1. Deterministic Finite Automata:-

finite automata has an input tape, a finite control and a reading head combination as shown below:-

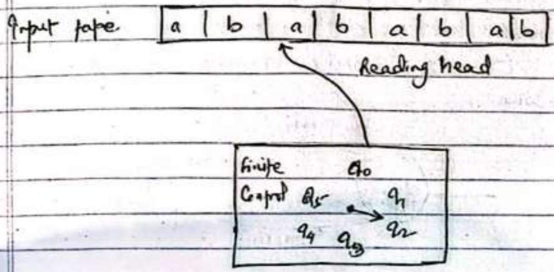


fig. FSA.

q_0 is the starting state.

Hence it reads (q_2, ababab). q_f is the finite length of string and can be determined so named as the deterministic finite automata (DFA).

Df:-

A deterministic finite automaton is a 5 (quintuple) $M = (K, \Sigma, \delta, s, F)$ where,

(i) $K \Rightarrow$ is a finite set of states

(ii) Σ is an alphabet.

(iii) $s \in K$ is the initial state

(iv) $F \subseteq K$ is the set of final states or accepting states

& (v) δ, the transition function, is a function from $K \times \Sigma$ to K [$K \times \Sigma \rightarrow K$].

q_n discrete.
$K \rightarrow S$
$\Sigma \rightarrow I$
$\delta \rightarrow f$
$F \rightarrow A$
$s \rightarrow \alpha$

Design:-

Pb(1). Design a deterministic finite automaton (DFA) M that accepts precisely those strings over alphabet $\Sigma = \{a, b\}$ that contains no a's.

Soln

The idea is to use two states,

A: An a was found

NA: No a's were found.

The transition diagram be,

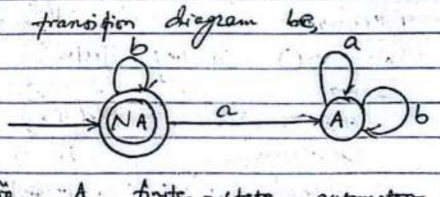


Fig :- A finite-state automaton that accepts precisely those strings over {a, b} that contain No. a's. [Note: This accepts the null string 'e'] pb (2.) Design a finite-state automaton that accepts those strings over alphabet $\Sigma = \{a, b\}$ that contain an odd number of a's.

Soln:- Say the states are:

(i) E: An even number of a's was found

(ii) O: An odd number of a's was found.

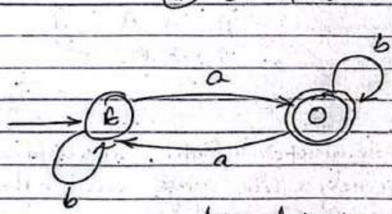


fig- Automaton accepting odd no of a's.

Notes:-

To design the automaton following patterns are considered.

(1). →○ or >○ Starting state

(2). →○→ Intermediate state or (○)

(3). →◎ → final state representation.

pb (3.) Draw the transition diagram of the finite state automaton $M = (K, \Sigma, \delta, s, F)$, where

$\Sigma = \{a, b\}$, $K = \{q_0, q_1, q_2\}$,

$F = \{q_2\}$, $s = q_0$ and the transition function is given by,

$K \backslash \Sigma$	δ a	b
q_0	q_0	q_1
a q_1	q_0	q_2
ab q_2	q_0	q_2

Soln:-

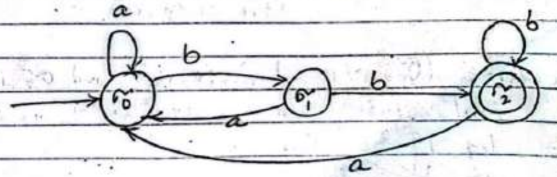
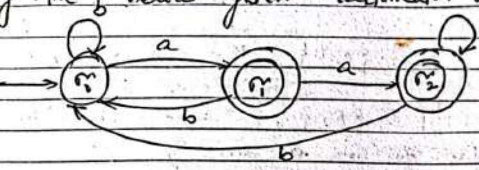


fig- The transition diagram of the above automaton.

pb (4). Test that the substring $\omega =$ "abaa" is accepted by the below given automaton or not?



Sol:-

If M is given the input "abaa", its initial configuration is $(q_0, abaa)$.

Now, $(q_0, abaa) \vdash_M (q_1, baa)$
$\vdash_M (q_0, aa)$
$\vdash_M (q_1, a)$
$\vdash_M (q_2, e)$

Therefore, $(q_0, abaa) \vdash_M^* (q_2, e)$, and q_2 is the final state so, "abaa" is accepted by M.

pb (5). Design a deterministic finite automaton (DFA) M that accepts the language given by $L = \{ \omega \in \{a, b\}^* : \omega$ has even number of b$\}$

Sol:-

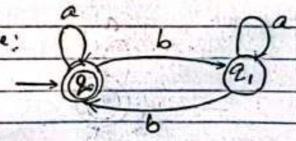
Let $M = (K, \Sigma, \delta, s, F)$ be required DFA;
where $K = \{q_0, q_1\}$

$\Sigma = \{a, b\}$
$s = q_0$
$F = \{q_0\}$

And transition function δ is as follows:

transition table →	State	a	b
	q_0	q_0	q_1
	q_1	q_1	q_0

Then the state diagram be:



let us test above automaton for string "bab"

$(q_0, bab) \vdash_M (q_1, ab)$
$\vdash_M (q_1, b)$
$\vdash_M (q_0, e)$

i.e. $(q_0, bab) \vdash_M^* (q_0, e)$ and q_0 is the final or accepting state, so, the substring "bab" is accepted by above automaton.

