* kleenes theorem Part 2

A language accepted by finite automata is regular.

Proof :

If $M = (Q, \Sigma, q_0, A, \delta)$ is finite automata.

① If we consider p and q as two sets then by using mathematical induction we can take union of it. which is regular.

② If p is an initial state then transition from p to q should be going from p to q in k state, if k is large enough then L will be L (p, q).

③ Let consider n states in the from 1 to n and path going throughs. S.

Now consider the string x, y, z and $x = yz$

$$\delta^*(p, y) = s$$

$$\delta^*(s, z) = q$$

Now,

$$p \xrightarrow{a} q \xrightarrow{b} r$$

If Language $L(P,q)$ for n states then

$$\boxed{L(P,q,n) = L(P,q)}$$

where n is not higher i.e., no of states higher than n.

if $L(P,q,n)$ is regular then $L(P,q,j)$ is also regular if

$$\boxed{0 \leq j \leq n}$$

if $j \leq n$ then

$$L(P,q,n) = L(P,q,j)$$

while proning basic steps i.e., $L(P,q,0)$ is regular.

i.e., the state does not contain more than 0 state i.e., it contain only one symbol i.e, finite so, $L(P,q,0)$ is regular.

By induction hypothesis,
if.

$L(P,q,k)$ is regular.

then

$L(P,q,k+1)$ is also regular.

$L(p, q, k+1)$

The path from p to q doesn't contain path higher than $(k+1)$ state.

Let consider ywz be the states in such a way

y ⇒ from state p to 1st to q

w ⇒ from $(k+1)$ to q

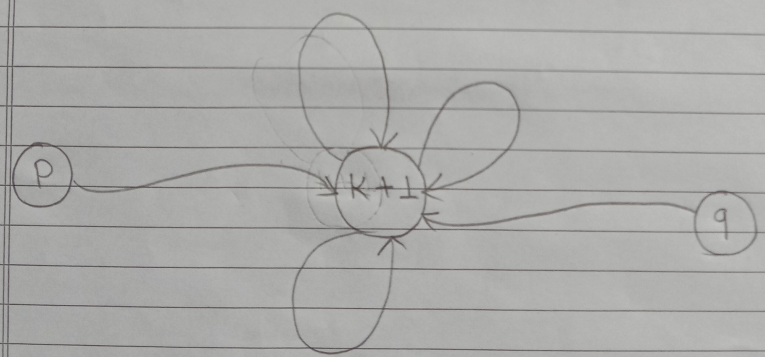
z ⇒ from $(k+1)$ to back itself.

i.e.,

$y \in L(p, k+1, k)$

$w \in L(k+1, q, k)$

$z \in L(k+1, k+1, k)^*$



it follows either of following two cases

$$x \in L(p,q,k) \cup L(p,k+1,k)\ L(k+1,q,k)\ L(k+1,k+1,k)^* \quad \text{--- (1)}$$

on the other hand by using (1)

$$\boxed{x \in L(p,q,k+1) = L(p,q,k) \cup L(p,k+1,k)\ L(k+1,q,k)\ L(k+1,k+1,k)^*}$$

From the above formula we have proved it is regular because it satisfy the union, concatenation and kleene operation.

Page No.:
Date: / /

* Minimum state FA for Regular Language

Removing / minimizing the original states in the language so, original meaning should not change.

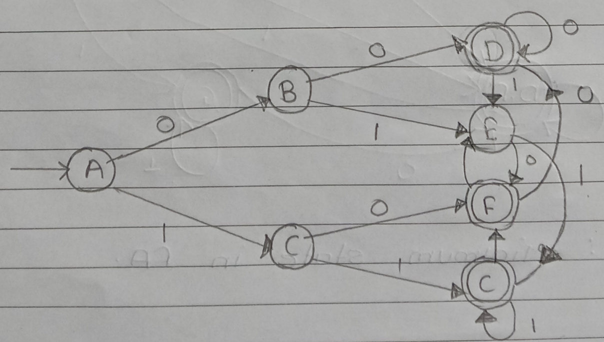
Algorithm :

① Eliminate any state that can't be reached from initial state.

② Then partation then remaining states into blocks so that all states in the same blocks are equivalent no pair of states from different block are equivalent.

