Theory of computation :—

Symbol :- a, b, 0, 1, 手, 女, 木, 웃

↓ derives

"Σ" Alphbet — {a, b}, {a, b --- z}
{0, 1}, {0, 1 - - - 9}

↓ makes

a String (collection of Alphabet) :-

a, ab, aaa, abc, abb - - - -

Q) How many strings we can create using a alphabet set?

Language :- collection of valid words (strings)

for eg:-
$\Sigma = \{a, b\}$

a language L_1 = set of all string of length of "2".

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

$L2 =$ Length "3" over Σ

$= \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$ finite

$L3 =$ set of all string where every string starts with "a".

$= \{a, aa, ab, aab, aba, abb \ldots\}$

infinite

Power of Σ :-

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

$\Sigma^1 =$ set of all string over 'Σ' of length '1'

$= \{a, b\}$

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

$\Sigma^n = \{$ all 'n' length strings over $\{a, b\}\}$

$\Sigma^0 =$ set of all string with length 0

= know as 'ε'

& $|\varepsilon| = 0$

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots\cdots$$
$$= \{\varepsilon\} \cup \{a,b\} \cup \{ab, aa, ba, bb\} \cup \cdots\cdots$$

Σ^* = set of all string over $\{a, b\}$
(infinite)

Take the Example of "C" lang :-

Char set : -
$\{a, b \cdots z, A, B \cdots Z, 0, 1 \cdots 9,$
$+, *, @ \cdots\}$

```
void main () {
        int a, b;
          !
}
```

Program which is considerd as string in ToC.

"C" prog. lang is set of all valid prog.

Q How many prog. possible in "c" lag?
Infinte

$= \{P_1, P_2, P_3 \cdots \cdots\}$

if we take any prog "P_n" it must belongs to the valid set.

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots\ldots$
$= \{\varepsilon\} \cup \{a,b\} \cup \{ab, aa, ba, bb\} \cup \ldots\ldots$

Σ^* = set of all string over $\{a, b\}$
(infinite)

Take the Example of "C" lang :-

Char set :-
{a, b - - - z, A, B - - - Z, 0, 1 - - - 9,
+, *, @ - - - }

```
void main () {
        int a, b;
           !
}
```

Program which is considered as string in ToC.

"C" prog. lang is set of all valid prg.

Q. How many prog. possible in "c" lag?
Infinte

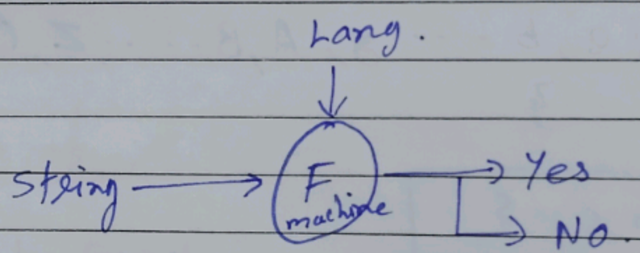
= {P_1, P_2, P_3 . . . . - - }

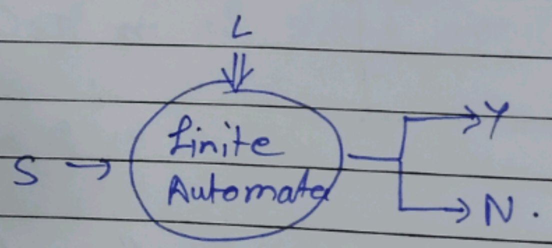
if we take any prog "P_n" it must belongs to the valid set.

The infinite no. of prog. we can't store in memory so "Pr" should be validted using some generic tool.

if Lang is finite the strings can be stored. But if it is infinite then it is not possible.

so machine should be like this.





Can't
would be
tool.

can be
e then

this.

machine (The finite state machine)
or
finite Auto mata

○ ⇒ state

→ ⇒ transition

a, b, 0, 1 ⇒ I/P symbols.

◎ ⇒ final state

→○ ⇒ initial state.

↻ → self loop transition.