Q(6). Design a DFA that accepts a language given by $L = \{ w \in \{a, b\}^* : w$ does not contain three consecutive b's$\}$.

Solution:- Let $M = \{K, \Sigma, \delta, s, F\}$ be required DFA

Where,

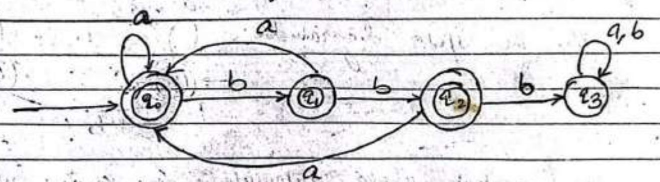
$K = \{ q_0, q_1, q_2$

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

$s = q_0$

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

Then the transition diagram be

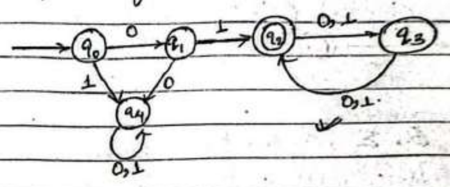


And the transition function δ is given by.

q	σ	$\delta(q, \sigma)$
q_0	a	q_0
q_0	b	q_1
q_1	a	q_0
q_1	b	q_2
q_2	a	q_0
q_2	b	q_3
q_3	a	q_3
q_3	b	q_3

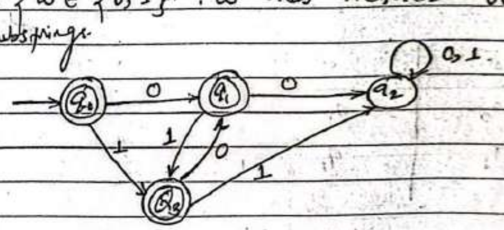
Pb(7). Design a DFA that accepts a language by $L = \{ w \in \{0, 1\}^* : w$ begin's with 01 a[nd] having even length$\}$.

Sol:- Transition diagram be,



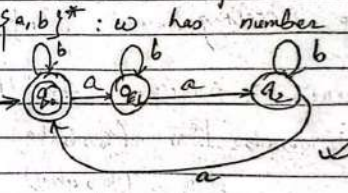
Pb 8 Design a DFA that accepts a language $L = \{ w \in \{0, 1\}^* : w$ has neither 00 nor [11] substrings.

Sol:-



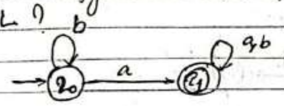
Pb(9) Design a DFA that accepts a language gi[ven] $L = \{ w \in \{a, b\}^* : w$ has number of a multip[le]

Sol:-



(10) Construct a DFA that recognize a language L with expression $b^*a(a \cup b)^*$ and show the processing for the string w = "bbaba" in terms of configuration transition. Does w belongs to L?

Sol:



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

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

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

δ	a	b
q_0	q_1	q_0
q_1	q_1	q_1

Starting state $q_0 = q_0$

final or accepting state $F = \{q_1\}$

The configuration transition for "bbaba" be:

q_0, bbaba $\vdash_D (q_0, baba)$

$\vdash_D (q_0, aba)$

$\vdash_D (q_1, ba)$

$\vdash_D (q_1, a)$

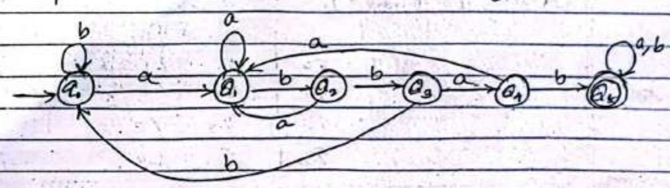
$\vdash_D (q_1, e)$

Hence, $(q_0, bbaba) \vdash_D^* (q_1, e)$ and q_1 is the final (accept) state, hence, "bbaba" $\in L$.

(11) Construct a DFA that recognizes a language over the alphabet $\Sigma = \{a, b\}$, Where,

$L = \{w : w \in \Sigma^*, w$ contains substring abbab$\}$.

Sol:



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

Where, $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

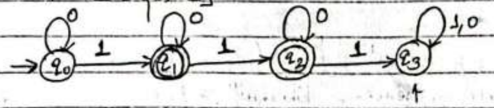
$\Sigma = \{a, b\}$ $\quad F = \{q_5\}$

and transition function δ is given by:

δ	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_4	q_0
q_4	q_1	q_5
q_5	q_5	q_5

(12) Construct DFA for Language
$L = \{ w : w \in \Sigma^*,$ w has exactly one or two 1 $\}$
and $\Sigma = \{0, 1\}$

Sol:-



$M = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{0, 1\}$
$F = \{q_1, q_2\}$
and δ is given by,

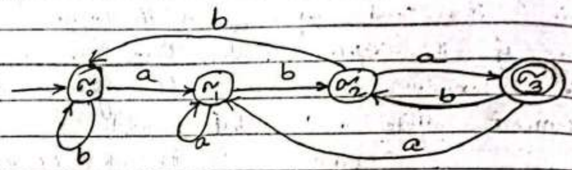
δ	0	1
q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_3
q_3	q_3	q_3

(13). Construct DFA to accept the substring $w =$ 'baa' where w is the string with baa.

