Chapter-7

Finite State Machine

$M_1 = \{Q, \Sigma, \delta, g, S, O\}$

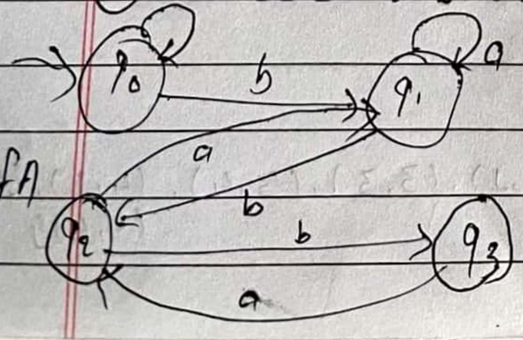
K/Q = Set of States
Σ = inputs
δ = transition function
g = Output function
I/S = Starting State
O = Output Symbol

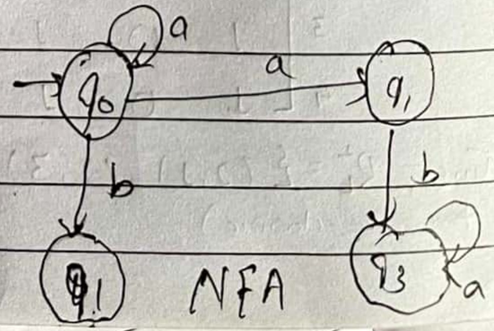
Finite State Automata
$F = \{Q, \Sigma, \delta, I, F\}$

Q = Set of states
Σ = inputs
δ = transition function
S/I = Initial State
A/F = Final State

Two Types of Automata (Finite State)
(I) Deterministic Finite State Automata (DFA)
(II) Non-Deterministic Finite State Automata (NFA)

DFA





NFA

Automata theory is the study of abstract computational devices. Abstract devices are models of real computations. It is the study of abstract machines and their properties providing a mathematical notation of a computer. Examples of such models are: finite automata.,

Finite Automata
These are used in text processing, compilers & hardware design

Context Free Grammars
These are used to define programming languages and in AI.

Turing Machines
These form a simple abstract model of a real computer such as your PC at home.

Basic Concepts of Automata Theory
=> Alphabets (Σ)
Alphabets is a finite non-empty set $\{a, b, c\}$ of symbols. The symbols can be letters such as A, B, C or digits {1, 2, 3 ...} or special characters like {\$, #, *, + . . .}

Date
Page

⇒ Strings:
String is a finite sequence of symbols taken from some alphabets.
Example: 0010 which is a sum of binary digits.

Length of String
The length of string w is denoted by as |w|, is the number of symbols in w. Example: w= computation
|w| = 11.

Empty String
It is a string with 0 occurrence of symbol. It denoted by ε(epsilon). So length of |ε| = 0.

Power of Alphabeth
The set of all strings of certain length k from alphabeth is the k^{th} power of that alphabeth.

If $\Sigma = \{0, 1\}$ then
$\Sigma^0 = \{\epsilon\}$
$\Sigma^1 = \{0, 1\}$
$\Sigma^2 = \{00, 01, 10, 11\}$
$\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

Kleen Closure

The set of all strings over the alphabeth Σ (sigma) is called kleen closure of sigma and denoted as Σ^*. If $\Sigma = \{a, b\}$ then

$\Sigma^* = \{\epsilon, a, b, ab, abb, aabb \ldots\}$

Positive Closure (Σ^+)

The set of all string of alphabeth sigma except epsilon is said to be positive closure of sigma and denoted as Σ^+.

Example:

$\Sigma^+ = \{a, b, ab, aab, aabb, aabb, \ldots\}$

Prefix & Suffix of String

$w = abcd$

Suffix = bcd

Prefix = abc

Substring

A string S is called substring of w if it is obtained by removing 0 or more leading or trailing symbols in w

Example:

$w = abbab$

$s = ba$

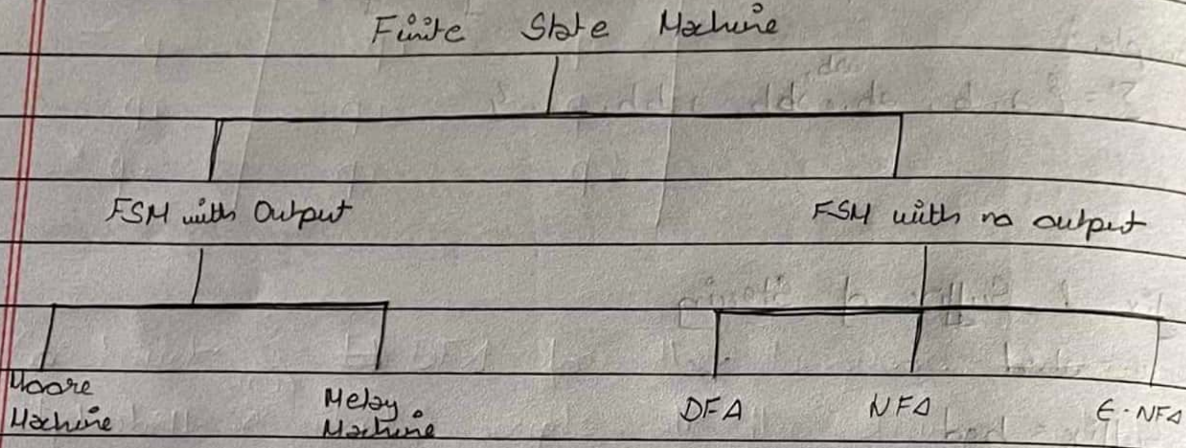
For proper substring

$w \neq s$.

