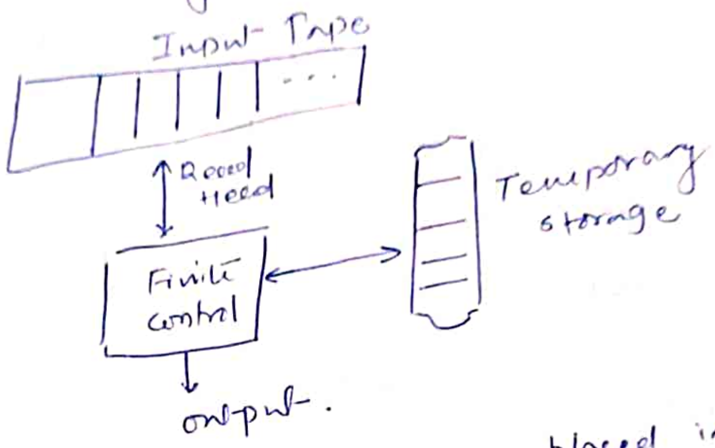
Finite Automata

- The main issue is how to represent the languages with finite specifications.
- An automation is an abstract computing device (or machine) which can be defined mathematically.
- A finite Automata or finite state automata is a language accepting device.



(i) it has an input tape. Input is placed in this tape.

(ii) it has a moveable read-head to read symbols from input string.

(iii) it can produce output of some form.

(iv) it may have temporary storage.

(v) it has a finite control unit, which can be in any one of a finite number of interval states at any point. It can change state in some defined manner determined by a transition function.

- Finite automata → with output (out) → DFA, NFA, ε-NFA
- Finite automata → with~~out~~ output → Moore Machine, Melay Machine.

① DFA :- Deterministic Finite Automata

→ A DFA is a type of finite automata in which the transitions are deterministic, in the sense that there will be exactly one transition from a state on an input symbol.

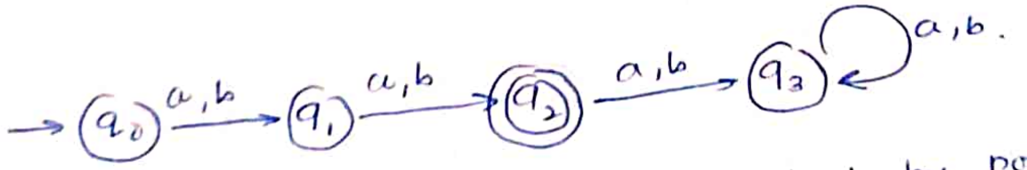
→ Formally, a DFA is a quintuple $M = (Q, \Sigma, \delta, q_0, F)$, where -

(i) Q - finite set of states.
(ii) Σ → is a finite set called input alphabet
(iii) $q_0 \in Q$, called initial/start state.
(iv) $F \subseteq Q$, called set of final/accept states
(v) $\delta : Q \times \Sigma \to Q$, called the transition function.

Eg:- Construct a DFA that accepts all string over $\Sigma = \{a, b\}$ of length two.

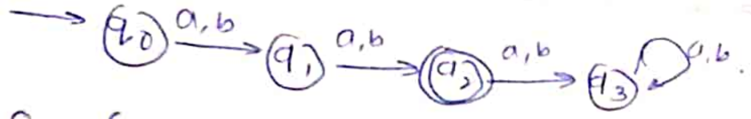
if $\Sigma = \{a, b\}$, then $L = \{aa, ab, ba, bb\}$

→ State transition diagram.



* Every state in Q is represented by node.
* If $\delta(q_i, a) = q_{i+1}$, then there is an arc from q_i to q_{i+1} labled with a.
* If there are multiple arcs from q_i to q_{i+1} labled $a_1, a_2, a_3, \ldots$, then we simply put only one arc labbled $a_1, a_2, a_3, \ldots$
* There is an arrow with no source into the initial state q_0.
* Final states are indicated by double circle.

$\therefore$



(i) $Q = \{q_0, q_1, q_2, q_3\}$

(ii) $\Sigma = \{a, b\}$

(iii) $F = \{q_2\} \subseteq Q$

(iv) $q_0 \in Q$

(v) $\delta : Q \times \Sigma \to Q$ $\quad\therefore$ q_3 is the dead state

* Transition Table :-

δ	a	b
$\to q_0$	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_3	q_3

* Extended Transition Function :-

→ The transition function $\delta : Q \times \Sigma \to Q$ assigns a state for each state and an input symbol.

→ The δ can be extended for all strings in Σ^*, as they

→ extended transition function $\hat{\delta} : Q \times \Sigma^* \to Q$ is defined recursively as follows:

(i) $\hat{\delta}(q, \epsilon) = q$

(ii) $\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$.