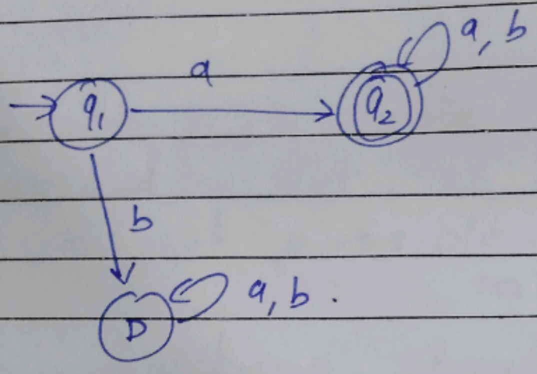
Q) set of all the string which starts with 'a'

Q) Represent "abaa" over $\Sigma = \{a, b\}$.

$$\rightarrow q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_3 \xrightarrow{a} q_4 \xrightarrow{a} q_5$$

DATE: __/__/__

Q) L = set of all string which starts with 'a' over
$\Sigma = \{a, b\}$.



Finite Automata :- (Types)

- FA with o/p
  - moore m/c
  - mealy m/c
- FA without o/p
  - DFA (determinsic Finete Auto.)
  - NFA (Non-determinsitc —)
  - ε-NFA (ε — NFA)

DFA :-

Tuples :-

$$(Q, \Sigma, \delta, q_0, F)$$

set of all states

set of I/P symbols

Transition funct^{n}

initial state

set of final state

Q = set of all states

Σ = set of I/P symbols

δ = Transition function

q_0 = initial state

F = set of all final state.

Transition function :-

$$\delta :- \quad Q \times \Sigma \rightarrow Q$$

$$\{q_1, q_2, D\} \times \{a, b\} \rightarrow \{q_1, q_2, D\}$$

$$(q_1 \times a) \longrightarrow (q_2)$$

$$(q_2 \times b) \longrightarrow (q_2)$$

$$(q_1 \times b) \longrightarrow (D)$$

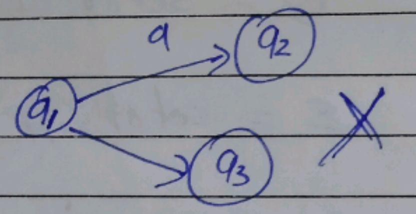
⋮

> in DFA for a single I/P from a state it goes to only one next state.

in other wods,

The DFA defines the next state with out any ambiguity for a single I/P.

Note; in DFA

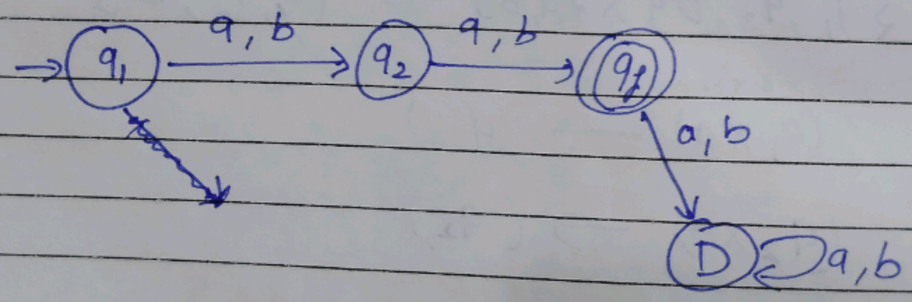


this should not happen.

Q) Construct a DFA, that accept set of all string over {a,b} of length 2.

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

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



The Final defn of DFA

every state should consist out edges form it with all the possible i/p & the with i/p symbol ~~it~~ ~~sho~~ the next state should not be ambigous.

String acceptance :-

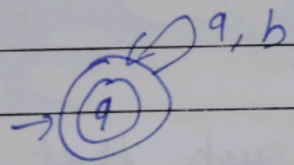
scan the entire string with the help of FA & we reach to the final state from initial state then string is accepted.

Lang. acceptance :-

A FA is said to accept a lang. if all the string in the lang are accepted & all the string not in the lang. should be rejected.

Rejection of non-valid strings are also important & it has the same weightage as acceptance.