Language

A language L over an alphabeth Σ is subset of all strings that can be formed out of Σ. Example: L = set of all strings over Σ = {0,1} having equal number of zeros(0) and ones(1).

$L = \{\epsilon, 01, 0011, 000111, \dots\}$



Finite Automata

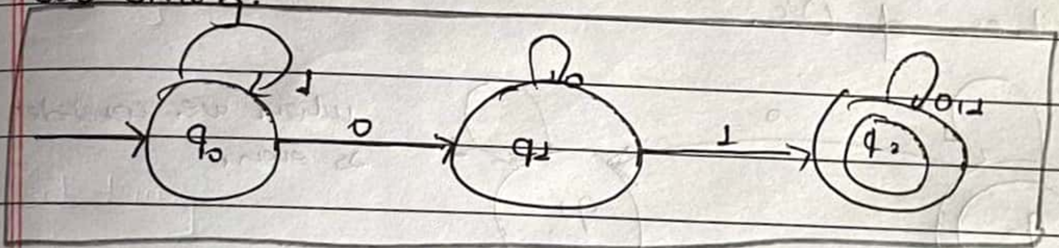
A finite automaton is a mathematical abstract machine that has a set of states and its control moves from state to state in response to the external inputs. The control may be deterministic or non-deterministic which distinguishes the class of automata as DFA or NFA.

Mathematically a finite automata F is represented by 5

tuples as $F = \{Q, \Sigma, \delta, S, F\}$
where,

Q is a finite set whose elements are called ~~set~~ states.
Σ is a finite set called the alphabeth whose elements are called symbols.
$\delta : Q \times \Sigma \to Q$ is a function called transition function. It takes state input symbol. It takes state and input symbols ~~as and~~ as arguments and returns a state as output.
S: is an element of Q which is called starting state
F: F is a subset of Q whose elements are called as ~~ex~~ accepting of final states.

Q.) Construct a DFA for language L which takes input over $\Sigma = \{0, 1\}$ which accepts those strings with 01 as sub-string.



where,

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1\}$

$S = \{q_0\}$

$I = \{q_2\}$

$\delta(q_0, 1) = q_0$

$\delta(q_0, 0) = q_1$

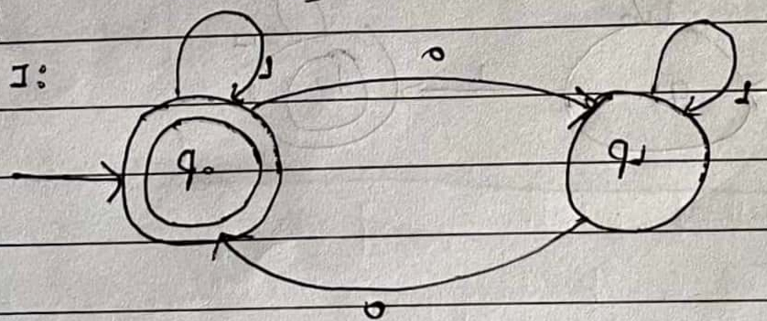
$\delta = (q_1, 0) = q_1$
$\delta = (q_1, 1) = q_2$
$\delta = (q_2, 0) = q_2$
$\delta = (q_2, 1) = q_2$
(Fig 1)

Transition ~~Function~~ Table

State	Inputs 0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$* q_2$	q_2	q_2

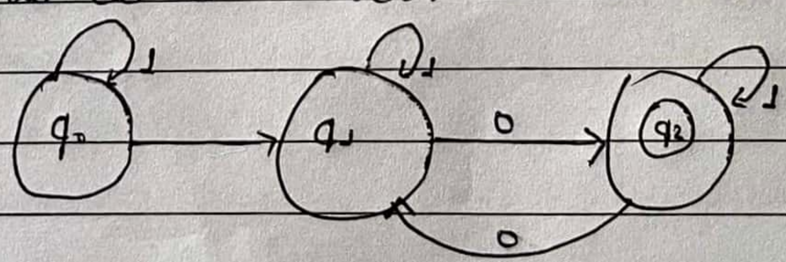
Q.) Construct a ~~tab~~ DFA for the language ~~over~~ L over the alphabet (0,1) which accepts all the strings which have even number of 0's.

Sol:
Case I:



when we consider z as even.

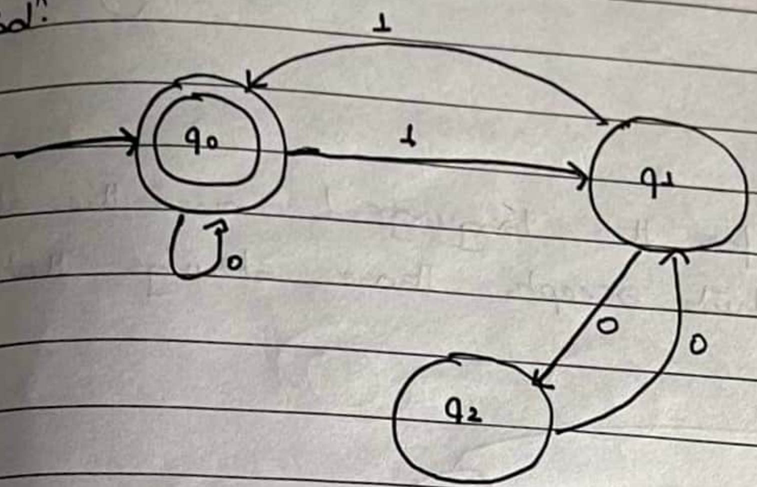
Case II: when we do not consider zero as even.



