C) even no of b & odd no of a
× (D) odd no of a & b.

* **Notations:-**
  1. Transition diagram
  2. Transition table.



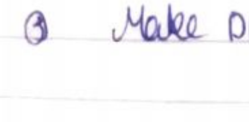
O/P / I/P	0	1
q_0	q_1	q_0
q_1	q_1	q_1

* **Language to DFA conversion:-**

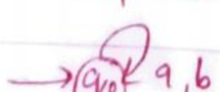
Step :- 1 ① Σ (Identify Input symbols)
Step :- 2 ② Write sample language $L = \{\ \}$
Step :- 3 ③ Make DFA.

NOTE :- Every DFA will accept one language but for every language there may be many number of DFA's, out of which the DFA contains minimum number of states that is called minimal DFA.

Que. construct a minimal DFA that accepts all strings of a's and b's including ϵ / λ.

→ 

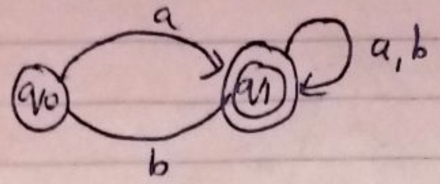
NOTE :- To represent null language



q_f → final state , Dead state → (DS) / (q_d)

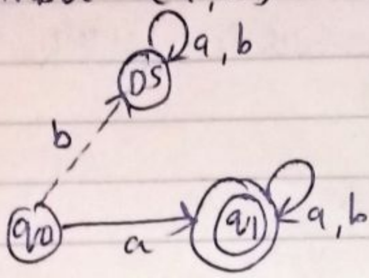
Que. construct a minimal DFA for all a's & b's excluding ϵ, λ or null transition.

→



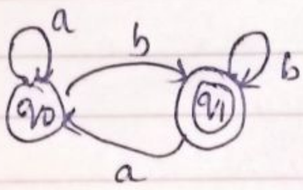
Que. Construct a minimal DFA where every string is starting with 'a', with i/p symbol {a, b}

→



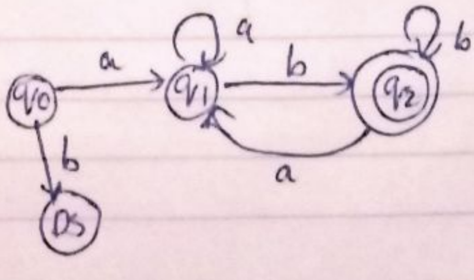
Que construct a DFA with each string ending with b over symbol {a, b}

→



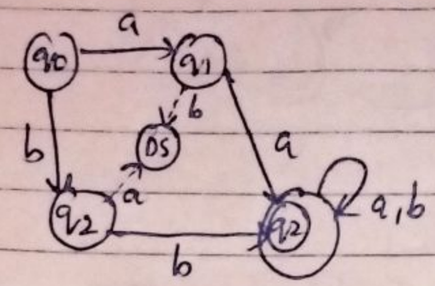
Que. Construct minimal DFA starting with a & ending with b.

→



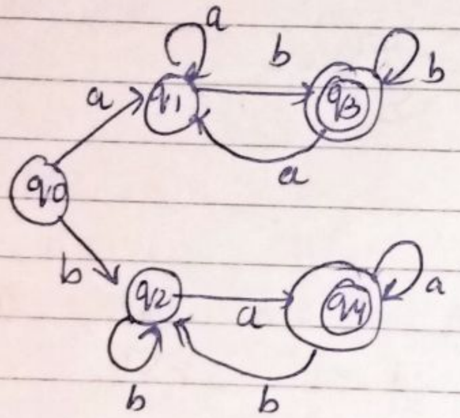
Que Construct DFA with first two symbol are same over symbol {a,b}

→

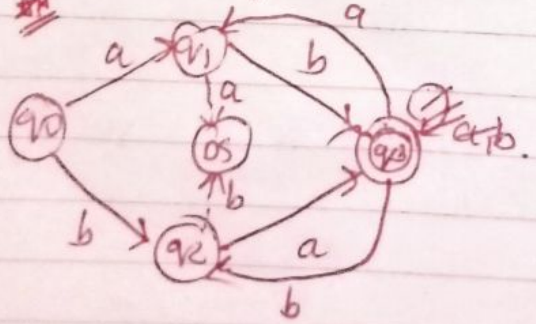


Que. Construct DFA with start & end with different symbols.

→

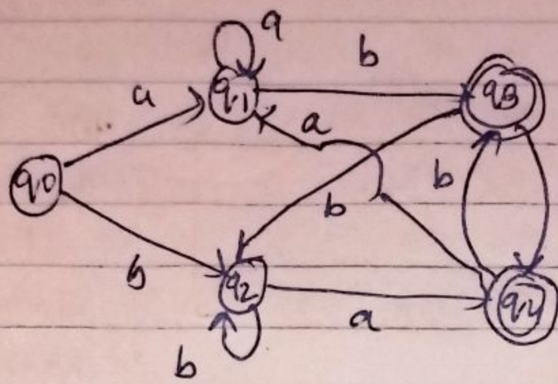


→ Four state diagram , don't accept baa , abb string.



Que construct DFA where last two symbol are different.

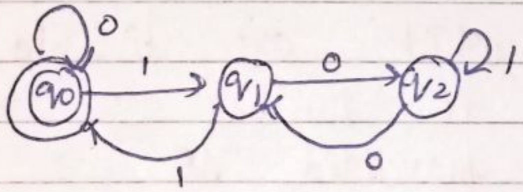
→



Que construct a minimal DFA that accepts all binary number which are divisible by 3.

→

O/P / I/P	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

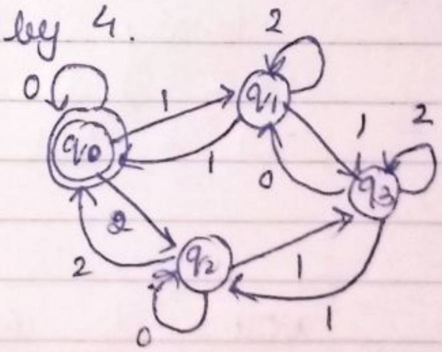


NOTE:- Divisible by n → n states

Que. Construct a DFA that accepts all base 3 numbers with are divisible by 4.

→

O/P / I/P	0	1	2
q_0	q_0	q_1	q_2
q_1	q_3	q_0	q_1
q_2	q_2	q_3	q_0
q_3	q_1	q_2	q_3



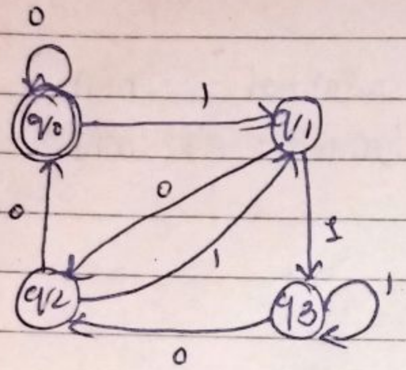
Que Construct DFA for binary number divisible by 4.