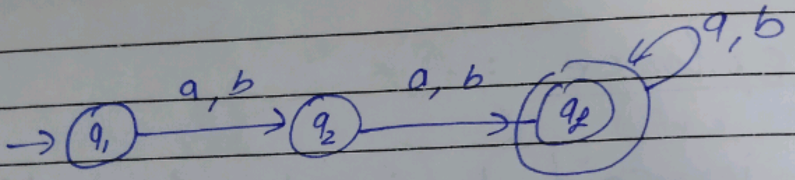
for eg..



The given Automata accept ever string which we throw on it. but it is not a valid Automata for the restricted strings. as it accept all the string - valid, invalid both.

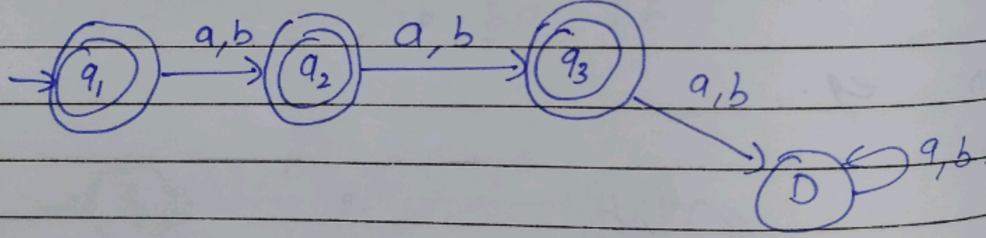
Q) Create a DFA over $\Sigma = \{a, b\}$ where $|w| \geq 2$ (~~no. of words~~ length of string must be "2" or greater.

$L = \{aa, ab, ba, bb, aaa, aba - - - \}$



Q DFA over ~~ab~~ $w = \{a, b\}$ where $|w| \leq 2$

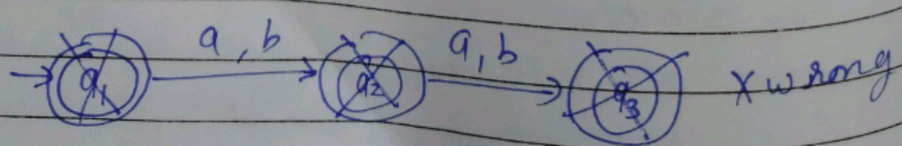
$L = \{\varepsilon, a, b, aa, ab, ba, bb\}$

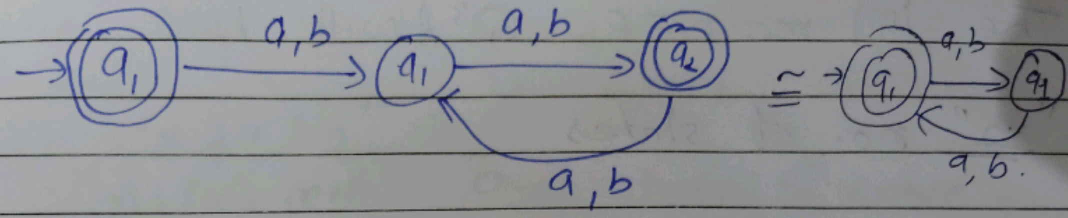


Q) DFA $w = \{a, b\}$ such that

$|w| \bmod 2 = 0$

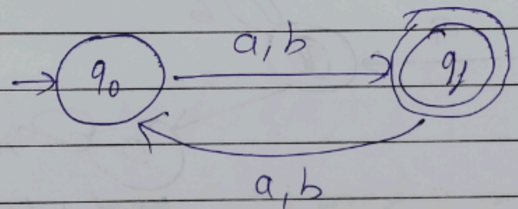
$L = \{\varepsilon, aa, bb, ab, ba, aaaa, bbbb - - - \}$





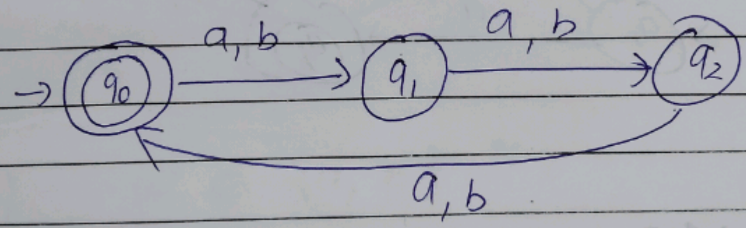
Q) DFA $w = \{a, b\}$ where

$|w| \bmod 2 = 1$.