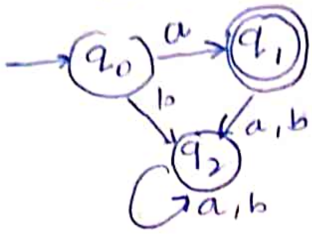
* Language of DFA :-

• The set of all strings accepted by the DFA M is said to be the language of the DFA i.e., denoted by $L(M)$.

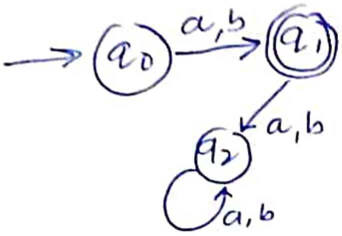
* DFA Designing :-

(i) $L = \{\epsilon\}$ $\Rightarrow$

(ii) $L = \{a\}$ $\Rightarrow$

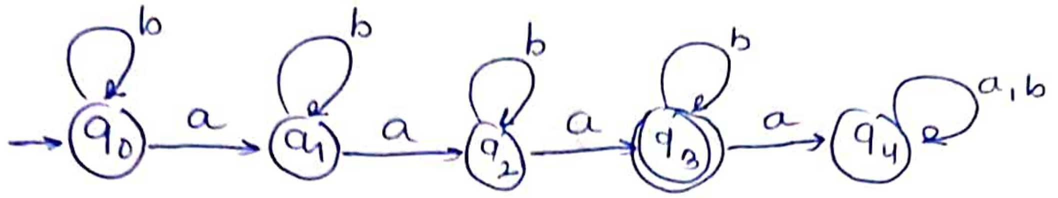


(iii) $L = \{a, b\}$ $\Rightarrow$

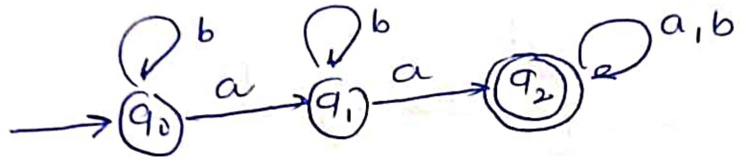


(iv)

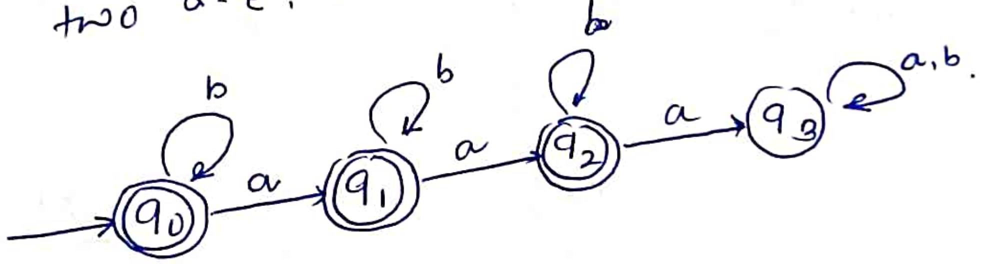
Problem :- $L = \{$ set of all strings contain exactly 3 a's $\}$. $\Sigma = \{a, b\}$.



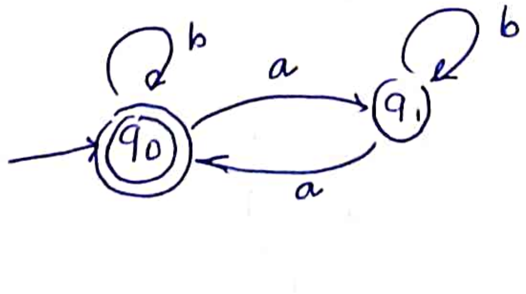
Problem :- $L = \{$ set of all strings contain at least two a's $\}$, $\Sigma = \{a, b\}$.



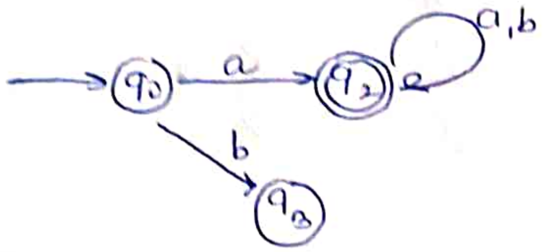
Problem :- $L = \{$ set of all strings contain at most two a's $\}$, $\Sigma = \{a, b\}$



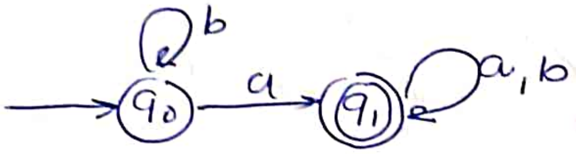
Problem :- $L = \{$ set of all strings contain even no of a's $\}$. $\Sigma = \{a, b\}$.