→

o/p | I/P	0	1
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_0	q_1
q_3	q_2	q_3

we can merge q_0 & q_2 in final state since q_0 & q_2 have same entries

This short cut method does not give minimal DFA



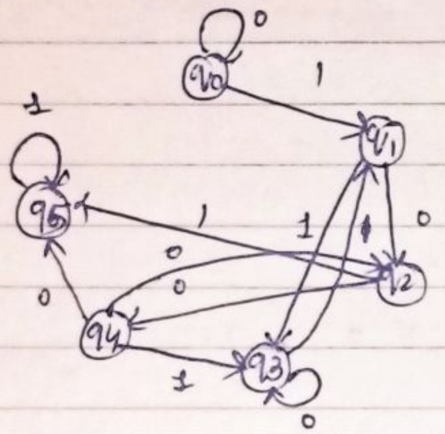
NOTE:- The DFA that accept all base-m number which are divisible by n, requires n number of states where n is not guaranted to be minimum.

Que Construct a DFA that accepts all binary number which are divisible by
(i) 2 & 3. (ii) 2 but not by 3
(iii) 3 but not by 2 (iv) not by 2 or 3.

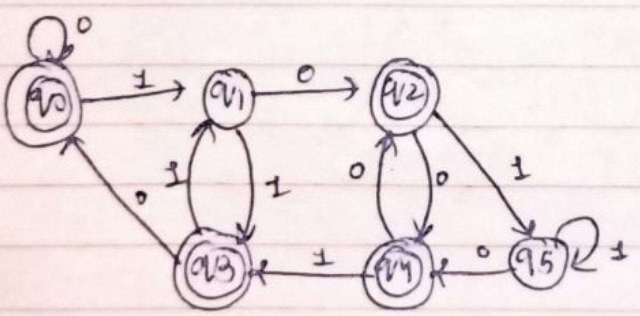
NOTE:- If we talk about two number then number of state = LCM(a,b).

(i)

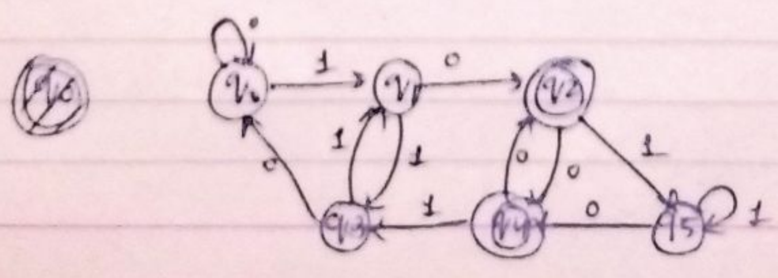
O/P | I/P	0	1
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_5
q_3	q_0	q_1
q_4	q_2	q_3
q_5	q_4	q_5



2 → 0, 2, 4
3 → 0, 3
final state = 0, 2, 3, 4



(ii) 2 but not by 3
2 → 0, 2, 4   final state = 2, 4
3 → 0, 3

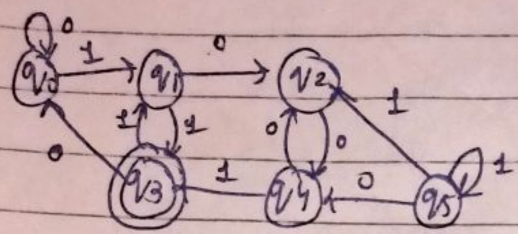


DELUXE
PAGE NO.:
DATE:

(iii) 3 but not by 2

2 → 0, 2, 4

3 → 0, 3 final state = 3

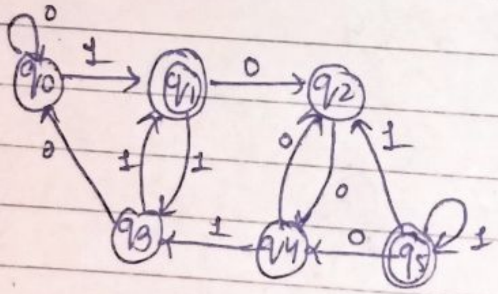


(iv) not by 2 or 3

2 → 0, 2, 4

3 → 0, 3

final state = 1, 5

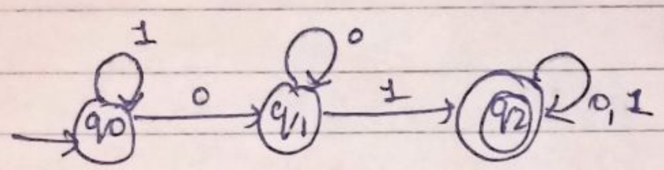


Que. Find the number of states in a DFA that accepts all hexadecimal numbers which are divisiby by 3 or 5 but not by 7.

→ ~~16~~ → Lcm(3,5,7) = 105

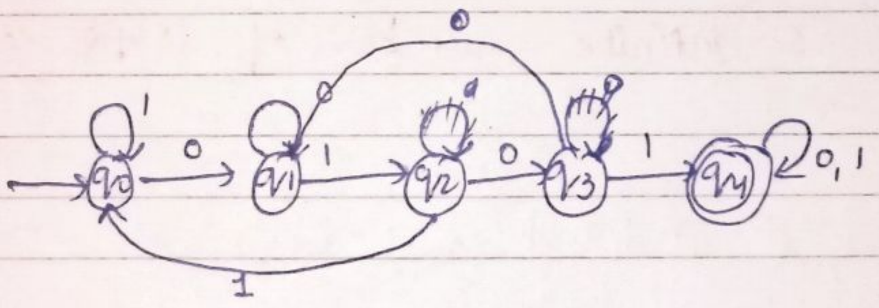
Que. Construct a minimal DFA that accepts all string of 0 & 1 where each string contains '01' as a substring.

→



Que. Construct a DFA with '0101' as a substring.

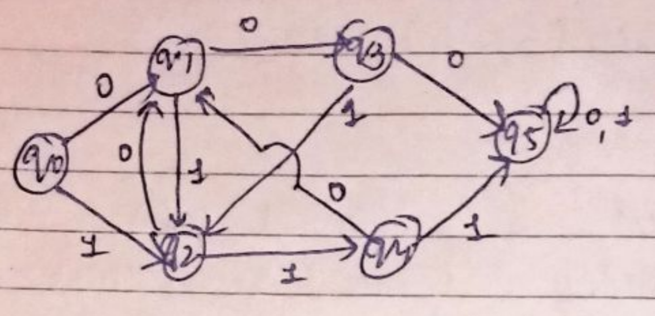
→



NOTE:- If length of substring is n, total no of state = n+1

Que construct a DFA where each string contais triplet "xxx" as a substring

→ xxx → 000 or 111



Que $L = \{ a^n b^n \mid n \geq 1 \}$ ↳ (ε, ab, aabb, aaabbb ...)

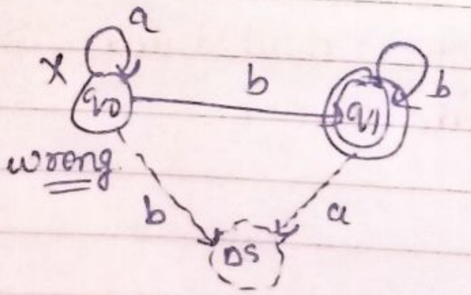
→ No DFA is made because DFA does not have enough power to compare.