Q) $w \in \{a,b\}^*$, $|w| \bmod 3 = 0$

$L = \{ \varepsilon, aaa, aab, aba, bbb \ldots\ldots \}$



it is(a) written same as.

$|w| \cong \bmod 3$

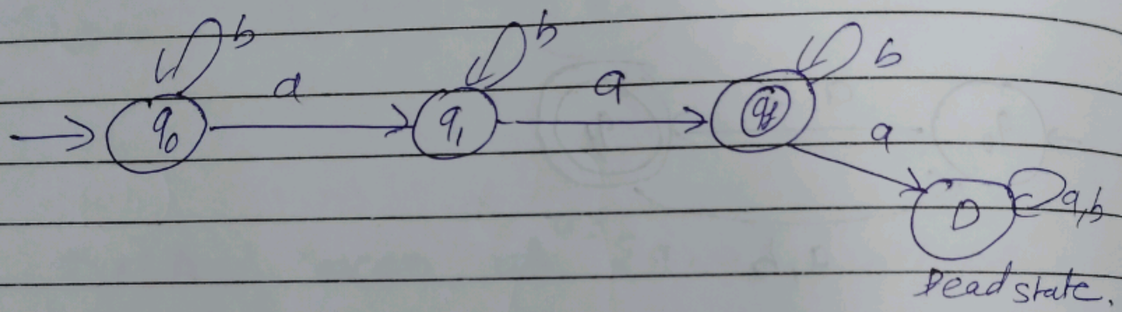
or.

$|w| \bmod 3 = 0$

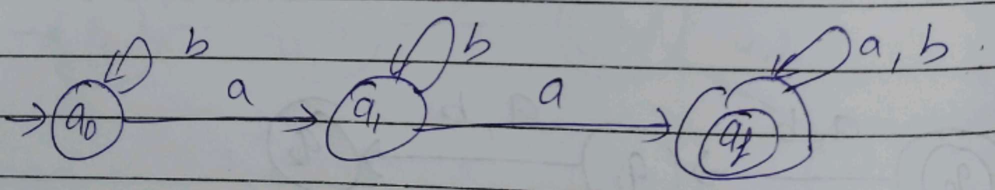
The $|w| \bmod n = 0$ Q's A will have "<u>n</u>" no. of states

