THEORY OF COMPUTATION

Deals with Finite Automater / Finite State Machine

Source program

↓

Lexical Analyzer

↓ Tokens

Syntax Analyzer

6/1/2018

Central Concepts of Automater Theory

Alphabet : Finite, non-empty set of symbols.

"Σ" denotes set of symbols.

Eg: $\Sigma = \{0, 1\}$ → Binary alphabet

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

$\Sigma = \{a, b, c, ---, z\}$ → English alphabet

String: Finite sequence of symbols chosen from particular alphabet.

"w" denotes a string.

Eg: w = 01010

Scanned by CamScanner

Empty String: It is a string with zero occurances of symbols.

"ϵ" [Epsilon] denotes Empty string.

Length of the String : The no. of position for the symbols in a string.

"$|W|$" denotes Length of the string.

eg. $W = 01010$

$|W| = 5$

$\boxed{|\epsilon| = 0}$

Power of an Alphabet: If Σ is an alphabet, the set of string of certain length is represented by exponential notation. $\Sigma^k \rightarrow$ set of string of length k.

eg: If $\Sigma = \{0, 1\}$

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

$\Sigma^2 = \{00, 01, 10, 11\}$

$\Sigma^3 = \{000, 001, 010, 100, 110, 101, 011, 111\}$

$\Sigma^* \rightarrow$ set of all string of any length over Σ.

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots$

$\Sigma^* = \epsilon \cup \{0, 1\} \cup \{00, 01, 10, 11\} \cup \{000, \ldots, 111\} \cup \ldots$

$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots, 111, 0000, \ldots, 1111, \ldots\}$

$\Sigma^+ \rightarrow$ Excluding Empty string

$\boxed{\Sigma^+ = \Sigma^* - \{\epsilon\}}$

Scanned by CamScanner

Language: The set of string all of which are taken from Σ^* where Σ is particular alphabet.

eg: SJCE IS IN MYSURU

$\Sigma_4 \quad \Sigma_2 \quad \Sigma_2 \quad \Sigma_6$

$$\boxed{L \subseteq \Sigma^*}$$

* Acc. to Concatenation of String

$\epsilon W = W\epsilon = W$

$L = \{ W \in \{0,1\}^* \mid$ we have even no. of zero's $\}$

$\therefore L = \{001, 1111, 1010111\}$

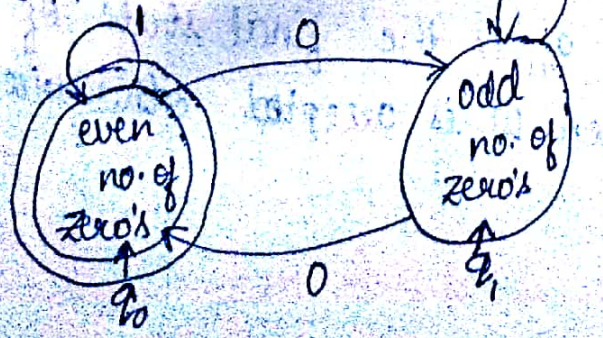
1) Finite Automater
   a) Deterministic FA (DFA)
   b) Non-deterministic FA (NFA)
   c) NFA with ϵ moves

2) Pushdown Automater
   a) Deterministic PDA
   b) PDA

3) Turing machine

$L = \{ W \in \{0,1\}^* \mid W$ has even no. of zero's $\}$

0110111 00



Scanned by CamScanner

I/P	O/P
q_0	$1 \to q_0$
q_0	$0 \to q_1$
q_1	$1 \to q_1$
q_1	$0 \to q_0$

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

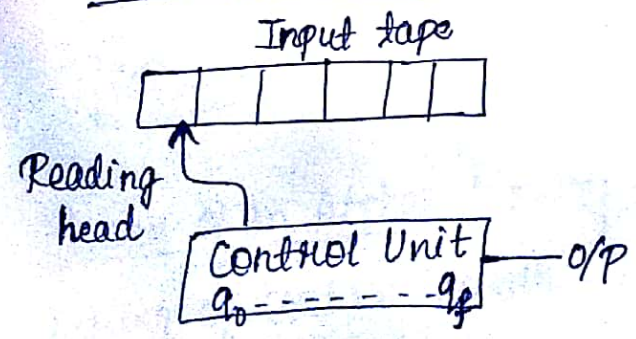
Alphabet $\Sigma = \{0, 1\}$

Transition $\delta : \underbrace{Q \times \Sigma}_{\text{Cartesian Product}} \to Q$

Start state : q_0

Final state : q_0

Finite Automaten



Once when all symbols are read, at that time if Control Unit is at Final state, then string is accepted, otherwise rejected.

Deterministic Finite Automaten

It is defined with 5 tuples $A = (Q, \Sigma, \delta, q_0, F)$

where, $q_0 \to$ Initial state, $q_0 \in Q$

$F \to$ Set of final states, $F \subseteq Q$

$Q \to$ Set of finite states

$\Sigma \to$ Finite set of input symbols

$\delta \to$ Transition function. $\delta : Q \times \Sigma \to Q$

$W = a_1, a_2, \ldots, a_n$

$\delta(q_0, a_1) = q_1$

$\delta(q_1, a_2) = q_2$

$\delta(q_2, a_3) = q_3$

$\vdots$

$\delta(q_{n-1}, a_n) = q_n$

If q_n is the final state of DFA, then W is accepted otherwise rejected.

Scanned by CamScanner

Eg: $A = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, \{q_0\}, \{q_2\})$

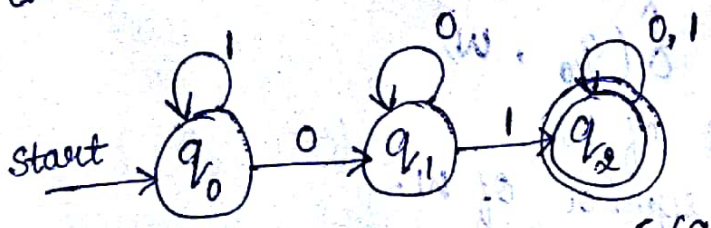
where δ is as follows,

$\delta(q_0, 0) = q_1$ $\quad$ $\delta(q_1, 0) = q_1$ $\quad$ $\delta(q_2, 0) = q_2$

$\delta(q_0, 1) = q_0$ $\quad$ $\delta(q_1, 1) = q_2$ $\quad$ $\delta(q_2, 1) = q_2$

Notations: Transition diagram → It is a graph

* Create a Node for every state in Q.



If $q \in Q$, $a \in \Sigma$ then $\delta(q, a) = p$

Transition table → Tabular representation of δ function.

δ	0	1
→ q_0	q_1	q_0
q_1	q_1	q_2
* q_2	q_2	q_2

1) * indicates the final state.

2) → indicates the start.

It accepts only the language consisting of the substring '01'.

General form.

δ	a_1	a_2	---	a_n
q_0	-	-	-	-
q_1	-	-	-	-
⋮	⋮	⋮	⋮	⋮
q_n	-	-	-	-

Scanned by CamScanner

Extending transition function to string ($\hat{\delta}$)

It describes what happen when we start any state and follow any input symbol.

$\hat{\delta}$ is constructed by δ.

* Define $\hat{\delta}$ on induction of the length of input string.

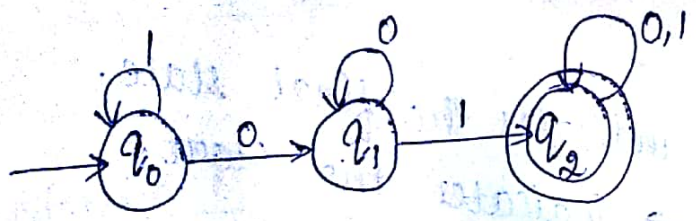
Basis: $\hat{\delta}(q, \epsilon) = q$

Induction: $W = xa$ $\quad \hat{\delta}(q_0, W)$

$a \rightarrow$ last i/p symbol of W.