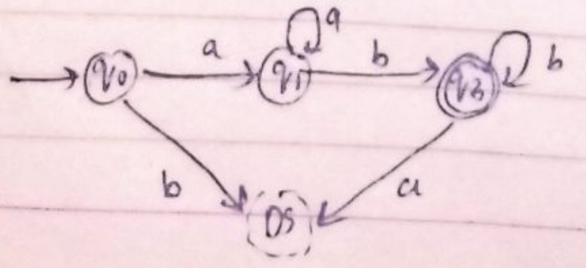
eg. aaabbbb

3 no of a's ↙ ↘ 4 no of b's

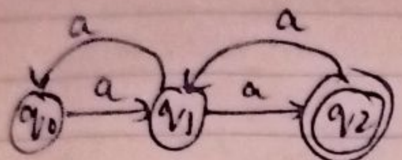
infinite number of loops can be done.

Que $L = \{ a^n b^m \mid n \geq 1 \ m \geq 1 \}$





Que. $L = \{ a^{2n} \mid n \geq 1 \}$



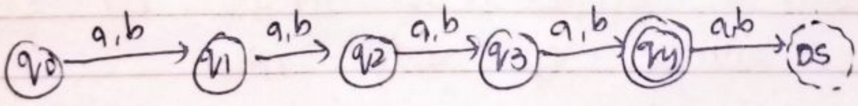
NOTE :- a^{Kn} $\rightarrow$ number of states $= K+1$

Que. How many maximum number of DFA we can construct using two state x & y, I/P Σ = {a, b} including null DFA ?

$\rightarrow$ Number of DFA's $= 2^n * n^{(2*n)}$

Que. construct a DFA all string of a, b where length of each string is exactly 4.

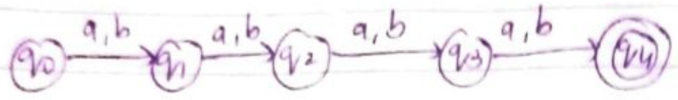
$\rightarrow$



NOTE: length of string of exactly n then no of states $= n+2$.

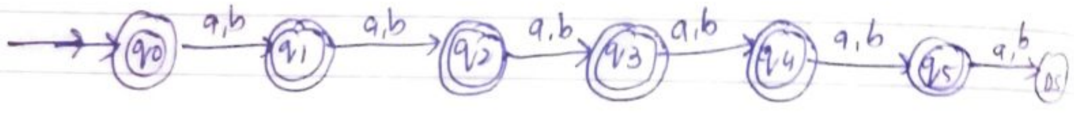
Que construct a DFA over a, b where length of string is atleast 4.

→



NOTE :- If length of string is n, then atleast n length then numbers of state = n+1.

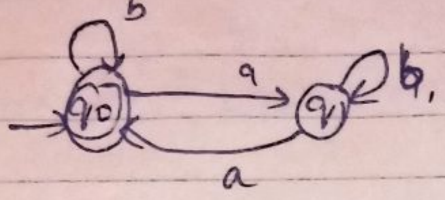
Que. E = {a, b}
length of string is atmost 5.



NOTE :- Atmost n length of string, number of state = n+2.

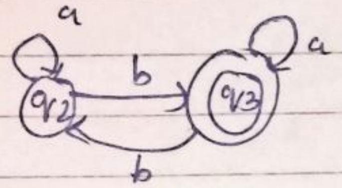
Que. Construct a minimal DFA that accepts all string of a, b where number of a's is even.

→



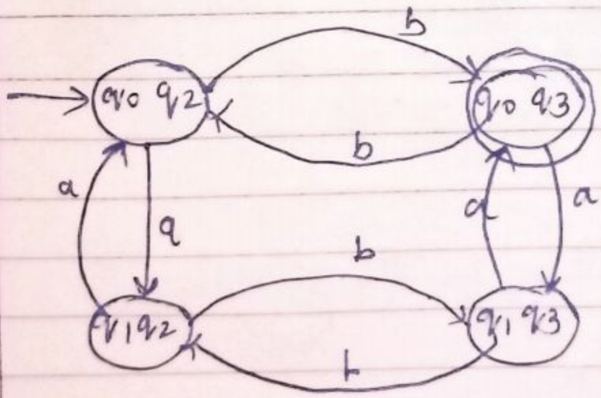
Que Number of b's are odd.

→



Que Construct a DFA where number of a is even and b is odd.

→ cross product = $\{q_0, q_1\} \times \{q_2, q_3\}$
= q_0q_2, q_1q_2, q_0q_2, q_1q_3

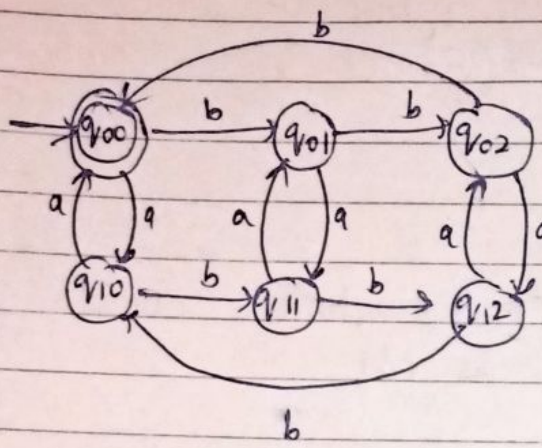


Que Construct a DFA that accepts all a's & b's where number of a's are even & no of b's are divisible by 3.

→ A state q_{xy} means $n_a \bmod 2 = x$, $n_b \bmod 3 = y$

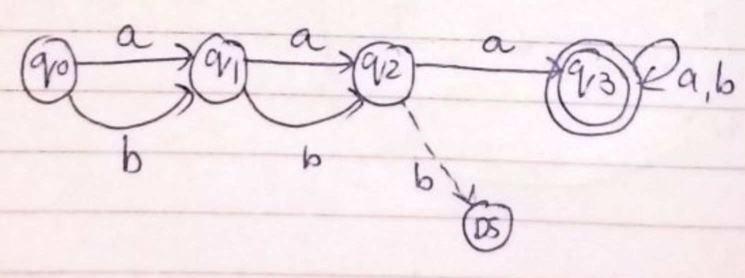
q_{00} means $n_a \bmod 2 = 0$, $n_b \bmod 3 = 0$

$q_{00} \times a \rightarrow q_{10}$
$q_{00} \times b \rightarrow q_{01}$ and so on.

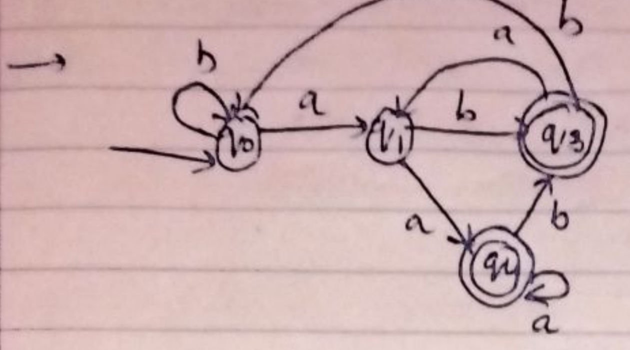


Que Construct a minimal DFA that accepts all a's & b's where 3rd input symbol is 'a'. while reading from left to right.