Sol:-

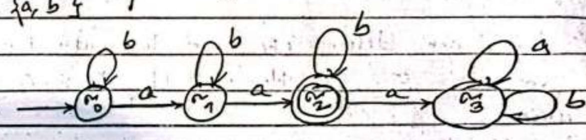
(14) Design DFA which has the acceptance of string on alphabet $\Sigma = \{a, b\}$ where stri with 'aba'.

Sol:-



(15) Design DFA which is capable to accept L which consists of exactly two a's on the alp $\Sigma = \{a, b\}$

Sol:-



Non- Deterministic Finite Automata:-

Intro: As in case of the DFA, the state changed from one to another is exactly determined. But, in NDFA the change of states are partially determined by the current state and input symbol. The state change is function of the state variables. Hence, there will be several possible "next-states" for a given combination of current state and input symbol.

NDFA reads the input string and may choose at each step to go into any of the legal next states, the choice is not determined by anything in the model, and is therefore said to be non-deterministic.

Defn A non- deterministic finite automaton (NDFA) is also a '5-' quintuple i.e. $M_{ND} = (K, \Sigma, \Delta, s, F)$, where,

(i) K is a finite set of states
(ii) Σ is an input alphabet
(iii) $s \in K$ is an initial state (starting state)
(iv) $F \subseteq K$ is a set of final states
(v) Δ is $\Delta: K \times (\Sigma \cup \{e\}) \rightarrow P(K)$ is transition function [$P(K) \Rightarrow$ power set of K].

$\therefore M_{ND} = (K, \Sigma, \Delta, s, F)$

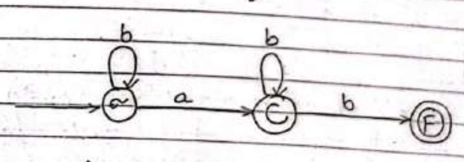
① Draw the transition diagram of the NDFA described by,

$\Sigma = \{a, b\}$
$K = \{\sigma, C, F\}$
$F = \{F\}$
$s = \sigma$

and Δ is given by,

$K \setminus \Sigma$	Δ: a	b
σ	$\{C\}$	$\{\sigma\}$
C	ϕ	$\{C, F\}$
F	ϕ	ϕ

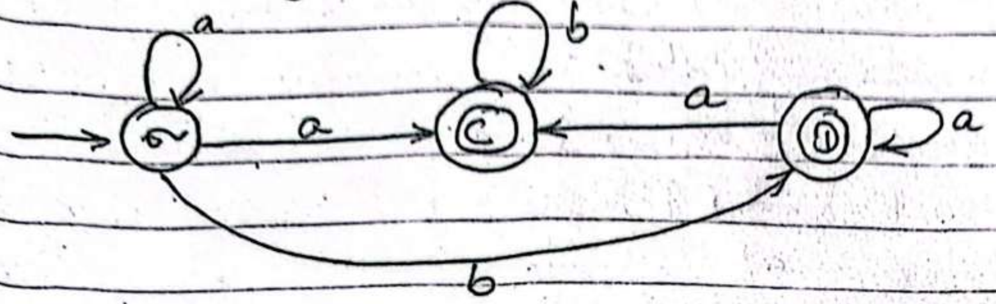
Soln: The transition diagram be:-



The productions are:-

$\sigma \rightarrow b\sigma$, $\sigma \rightarrow aC$
$C \rightarrow bC$, $C \rightarrow bF$
$F \rightarrow \lambda$ or e, ϵ or Λ.

NDFA given below, draw its transition table, listing all parameters.



The definition of the above NDFA be:

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

$s = \sigma$

$F = \{C, D\}$

$K = \{\sigma, C, D\}$

and Δ is given by.

Δ

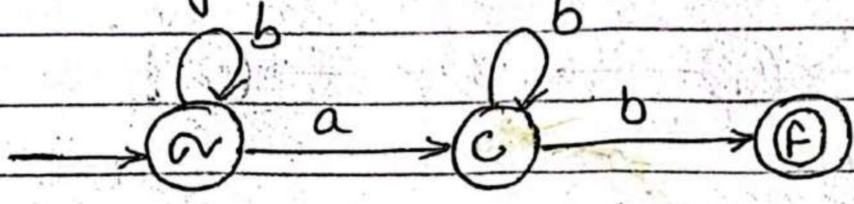
K \ Σ	a	b
σ	$\{\sigma, C\}$	$\{D\}$
C	ϕ	$\{C\}$
D	$\{C, D\}$	ϕ

Where, Σ = alphabet

s = Starting/initial state

F = final/accepting state

pb:
(3) Test that the substring α = "bbabb" is acce
the below given NDFA or not.