$x \rightarrow$ remaining i/p symbol of W.

$$\boxed{\hat{\delta}(q_0, W) = \delta(\hat{\delta}(q, x), a)}$$



$W = 101$ $\quad \hat{\delta}(q_0, 101) = ?$

$\hat{\delta}(q_0, \epsilon) = q_0$

$\hat{\delta}(q_0, 1) = \delta(\hat{\delta}(q_0, \epsilon), 1)$

$= \delta(q_0, 1)$

$= q_0$

$\hat{\delta}(q_0, 10) = \delta(\hat{\delta}(q_0, 1), 0)$

$= \delta(q_0, 0)$

$= q_1$

$\hat{\delta}(q_0, 101) = \delta(\hat{\delta}(q_0, 10), 1)$

$= \delta(q_1, 1)$

$= q_2 \in F$ $\quad$ Accepted string 101.

Scanned by CamScanner

$W = 100$, $\hat{\delta}(q_0, 100) = ?$

$\hat{\delta}(q_0, \epsilon) = q_0$

$\hat{\delta}(q_0, 1) = \delta(\hat{\delta}(q_0, \epsilon), 1)$

$= \delta(q_0, 1)$

$= q_0$

$\hat{\delta}(q_0, 10) = \delta(\hat{\delta}(q_0, 1), 0)$

$= \delta(q_0, 0)$

$= q_1$

$\hat{\delta}(q_0, 100) = \delta(\hat{\delta}(q_0, 10), 0)$

$= \delta(q_1, 0)$

$= q_1$ Rejected

Language of DFA

Set of strings accepted by DFA.

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA, the language of DFA is defined as $L(A)$.

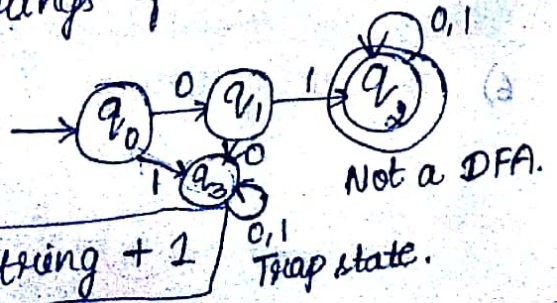
$L(A) = \{W \mid \hat{\delta}(q_0, W) \text{ is in } F\}$

i.e., Set of all string from start state to one of the accepting state.

If 'L' is $L(A)$ for some DFA, L is called as Regular Language.

Problems

1) Design DFA to accept set of strings of 0's & 1's ~~ending~~ starting with 01.



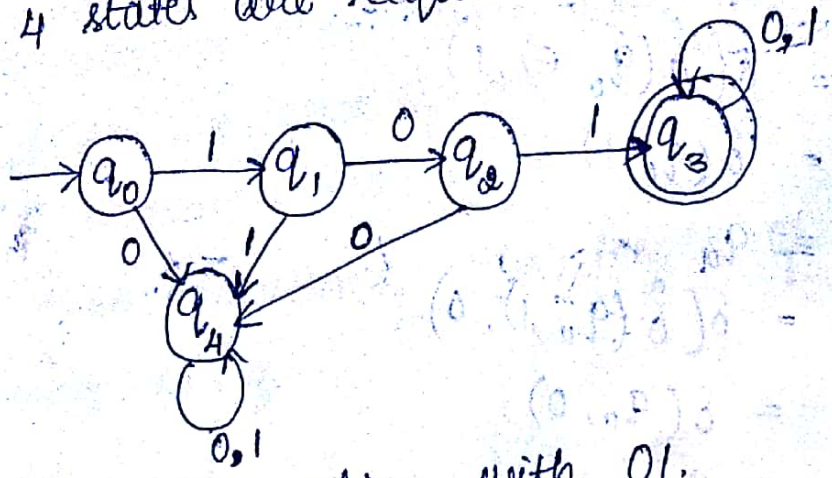
Not a DFA.

WKT, $A = (Q, \Sigma, \delta, q_0, F)$

Sets required are = Min. length of string + 1

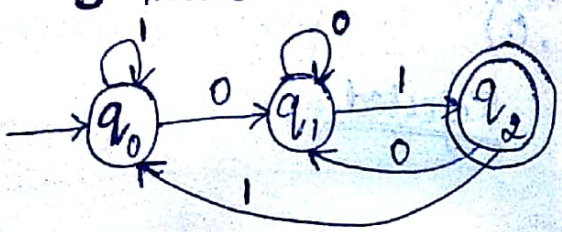
2) Define a DFA starting with 101.

4 states are required.



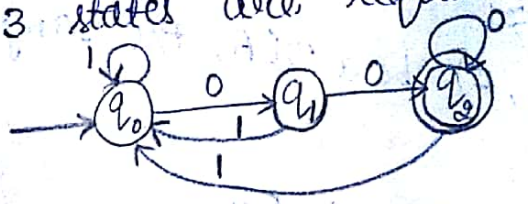
3) Define a DFA ending with 01.

3 states are required.



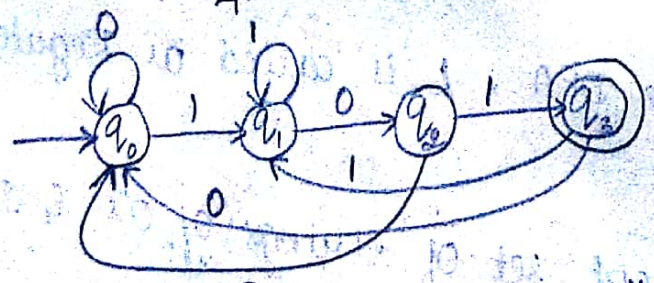
4) Ending with 00.

3 states are required

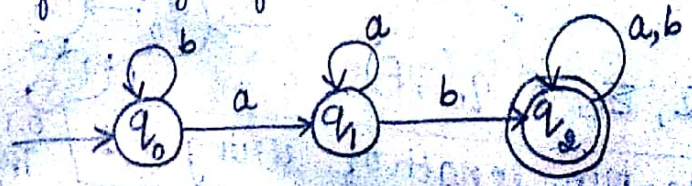


5) Ends with 101

4 states are required.

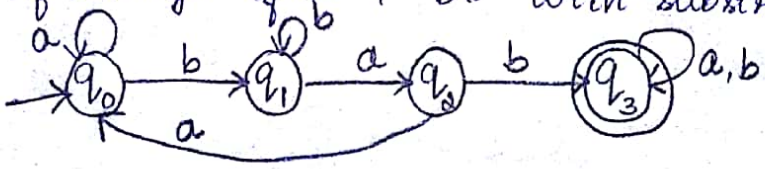


6) The set of strings of a's & b's with a substring ab.



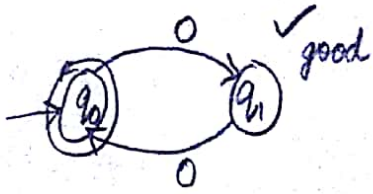
Scanned by CamScanner

7) The set of strings of a's & b's with substring bab.

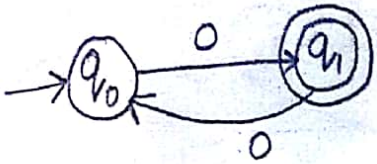


8) Set of string of even length zero's are $\Sigma = \{0\}$

(or)



9) Set of string of odd length zero's are $\Sigma = \{0\}$



10) Set of strings of 0's and 1's with atleast one zero.

Problems

1) Design DFA for the following Languages.