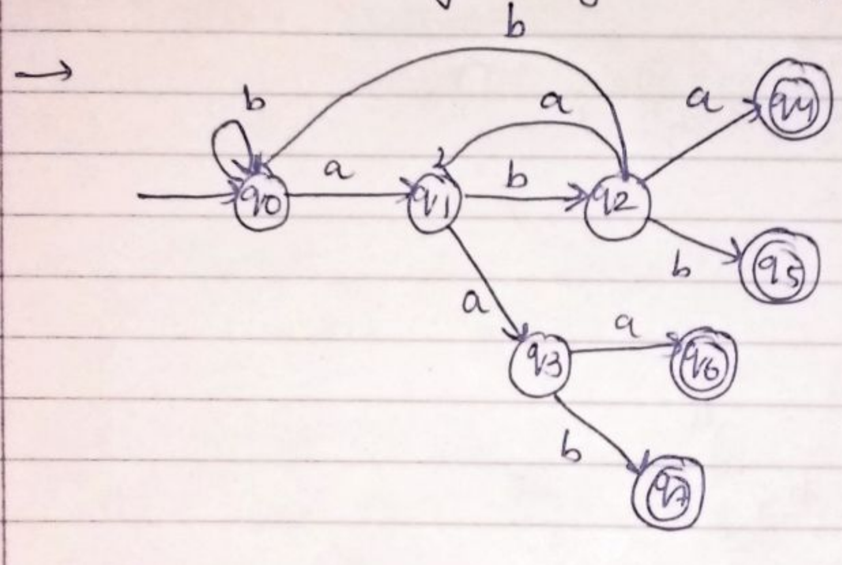
→



Que Construct a DFA where second input symbol is 'a' while reading right to left.



Que Construct a DFA where third input symbol is 'a' while reading right to left.



NOTE: DFA are very complex to design for some language whenever DFA is difficult, we can construct NFA.

* Complement of DFA:-

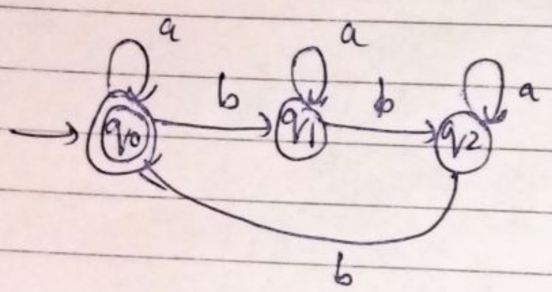
Method:

By interchanging final and non-final states, we can get complement of the DFA.

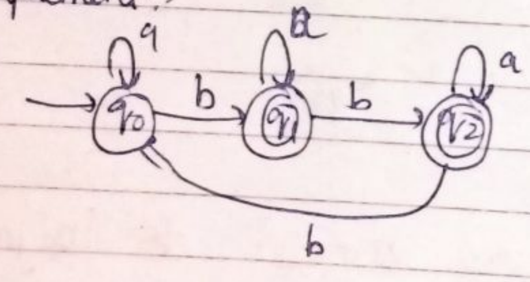
$$\boxed{L' = \Sigma^* - L}$$

Que. Construct a minimal complementing DFA for language over a, b where number of b's divisibly by 3.

→



Complement:-



* Decision Properties of Finite Automata:-

① Decidable Problem:-
A problem P is said to decidable if there exists some generalised algorithm to solve the problem.

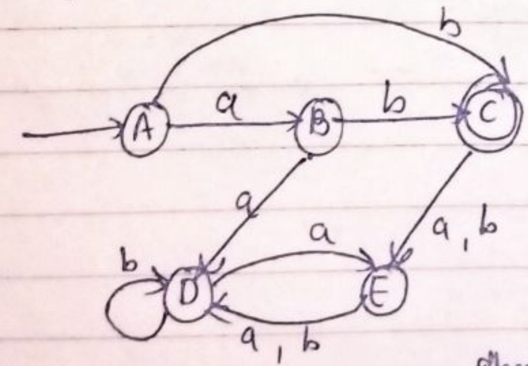
② Undecidable Problem:-
There is no generalised algorithm to solve the problem.

(i) Emptiness Probem:-
↳ checking whether language accepted by given DFA is empty or non-empty.

Step1. Eliminate all inaccessible or non-reachable states from automata.

Step2. In the resultant automata, if there exist atleast one final state then it accepts non empty language, otherwise accepts empty language.

Que



non-reachable state = D & E

There exists one final state.
Thus, it is non- empty

NOTE: Emptiness problem is a decidable problem.

(ii) Finiteness Problem:-

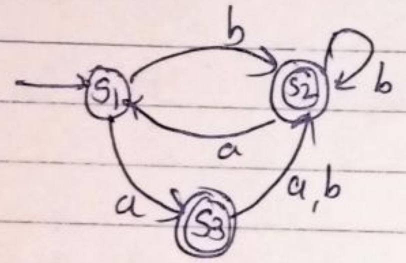
↳ checking whether given automata accepts finite language or infinite languages. This problem is also decidable, because of following steps-

Step 1. Eliminate all non-reachable state.

Step 2. Eliminate states from which we can not reach to final state.

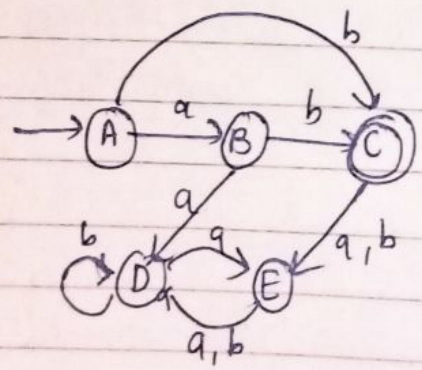
Step 3. In the resultant automata, there exists any loops, self loops or cycle then automata accepts infinite language otherwise finite languages.

Que



→ Infinite language

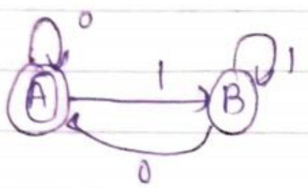
Que

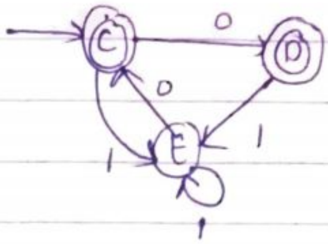


→ Finite language.

(iii) Equilance Problem:-
↳ Checking whether given two automata accepts same languages or not.

Imp Que



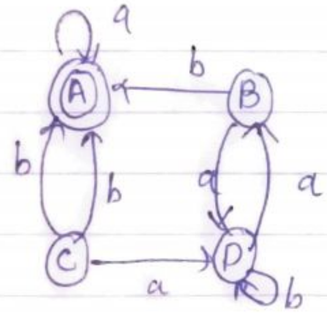


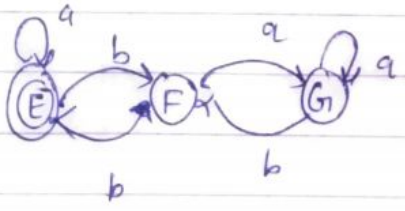
	0	1
AC	AD	BE
AD	AD	BE
BE	AC	BE

→ Equivalence language.

NOTE: Fore equilance problem, the combine table should have both as final- final or non-final - non-final.

Que.





	a	b
AE	AE	CF
BF	DG	AE
BG	DG	AF
CF		
CG		
DF		
DG		

→ A = final
F = non- final

→ Not equilance

(iv) Membership problem:-

↪ Checking whether given string is valid in a given automata or not.