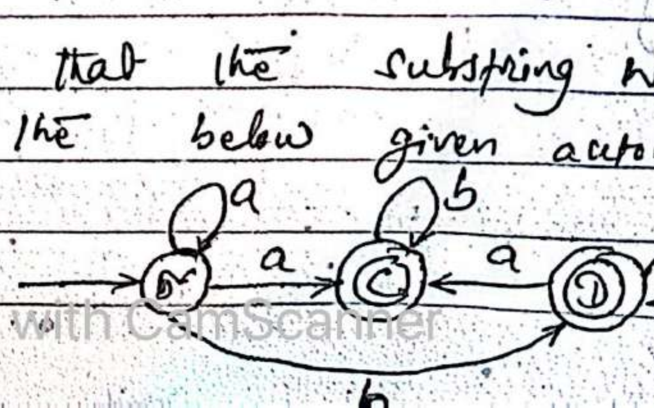


Sol:- The starting state $S = a$
The final state/accepting state $F = \{F\}$
To test the string, α = "bbabb" following to be carried out:

$$(a, bbabb) = \vdash_{M_N} (a, babb)$$
$$= \vdash_{M_N} (a, abb)$$
$$= \vdash_{M_N} (C, bb)$$
$$= \vdash_{M_N} (C, b) \text{ or } = \vdash_{M_N} (F,$$
$$= \vdash_{M_N} (C, e) \text{ or } = \vdash_{M_N} (F,$$

Here, NFA may indicate two or more states with th
input. So, the path at last may be (a, a, a,
which ends with accepting state for final s
F. Hence $(a, bbabb) \vdash_{M_N}^{*} (F, e)$ So the string
is accepted by this automaton. (NDFA).

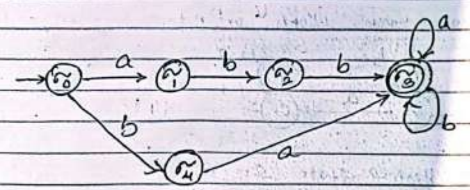
pb(4.) Test that the substring w = "abba" is acc
by the below given automaton (NDFA) or NOT.



[No].

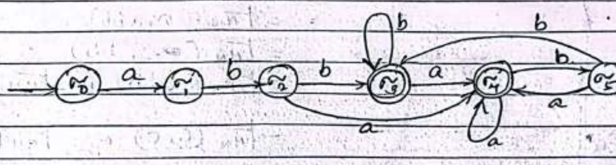
(5) Design NDFA which consists of the string either abb or ba.

Sol:



(6) Design NDFA which accepts the string over $\Sigma = \{a, b\}$, starting with ab but not ending with ab.

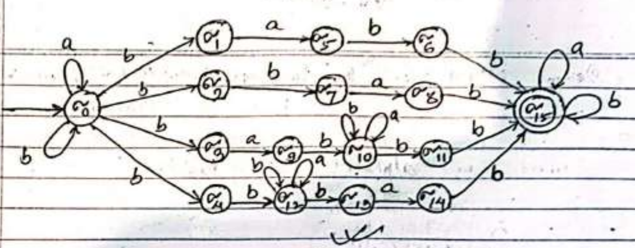
Sol:



(7) Design NDFA which accepts the string containing 'bab' and 'bb' as the substrings over the alphabet $\Sigma = \{a, b\}$.

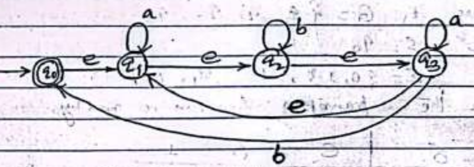
Sol:

Complex
To design for all class of strings



(8) Construct NFA for the regular expression $((a^* b^* a^*)^* b)^*$

Sol:



$K = Q = \{q_0, q_1, q_2, q_3\}$

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

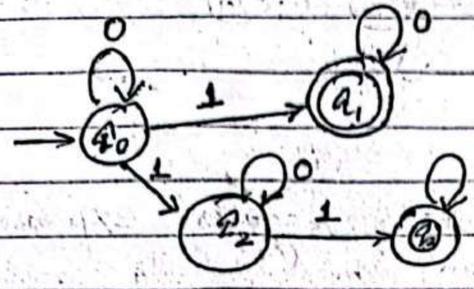
δ	e	a	b
q_0	$\{q_1\}$	ϕ	ϕ
q_1	$\{q_2\}$	$\{q_1\}$	ϕ
q_2	$\{q_3\}$	ϕ	$\{q_2\}$
q_3	$\{q_1\}$	$\{q_3\}$	$\{q_0\}$

$s = q_0 = q_0$ & $F = \{q_0\}$.

Ques (9) Design NDFA for the language $L = \{ w \in \Sigma^* ; w$ contains exactly one or two ones $\}$

where, the alphabet $\Sigma = \{0, 1\}$.

solⁿ:



Where $K = Q = \{q_0, q_1, q_2, q_3\}$

$S = q_0$

$\Sigma = \{0, 1\}$, $F = \{q_1, q_3\}$

& the transition function is given by,

δ	e	0	1
q_0	ϕ	$\{q_0\}$	$\{q_1, q_2\}$
q_1	ϕ	$\{q_1\}$	ϕ
q_2	ϕ	$\{q_2\}$	$\{q_3\}$
q_3	ϕ	$\{q_3\}$	ϕ.

Ques (10) Design NDFA for the regular expression $(a^* b^*)^*$ over the alphabet $\Sigma = \{a, b\}$.