Ex : from given table and diagram minimize FA the partation of states into equivalent blocks is,

({A} , {B} , {C , E} , {D , F} , {G})



on the given transaction diagram applying Table filling Algorithm we get following

X distinguishable states
blank square are equivalent.

	A	B	C	D	E	F
B	X					
C	X	X				
D	X	X	X			
E	X	X		X		
F	X	X	X		X	
G	X	X	X	X	X	X

Fig: Table of state Equivalence

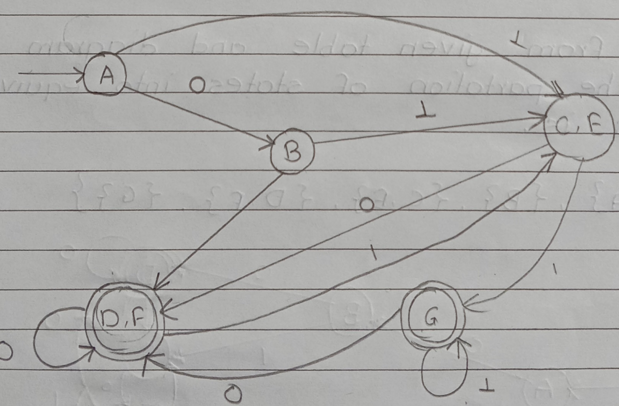


Fig: Minimum state in FA.

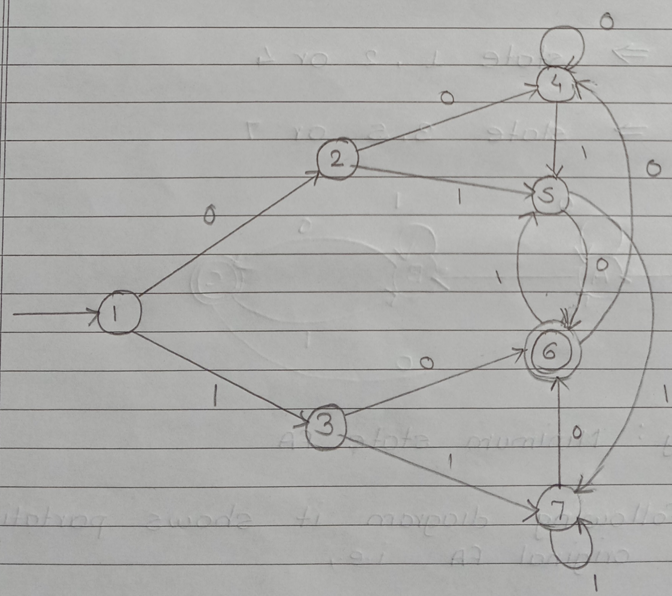
* Minimizing no of states in FA

Let $M = (Q, \Sigma, q_0, A, \delta)$ be the finite automata then while minimizing no of states

① consider a single state for equal states

② Remove unreachable states for initial state without affecting meaning of original state.

consider A finite automata with $M = (Q, \Sigma, q_0, A, \delta)$ with 7 state now.



a) digital finite automata [Transition diagram] input

		0	1	
	Λ	1	0	1	
	0	2	00	01	
State	1	3	10	11	
	00	4	00	01	
	01	5	10	11	
	10	6	00	01	← Accepting state
	11	7	10	11	