REGULAR EXPRESSIONS

* Regular expressions

↪ It is a simplest way of representing a regular language. For every regular language, we can construct equivalent regular expression by taking symbols as an iputs & operators as a (*, +, •).
{*, +, •}

Eg. ① $\Sigma = \{a\}^*$
$L^0 = \{\epsilon\}$
$L^1 = \{a\}$
$L^2 = \{aa\}$
$\Sigma^* = L^0 \cup L^1 \cup L^2 \ldots L^n = \{\epsilon, a, aa, aaa \ldots\} \cdot L^*$

$L^* = \{\epsilon, a, aa, \ldots\}$ $n \geq 0$ (klean closure)
$L^+ = \{a, aa \ldots\}$ $n \geq 1$ (positive closure)

② $\Sigma = \{a, b\}$
$L^* = \{\epsilon, a, b, ab, ba..\} = (a+b)^*$
$\downarrow$
OR

($L_1 L_2$ or $L_1 \cdot L_2$ → AND → b from L_2.)
↓
a comes from L_1

(For particular pattern (•) operator is used)

* Rules:-

1. $L = \{\} = \phi$
2. $L = \{\epsilon\} = \epsilon$
3. $L = \{a\} = a$
4. $L = \{a, b\} = a+b$
5. $L = \{ab\} = a \cdot b$
6. $\Sigma = \{a, b\}$
   $L = \{\epsilon, a, b, ba, ab \ldots\} = (a+b)^*$
7. $\Sigma = \{a, b\}$ (positive)
   $L = \{a, aa \ldots\} = L^+$
8. $\Sigma = \{a\}$ (kleen)
   $L = \{\epsilon, a, aa ..\} = L^*$

Que Find the regular expression that generate $\Sigma = \{0, 1\}$ where each string starting & ending with different symbols

→ Approach → $0(0+1)^*1$
$1(0+1)^*0$

$$0(0+1)^*1 + 1(0+1)^*0$$

Que $\Sigma = \{0, 1\}$ starting and ending with same symbol.

→ $0(0+1)^*0 + 1(0+1)^*1 + 0 + 1$

DELUXE
PAGE NO.:
DATE:

Que $\Sigma = \{0, 1\}$ where 3^{rd} input symbol is '0' reading from left to right

→ $(0+1)(0+1)\,0\,(0+1)^*$

Que Find the number of states in given regular expression.
$(a+b+c)(a+b+c) \ldots (n-2)$ times

→ $\underbrace{n-2}_{\text{exactly } (n-2) \text{ length.}} + 2 = n$

Que $\Sigma = \{0, 1\}$, length of string is atleast 4.

→ $(0+1)(0+1)(0+1)(0+1)^+$

<u>OR</u>

$(0+1)(0+1)(0+1)(0+1)(0+1)^*$

Que $\Sigma = \{a, b\}$ length of string is atmost 5

→ $(a+b+\epsilon) \cdot (a+b+\epsilon) \cdot (a+b+\epsilon) \cdot (a+b+\epsilon) \cdot (a+b+\epsilon)$

DELUXE
PAGE NO.:
DATE:

Que Length of $\Sigma = \{0,1\}$ and length of string is divisible by 4.

$\rightarrow$ $[(0+1)^*(0+1)(0+1)(0+1)]^*$

Atleast 4 length $[(0+1)(0+1)(0+1)(0+1)]^+$

Que. $\Sigma = \{0,1\}$ and length of string is odd.

$\rightarrow$ $(0+1)\cdot[(0+1)(0+1)]^*$

Que String $\Sigma = \{0,1\}$ starting with '0' and total length is odd., OR starting with '1' and length is even.

$\rightarrow$ $0[(0+1)(0+1)]^* + 1[(0+1)(0+1)]^*(0+1)]$

Que. Each string starting with 010 or 110

$\rightarrow$ $(0).(1).(0)(0+1)^* + (1)(1)(0)(0+1)^*$

Que Each string ending with (010) or 110

$\rightarrow$ $(a+b)^*(0)(1)(0) + (0+1)^*(1)(1)(0)$

Or

$(0+1)^*(010+110)$ Or $(0+1)^+(10)$

Que. $L = \{a^n b^n \mid n \geq 1\}$

→ ~~(a+b)~~ Regular expression is not possible becz, we cannot control value of n.

Que $L = \{a^n b^m \mid n, m \geq 0\}$

→ $(a^* . b^*)$

Que $L = \{a^n b^m \mid n, m \geq 1\}$

→ $a^+ . b^+$

Que. $L = \{a^n b^m \mid n+m \text{ is odd}\}$

→ if n = even m = odd

∴ $(aa)^* b(bb)^*$

if n = odd m = even

$a(aa)^* (bb)^*$

→ $a(aa)^*(bb)^* + (aa)^* b(bb)^*$

DELUXE
PAGE NO.:
DATE:

Que. $\Sigma = \{0,1\}$ where string contains atmost 4 zeros.

$\rightarrow 1^*(0^* + \epsilon)1^*(0+\epsilon)1^*(0+\epsilon)1^*(0+\epsilon)(1)^*$

Que. $\Sigma = \{0,1\}$ where no. of zeros is divisible by 3.

$\rightarrow (1^*01^*01^*01^*)^* + 1^+$

Que. $L = \{1, 2, 4, 8 \ldots 2^n\}$ where all no are in the form of binary.

$\rightarrow 0^* \cdot 1 \cdot (0^*)$

eg for hexadecimal representation

we represent 1 as 0001

Que $L = \{1, 2, 4, 8 \ldots 2^n\}$ where all string has base 1 and are uniary number.

$\rightarrow L = \{1^{2^n} \mid n \geq 1\}$

$\rightarrow$ no regular expression. because we can't do comparison..

Que Find RE. that generate all palindrome string over $\Sigma = \{अ\}$ odd length.

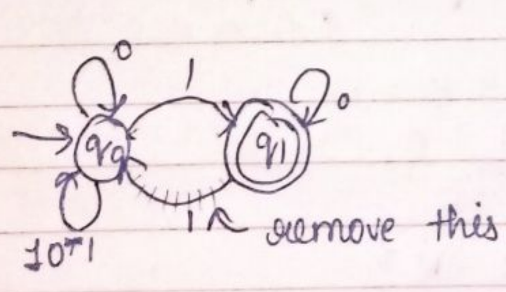
→ अ(अअ)*

NOTE: Palindrome over one symbol alphabet on a regular language can generate regular exp. But more than 1 symbol alphabet palindrome are not possible because it is non-regular.

Que Find RE that generate $\Sigma = \{0, 1\}$ where no. of '1' are odd in a string.

→ $0^* 1 0^* (0^* 1 0^* 1 0)^* 0^*$ (X)

DFA:



$q_0 \rightarrow 1 0^* 1$

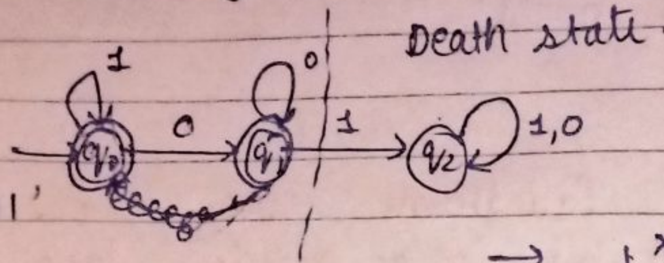
$(0 + 10^*1)^* \cdot (1) \cdot (0^*)$

→ '+' between RE on same state, otherwise between two state apply '.'.