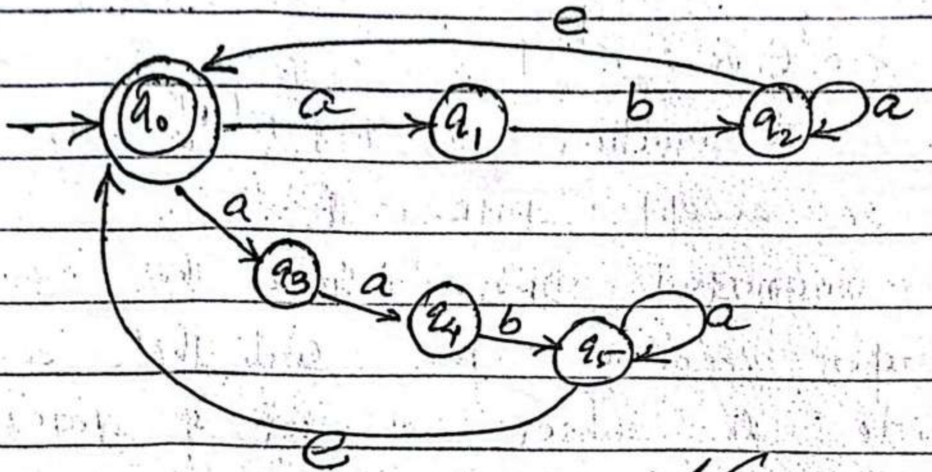
solⁿ:

$CH_2 - 21$

$(aabb)^*$
$(aaa\ bbb)^*$ $\quad a^* b^*$ $\quad a \cdot b$ $\quad \cup$

(ii) Design NDFA for the regular expression ((ab ∪ aab

Sol^n



Theorem:-

Each Non-deterministic (NDFA) finite Automaton, there an equivalent deterministic finite automaton (DFA).

Proof:- let us Consider N be NFA, where,

$N = (Q, \Sigma, \delta, q_0, F)$ [$\because K = Q,\ S = q_0$]

and say $D = (Q', \Sigma', \delta', q_0', F')$ be the DFA states of D be power-set of satetes of N. i.e

$$Q' = P(Q) = 2^{Q}$$

So, $\Sigma' = \Sigma$

$$q_0' = E(q_0);\ E(q_0) = \{q \in Q : (q_0, e) \vdash_N^* (q, e)\}$$

i.e. initial state of D is set of initial states of N.

Transition function,

$$\delta'(R', a) = \{q : q \in Q,\ q \in E(\delta(r', a)), r'$$

δ' is a unique mapping.

$F' = \{R' \in Q' : R' \cap F \neq \phi\}$

i.e. the machine accepts R' if any element of R' is an accept state in F.

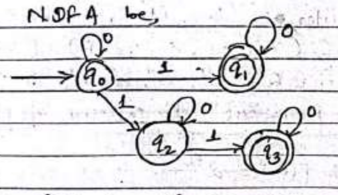
The, so constructed DFA follows the sequence of states upon reading input such that each state corresponds to subset of set of states NFA would occupy.

Q(1). Construct NDFA for the Language,

$L = \{w : w \in \Sigma^*, w$ contains exactly one or two 1's$\}$.

Then convert into corresponding DFA.

Soln- NDFA be,



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

where,

$K = Q = \{q_0, q_1, q_2, q_3\}$

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

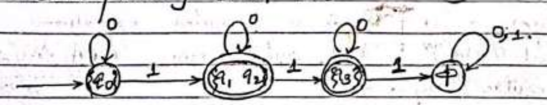
and the transition function is given by,

δ	0	1	e
q_0	$\{q_0\}$	$\{q_1, q_2\}$	ϕ
q_1	$\{q_1\}$	ϕ	ϕ
q_2	$\{q_2\}$	$\{q_3\}$	ϕ
q_3	$\{q_3\}$	ϕ	ϕ.

and $S = q_0 = q_0$

$F = \{q_1, q_3\}$.

The corresponding DFA be,



$K = Q' = \{\{q_0\}, \{q_1, q_2\}, \{q_3\}, \phi\}$

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

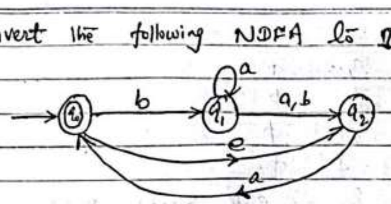
δ'	0	1
$\{q_0\}$	$\{q_0\}$	$\{q_1, q_2\}$
$\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_3\}$
$\{q_3\}$	$\{q_3\}$	ϕ
ϕ	ϕ	ϕ

$S = q_0' = E(q_0) = \{q_0\}$

$F' = \{\{q_1, q_2\}, \{q_3\}\}$.

Pb. HW

(2.) Convert the following NDFA to DFA.



Pb (3.) design a DFA for a regular expression $(ab \cup aba)^*$. And also design NFA for same regular expression.

Sol.

DFA → let $M = (K, \Sigma, \delta, s, F)$ be required DFA.