fig : Transition table

In given table state 6 is accepting state so, considered as C

A ⇒ state 1, 2 or 4

B ⇒ state 3, 5 or 7

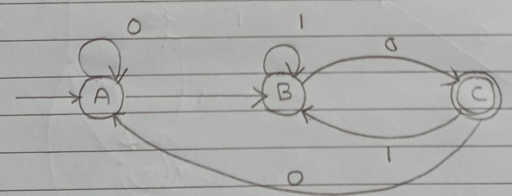


fig : Minimum state FA

In Following diagram it shows portation with original FA i.e.,

L1, L2, L3, L4, L5, L6, L7

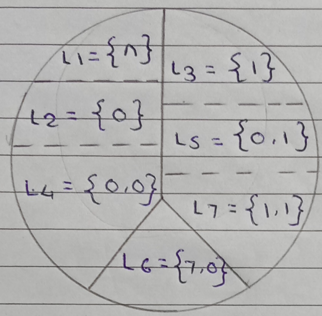


fig : original finite automata

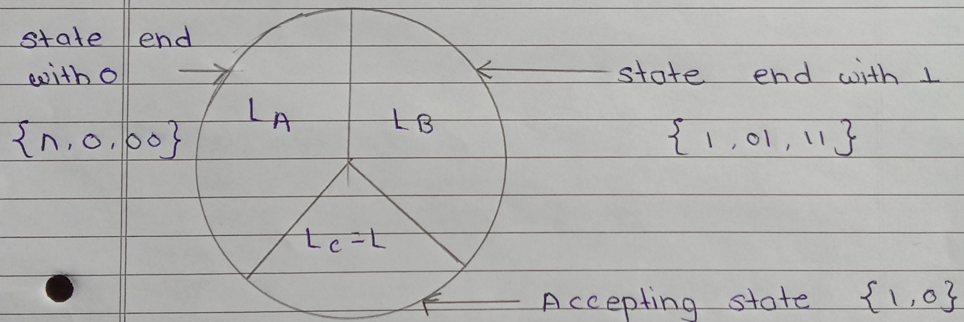
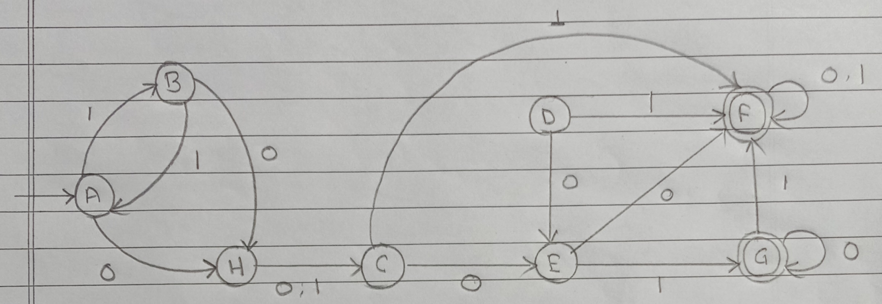


fig : Minimal state in FA

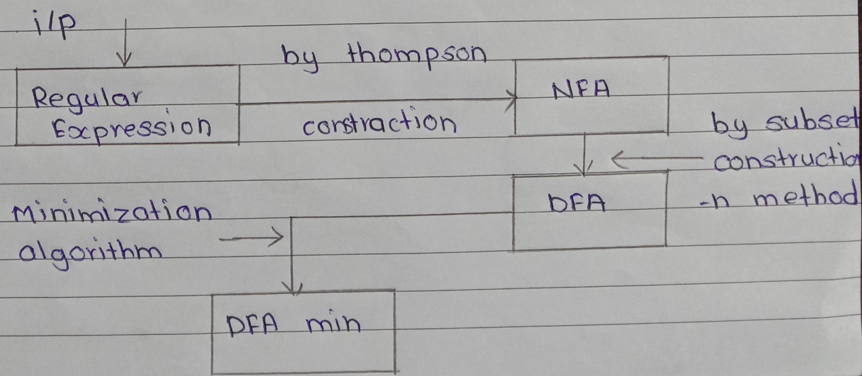
Hence it is minimal FA , of given FA

* DFA Minimization



- Text processing - larger data set
- compiler
- smoller DFA require less space and time
- Sare resources

* Compiler conversion from Regular Expression to DFA min



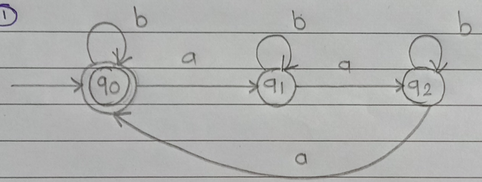
* Draw DFA for following :