DELUXE
PAGE NO. :
DATE:

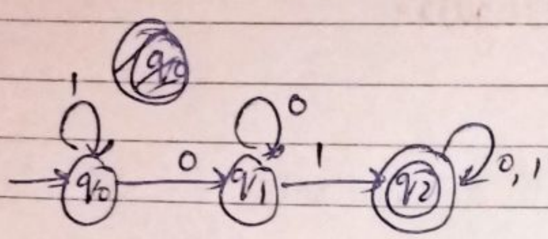
*Que Find R.E. that generates all string of 0's and 1's where each string does not contain '01' as substring

→ does not contain '01'



→ $1^* 0^*$

→ contains '01'



Que

* Identities of Regular Expression :-
(R → R.E.)

1. $R + \phi = \phi + R = R$
2. $\phi \cdot R = R \cdot \phi = \phi$
3. $R \cdot \epsilon = \epsilon R = R$
4. $R \cdot R^* = R^* R = R^+$
5. $R \cdot R^* + \epsilon = \epsilon + RR^* = R^*$
6. $(R^*)^+ = (R^+)^* = (R^*)^* = R^*$
7. $\phi^* = \epsilon$
8. $\phi^+ = \phi$
9. $\epsilon^* = \epsilon$
10. $\epsilon^+ = \epsilon$

(11) $(PQ)^*P = P(QP)^*$

** (12) $(a+b)^* = (a^* + b^*)^*$

$= (a^* + b)^*$

$= (a + b^*)^*$

$= (a^* \cdot b^*)^*$

$= a^*(ba^*)^*$

$= b^*(ab^*)^*$

Que. $P = ((01)^* + 1)^*$

$Q = (01)^*(1(01)^*)^*$

For the given P, Q, which of the following is correct?

(A) $P = Q$

(B) $P \subset Q$

(C) $P \supset Q$

(D) None.

$\rightarrow$ $\cancel{(01)}$ $(a+b)^*$

$(a+b)^* = a^*(ba^*)^*$

$P = Q$

$a = (01)^*$

$b = 1$

$((01)^*)^* = (01)^*$

$(R^*)^* = R^*$

PAGE NO.:
DATE:

Que For given R, S, & T :

$R = 1(0+1)^*$

$S = 11^* 0$

$T = 1^* 0$

which of the following is true ?

(A) $L(S) \subset L(R)$ & $L(T) \subset L(S)$

(B) $L(R) \subseteq L(S)$ & $L(T) \subseteq L(R)$

(C) $L(S) \subset L(R)$ & $L(S) \subset L(T)$

(D) None.

Que $(ab + aa + baa)^*$

which option is correct.

(1) (A) a b a a b a a a b a a

(2) (B) a a a a b a a a a

(3) (C) b a a a a a b a a a a b

(4) (D) b a a a a a b a a

(A) 1 & 2

(B) 1 & 3

(C) 2 & 3

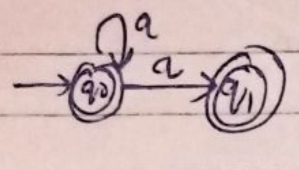
(D) None.

It can generate A, B, D
(1), (2), (4)

* NDFA or NFA:-

In NFA, from each and every state, on every I/P symbol there may be 0 number of transition or more than 1 transition or only one transition.

eg



→ b does not have any transition

→ a has 2 transitions from one state.

↳ DFA → $Q \times \Sigma \rightarrow 1$

NFA → $Q \times \Sigma \rightarrow \{0, 1, 2, \ldots\}$

NFA = $\{Q_0, \Sigma, \delta, q_0, F\}$

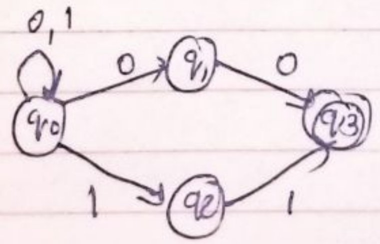
$\delta : Q \times \Sigma \rightarrow 2^Q$

or

$P(Q)$ → Power set of Q.

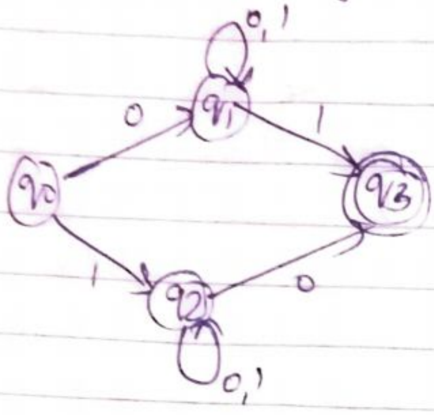
Que construct the NFA $\Sigma = \{0,1\}$ where last two symbols are same in every state.

→



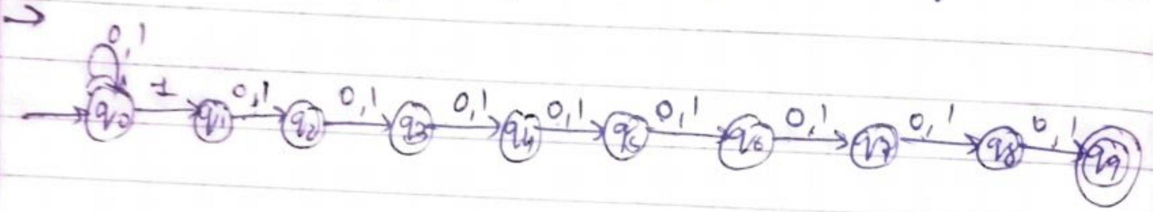
Que. Make the NFA where starting and ending with different symbols

→



Que. Construct the NFA where 9th symbol is '1' while reading from right to left.

→

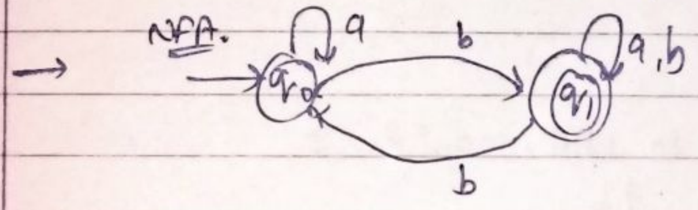


Conditions	No of state in DFA	No of states in NFA.
① when the length of string is n $\lvert w \rvert = n$	$n+2$	$n+1$
② $\lvert w \rvert \leq n$ (atmost)	$n+2$	$n+1$
③ $\lvert w \rvert \geq n$ (atleast)	$n+1$	$n+1$

Condition	DFA	NFA
(4) $\|w\| \bmod n$	n	n
(5) n^{th} i/p sym from LHS	$n+2$	$n+1$
(6) n^{th} i/p sym from RHS	2^n	$n+1$

* NFA TO DFA :-

Que Construct the DFA for given NFA :-



NFA transition table.

S | I/P	a	b
q_0	q_0	q_1
q_1	q_1	q_0, q_1

Step 1: Convert given NFA to NFA transition table.