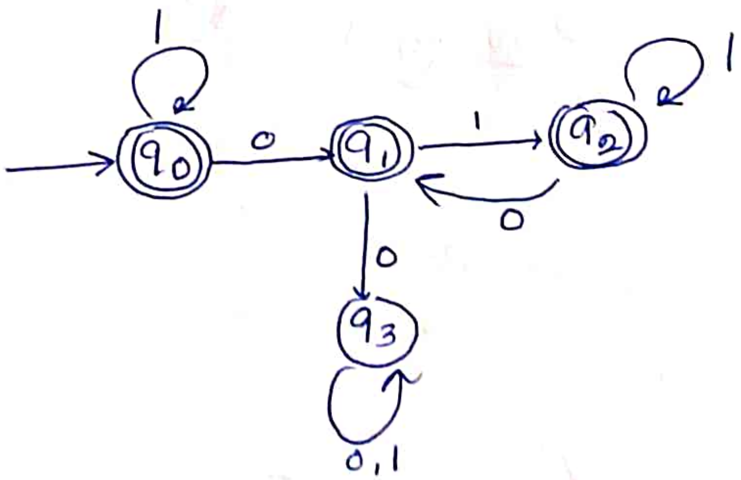
Problem :- $L = \{$set of all strings starting with a's$\}$
$\Sigma = \{a, b\}$.



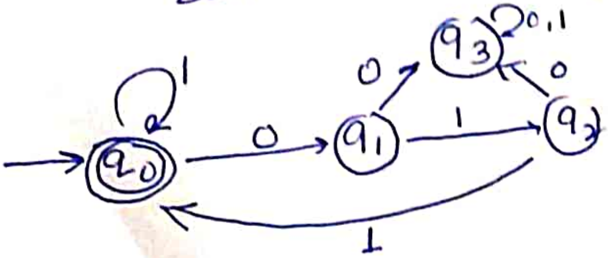
Problem :- $L = \{$set of all strings contain at least one a's$\}$, $\Sigma = \{a, b\}$.



Problem :- $L = \{$set of all string does not contain any consecutive 0's$\}$, $\Sigma = \{0, 1\}$



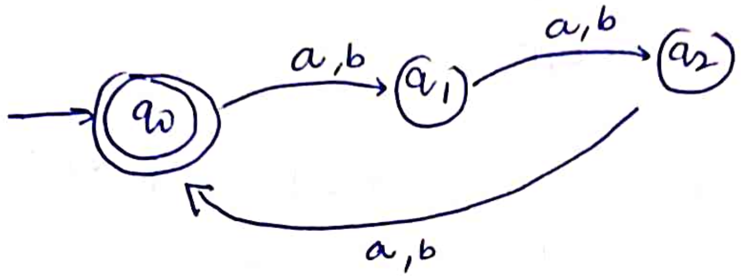
Problem :- $L = \{$Set of all strings in which every zero is followed by at least 2 1's$\}$. $\Sigma = \{0, 1\}$



Problem :- Construct a DFA that accepts all strings over $\{a,b\}^*$ of length divisible by three.

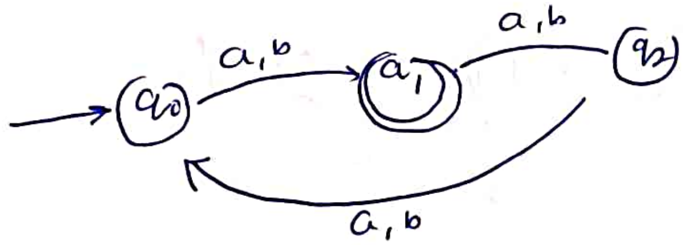
$\therefore \; L = \{ w \in \{a,b\}^* \mid |w| \bmod 3 = 0 \}$

$L = \{ \epsilon, aaa, aab, \ldots, aaaaa, aaaaab, \ldots \}$



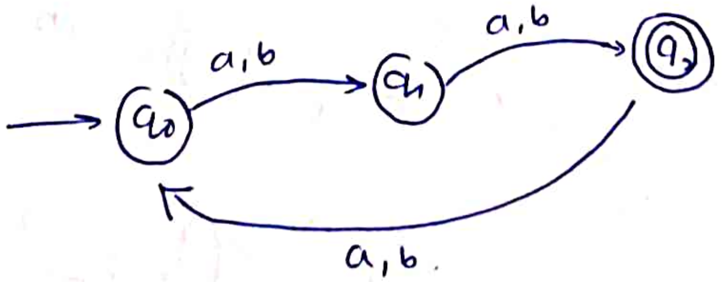
Problem :- $L = \{ w \in \{a,b\}^* \mid |w| \bmod 3 = 1 \}$