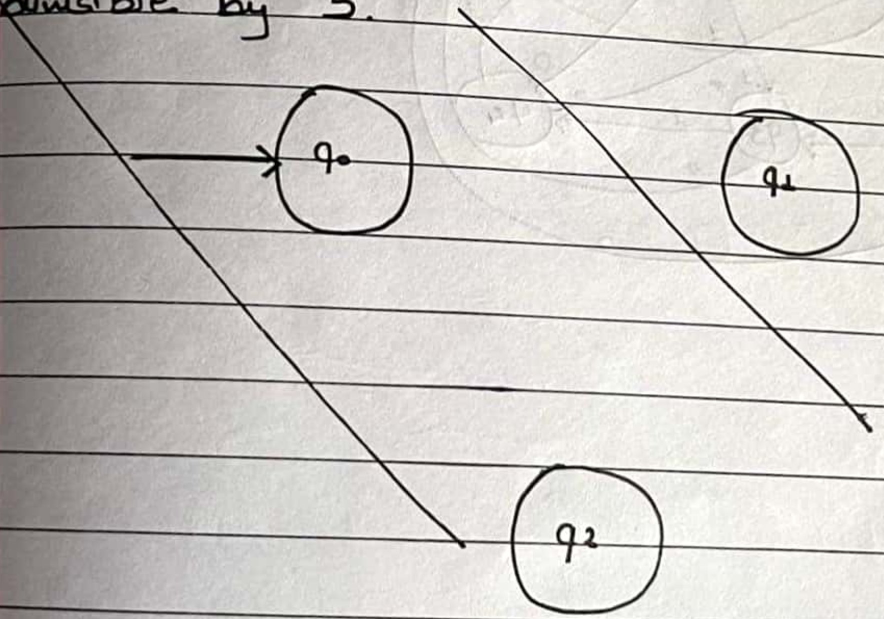
Q1) Construct a DFA for the language L over the string $\Sigma = \{0,1\}$ which accepts those string that are divisible by 3.

Q2) Construct a DFA for the languag L over the string $\Sigma = \{0,1\}$ which accepts those string all the odd 0's and odd 1's.

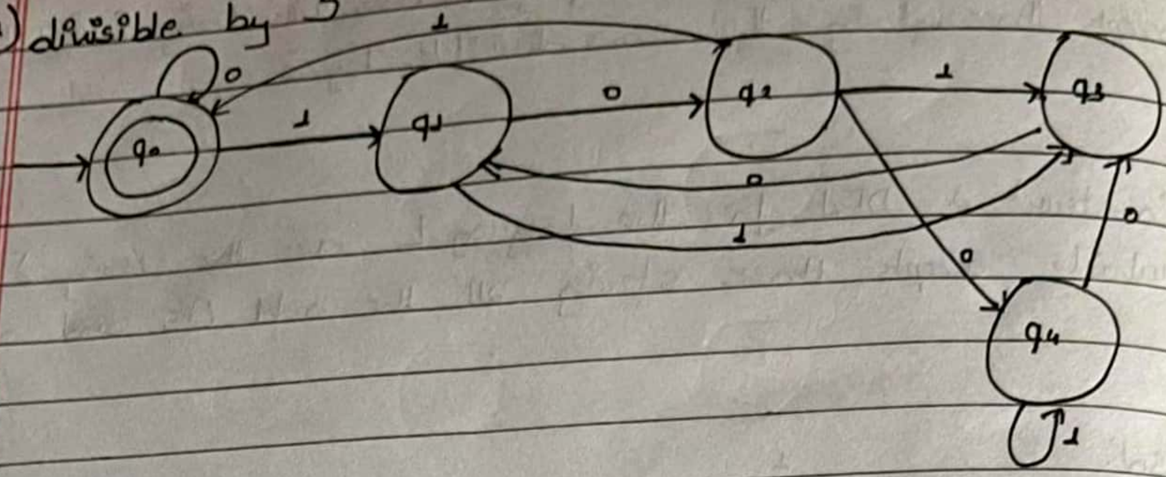
Q1) Sol^n:



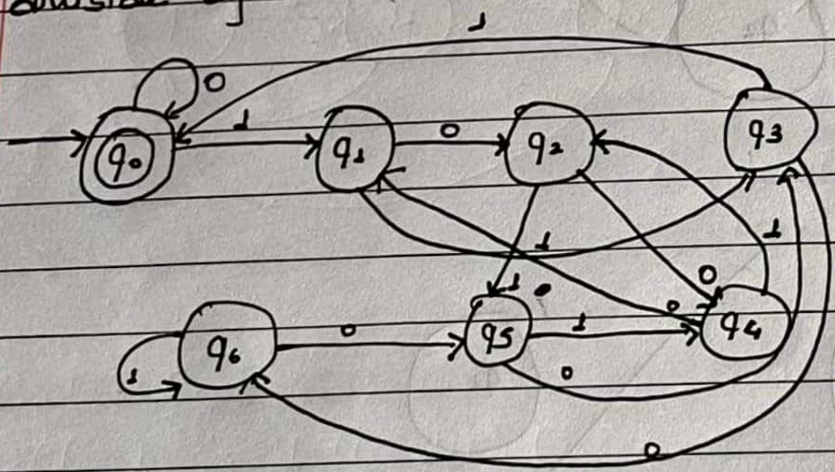
Q.) divisible by 5.