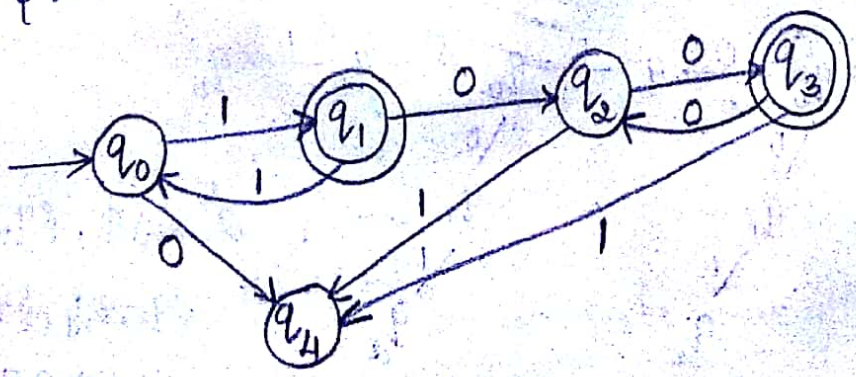
a) $L = \{ W \mid W$ has odd no. of 1's followed by even no. of 0's $\}$

b) $L = \{ W : |W| \bmod 3 = 0 ,\ W \in \{a,b\}^* \}$

c) Set of strings of a's & b's doesnot ends with string "abb".

d) $L = \{ W \mid W$ has even no. of a's and even no. of b's $\}$

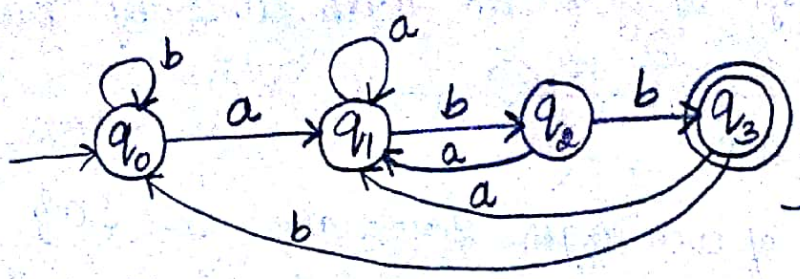
a)



111
1110000

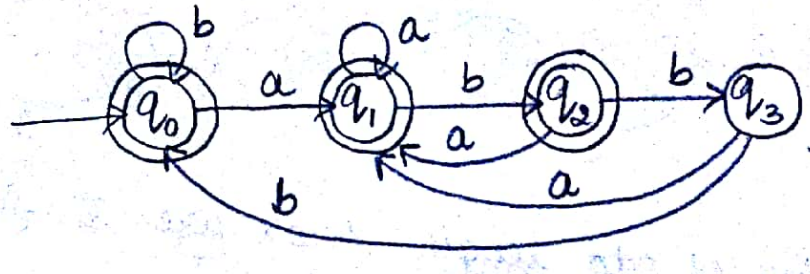
b)

c)



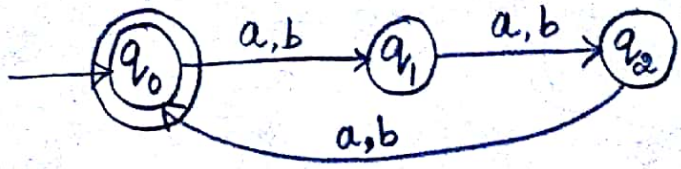
a, b, ab, aa

→ Ends with abb



→ Doesnot ends with abb

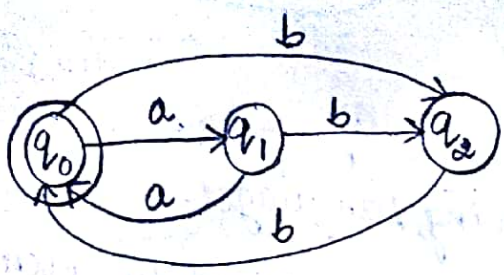
b)



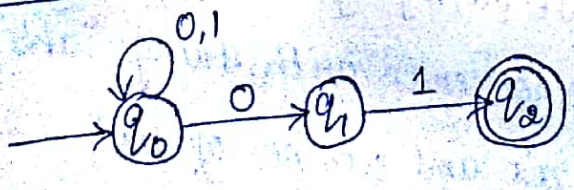
aaa, aba bbb

d)

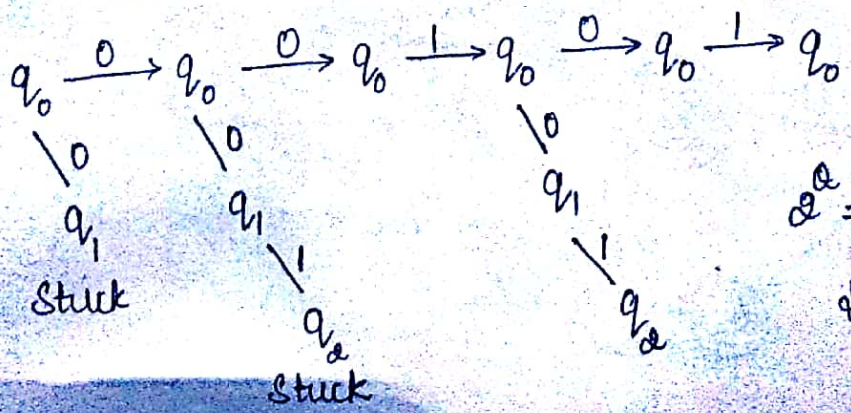
aa, bb, aabb, abb



Non-Deterministic Finite Automaton



00101



$2^Q = \{\phi, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_1, q_2\}, \{q_0, q_2\}, \{q_0, q_1, q_2\}\}$

$Q = 3$

$2^Q = 2^3 = 8$

Scanned by CamScanner

Definition

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

where, $Q \rightarrow$ Finite set of states

$\Sigma \rightarrow$ Finite set of input symbols

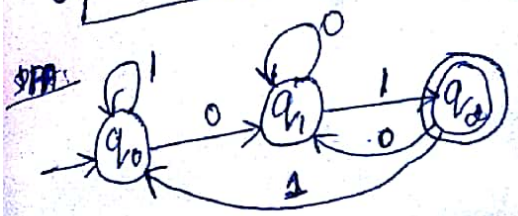
$\delta \rightarrow$ Transition function

$q_0 \rightarrow$ Initial state, $q_0 \in Q$

$F \rightarrow$ Set of final states, $F \subseteq Q$

$\delta : \boxed{Q \times \Sigma = 2^Q}$

Eg:



00101

$q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_1 \xrightarrow{1} q_2$

* DFA is more efficient than NFA.
* More options are there in NFA

Extending Transition function : $\hat{\delta}$

$\hat{\delta}$ is defined by δ – induction on the length of the input string.

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

Induction: $w = xa$

$$\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), a)$$

$$\hat{\delta}(q, x) = \{P_1, P_2, \ldots, P_n\}$$

$$\hat{\delta}(q, w) = \bigcup_{i=1}^{w} \delta(P_i, a) = \{\phi, r_1, \ldots, r_j\}$$

$$\hat{\delta}(q, w) = \delta(P_1 a) \cup \delta(P_2 a) \cup \ldots \delta(P_n \cup a)$$

Scanned by CamScanner

For $w = 001$

$\delta(q_0, \epsilon) = q_0$

$\hat{\delta}(q_0, \epsilon) = \delta(\hat{\delta}(q_0, \epsilon), 0)$

$= \delta(q_0, 0) = \{q_0, q_1\}$

$\hat{\delta}(q_0, 00) = \delta(\hat{\delta}(q_0, 0), 0)$

$= \delta((q_0, q_1), 0)$

$= \delta(q_0, 0) \cup \delta(q_1, 0)$

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

$= \{q_0, q_1\}$

$\hat{\delta}(q_0, 001) = \delta(\hat{\delta}(q_0, 00), 1)$

$= \delta((q_0, q_1), 1)$

$= \delta(q_0, 1) \cup \delta(q_1, 1)$

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

$= \{q_0, q_2\}$

If this contains one of the state, then this is accepted as a string.

<u>Language of NFA</u> <u>Regular Language</u>

Let $N = \{Q, \Sigma, \delta, q_0, F\}$ be a NFA then language of NFA defined as follows.