Q) $w \in \{a,b\}^*$, $n_a(w) = 2$

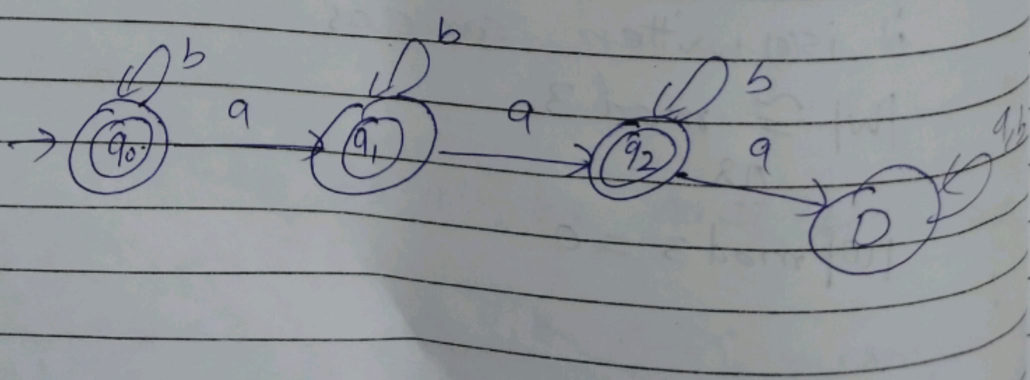
$L = \{aa, baa, aba, aab \ldots\}$



Q) $n_a(w) \geq 2$



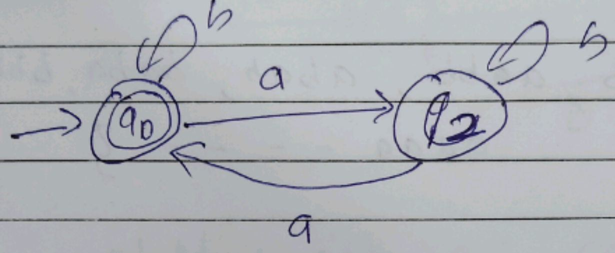
Q) $n_a(w) \not> 2$



Q) no. of 'a' must be even. over $\Sigma = \{a,b\}$

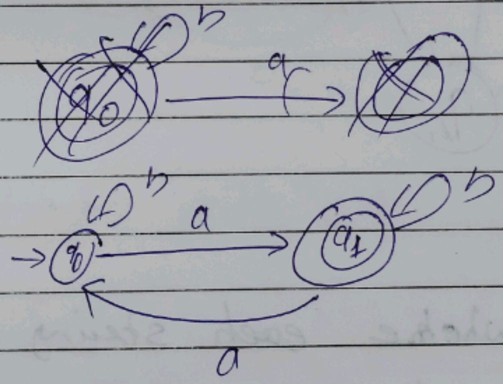
$w \in \{a,b\}^*$

$n_a(w) \bmod 2 = 0$ or $n_a(w) \equiv 0 \bmod 2$.



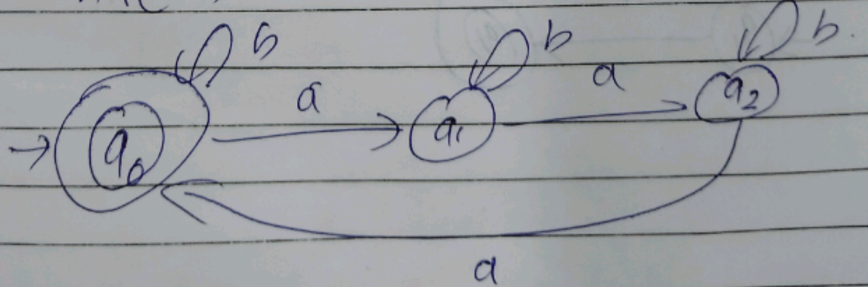
Q) $w \in \{a,b\}^*$

$n_a(w) \bmod 2 = 1$.



Q) $w \in \{a,b\}^*$

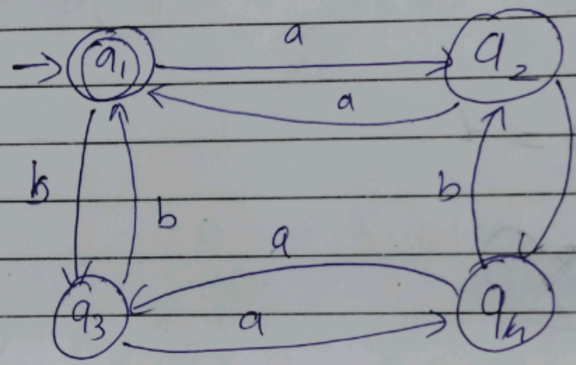
$n_a(w) \bmod 3 = 0$.



Q) minimal DFA $w \in \{a,b\}^*$

$$\begin{cases} n_a(w) \bmod 2 = 0 \\ \& \\ n_b(w) \bmod 2 = 0 \end{cases}$$

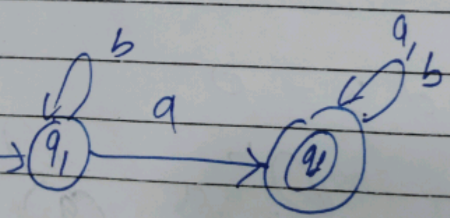
$L = \{\varepsilon, aa, bb, aabb, abab, baba, bbbb, aaaa - - - - \}$