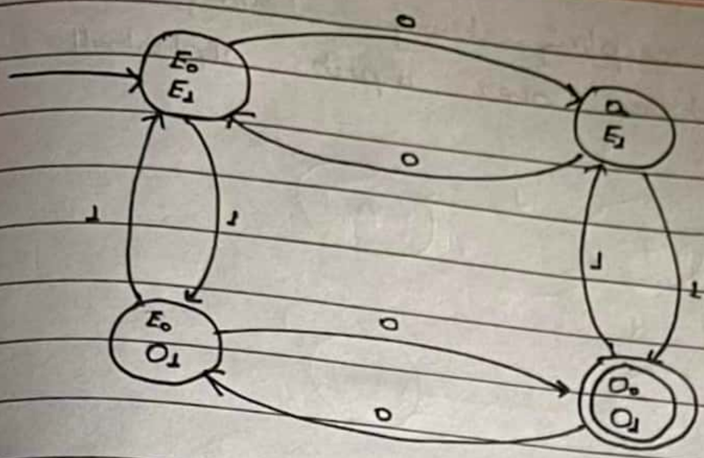
Q) divisible by 5



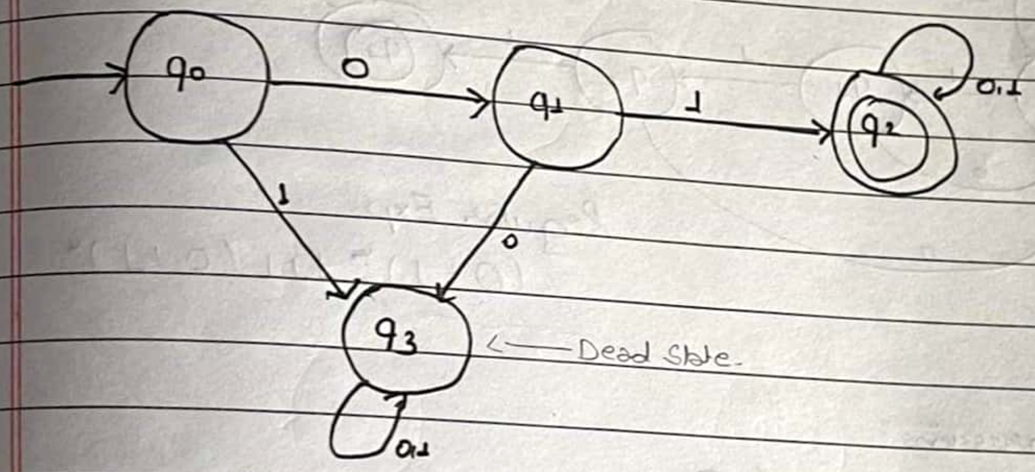
Q.) Construct a DFA for the language L over the alphabet string Σ={0,1} which accepts those strings that are divisible by 7.



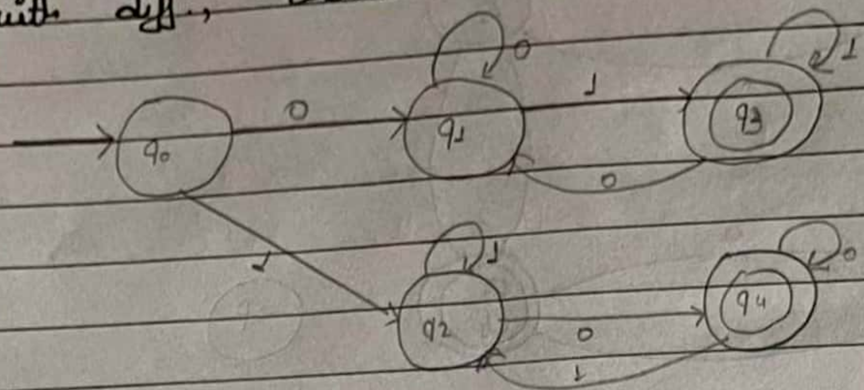
Q) Sol:



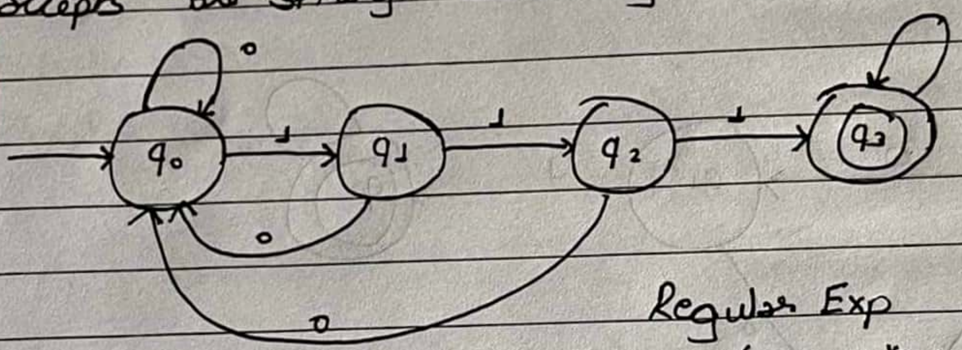
Q.) Draw a DFA for the language accepting strings starting with 01 over input $\Sigma = \{0, 1\}$



Q.) Draw a DFA for accepting string starting and ending with diff., characters over input alphabets $\Sigma = \{0, 1\}$



Construct a DFA for the language over $\Sigma \{0, 1\}$ that accepts the strings containing three consecutive 1s.