① $\{ w \mid n_a \bmod 3 = 0 \}$

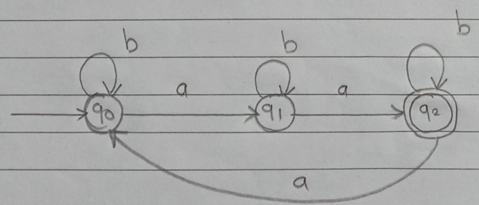
② $\{ w \mid n_a \bmod 3 > 1 \}$

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

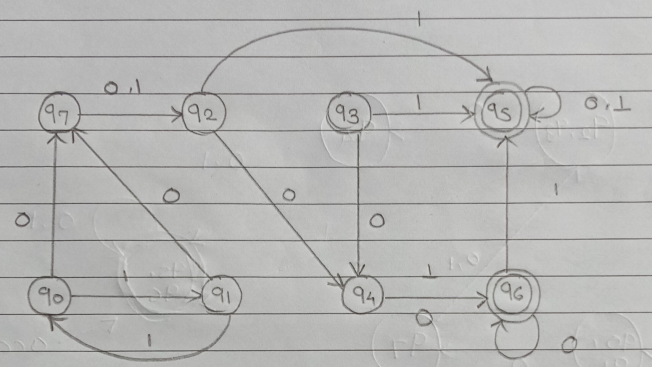
①



②



* Reduce the DFA



$S_1 = \{q_5, q_6\}$

$S_2 = \{q_0, q_1, q_2, q_3, q_4, q_7\}$

while making transitions we get,

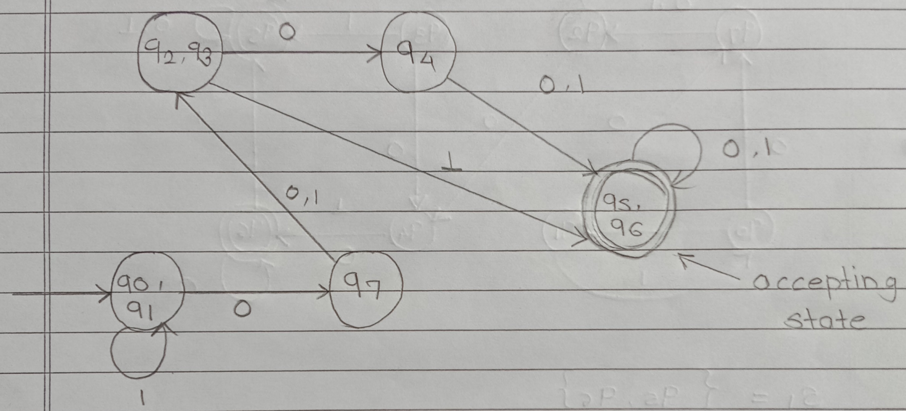
$S_1 = \{q_5, q_6\}$ $\qquad$ $S_2 = \{q_0, q_1\}$

$S_3 = \{q_2, q_3\}$ $\qquad$ $S_4 = \{q_4\}$

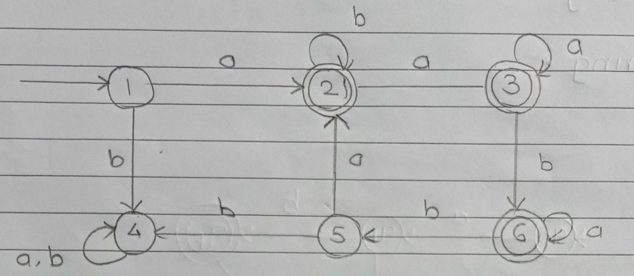
$S_5 = \{q_7\}$

Lucky Page No.:
Date : / /

Minimize DFA states in given DFA

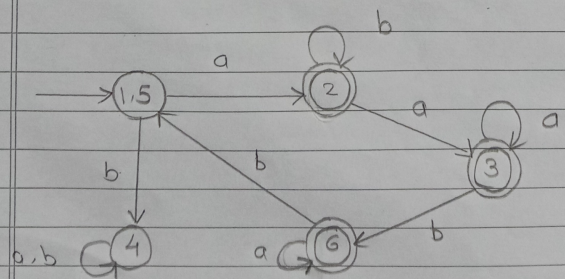


* Reduce the DFA



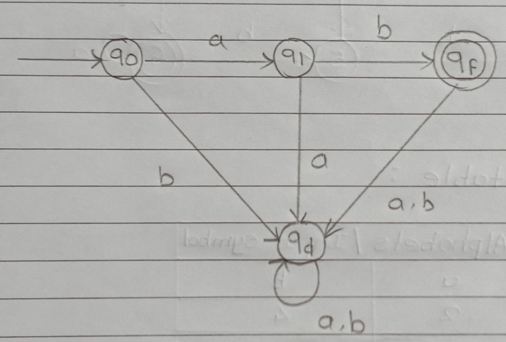
Transition table :