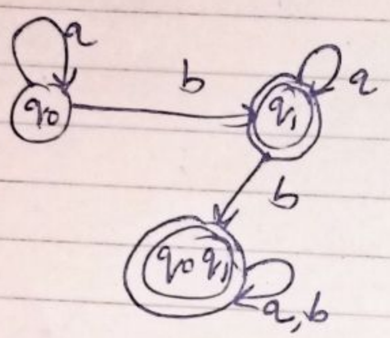
DFA transition table

	a	b
q_0	q_0	q_1
q_1	q_1	$[q_0\, q_1]$
$[q_0 q_1]$	$[q_0, q_1]$	$[q_0 q_1]$

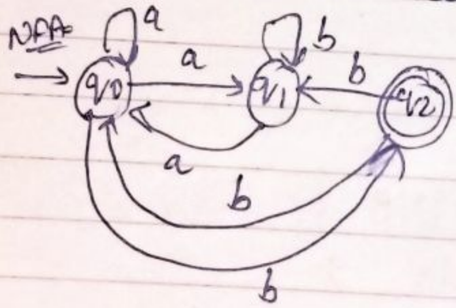
→ [if row has new state entry then bring that new state to new row entry.]

Step 2: Construct DFA transition table

Step 3: Draw DFA.



Ques: Convert NFA to DFA equivalent.

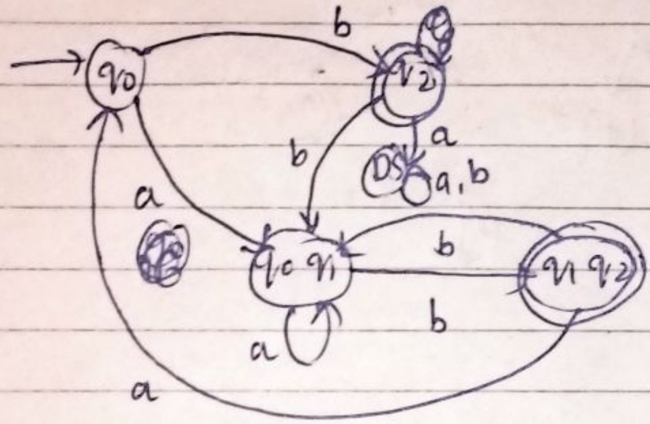


NFA table:-

	a	b
q_0	$[q_0, q_1]$	q_2
q_1	q_0	$[q_1]$
q_2	-	$[q_0\, q_1]$

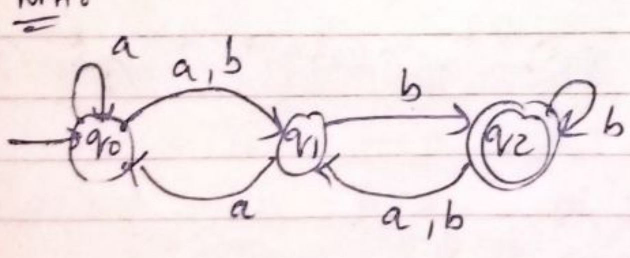
DFA table :-

	a	b
q_0	$[q_0 q_1]$	q_2
$[q_0 q_1]$	$[q_0 q_1]$	$[q_1 q_2]$
q_2	DS	$[q_0 q_1]$
$[q_1 q_2]$	q_0	$[q_0 q_1]$
$\rightarrow [q_0 q_1]$	$[q_0 q_1]$	$[q_1 q_2]$ ×
q_0	$[q_0 q_1]$	$[q_2]$ ×
DS	DS	DS



Que. Convert NFA to DFA equivalent :-

→ NFA.

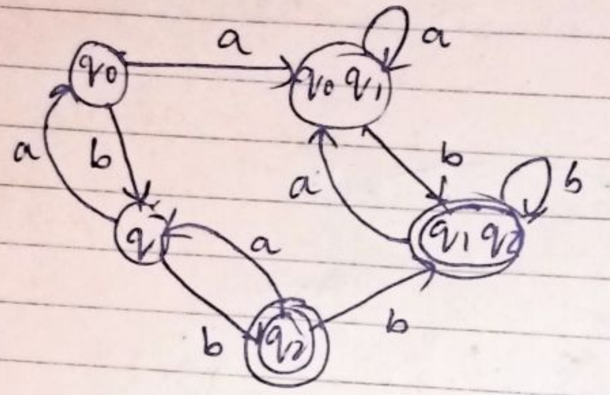


NFA table :-

	a	b
q_0	$[q_0 q_1]$	q_1
q_1	q_0	q_2
q_2	q_1	$[q_2 q_1]$

DFA table:-

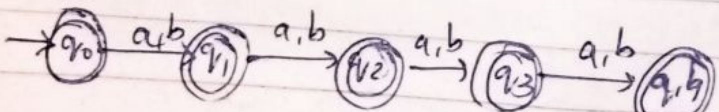
	a	b
q_0	$[q_0 q_1]$	q_1
$[q_0 q_1]$	$[q_0 q_1]$	$[q_1 q_2]$
q_1	q_0	q_2
$[q_1 q_2]$	$[q_0 q_1]$	$[q_1 q_2]$
q_2	q_1	$[q_1 q_2]$



* ϵ - NFA :-

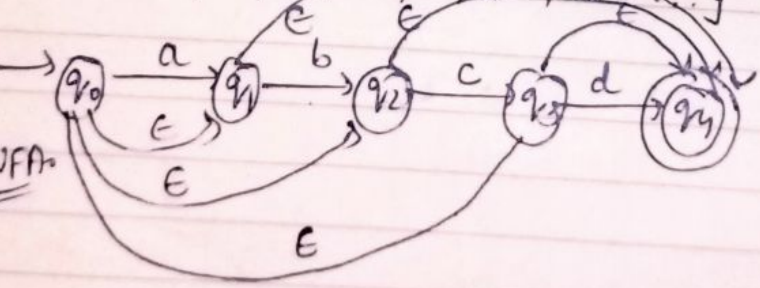
Que Construct a NFA that accepts all substrings of given 4 length string.

→



X (wrong)

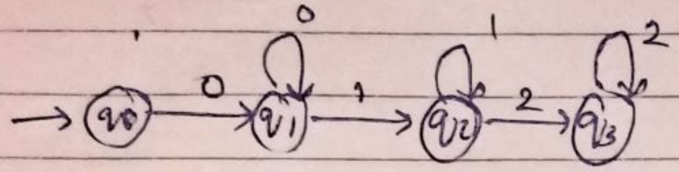
Sub. {ϵ, a, b, c, d, a ...}



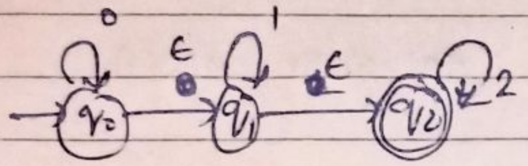
ϵ-NFA.

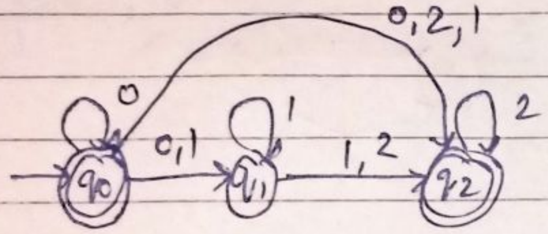
Que Construct the NFA for the $L = \{0^n 1^m 2^p \mid n \geq 0, m \geq 0, p \geq 0\}$

→



× (wrong.