Regular Exp
$(0+1)^* 111 (0+1)^*$

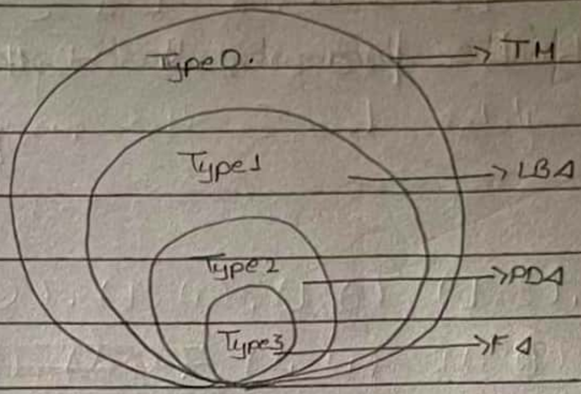
Regular Expressions:
$0(0+1)^* + 1(0+1)^* 0$

Kleen exp of $(11)^*$
$= \{\epsilon, 11, 1111, 111111, 11111111, \ldots\}$

$(0+1)^*$	$(01)^*$
$= \{\epsilon, 0, 1, 01, 10, 11, 00, 000 \ldots\}$	$= \{\epsilon, 01, 0101, 010101, \ldots\}$

Capital Letters ⇒ Non-Terminal
Small Letters ⇒ Terminal

CHOMSKY'S Hierarchy of Grammer



(Unrestricted Grammar)
(Context Sensitive Grammar)
(Context Free Grammar)
(Regular Grammar)

~~Type~~ **Type 3:**

$X \to a$

or

$X \to aY$

Type 1

$AB \to AbBC$

$A \to bCA$

$B \to b$

$\alpha AB \to \alpha\gamma B$

Type 2:

$A \to \gamma$

or

$A \to Xa$

or

$X \to a$

or

$X \to \epsilon$

Type 0

$\alpha \to \beta$

Eg:

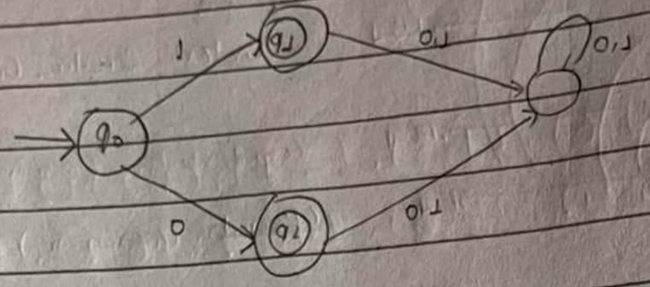
$S \to ACaB$

$BC \to aCB$

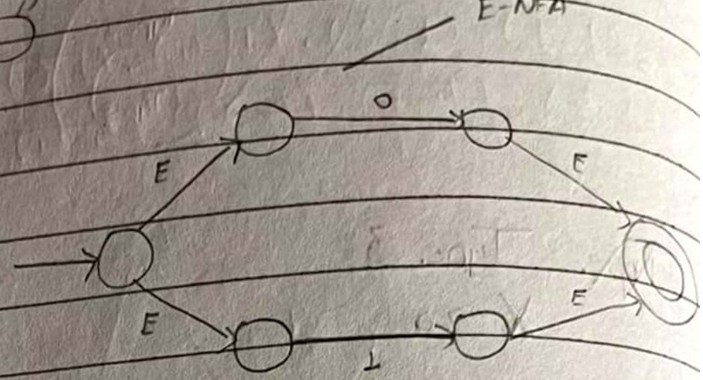
$aD \to Db$

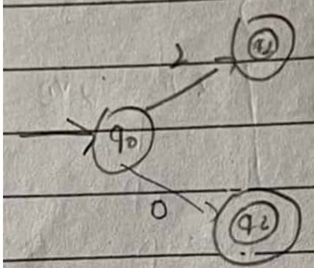
Regular Expressions:

R.E.: $(1+0)$: The finite Automata must accept either 1 or



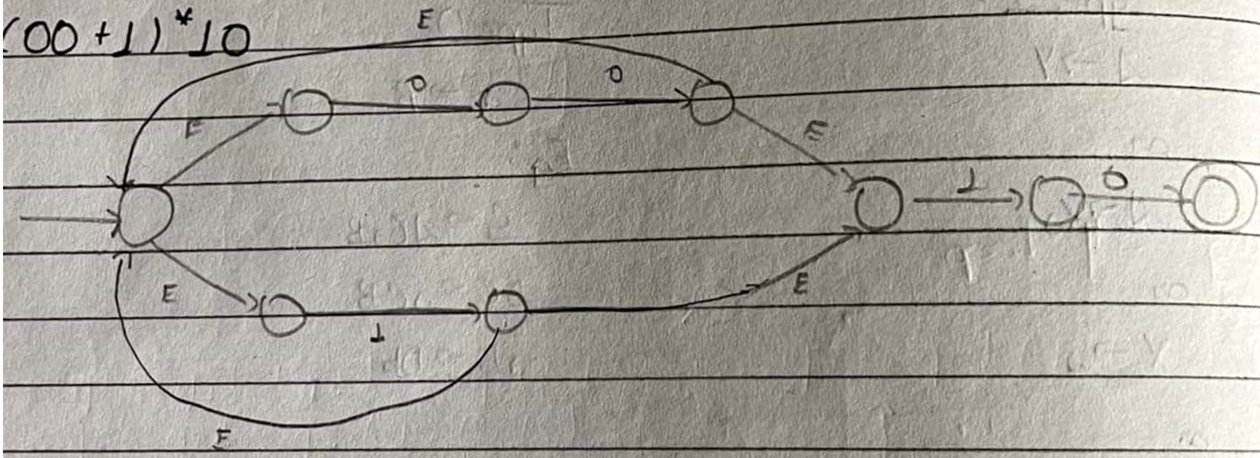
E-NFA





NFA

$(00+1)^*10$



$(00)^*$

Arden's Theorem :-

Let p and q be the regular expressions over the alphabet Σ, if p does not contain any empty string then $r = q + rp$ has a unique solution $r = qp^*$.

Here,

$r = q + rp$

or, $r = q + (q + rp)p$

$= q + qp + rpp$

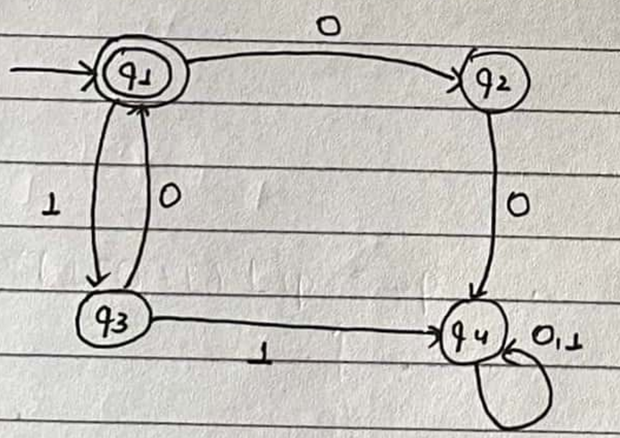
$= q + qp + (q + rp)pp$

$= q + qp + qpp + rpp$

$= q + qp + qpp + qppp + \ldots$

$= q\{\epsilon + p + pp + ppp + pppp + \ldots\}$

$= q \cdot p^*$



$r = q + rp \Rightarrow r = qp^*$

Let the equations are :

$q_1 = q_2 1 + q_3 0 + \epsilon$ — (i)

$q_2 = q_1 0$ — (ii)

$q_3 = q_1 1$ — (iii)

$q_4 = q_2 0 + q_3 1 + q_4 0 + q_4 1$ — (iv)

$q_1 = q_2 1 + q_3 0 + \epsilon$

$= q_1 01 + q_1 10 + \epsilon$

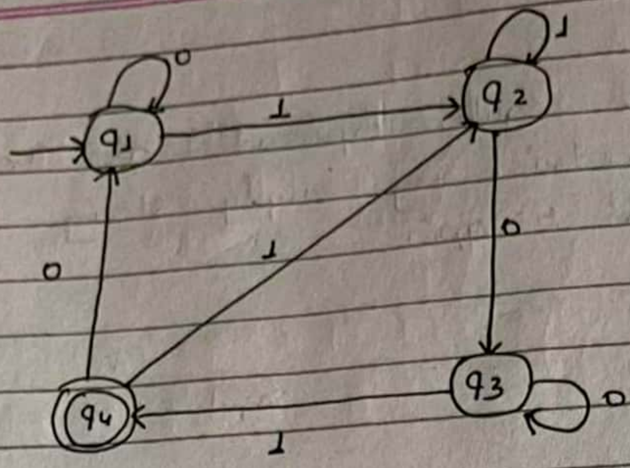
$= \epsilon + q_1 01 + q_1 10$

$q_1 = \epsilon + q_1 (01 + 10)$

Thus, by Arden's theorem we have,

$q_1 = \epsilon . (01 + 10)^*$

$q_1 = (01 + 10)^*$



Let the equations are:

$q_1 = q_1 0 + q_4 0 + \epsilon$

$q_2 = q_1 1 + q_2 1 + q_4 1$

$q_3 = q_2 0 + q_3 0 \Rightarrow q_3 = q_2 0 + q_3 0$

$q_4 = q_3 1 \qquad q_3 = q_2 0 0^*$

$q_4 = q_3 1$

$\boxed{\cancel{= (q_2 0 + q_3 0) 1}}$

$= q_2 0 0^*$

$= q_2 0^+$

$q_2 = q_1 1 + q_2 1 + q_2 0^+ 1 \quad , \quad q_4 = q_1 1 (1 + 0^+ 1)^*$

$q_2 = q_1 1 + q_2 (1 + 0^+ 1)$

$q_2 = q_1 1 (1 + 0^+ 1)^*$

$q_1 = q_1 0 +$

$= q_1 0 + q_1 1 (1 + 0^+ 1)^* 0 + \epsilon$

$q_1 = \epsilon + q_1 (0 + (1 + 0^+ 1)^* 0)$

$q_1 = (0 + (1 + 0^+ 1)^* 0)^*$

Sol^n:
Let a and b be two parallel lines. The relation "is parallel to" is an equivalence relation as it is reflexive, symmetric and transitive.
Let R be the relation "is parallel to."

Proof:-
Reflexive :- A line will always be parallel to itself as it is coincide by itself. Thus for every line a, (a,a) ∈ R i.e. R is reflexive.