$L = \{ a, b, aaaa, bbbb, \ldots \}$

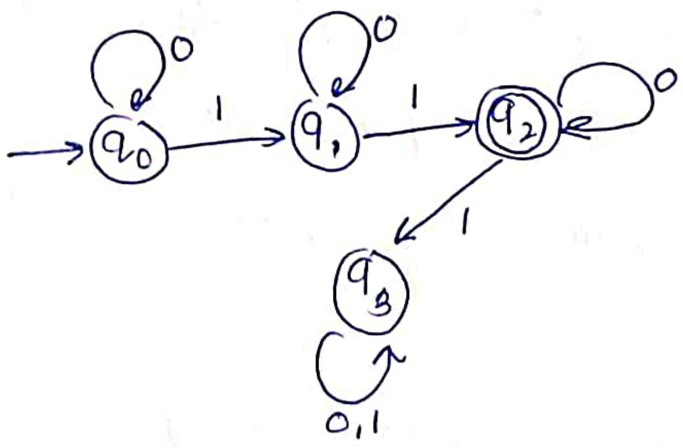


Problem :- $L = \{ w \in \{a,b\}^* \mid |w| \bmod 3 = 2 \}$

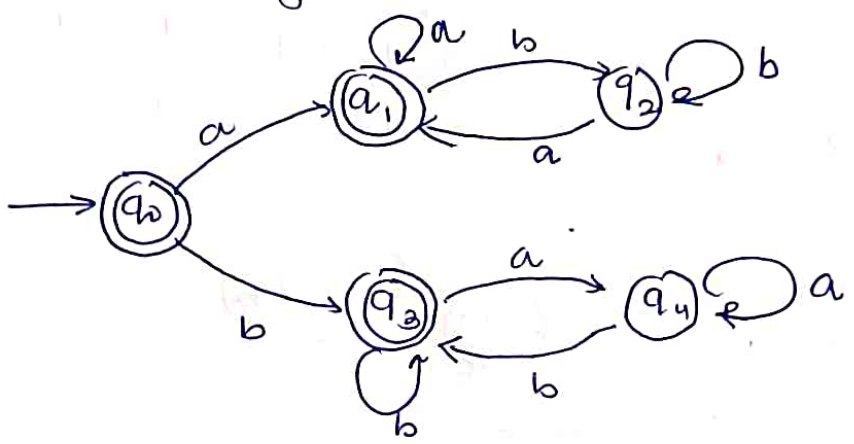
$L = \{ aa, ab, bb, ba, aaaaa, aaaab, \ldots \}$



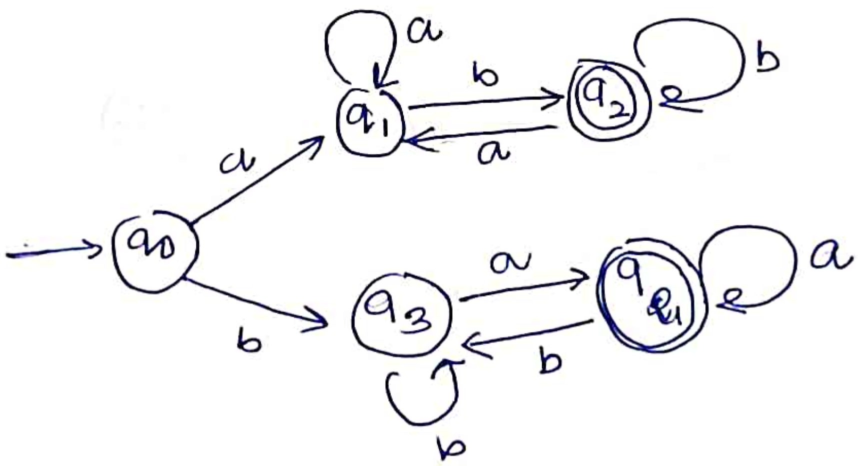
Problem :- L contains exactly 2 1's. $\Sigma = \{0,$



Problem :- L is set of string start and ends with same symbols. $\Sigma = \{a,b\}$.

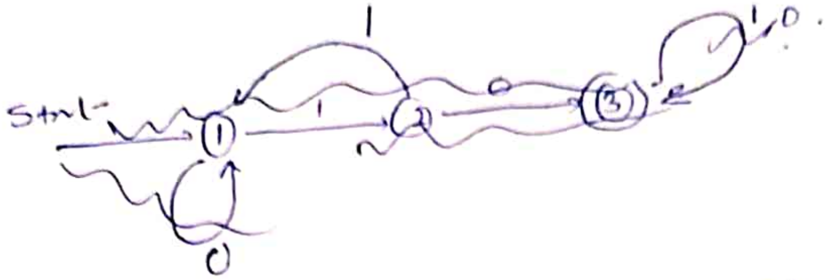


Problem :- L is set of all strings starting & Ending with different symbols.