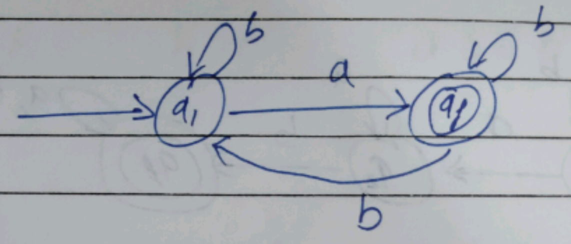
Q) $w \in (a,b)^*$ where each string contain 'a'

a

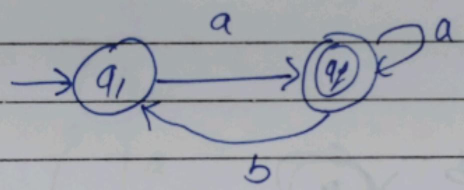
A :-



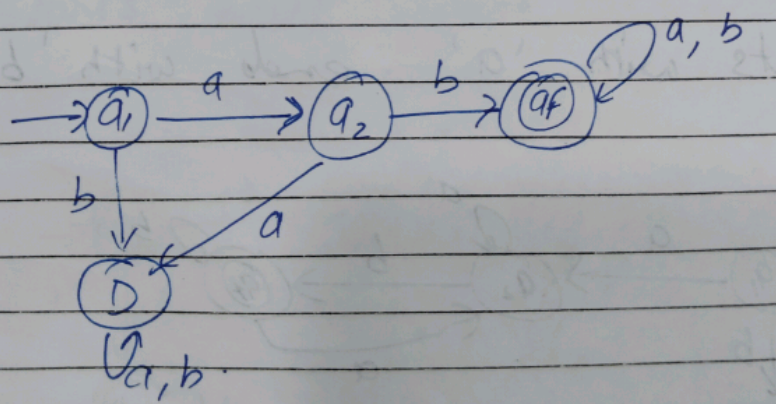
Q) set of all the string over $w \in (a,b)^*$ that ends with 'a'



Q) starting with 'a' containing 'a' & end with 'a'.

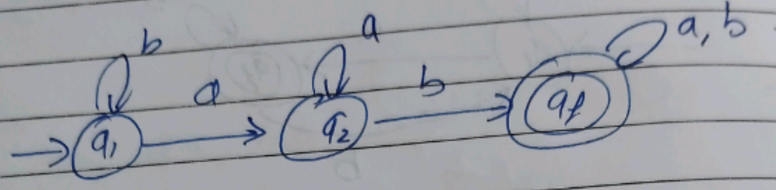


Q) starts with 'ab'

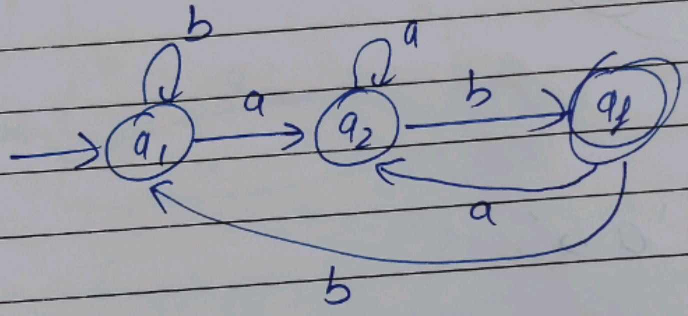


Q) Set of all string that contains 'ab' as substring.

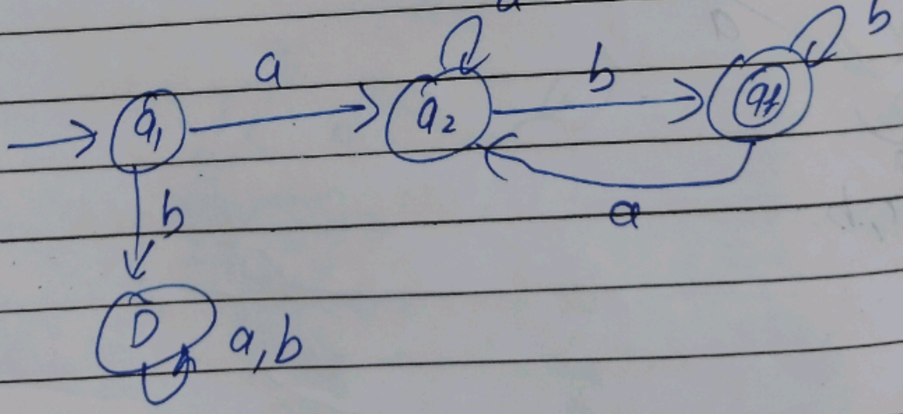
A.