Symmetric :- If a and b are two parallel lines then a is parallel to b and conversely b is also parallel to a i.e., both (a,b) and (b,a) ∈ R. So, R is symmetric.

Transitive :- Let a, b and c be three parallel lines. If a is parallel to b and b is parallel to c then in the same plane a would be parallel to c i.e., if (a,b) ∈ R and (b,c) ∈ R, (a,c) ∈
Thus, R is transitive.

Hence, the provided relation R which is "is parallel to" is an equivalence relation.

Poset:

Partial Order Relation:

A relation on set A, denoted by R is said to be partial order if it is reflexive, anti-symmetric and transitive.

A relation R on set A is called partial order if it is ref

Example:

Let Z^+ be the set of positive integers. The usual relation $\leq$ is partial order on Z^+. Thus $(Z^+, \leq)$.

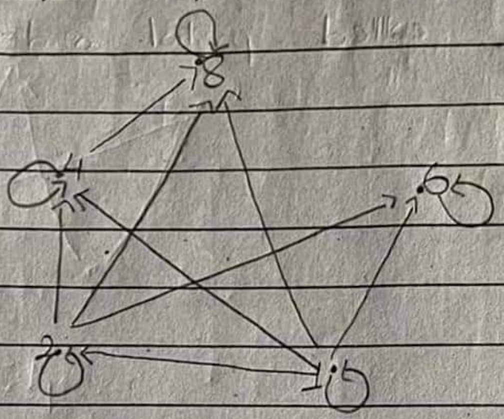
Hasse Diagram

Let us consider $S = \{1, 2, 4, 6, 8\}$ and relation R is defined on set S such that $R = \{(a,b) \mid a \text{ divides } b\}$. Construct Hasse Diagram of the given set S and relation R are

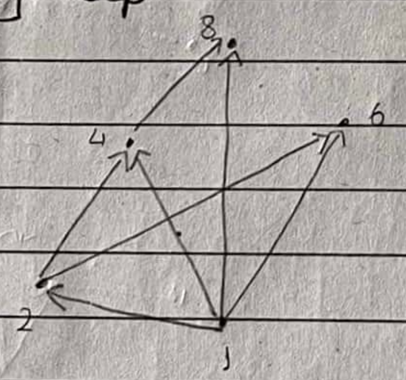
sol:

$R = \{(1,1), (1,2), (1,4), (1,6), (1,8), (2,1), (2,2), (2,4), (2,6), (2,8), \cancel{(4,1)}, \cancel{(4,2)}, (4,4), (4,8), \cancel{(4,8)}, (6,6), (8,8)\}$

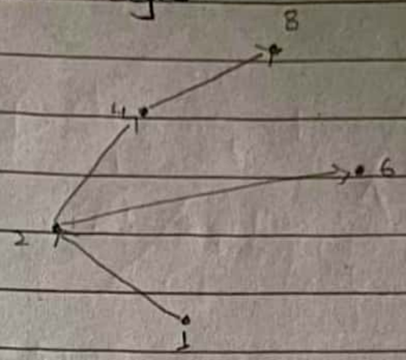
i)



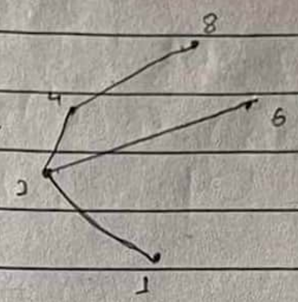
Removing Self-Loop



iii) Removing Transitive Edges.



iv) Removing Direction



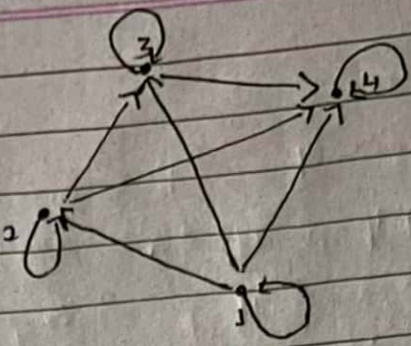
Simplest Possible Diagram of a Graph which has sufficient information is called Hasse Diagram.

7.) $S = \{1, 2, 3, 4\}$, $R = \leq$
$(S, R) =$ poset P
Construct Hasse Diagram.
sol^n:

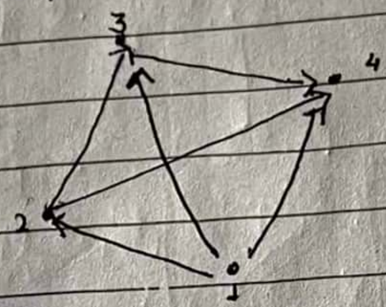
$R = \{(1,1), (1,2), (1,3), (1,4), (2,3), (2,4), (2,2), (3,3), (3,4), (4,4)\}$

$\therefore$ R is a partial order because it is reflexive, transitive & anti-symmetric.

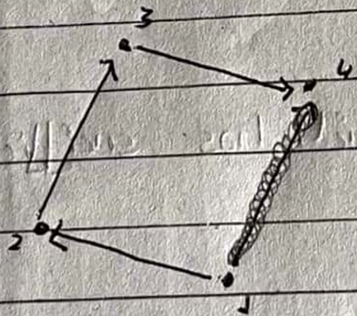
i)



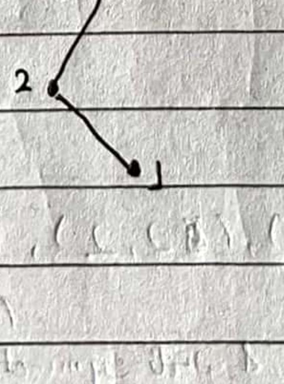
ii) Removing Self-Loop



iii) Removing Transitive Edges.



iv) Removing Direction



Minimal and Maximal Element of Poset.