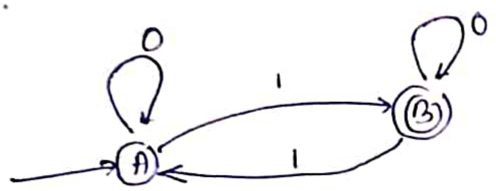


∴ To solve →

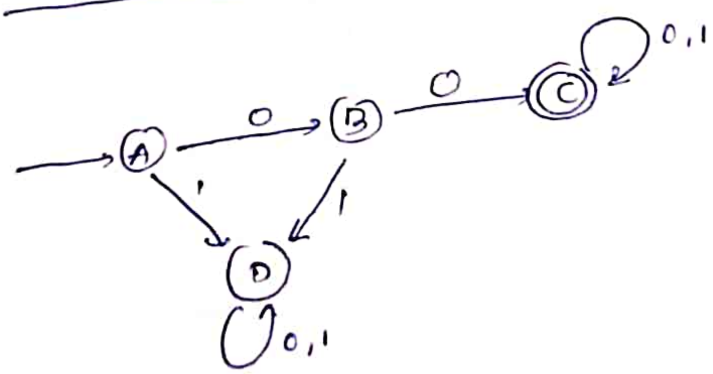
∴ Design a DFA which accept a string which have a sub string 10. $\Sigma = \{0,1\}$.



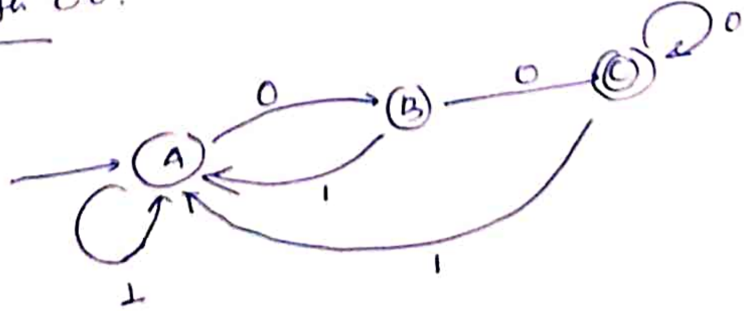
(1) Accepts all string over $\{0,1\}$ with odd number of 1's.



(2) Start with 00:

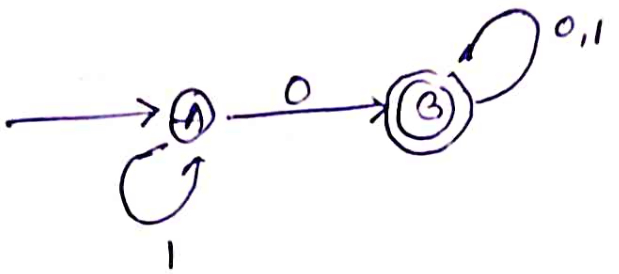


(3) End with 00:

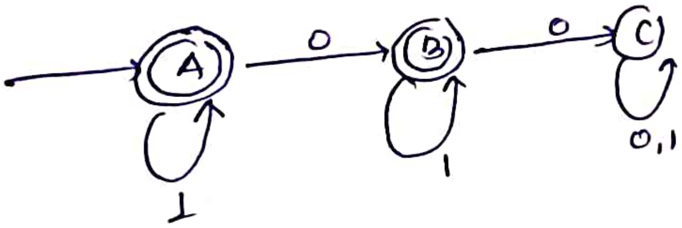


1. All binary string with at least one 0.
2. All binary string with at most one 0,
3. All binary strings ending and starting with 0's,

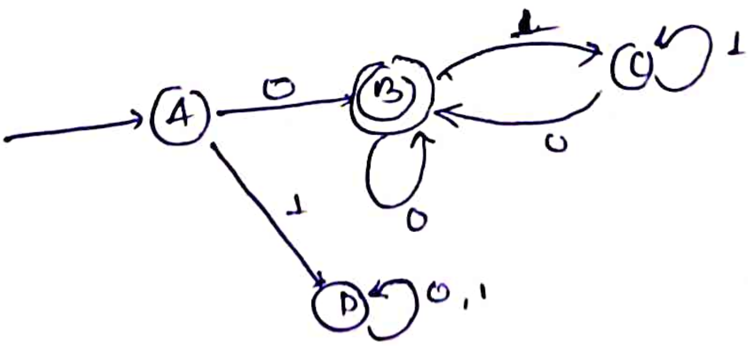
1)