Where,

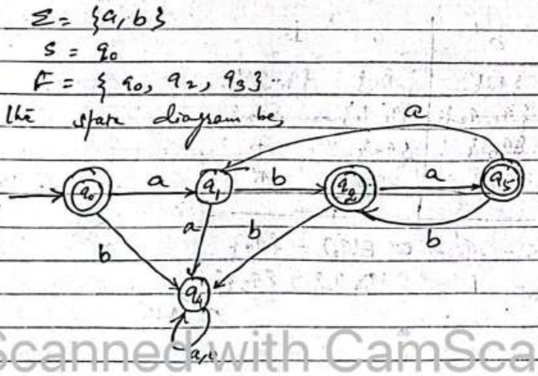
$K = \{q_0, q_1, q_2, q_3, q_4\}$

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

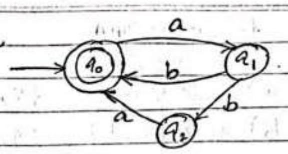
$s = q_0$

$F = \{q_0, q_2, q_3\}$

Then the state diagram be,

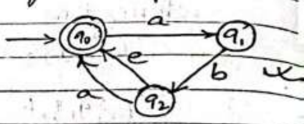


The NFA be,



[Note- 3
are

Pb (4.) Construct NFA for the regular expression $(ab \cup aba)^*$



Application of Theorem:-

Theorem " For each nondeterministic finite automo
is an equivalent deterministic finite automaton".

Steps:-

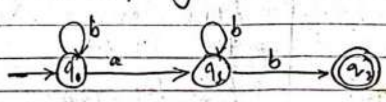
1> Observe all the transitions from starting star
every symbols in Σ.

2> For New states appeared in step-1, check
transitions for every Σ.

3> Repeat step 2 till new states are appear

*Imp

Pb (1.) Convert the following NFA into equivalent



Sol:-

Step-1. The initial state is q_0, so we ca

Write $\{q_0\}$ for new DFA.

Then $\delta'(\{q_0\}, a) = \delta(q_0, a) = \{q_1\}$ New state

$\delta'(\{q_0\}, b) = \delta(q_0, b) = \{q_0\}$ old state

step-2. For new state $\{q_1\}$ appeared in step-1 we have to write:

$\delta'(\{q_1\}, a) = (\delta(q_1, a)) = \phi$

$\delta'(\{q_1\}, b) = \delta(q_1, b) = \{q_1, q_2\}$
New state.

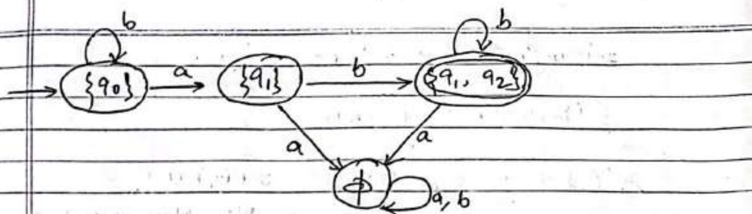
step-3:-

$\delta'(\{q_1, q_2\}, a) = \delta(q_1, a) \cup \delta(q_2, a)$
$= \phi \cup \phi = \phi$

$\delta'(\{q_1, q_2\}, b) = \delta(q_1, b) \cup \delta(q_2, b)$
$= \{q_1, q_2\} \cup \phi$
$= \{q_1, q_2\}$

We may draw the following transition diagram and table:

δ/Σ	a	b
$\rightarrow \{q_0\}$	$\{q_1\}$	$\{q_0\}$
$\{q_1\}$	ϕ	$\{q_1, q_2\}$
$\{q_1, q_2\}$	ϕ	$\{q_1, q_2\}$



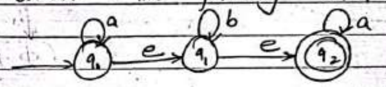
Conversion of epsilon (ϵ/ϵ) based NDFA(ϵ-NFA) to DFA

steps:-

1) Find out the ϵ-closure of starting state and calculate union of ϵ-closure of each transitions for every symbols of Σ.

2) Repeat the same process till new states are appeared.

Q) ① Convert the following NFA to DFA



Sol^n Here, ϵ/e closure of q_0 is Eclose(q_0) or
$E(q_0) = \{q_0, q_1, q_2\}$

Similarly,
$E(q_1) = \{q_1, q_2\}$
$E(q_2) = \{q_2\}$

Now, $\delta'(\{q_0, q_1, q_2\}, a) = E(q_0) \cup \phi\ E(q_2)$

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

$= \{q_0, q_1, q_2\}$ old state.

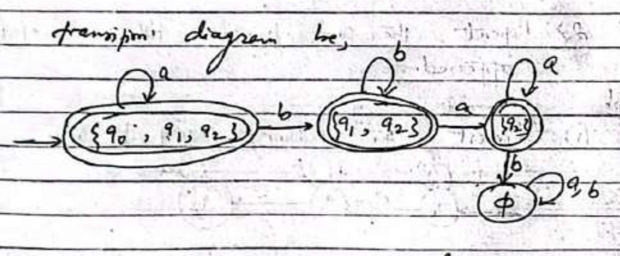
$\delta'(\{q_0, q_1, q_2\}, b) = \phi \cup \epsilon(q_1) \cup \phi$

$= \{q_1, q_2\}$ New state.

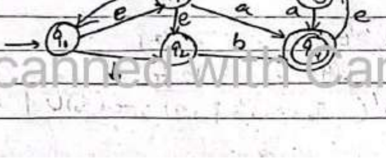
Similarly, we can draw following transition table.