Q) ends with 'ab'



Q) starts with 'a' ends with 'b'



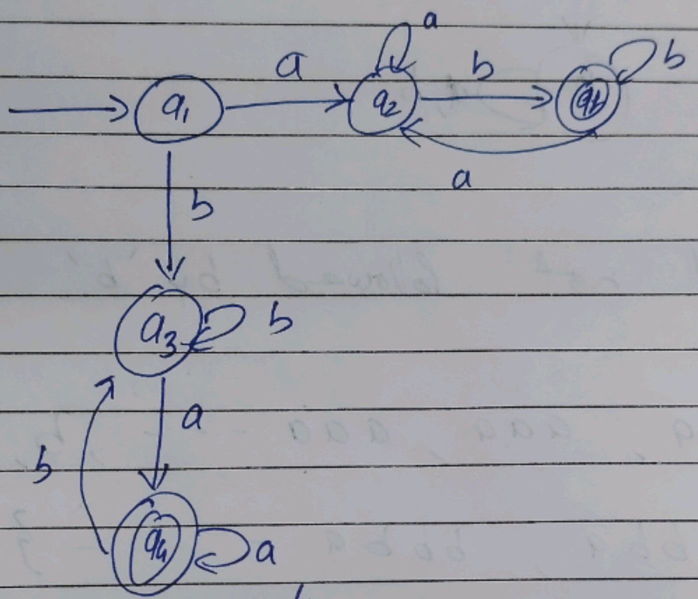
Q) Set o
& and

L = { a

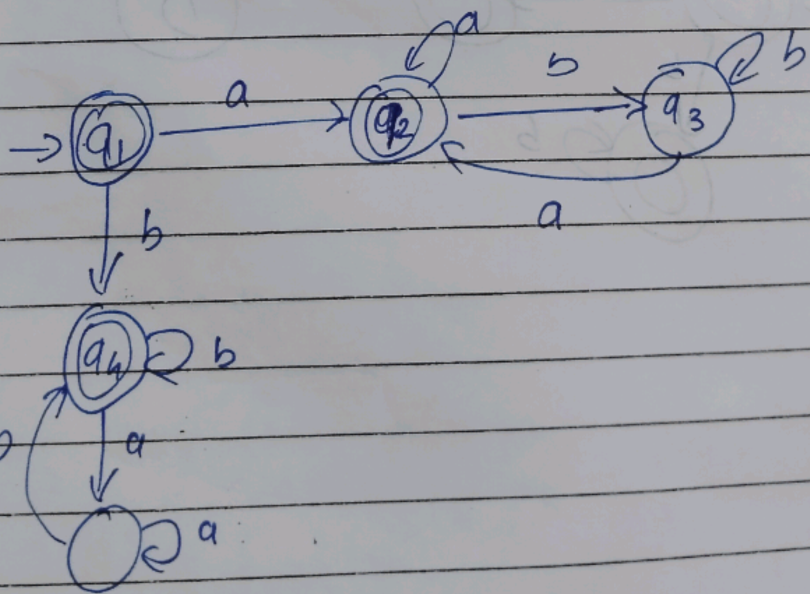
Q) S

Q) Set of all string ~~which wit~~ which starts & and ends with different symbols.

L = {ab, ba, abbb, aaab ...}

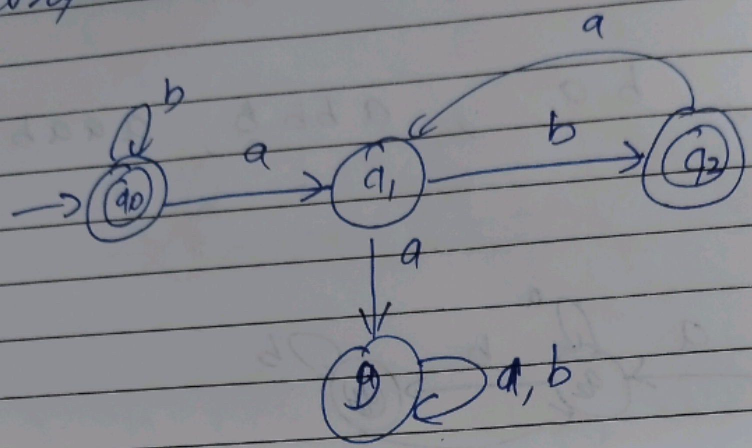


Q) Starts and ends with same symbols.