2)

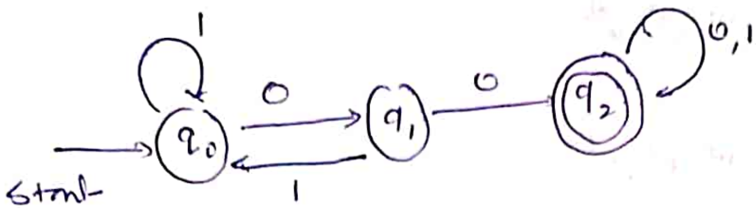


3)

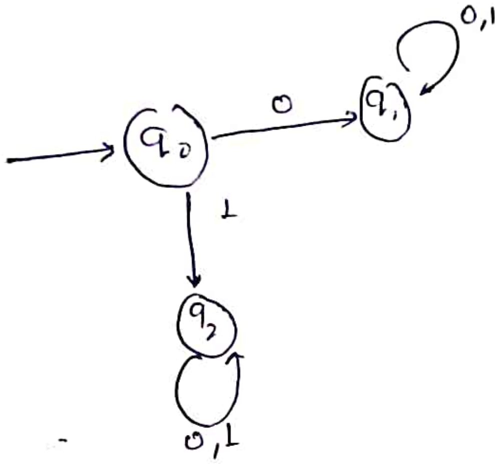


Lab - 8 CP (Section CSIT-C) 20/04/2024.

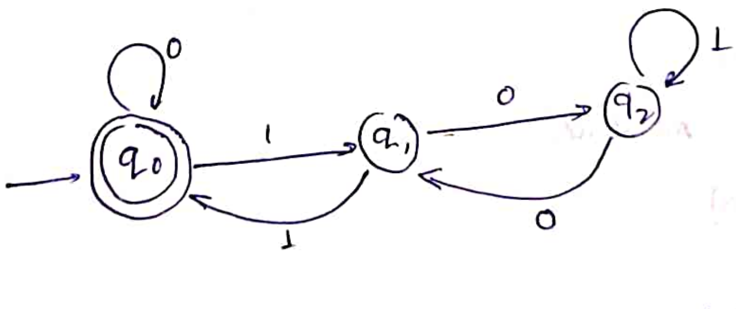
(1) $L = \{w \in \{0,1\}^* \mid w$ contains 00 as substring.



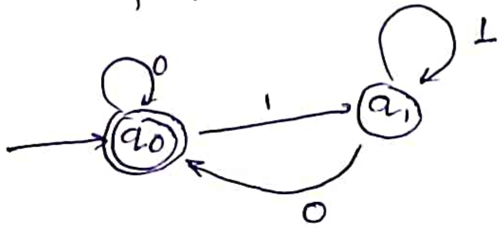
(2) $L = \{w \in \{0,1\}^* \mid$ the starting char in w is 0$\}$.



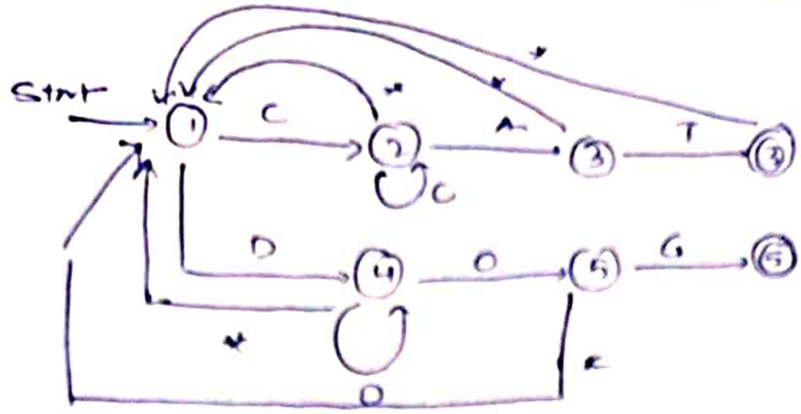
(3) $L = \{w \in \{0,1\}^* \mid w$ is a string which is multiple of 3$\}$