δ/Σ	a	b
$\rightarrow \{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_1, q_2\}$
$\{q_2\}$	$\{q_2\}$	ϕ

The transition diagram be,



Pb ② Convert the following NFA to DFA.

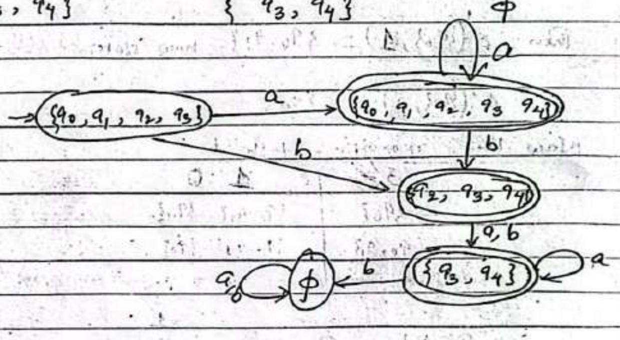


The epsilon closure of $\{q_0\}$ be.

$\epsilon(q_0) = \{q_0, q_1, q_2, q_3\}$ [∵ q_4 is not because ϵ can not link q_1 ...

Now the transition table be.

δ/Σ	a	b
$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3, q_4\}$	$\{q_2, q_3, q_4\}$
$\{q_0, q_1, q_2, q_3, q_4\}$	$\{q_0, q_1, q_2, q_3, q_4\}$	$\{q_2, q_3, q_4\}$
$\{q_2, q_3, q_4\}$	$\{q_3, q_4\}$	$\{q_3, q_4\}$
$\{q_3, q_4\}$	$\{q_3, q_4\}$	ϕ



pb 5. Construct NFA for the Language

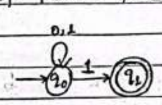
(i.) $L = \{ \omega \in \Sigma^* : \text{Last symbol of } \omega \text{ is } 1 \}$

(ii) $L = \{ \omega \in \Sigma^* : \omega \text{ ends with substring '100'} \}$

Convert them to DFA.

Sol

(i)



δ / Σ	0	1
q_0	q_0	$\{q_1, q_2\}$
q_1	ϕ	$\{q_1\}$

The Conversion of this NFA to DFA is achieved by,

$E(q_0) = \{q_0\}$ for new DFA.

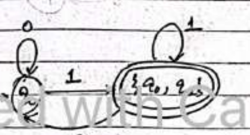
Then $\delta'(\{q_0\}, 1) = \{q_0, q_1\}$ new state.

$\delta'(\{q_0\}, 0) = \{q_0\}$

Now the transition table be

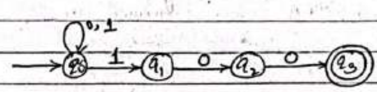
δ' / Σ'	1	0
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0\}$

$\therefore$ DFA be.



Transition diagram.

(ii.) The NDFA be:



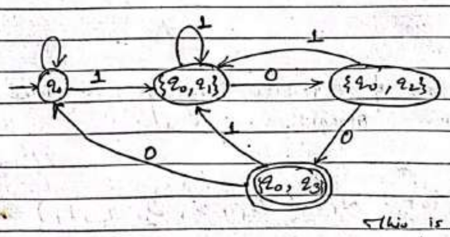
Now, $E(q_0) = \{q_0\}$

$\delta'(\{q_0\}, 0) = \{q_0\}$ old state.

$\delta'(\{q_0\}, 1) = \{q_0, q_1\}$ A new state.

Hence the transition table be:

δ' / Σ'	0	1
$\rightarrow \{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$
$\{q_0, q_2\}$	$\{q_0, q_3\}$	$\{q_0, \cancel{q_1}\}$ $\{q_0, q_2\}$
$\{q_0, q_3\}$	$\{q_0\}$	$\{q_0, q_1\}$



This is the Transition diagram of desired DFA.

Such that,

$$q_0 = q_0' \ , \ Q = q' \cup q'' \ , \ F = F''$$

$$\delta = \delta' \cup \delta'' \cup \{ \delta(q, e) = q_0'', q \in F' \}$$

Then let $w = w_1 w_2$, $w_1 \in L(N_1)$ & $w_2 \in$

Then,

$$(q_0, w) = (q_0', w)$$

$$(q_0, w_1 w_2) \vdash_N^* (q, w_2), \ q \in F_1$$

because,

$$q_0 = q_0' \text{ and } (q_0', w_1) \vdash_N^* (q, e)$$

but,

$$(q, w_2) \vdash_N^* (q_0'', w_2) \ , \ q \in F_1$$

because,

$$\delta(q, e) = q_0'' \text{ if } q \in F_1$$

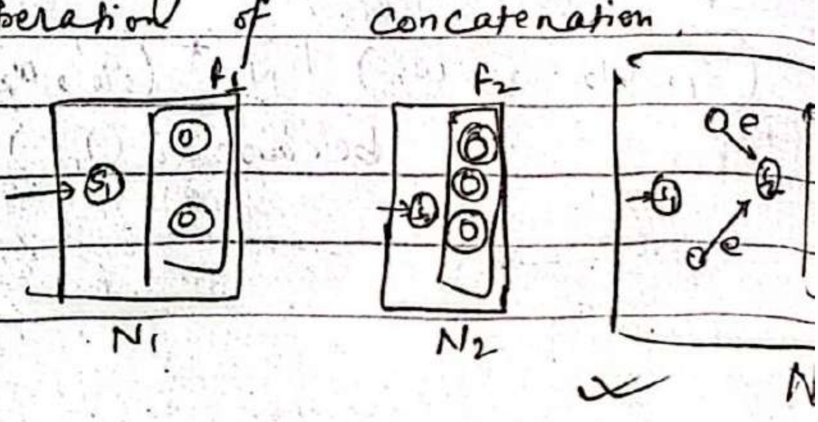
Then $(q_0'', w) \vdash_N^* (q, e)$, $q \in F$