State	Alphabets / Input symbol	
	a	b
1	2	4
2	3	2
3	3	6
4	4	4
5	2	4
6	6	5

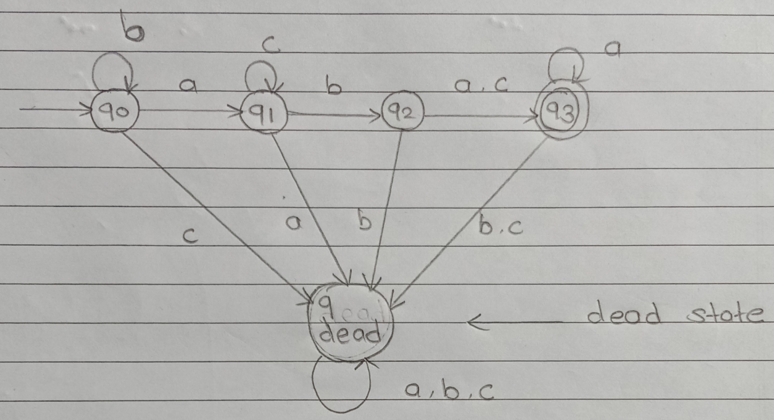


Exactly 1 a's and 1 b's in string

string :



* Dead states :



	a	b	c
q_0	q_1	q_0	q dead
q_1	q dead	q_2	q_1
q_2	q_3	q dead	q_3
q_3	q_3	q dead	q dead
q dead	q dead	q dead	q dead

The above transition Diagram connecting q_0 as initial state, q_3 accepting state. The transition which are not going to other state with alphabets will goes to dead state.

Page No.: Date: / /

* construct NFA - Λ to NFA

① Remove All Λ transactions in Resultant NFA

② If initial state in NFA - Λ is reaching to an accepting state of NFA - Λ then in NFA initial state will become accepting state with previous accepting state seprately.

Ex :

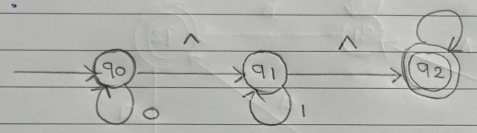


fig : state diagram

Transation table for given state diagram

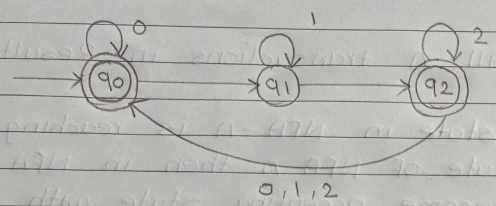
	0	1	2	Λ
$\rightarrow q_0$	$\{q_0\}$	$\emptyset$	$\emptyset$	$\{q_1\}$
q_1	$\emptyset$	$\{q_1\}$	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$	$\{q_2\}$	$\emptyset$

While converting NFA - Λ to NFA we need to convert NΛ to NFA by removing Λ transition in NFA - Λ

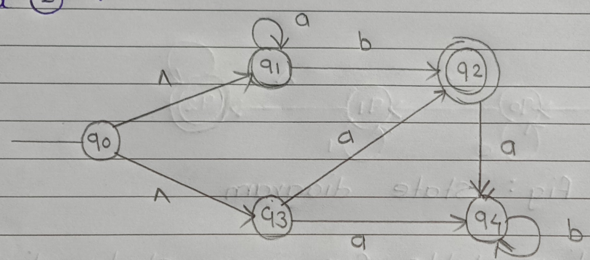
	0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	$\emptyset$	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$	$\{q_2\}$

fig : state diagram

Resultant NFA



Ex ② :