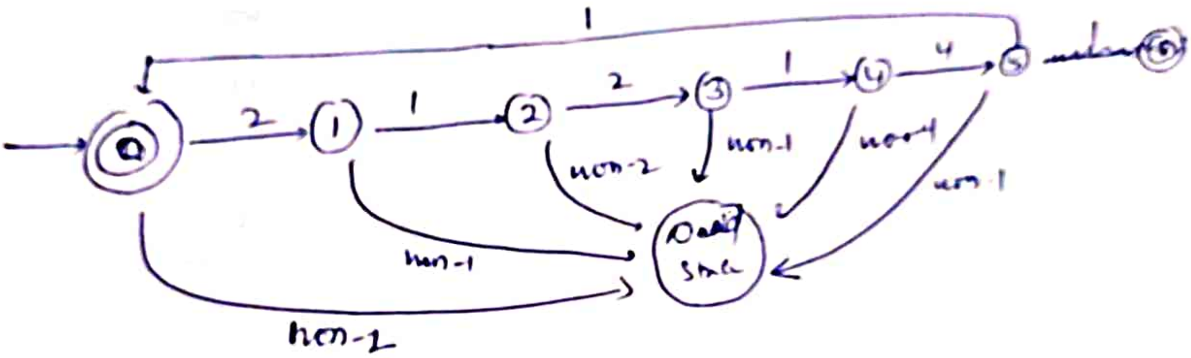
(4) $L = \{w \in \{0,1\}^* \mid w$ is a string which is multiple of 2$\}$



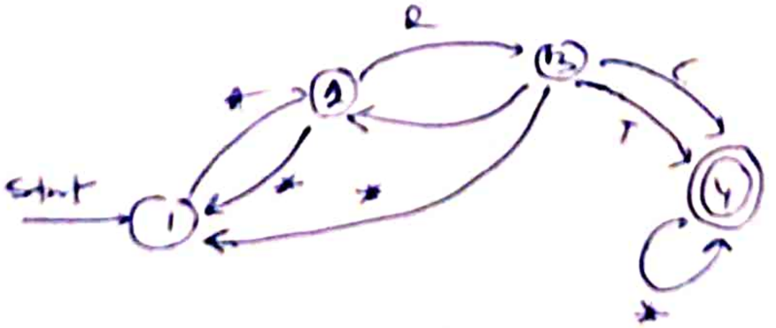
(5) Design a FSA to accept CAT and DOG alone.



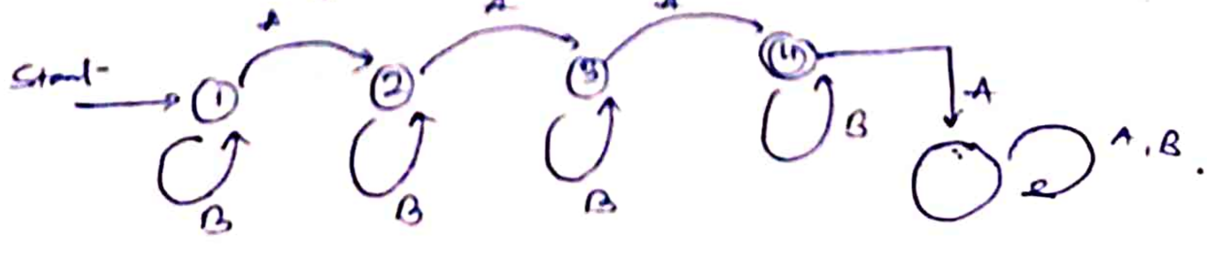
(6) Accept strings consisting of only zero or more repetitions of 212141.



(7) Accepts strings containing ART or ARC anywhere.