$L(N) = \{w \mid \hat{\delta}(m, w) \cap F \neq \phi\}$

Scanned by CamScanner

Equivalence of DFA and NFA [$L(N) = L(D)$]

Subset Construction Algorithm :-

Input : $N = (Q_N, \Sigma, \delta_N, q_N, F_N)$

Output : Equivalent $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$

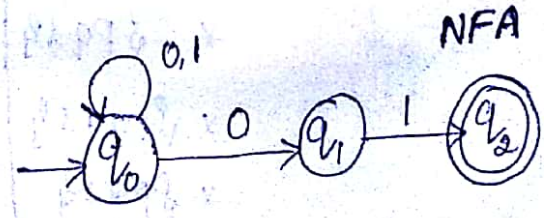
such that, $L(N) = L(D)$

Method:

* Σ is same for both the Automaton.
* q_D – a state, is set containing only the initial state of N.
* Q_D – Subset / Powerset of Q_N.
  Consider only the state, which are reachable from the initial state.
* F_D – Set of subset of S of Q_N, such that $S \cap F_N \neq \phi$
* δ_D – $S \subseteq Q_N$, $a \in \Sigma$

$$\delta_D(S, a) = \cup\, \delta_N(P, a), \quad P \text{ is in } S.$$

δ_N	0	1
q_0	$\{q_0 q_1\}$	$\{q_0\}$
q_1	ϕ	$\{q_2\}$
* q_2	ϕ	ϕ

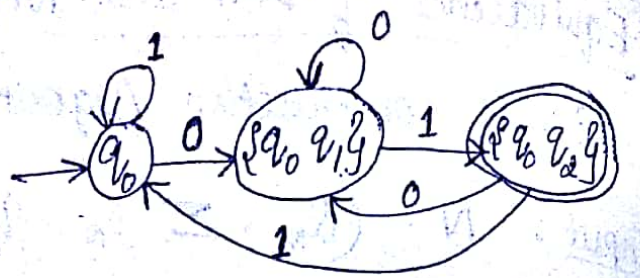


$$\boxed{\begin{aligned} \delta_D(\{q_0 q_1\} 0) &= \delta_N(q_0, 0) \cup \delta_N(q_1, 0) \\ &= \{q_0 q_1\} \cup \{\phi\} \\ &= \{q_0 q_1\} \end{aligned}}$$

δ_D	0	1
ϕ	ϕ	ϕ
→ $\{q_0\}$	$\{q_0 q_1\}$	$\{q_0\}$
$\{q_1\}$	ϕ	$\{q_2\}$
* $\{q_2\}$	ϕ	ϕ
→ $\{q_0 q_1\}$	$\{q_0 q_1\}$	$\{q_0 q_2\}$
$\{q_0 q_2\}$	$\{q_0 q_1\}$	$\{q_0\}$
$\{q_1 q_2\}$	ϕ	$\{q_2\}$
$\{q_0 q_1 q_2\}$	$\{q_0 q_1\}$	$\{q_0 q_2\}$

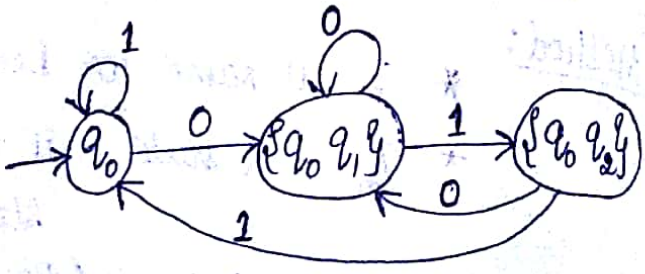
Scanned by CamScanner

δ_D	0	1
$\rightarrow \{q_0\}$	$\{q_0 q_1\}$	$\{q_0\}$
$\{q_0 q_1\}$	$\{q_0 q_1\}$	$\{q_0 q_2\}$
* $\{q_0 q_2\}$	$\{q_0 q_1\}$	$\{q_0\}$



Lazy Method

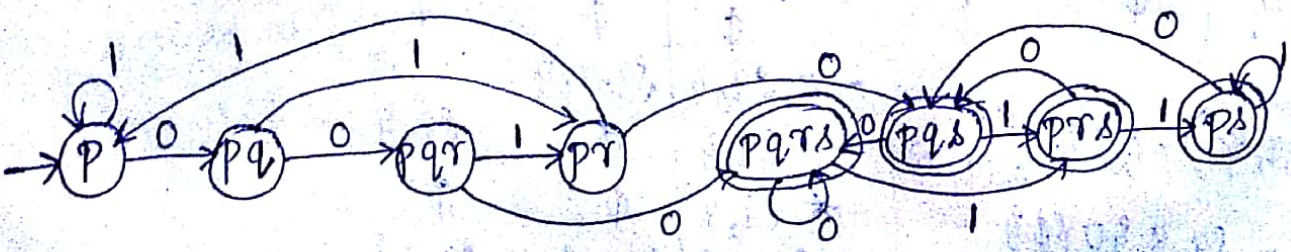
δ_D	0	1
$\rightarrow \{q_0\}$	$\{q_0 q_1\}$	$\{q_0\}$
$\{q_0 q_1\}$	$\{q_0 q_1\}$	$\{q_0 q_2\}$
$\{q_0 q_2\}$	$\{q_0 q_1\}$	$\{q_0\}$



*

	0	1
$\rightarrow p$	$\{pq\}$	$\{p\}$
q	$\{r\}$	$\{r\}$
r	$\{s\}$	ϕ
* s	$\{s\}$	$\{s\}$

	0	1
$\rightarrow p$	$\{pq\}$	$\{p\}$
$\{pq\}$	$\{pqr\}$	$\{pr\}$
$\{pqr\}$	$\{pqrs\}$	$\{pr\}$
$\{pr\}$	$\{pqs\}$	$\{p\}$
* $\{pqs\}$	$\{pqrs\}$	$\{prs\}$
* $\{pqrs\}$	$\{pqrs\}$	$\{prs\}$
* $\{prs\}$	$\{pqs\}$	$\{ps\}$
* $\{ps\}$	$\{pqs\}$	$\{ps\}$

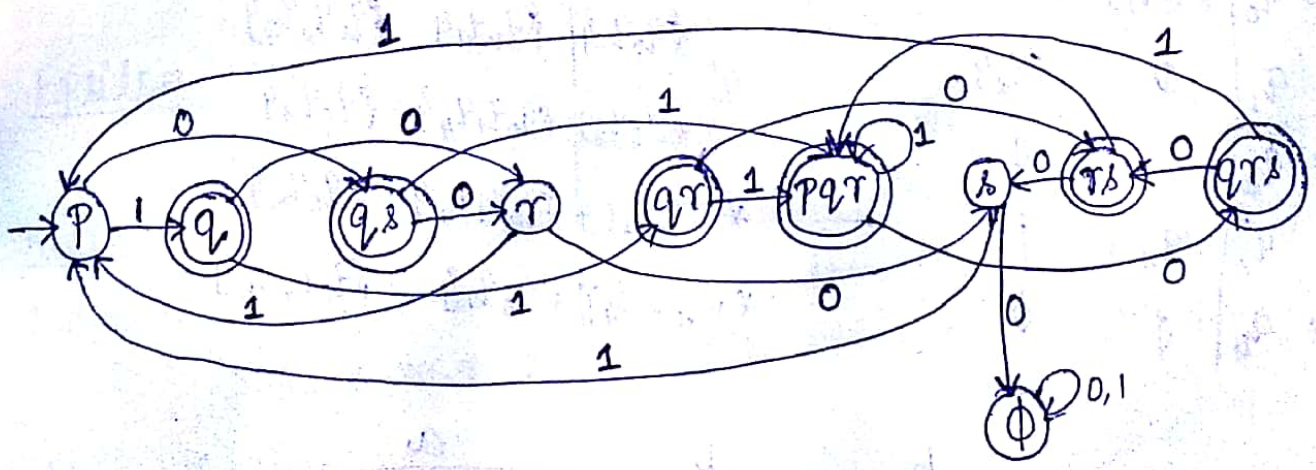


Scanned by CamScanner

19/1/2018

*	0	1
→ p	{q,s}	{q}
*q	{r}	{q,r}
r	{s}	{p}
*s	ϕ	{p}

	0	1
→ {p}	{q,s}	{q}
* {q}	{r}	{q,r}
* {q,s}	{r}	{p,q,r}
{r}	{s}	{p}
* {q,r}	{r,s}	{p,q,r}
* {p,q,r}	{q,r,s}	{p,q,r}
* {s}	ϕ	{p}
* {r,s}	{s}	{p}
* {q,r,s}	{r,s}	{p,q,r}