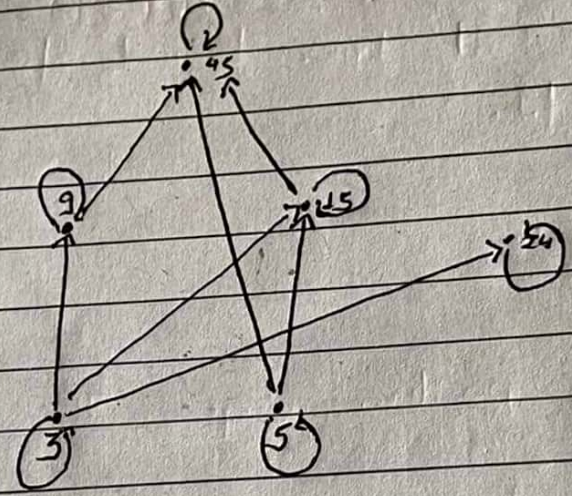
What are the maximal and minimal elements for the poset ({3, 5, 9, 15, 24, 45}, /)? (a divides B)

Sol:

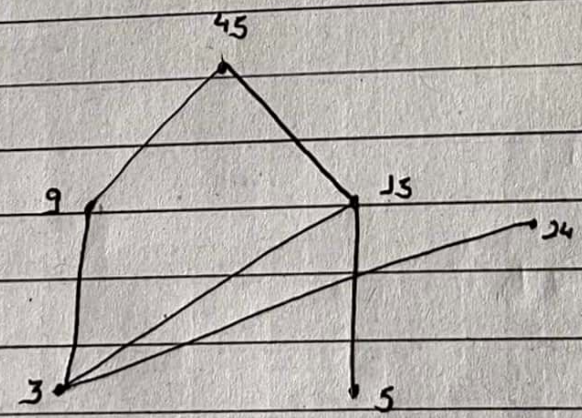
R = {(3, 9), (3, 24), (5, 15), (5, 45), (3, 3), (15, 5), (9, 9), (24, 24), (45, 45), (9, 45), (3, 15), (15, 45)}

Hasse Diagram

i)

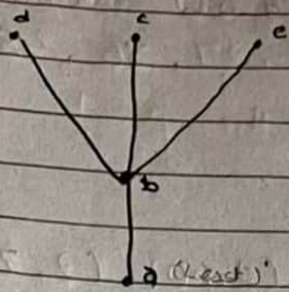


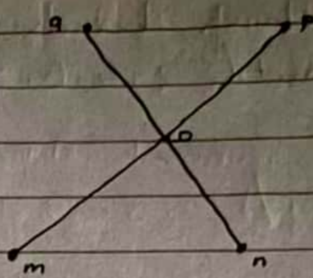
ii)

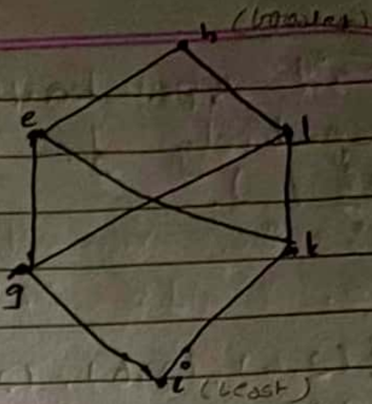


Maximal Element = 45, 24

Minimal Element = ~~3,24~~ 3, 5





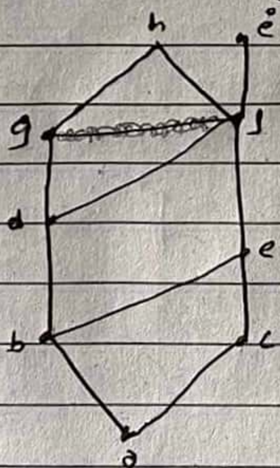


Max, = c, d, e
Min, = a

Max., = q, p
Min., = m, n

Max., = h
Min., = i

Q.) Find the upper bounds and lower bounds for {a, b, c}, {i, h} and {a, c, d, j}.



Lower bound of {a, b, c} = a
Upper bound of {a, b, c} = e, j, i, h

Lower bound of {i, h} = j, e, c, a, d, b
Upper bound of {i, h} = ∅

GLB & LUB {b, d, g}
g & h are the upperbound of {b, d, g}
g is the least upper bound
b & a are lower bound of {b, d, g} & b is the greatest ~~upper~~ lower bound.

Q.) Find upper bounds and lower bounds for $\{2, 9\}$ and $\{60, 72\}$ respectively for the poset $(\{2, 4, 6, 9, 12, 18, 27, 36, 48, 60, 72\}, |)$.

Sol:

$R = \{(2,4), (2,6), (2,12), (2,18), (2,36), (2,48), (2,60), (2,72),$
$(4,12), (4,36), (4,48), (4,72), (6,12), (6,18), (4,60), (6,36),$
$(6,48), (6,60), (6,72), (9,18), (9,27), (9,36), (9,72), (12,36),$
$(12,48), (12,60), (12,72), (18,36), (18,72), (36,72), (2,2), (4,4),$
$(6,6), (9,9), (12,12), (18,18), (27,27), (76,36), (48,48), (60,60), (72,72)$