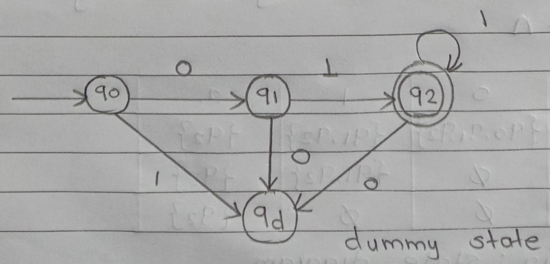
Optimum DFA

① unreachable state

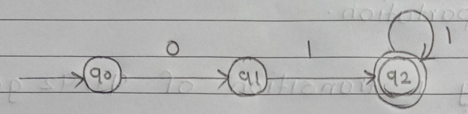
② Read state / Dummy

③ Non- distinguishable or indistinguishble or equivalent state

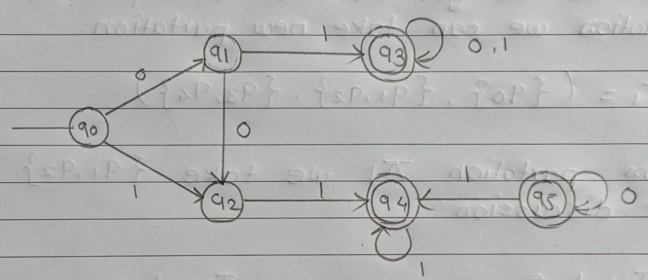
① Minimize the DFA



Minimized DFA



Ex ③ :



We have to make partation as per the type of states

$$\pi = (\{q_0, q_1, q_2\}, \{q_3, q_4\})$$

Now take first partations ~~for~~ $\{q_0, q_1, q_2\}$

① Take transition for 1^{st} partation

Input 0	Input 1
$\delta(q_0, 0) = q_1$	$\delta(q_0, 1) = q_2$
$\delta(q_1, 0) = q_2$	$\delta(q_1, 1) = q_3$
$\delta(q_2, 0) = \phi$	$\delta(q_2, 1) = q_4$

For input 1, state q_0 goes to q_2 which is part of partation.

We obseriving 1 transition of q_1, q_2 goes to other partation

Now, as we get transition with previous partation we can take new partation.

$$\pi_1 = (\{q_0\}, \{q_1, q_2\}, \{q_3, q_4\})$$

From partation π_1 we take $\{q_1, q_2\}$ for discussion

Input 0	Input 1
$\delta(q_1, 0) = q_2$	$\delta(q_1, 1) = q_3$
$\delta(q_2, 0) = \emptyset$	$\delta(q_2, 1) = q_4$

Both q_1 and q_2's 1 transition are going to final / Accepting state.

i.e. $q_3 \in F$, also $q_4 \in F$

Both q_1, q_2 goes to other states will transition 1 thus we not need to partation q_1 and q_2 once again

Hence they will become indistinguishble state

Now take $\{q_3, q_4\}$ together for finding transition with 0 and 1

Input 0

$\delta(q_3, 0) = q_3$

$\delta(q_4, 0) = \phi$

Input 1

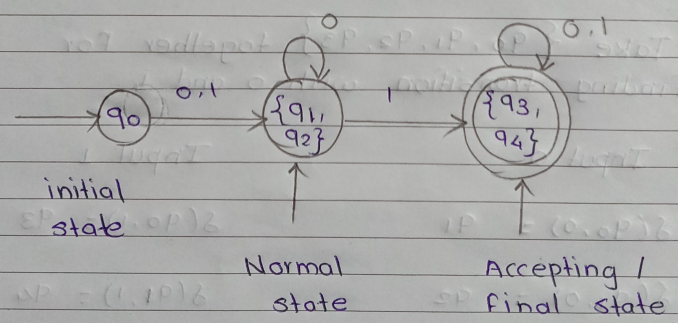
$\delta(q_3, 1) = q_3 \in F$

$\delta(q_4, 1) = q_4 \in F$

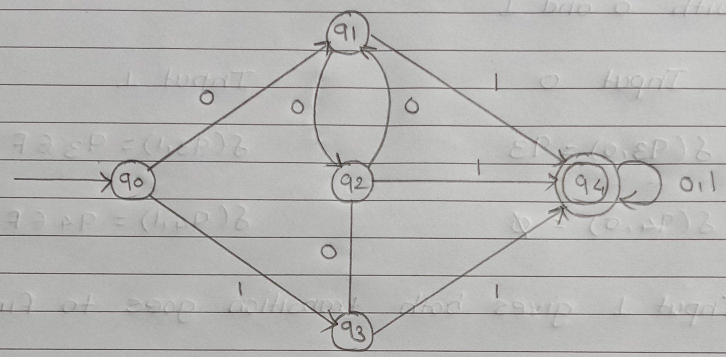
Input 1 gives both transition goes to final state

so partation π_1 will be final result i.e,

$\pi_1 = (\{q_0\}, \{q_1, q_2\}, \{q_3, q_4\})$



Ex ④ :