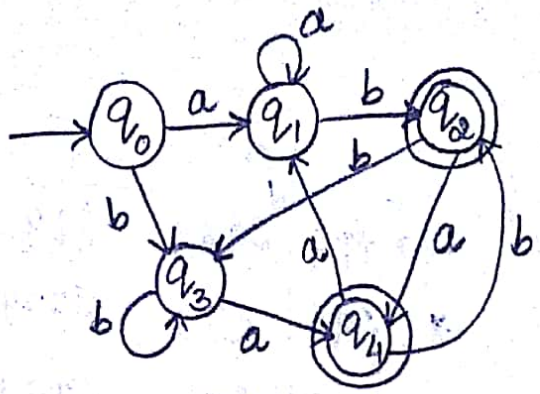
NFA

Problems

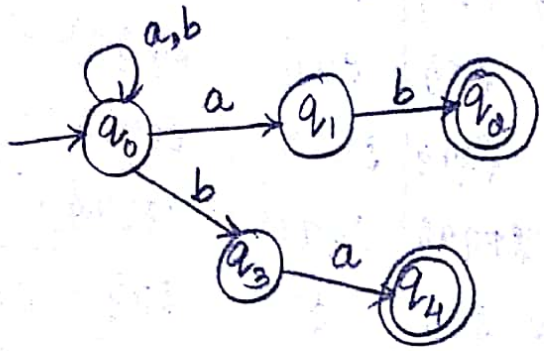
Q) Create DFA to accept the following :-

1) $L = \{ w(ab+ba) \mid w \in \{a,b\}^* \}$

Scanned by CamScanner

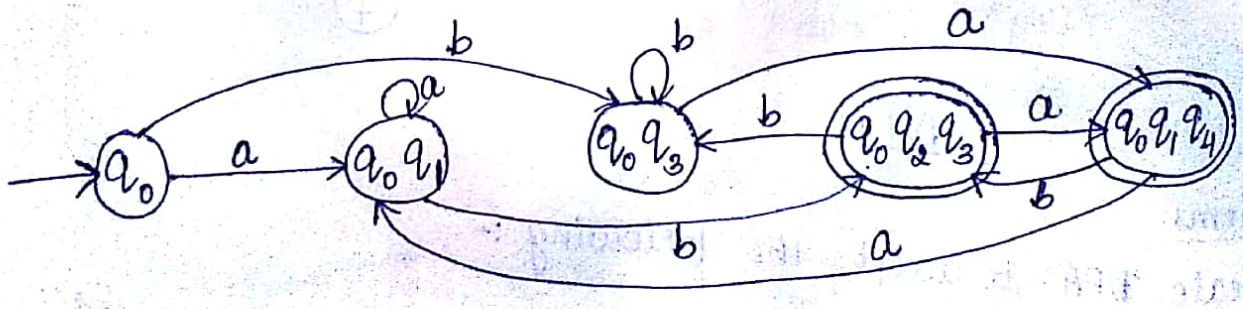


NFA:-

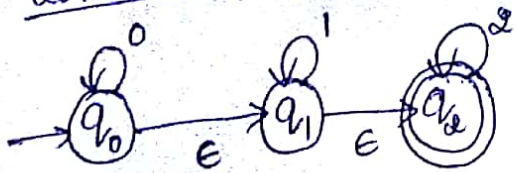


	a	b
→ q_0	$\{q_0 q_1\}$	$\{q_0 q_3\}$
q_1	ϕ	$\{q_2\}$
* q_2	ϕ	ϕ
q_3	q_4	ϕ
* q_4	ϕ	ϕ

	a	b
→ $\{q_0\}$	$\{q_0 q_1\}$	$\{q_0 q_3\}$
$\{q_0 q_1\}$	$\{q_0 q_1\}$	$\{q_0 q_2 q_3\}$
$\{q_0 q_3\}$	$\{q_0 q_1 q_4\}$	$\{q_0 q_3\}$
* $\{q_0 q_2 q_3\}$	$\{q_0 q_1 q_4\}$	$\{q_0 q_3\}$
* $\{q_0 q_1 q_4\}$	$\{q_0 q_1\}$	$\{q_0 q_2 q_3\}$



20/1/2018



→ This is zero or more no. of 0's followed by zero or more 1's followed by 0's or 2's.

Scanned by CamScanner

NFA with ϵ moves: (ϵ-NFA)

Definition: $E = \{Q, \Sigma, \delta, q_0, F\}$

where, $Q \rightarrow$ Finite set of states

$\Sigma \rightarrow$ Finite set of input symbol

$\delta \rightarrow$ Transition function, $Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$

$q_0 \rightarrow$ initial states, $q_0 \in Q$

$F \rightarrow$ Set of final states, $F \subseteq Q$

From the above example:-

δ_E	ϵ	0	1	2
$\rightarrow q_0$	$\{q_1\}$	$\{q_0\}$	ϕ	ϕ
q_1	$\{q_2\}$	ϕ	$\{q_1\}$	ϕ
$* q_2$	ϕ	ϕ	ϕ	$\{q_2\}$

Epsilon Closures:- [E-close (q)]

Epsilon closures is defined recursively as follows.

Basis:- q is in E-close (q)

Induction:- If p is in E-close (q), then there is a transition from p to r on ϵ, then add r to E-close (q).

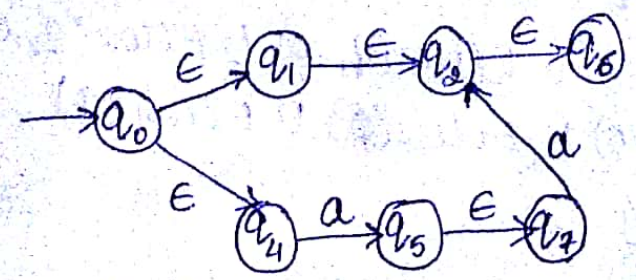
From the Eg:

E-Close $(q_0) = \{q_0, q_1, q_2\}$

E-Close $(q_1) = \{q_1, q_2\}$

E-Close $(q_2) = \{q_2\}$

Eg 2:



From the figure,

Scanned by CamScanner

$E\text{-close}(q_0) = \{q_0, q_1, q_2, q_4\}$

$E\text{-close}(q_1) = \{q_1, q_2, q_6\}$

$E\text{-Close}(q_2) = \{q_2, q_6\}$

$E\text{-Close}(q_4) = \{q_4\}$

$E\text{-Close}(q_5) = \{q_5, q_7\}$

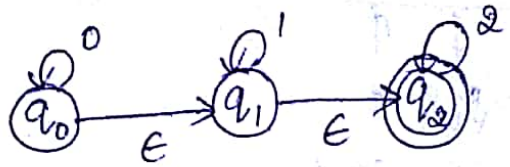
$E\text{-Close}(q_6) = \{q_6\}$

$E\text{-Close}(q_7) = \{q_7\}$

$\hat{\delta}$ - Extended transition function for ϵ - NFA

Basis: $\hat{\delta}(q, \epsilon) = E\text{-Close}(q)$

Induction:



$W = xa \qquad \hat{\delta}(q, w)$

$\hat{\delta}(q, x) = \{P_1, P_2, ---, P_k\}$

$\hat{\delta}(q, w) = \bigcup_{i=1}^{k} \delta(P_i, a) = \{r_1\ r_2 --- r_m\}$

$\hat{\delta}(q, w) = \bigcup_{i=1}^{m} E\ Close(r_i) = \{q_0\ q_1 --- q_n\}$

Let $w = 012$, $x = 01$, $a = 2$

$\hat{\delta}(q, \epsilon) = E\text{-Close}(q_0) = \{q_0\ q_1\ q_2\}$

$\hat{\delta}(q_0, 0) = \delta(\hat{\delta}(q_0, \epsilon), 0)$

$= \delta\{q_0\ q_1\ q_2, 0\}$

$= E\ Close(q_0, 0) \cup E\ Close(q_1, 0) \cup E\ Close(q_2, 0)$

$= E\ Close(q_0, 0) \cup E\ Close(\phi) \cup E\ Close(\phi)$

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

Scanned by CamScanner

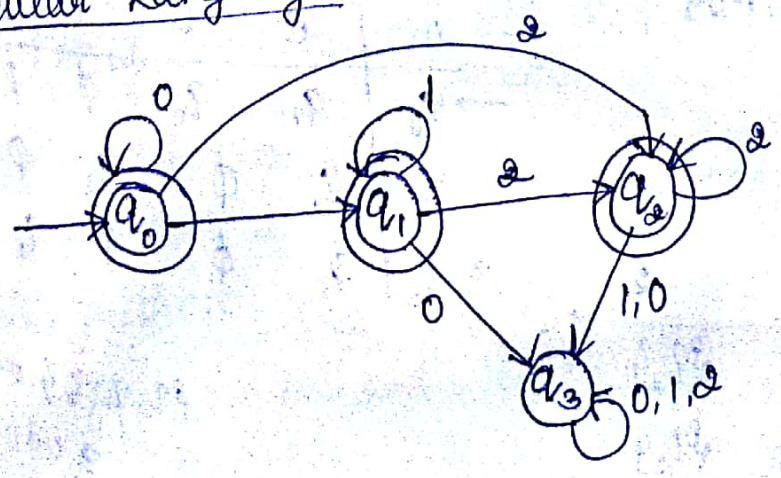
$$\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1)$$
$$= \delta(q_0 q_1 q_2, 1)$$
$$= E\,Close(q_0, 1) \cup E\,Close(q_1, 1) \cup E\,Close(q_2, 1)$$
$$= E\,Close(\phi) \cup E\,Close(q_1, 1) \cup E\,Close(\phi)$$
$$= \{q_1\ q_2\}$$

$$\hat{\delta}(q_0, 012\} = \delta(\hat{\delta}(q_0, 01), 2)$$
$$= \delta(q_1 q_2, 2)$$
$$= E\,Close(q_1, 2) \cup E\,Close(q_2, 2)$$
$$= E\,Close(\phi) \cup E\,Close(q_2, 2)$$
$$= \{q_2\}$$

Language of ε NFA

$$E = (Q, \Sigma, \delta, q_0, F)$$
$$L(E) = \{w \mid \hat{\delta}(q_0, w) \cap F \neq Q\}$$

Regular Language:



* After 1→0 shd not come
* After 2→0 & 1 shd not come.

So, unlike other cases, we cannot put the arrow to q_0 & q_1 for 0 and 1 cases for q_2.

Elimination of ε-tranistion

Input: $E = (Q_E, \Sigma, \delta_E, q_E, F_E)$

Output: Equivalent DFA, $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$

Method: * Σ same for both automator.

Scanned by CamScanner

* Q_D - set of subset of Q_E, only the accessible state are E close of subset Q_E.

* q_D - E Close (q_E)
* F_D - $\{S \mid S \in Q_D,\ S \cap F_E \neq \phi\}$
* $\delta_D(S, a)$, $S \subseteq Q_D$, $a \in \Sigma$

$$S = P_1\ P_2 \cdots P_n$$

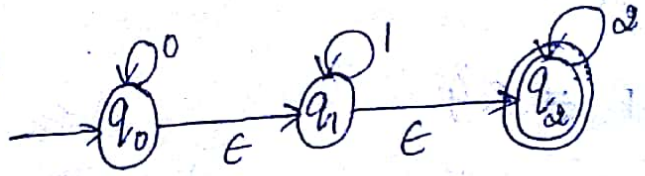
$$\delta_D(S, a) = \bigcup_{i=1}^{n} \delta_E(P_i, a)$$

$$= r_1\ r_2 \cdots r_m$$

$$\delta_D(S, a) = \bigcup_{j=1}^{m} \text{E Close}(r_j)$$

$$= q_1\ q_2 \cdots q_n$$

Eg: 1)



E Close (q_0) = $\{q_0, q_1, q_2\}$

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

E Close (q_2) = $\{q_2\}$

δ	ϵ	0	1	2
$\rightarrow q_0$	q_1	q_0	ϕ	ϕ
q_1	q_2	ϕ	q_1	ϕ
$* q_2$	ϕ	ϕ	ϕ	q_2

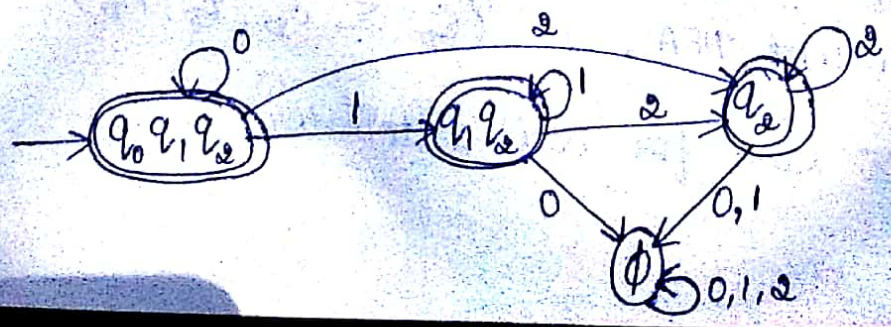
δ_D	0	1	2
$\rightarrow * \{q_0 q_1 q_2\}$	$\{q_0 q_1 q_2\}$	$\{q_1 q_2\}$	$\{q_2\}$
$* \{q_1 q_2\}$	ϕ	$\{q_1 q_2\}$	$\{q_2\}$
$* \{q_2\}$	ϕ	ϕ	$\{q_2\}$

$\because S \cap F_E$

$\{q_0\ q_1\ q_2\}$ on 0

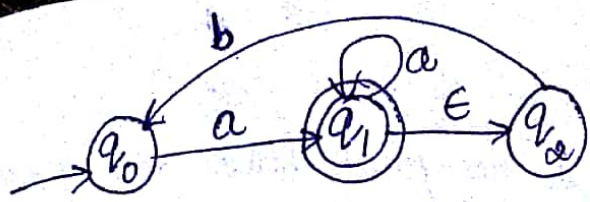
$q_0 \cup \phi \cup \phi = q_0$

E Close $(q_0) = \{q_0 q_1 q_2\}$



Scanned by CamScanner

2)



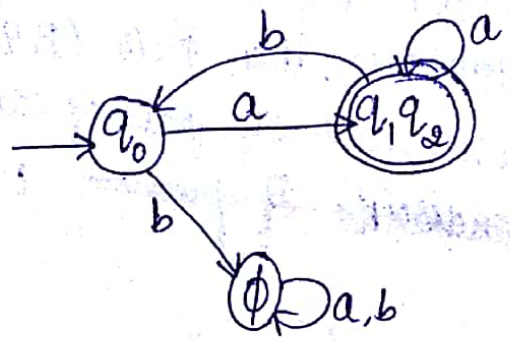
δ	ϵ	a	b
$\rightarrow q_0$	ϕ	q_1	ϕ
$* q_1$	q_2	q_1	ϕ
q_2	ϕ	ϕ	q_0

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

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

$E\,close(q_2) = \{q_2\}$

δ_D	a	b
$\rightarrow \{q_0\}$	$\{q_1 q_2\}$	ϕ
$* \{q_1 q_2\}$	$\{q_1, q_2\}$	$\{q_0\}$



<u>Indistinguishable State</u> (equivalent)

Let p and q ~~states of DFA~~ are said to be indistinguishable, iff

$\hat{\delta}(p, w) \in F$ implies $\hat{\delta}(q, w) \in F$

and

$\hat{\delta}(p, w) \notin F$ implies $\hat{\delta}(q, w) \notin F$, for all $w \in \Sigma^*$

<u>Distinguishable State</u>

Let p and q are the states of DFA are said to be distinguishable, iff

$\hat{\delta}(p, w) \in F$ and $\hat{\delta}(q, w) \notin F$

(or)

$\hat{\delta}(p, w) \notin F$ and $\hat{\delta}(q, w) \in F$, for atleast one $w \in \Sigma^*$

By Using Table Filling Algorithm, easily no. of states can be decreamented.

Scanned by CamScanner

Table filling Algorithm

Basis: If p is final state and q is non-final state, the pair (p,q) is distinguishable.

Induction: Let p and $q \in Q$, $a \in \Sigma$.

$r = \delta(p,a)$ and $s = \delta(q,a)$. (r,s) is known to be as distinguishable, then pair (p,q) is distinguishable.

X → Distinguishable pair of states

✓ → Undistinguishable pair of states.