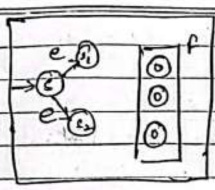
because,

$$(q_0'', w_2) \vdash_N^* (q, e) \ , \ q \in F''$$

Thus, regular languages are closed under operation of concatenation.

Illustration





M

for above figure,

$M = (K, \Sigma, \Delta, s, F)$

$K = K_1 \cup K_2 \cup \{s\}$

$F = F_1 \cup F_2$

$\Delta = \Delta_1 \cup \Delta_2 \cup \{(s, e, s_1), (s, e, s_2)\}$.

(b.) statement:-

"Regular Language is closed under Concatenation operation"

proof:- Let $N_1 = (Q', \Sigma, \delta', q_0', F')$

$N_2 = (Q'', \Sigma, \delta'', q_0'', F'')$

be the NDFA that recognizes languages $L(N_1)$ & $L(N_2)$ respectively.

Let N be a NDFA

Where, $N = (Q, \Sigma, \delta, q_0, F)$

Such that,

$q_0 = q_0'$, $Q = q' \cup q''$, $F = F''$

$\delta = \delta' \cup \delta'' \cup \{\delta(q, e) = q_0'', q \in F'\}$

Then let $w = w_1 w_2$, $w_1 \in L(N_1)$ & $w_2 \in L(N_2)$

Then,

$(q_0, w) = (q_0', w)$

$(q_0, w_1 w_2) \vdash_N^* (q, w_2)$, $q \in F_1$

because,

$q_0 = q_0'$ and $(q_0', w_1) \vdash_N^* (q, e)$

but,

$(q, w_2) \vdash_N^* (q_0'', w_2)$, $q \in F_1$

because,

$\delta(q, e) = q_0''$ if $q \in F_1$

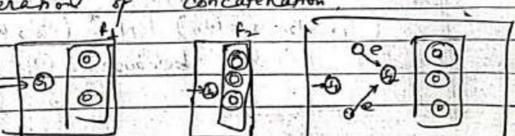
Then $(q_0'', w) \vdash_N^* (q, e)$, $q \in F$

because,

$(q_0'', w_2) \vdash_N^* (q, e)$, $q \in F''$

Thus, regular languages are closed under operation of concatenation.

Illustration



N_1 N_2 N

(c) Statement:

Class of regular language is closed under Kleene star operation.

proof:-

let $N_1 = (Q_1, \Sigma, \delta_1, q_0', F_1)$ be the NFA that recognizes the language $L(N_1)$

Construct N such that

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

Where,

$$Q = Q_1 \cup \{q_0\}$$

$$\delta = \delta_1 \cup \{\delta(q_0, e) = q_0', \delta(q, e) = q_0, q \in F_1\}$$

$$F = \{q_0\} \cup F_1$$

let, $\omega = \omega_1 \cdots \omega_{n-1}\omega_n$ where $\omega_i \in L(N_1)$

Then,

$$(q_0, \omega) \vdash_N^* (q_0', \omega); \quad \delta(q_0, e) = q_0'$$

$$(q_0, \omega_1, \omega_2, \cdots \omega_n) \vdash_N^* (q, \omega_2 \cdots \omega_n); \ q \in F$$

because,

$$(q_0', \omega_1) \vdash_N^* (q, e), \ q \in F; \omega_1 \in L(N_1)$$

$$(q_1, \omega_2 \cdots \omega_n) \vdash_N^* (q_0', \omega_2 \cdots \omega_n), \ q \in F$$

because $\delta(q, e) = q_0'$, $q \in F_1$

$$(q_0', \omega_2 \cdots \omega_n) \vdash_N^* (q, \omega_3, \omega_n), \ q \in F$$

because $(q_0', \omega_2) \vdash_N^* (q, e)$, $q \in F_1$, $\omega_2 \in L(N_1)$.

Similarly,

$$(q_0', \omega_n) \vdash_N^* (q, e), \ q \in F$$

i.e.

$$(q_0, \omega) \vdash_N^* (q, e), \ q \in F$$

Thus the class of regular language is closed u[nder] Kleene star operation.

(d) Statement:-

"Class of regular language is closed under Complementation."

proof:- let $N_1 = (Q_1, \Sigma, \delta_1, q_0', F_1)$ be a NFA and $L(N_1)$ be language recognized by N_1

Then, let us define another NFA as,

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

Such that,

$$Q = Q_1$$

$$\delta = \delta_1$$

$$q_0 = q_0'$$

$$F = Q_1 - F_1$$