Now partation will be,

$\pi = (\{q_0, q_1, q_2, q_3\}, \{q_4\})$

Take $\{q_0, q_1, q_2, q_3\}$ together for finding transition with 0 and 1

Input 0	Input 1
$\delta(q_0, 0) = q_1$	$\delta(q_0, 1) = q_3$
$\delta(q_1, 0) = q_2$	$\delta(q_1, 1) = q_4$
$\delta(q_2, 0) = q_1$	$\delta(q_2, 1) = q_4$
$\delta(q_3, 0) = q_2$	$\delta(q_3, 1) = q_4$

Next partation will be,

Page No.:
Date: / /

$\pi_1 = (\{q_0\}, \{q_1, q_2, q_3\}, \{q_4\})$

Take transition of $\{q_1, q_2, q_3\}$ with Σ

Input 0	Input 1
$\delta(q_1, 0) = q_2$	$\delta(q_1, 1) = q_4 \in F$
$\delta(q_2, 0) = q_1$	$\delta(q_2, 1) = q_4 \in F$
$\delta(q_3, 0) = q_2$	$\delta(q_3, 1) = q_4 \in F$

Hence the final partation is

$\pi_1 = (\{q_0\}, \{q_1, q_2, q_3\}, \{q_4\})$

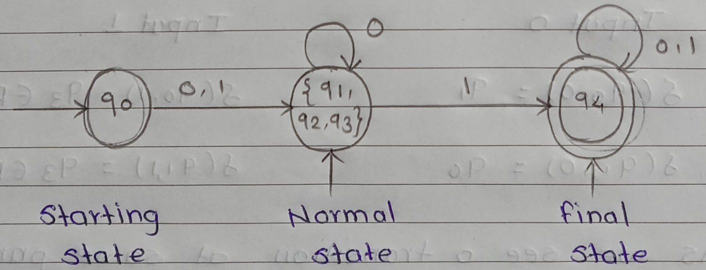
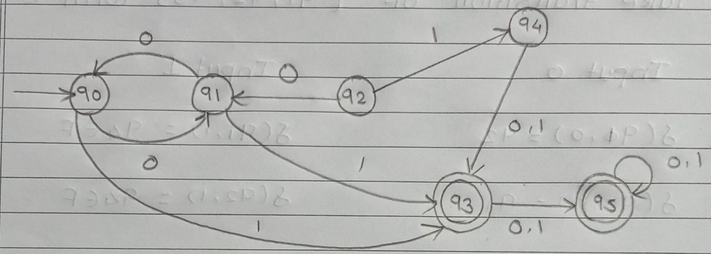


fig: Minimized DFA.

Ex ③ :



Partation will be

$$\pi = (\{q_0, q_1\}, \{q_3, q_5\})$$

Take $\{q_0, q_1\}$ and their transition

Input 0	Input 1
$\delta(q_0, 0) = q_1$	$\delta(q_0, 1) = q_3 \in F$
$\delta(q_1, 0) = q_0$	$\delta(q_1, 1) = q_3 \in F$

As we see 0 transition at same partation hence not required to divide Also 1 transition goes to final state hence it will remain as it.

Page No.:
Date: / /

Now goes for $\{q_3, q_5\}$

Input 0

$\delta(q_3, 0) = q_5 \in F$

$\delta(q_5, 0) = q_5 \in F$

Input 1

$\delta(q_3, 1) = q_5 \in F$

$\delta(q_5, 1) = q_5 \in F$

so final partation is.

$\pi = (\{q_0, q_1\}, \{q_3, q_5\})$

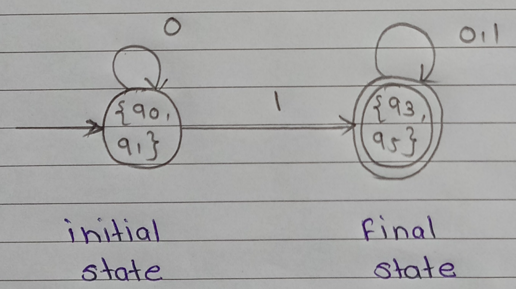


Fig: Minimized DFA.