1)

	0	1
→ A	B	A
B	A	C
C	D	B
* D	D	A
E	D	F
F	G	E
G	F	G

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

Equivalent pair of states,

(A,G), (BF), (CE)

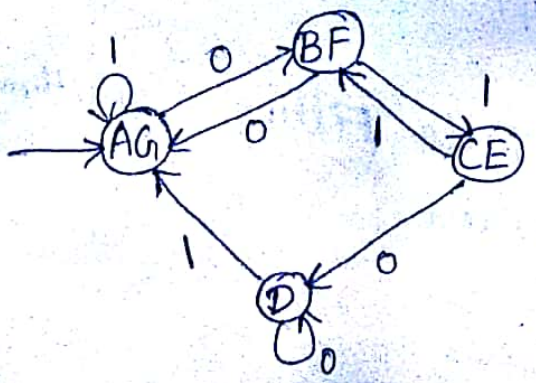
Consider state of the given DFA

$\check{A}, \check{B}, \check{C}, D, \check{E}, \check{F}$

⇓

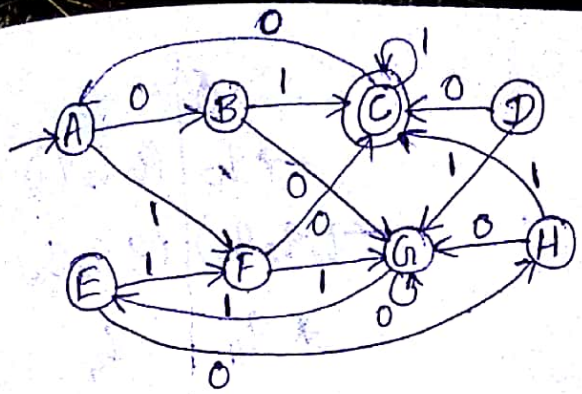
(AG), (BF), (CE), D → The states of minimized DFA.

δ	0	1
→ (AG)	(BF)	(AG)
(BF)	(AG)	(CE)
(CE)	D	(BF)
* D	D	(AG)



Scanned by CamScanner

2)

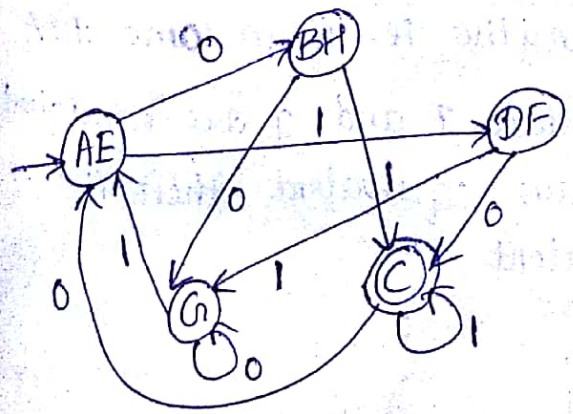


δ	0	1
→ A	B	F
B	G	C
* C	A	C
D	C	G
E	H	F
F	C	G
G	G	E
H	G	C

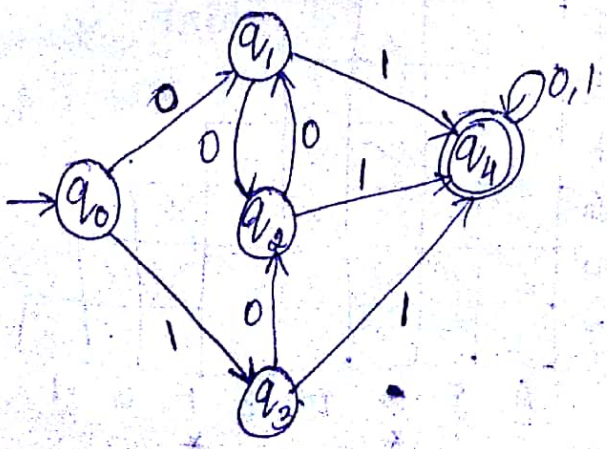
	A	B	C *	D	E	F	G
B	X						
* C	X	X					
D	X	X	X				
E	✓	X	X	X			
F	X	X	X	✓	X		
G	X	X	X	X	X	X	
H	X	✓	X	X	X	X	X

(AE), (BH), (DF), C, G

δ	0	1
→ (AE)	(BH)	(DF)
(BH)	G	C
(DF)	C	G
* C	AE	C
G	G	AE



3)



δ	0	1
→ q_0	q_1	q_3
q_1	q_2	q_4
q_2	q_1	q_4
q_3	q_2	q_4
* q_4	q_4	q_4

Scanned by CamScanner

	q_0	q_1	q_2	q_3
q_1	X			
q_2	X	✓		
q_3	X	✓	✓	
* q_4	X	X	X	X

δ	0	1
→ q_0	$(q_1 q_2 q_3)$	$(q_1 q_2 q_3)$
$(q_1 q_2 q_3)$	$(q_1 q_2 q_3)$	q_4
* q_4	q_4	q_4

q_0, q_1, q_2, q_3, $*q_4$

$(q_1 q_3)$, $(q_1 q_2)$, (q_2, q_3), q_0, q_4

Booz of its transitive property, we can combine $(q_1 q_3)$, $(q_1 q_2)$, $(q_2 q_3)$ as $(q_1 q_2 q_3)$

* <u>Theorem:</u>

The equivalent of states is transitive ie., if in some DFA $A = (Q, \Sigma, \delta, q_0, F)$.. we find that states p and q are equivalent and we also find that q and r are equivalent, then it must be that p and r are equivalent.

δ	0	1
A	B	E
B	C	F
* C	D	H
D	E	H
E	F	I
* F	G	B
G	H	B
H	I	C
* I	A	E

	A	B	C	D	E	F	G	H
B	X							
* C	X	X						
D	✓	X	X					
E	X	✓	X	X				
* F	X	X	✓	X	X			
G	✓	X	X	✓	X	X		
H	X	✓	X	X	✓	X	X	
* I	X	X	✓	X	X	✓	X	X

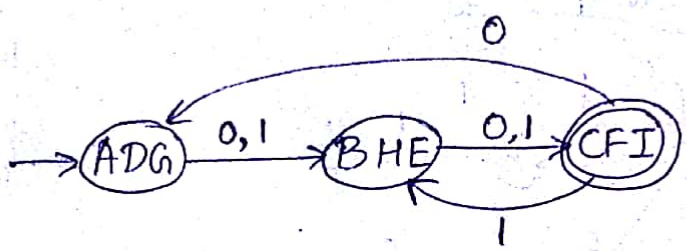
* *

Scanned by CamScanner

(AD), (BH), (DG), (B,E), (CF), (EH), (FI), (AG), (CI)

(ADG), (BHE), (CFI)

δ	0	1
→(ADG)	(BHE)	(BHE)
(BHE)	(CFI)	(CFI)
*(CFI)	(ADG)	(BHE)



Test Pattern

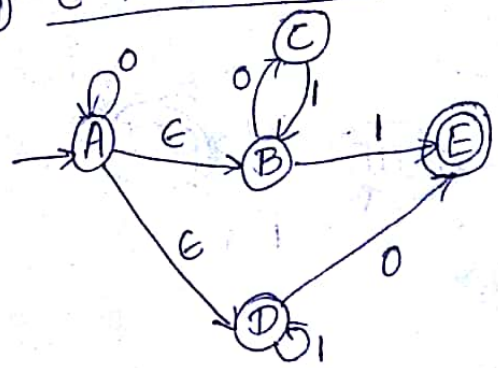
1) Define following terms with an example.
(i) Alphabet (ii) String (iii) Power of Alphabet
(iv) Language (v) DFA (vi) NFA (vii) ε-NFA
(vii) Extend transition function of DFA, NFA, ε-NFA
(ix) ε Closure (Basis & Induction)
(x) Distinguishable & Indistinguishable states.

2) Design DFA. Compute $\hat{\delta}(\ )$
3) Design NFA
4) Write Subset Construction Algorithm. Conversion from NFA to DFA.
5) Write algorithm to eliminate ε-transition and Convert ε-NFA to DFA.
6) Write Table filling Algorithm (Basis & Induction).
Minimize following DFA.