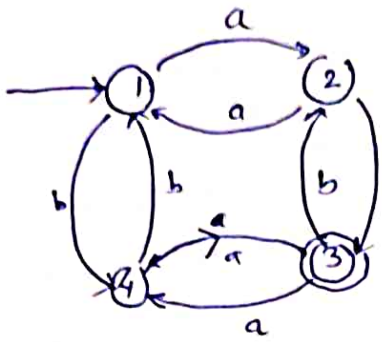


(8) Accepts string which have exactly 3 A's. $\Sigma = \{A, B\}$

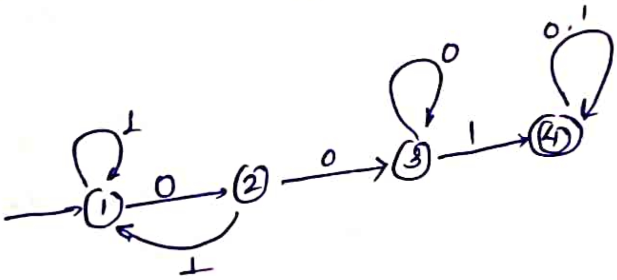


:- Lab Questions :-

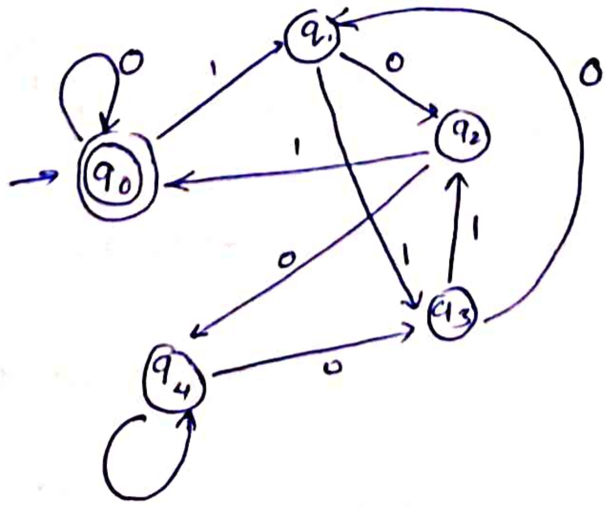
① Design a finite automata that will accept any string of odd number of a's and b's.



② Design a finite automata that will accept string having "001" as substring.



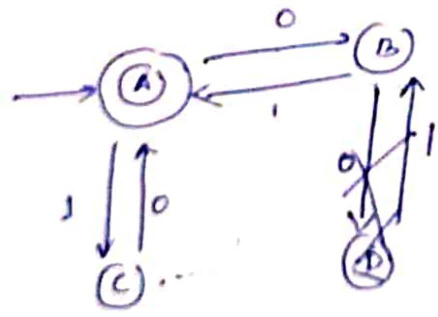
③ Design a finite automata that recognize binary number that are multiple of 5.



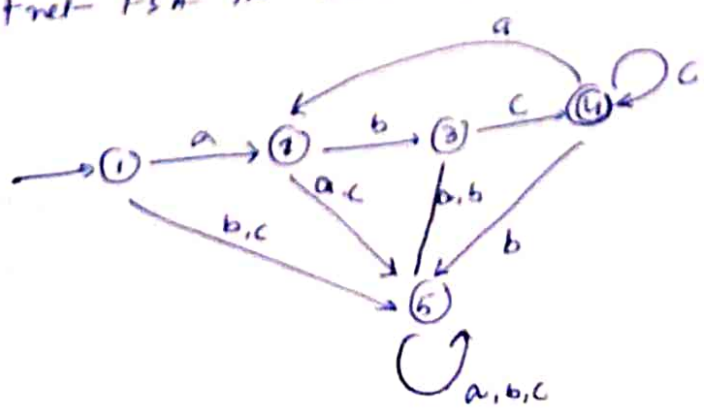
(4) Design a FSA that accept any string without an odd number of consecutive 0's after an odd number of consecutive 1's.

(5) Construct a finite automata accepting the set of all strings of zeros and ones, with at most one pair of consecutive zeros and at most one pair of consecutive ones.

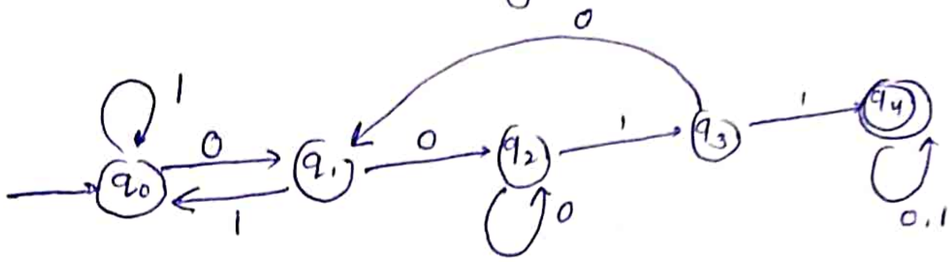
(6) Construct a FSA that will accept all the string containing equal number of zeros and ones, and no prefix of the string should contain two more zeros than ones, or two more ones than zeros.



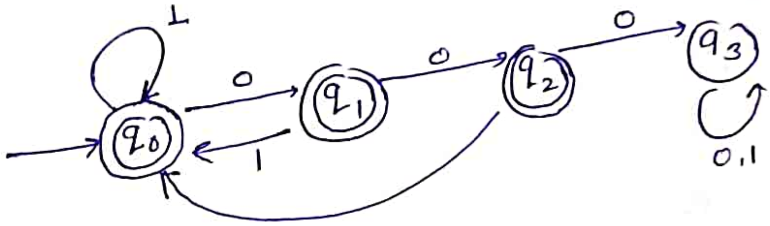
(7) Construct FSA for $(abc^+)^+$



(8) Construct DFA to accept a string containing two consecutive zeroes followed by two consecutive ones.



(9) Construct DFA that should not contain three consecutive zeroes.



(10) Construct DFA to accept all strings $(0+1)^*$ with an equal number of 0's and 1's such that each prefix has at most one ~~zoo~~ more zero than ones and at most one more one than zeroes.