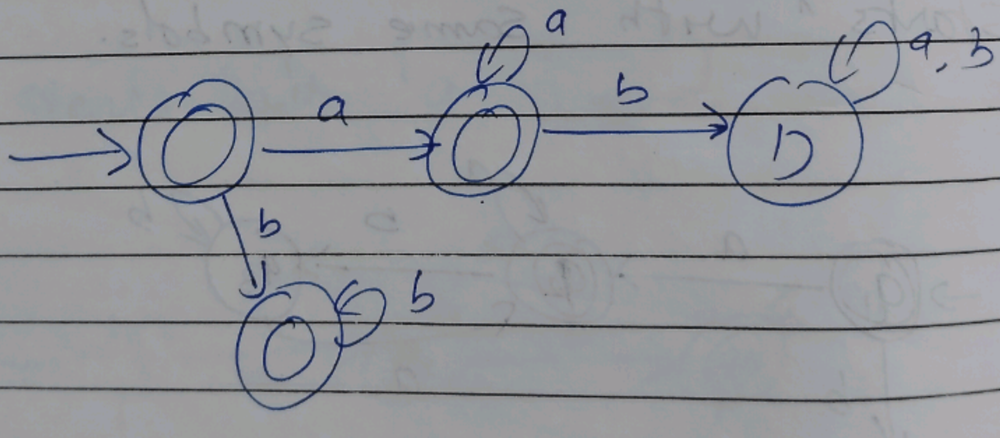
DATE: __/__/__

Q every 'a' should be followed by 'b' i.e ab

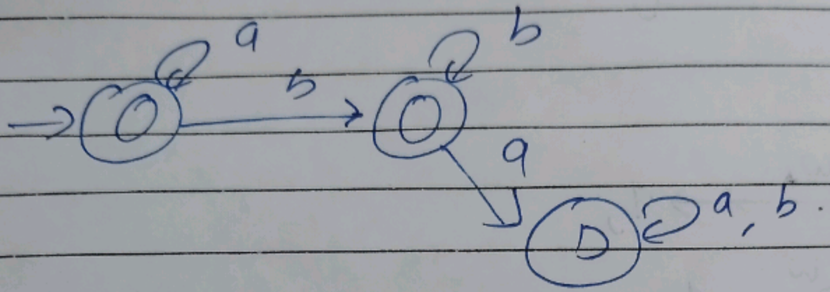


Q 'a' should not followed by 'b'

L = { ε, a, aaa, aaa ---, b, bb..
ba, bba, bbba - - - }



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



NFA

Q $q_i \xrightarrow{w} q_j$ DFA.

$q_i \xrightarrow{w} q_j$

$q_i \xrightarrow{w} q_k$

$\vdots$

$q_i \xrightarrow{w} \infty$

Q

Tuples :-

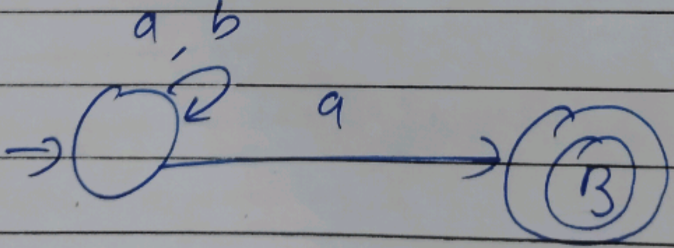
$(Q, \Sigma, \delta, q_0, F)$

all others are same as DFA.

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

$\boxed{2^Q = \text{Power set}}$

Q) ends with 'a'



what

$Q \varepsilon =$

PAGE NO.: ________
DATE: ___/___/___

what is $\underline{2^Q}$?

$Q = \{A, \}$, $\Sigma = \{a, b\}$.

$$Q \times \Sigma \qquad\qquad\qquad 2^Q$$

$$\begin{Bmatrix} A, a \\ A, b \\ B, a \\ B, b \end{Bmatrix} = \begin{pmatrix} \phi \\ \{A\} \\ \{B\} \\ \{C\} \\ \{A, B\} \end{pmatrix}$$