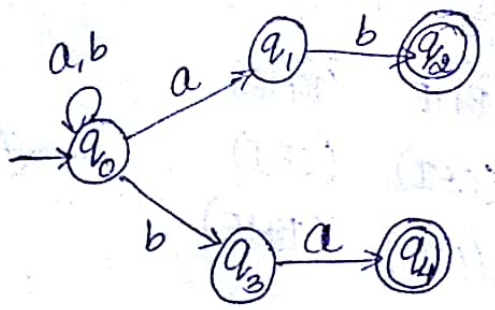
Scanned by CamScanner

Problems

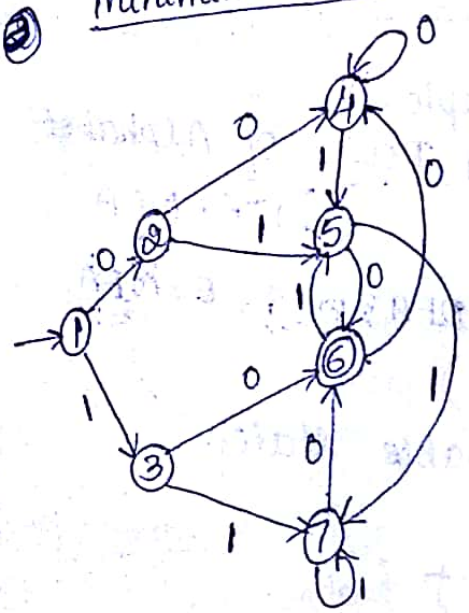
① ∈-NFA to DFA



② NFA to DFA



③ Minimize DFA



④ Design DFA to accept all string that contain atleast one of the two substring aba and bba over $\Sigma = \{a, b\}^*$

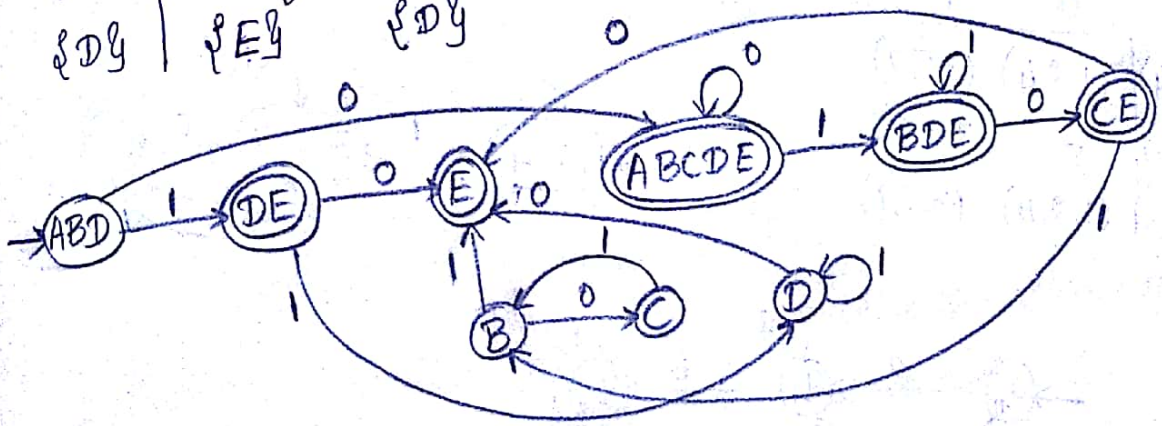
Answers

① E-Close (A) = {A, B, D}
E-Close (B) = {B}
E-Close (C) = {C}
E-Close (D) = {D}
E-Close (E) = {E}

δ	∈	0	1
→ A	{BD}	A	ϕ
B	ϕ	C	E
C	ϕ	ϕ	B
D	ϕ	E	D
* E	ϕ	ϕ	ϕ

Scanned by CamScanner

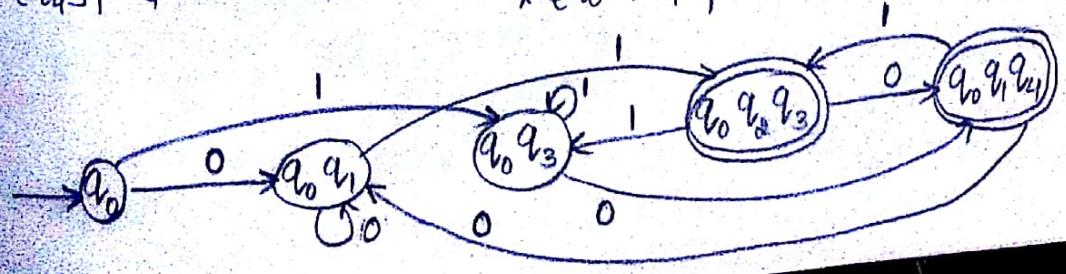
δ_D	0	1
→ {ABD}	{ABCDE}	{DE}
* {DE}	{E}	{D}
* {E}	ϕ	ϕ
* {ABCDE}	{ABCDE}	{BDE}
* {BDE}	{CE}	{BDE}
* {CE}	{E}	{B}
{B}	{C}	{E}
{C}	ϕ	{B}
{D}	{E}	{D}



(2)

δ_N	a	b
→ $\{q_0\}$	$\{q_0 q_1\}$	$\{q_0 q_3\}$
$\{q_1\}$	ϕ	$\{q_2\}$
* $\{q_2\}$	ϕ	ϕ
$\{q_3\}$	$\{q_4\}$	ϕ
* $\{q_4\}$	ϕ	ϕ

δ_D	a	b
→ $\{q_0\}$	$\{q_0 q_1\}$	$\{q_0 q_3\}$
$\{q_0 q_1\}$	$\{q_0 q_1\}$	$\{q_0 q_2 q_3\}$
$\{q_0 q_3\}$	$\{q_0 q_1 q_4\}$	$\{q_0 q_3\}$
* $\{q_0 q_2 q_3\}$	$\{q_0 q_1 q_4\}$	$\{q_0 q_3\}$
* $\{q_0 q_1 q_4\}$	$\{q_0 q_1\}$	$\{q_0 q_2 q_3\}$



Scanned by CamScanner

③

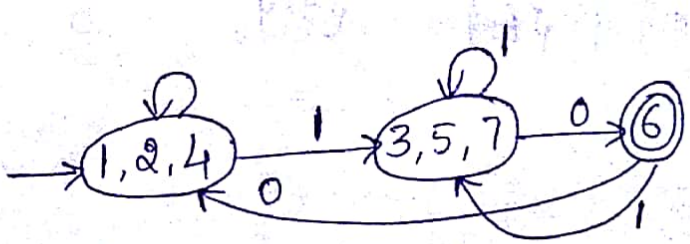
δ	0	1
1	2	3
2	4	5
3	6	7
4	4	5
5	6	7
* 6	4	5
7	6	7

	1	2	3	4	5	6 *
2	✓					
3	X	X				
4	✓	✓	X			
5	X	X	✓	X		
* 6	X	X	X	X	X	
7	X	X	✓	X	✓	X

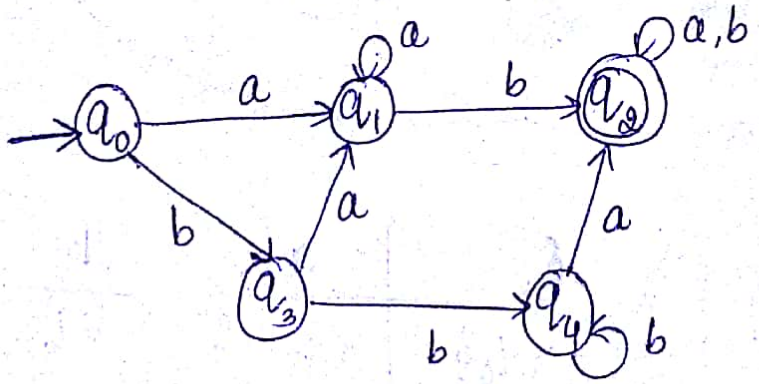
(1,2), (1,4), (5,7), (2,4), (3,5), (3,7)

(1,2,4), (3,5,7), 6

δ	0	1
→(1,2,4)	(1,2,4)	(3,5,7)
(3,5,7)	6	(3,5,7)
* 6	(1,2,4)	(3,5,7)



④



Scanned by CamScanner