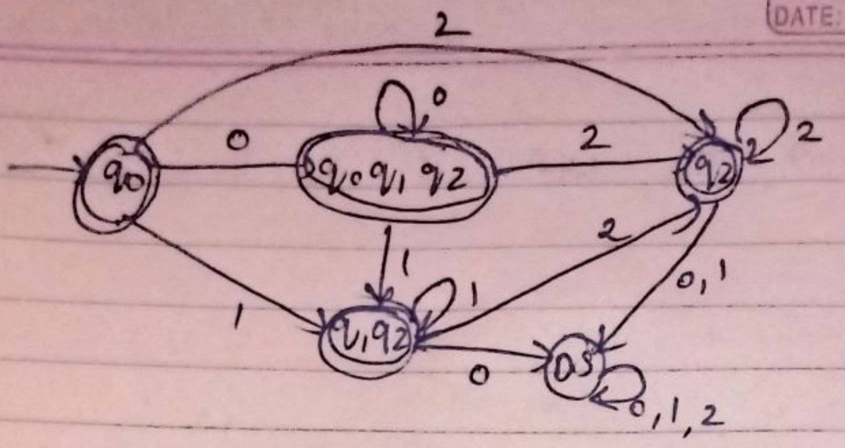
NFA

NFA table:-

	0	1	2
q_0	$[q_0, q_1, q_2]$	$[q_1, q_2]$	q_2
q_1	–	$[q_1, q_2]$	q_2
q_2	–	–	q_2

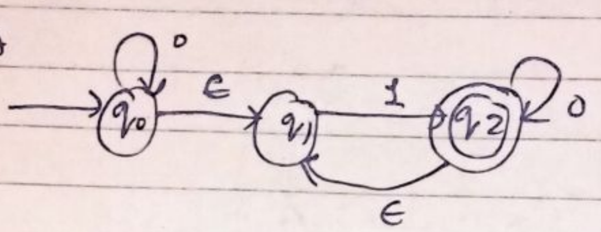
DFA table:-

	0	1	2
q_0	$[q_0 q_1 q_2]$	$[q_1 q_2]$	q_2
$[q_0 q_1 q_2]$	$[q_0 q_1 q_2]$	$[q_1, q_2]$	q_2
$[q_1, q_2]$	DS	$[q_1, q_2]$	q_2
q_2	DS	DS	q_2
DS	DS	DS	DS

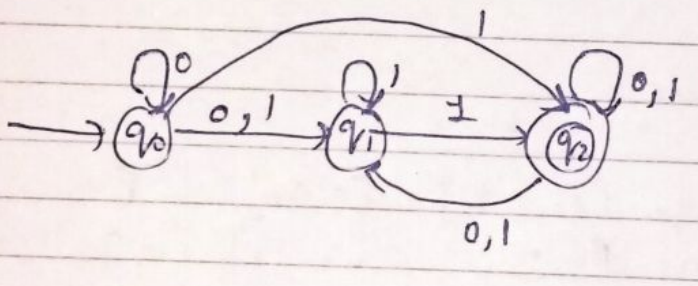


Que Construct NFA to given ϵNFA.

→ ϵ NFA

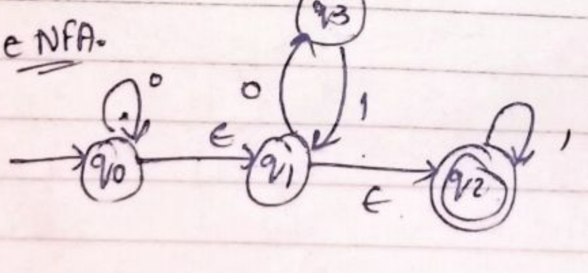


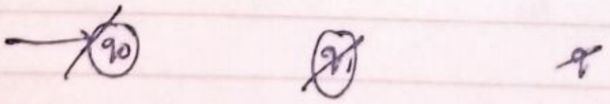
NFA.

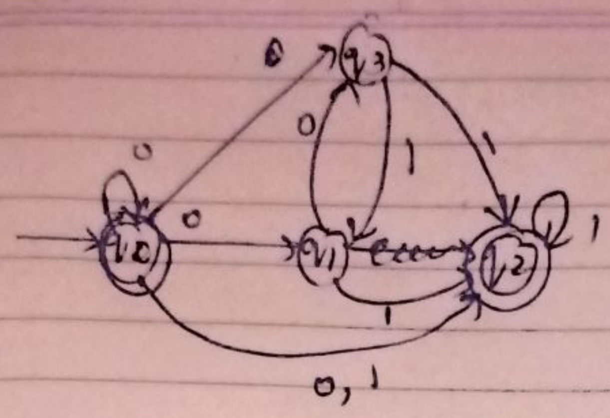


Que Construct NFA to given ϵ-NFA.

→ ϵ NFA.





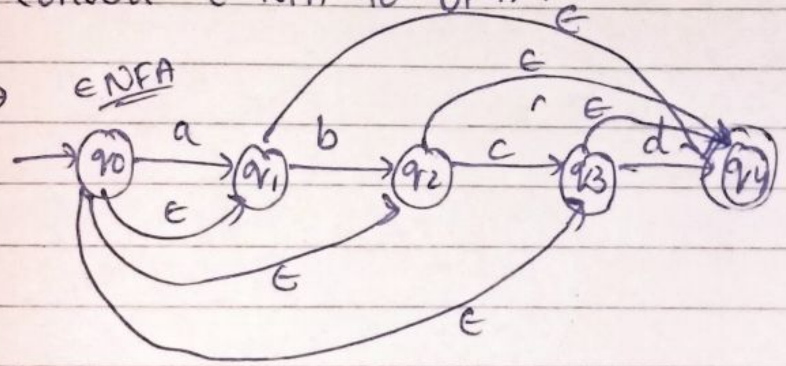


Que Construct the equivalent DFA for $L = \{a^n b^m c^p d^q \mid n \geq 0, m \geq 0, p \geq q, q \geq 1\}$

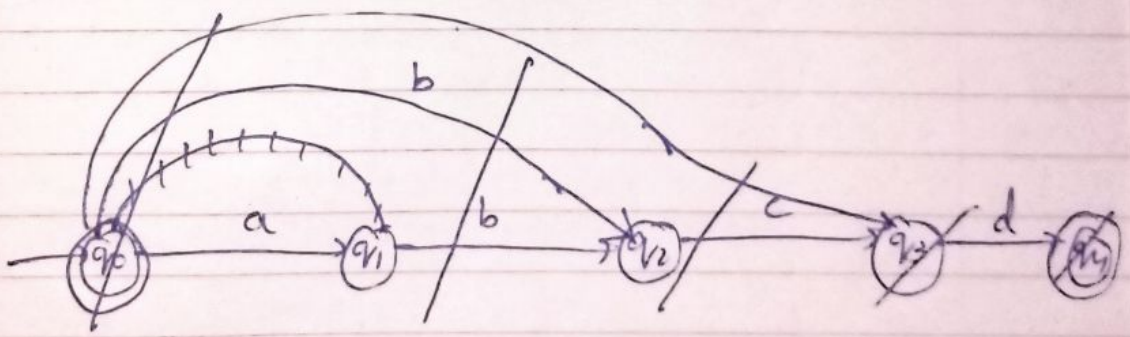
→ Already done.

Que Convert ε NFA to DFA :-

→ ε NFA



→



* $a w b \mid w \in (a,b)^* \rightarrow$ stard with 'a' and end with 'b'
* $n_a(w)/3 \rightarrow$ no of a divisible by 3 in string.