Automata Theory and Computability - 18CS54

1) Theory of Computation 2) Computability 3) Complexity

Automata | Formal Language

Automata
- → Finite
- → Push Down
- → Linear Bounded
- → Turing Machine

\# Formal Languages : Some mathematical model or formula
1) Regular 2) Context free 3) Context-Sensitive
4) Recursively Enumerable.

Introduction to Finite Automata

- These are computing devices that accept/recognise regular languages and are used to model operations of many systems we find in practice.

July/2019 Jan 2019 * Σ - It is alphabet (Stands for alphabet)
It is defined as a set of finite symbols.
Ex: 1) $\Sigma = \{0, 1\}$ Binary alphabet
2) $\Sigma = \{0, 1, 2, 3, \ldots 9\}$ Set for decimal numbers
3) $\Sigma = \{a, b, c, d, \ldots z\}$ Alphabet set for English alph.

July/2019 Jan 2019 * S - Stands for String
It is defined as a finite sequence of symbols derived from alphabet (Σ).

Ex. i) Consider $\Sigma = \{0,1\}$
Strings that can be derived are
0, 1, 01, 001, 101, 1001, 101010, ...

* ε - Epsilon - It is a special string
It is called the null string or empty string.
Its length is always 0. ($|\varepsilon| = 0$)

* Σ^* $*$ → It can be zero or more. (called Kleen star operator)

Jan 2018 Σ^* → Its the power of alphabet
$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$

Ex. i) $\Sigma = \{0,1\}$
Σ^0 = Set of string of length 0
Σ^1 = Set of string of length 1
$\Sigma^* = \{0, 1, 00, 11, 10, 01, 000, 001, \ldots 1111, \ldots\}$
It is defined as set of strings of all possible strings length derived from Σ.

July/2019 Jan 2019 * L - It stands for Language
It is defined as the set of strings derived from Σ^* (Sigma Star) i.e. $L \subseteq \Sigma^*$
Ex. i) Consider $\Sigma = \{a, b\}$
→ L_1 = Set of strings whose length is odd.
$L_1 = \{a, b, aaa, bbb, ababa, \ldots, bbbbb, \ldots\}$
→ L_2 = Set of strings begin with a and end with b
$L_2 = \{ab, ababb, aaab, aabaabb, \ldots\}$

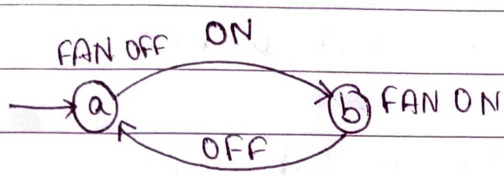
Symbols to draw an Automata.

→○ → Initial State

→ → Transition

○ → Intermediate State

◎ → End / final State



Strings

Functions	Relations
- Length	- Substrings
- Concatenation	- Proper Substrings
- Replication	- Prefix, Proper prefix
- Reversal	- Suffix, Proper Suffix

Functions :

★ Length of Strings : No. of symbols in the given strings.
Ex. $|101| = 3$, $|\varepsilon| = 0$

Jan 2018 ★ Concatenation of String : Concatenation of two strings ($S \| t$) is defined as string appending t to S.
Ex. $X = ab$, $Y = c \Rightarrow XY = abc$

★ Replication : Replication of string w is defined as

$$w^0 = \varepsilon$$

$$w^{i+1} = w^i . w \qquad \text{Eg. } a^3 = aaa$$

(Repeating the occurence of w 'i' no. of times).

★ Reversal of String w - For each string w, the reverse is defined as

i) if $|w| = 0$, then $w = w^R = \varepsilon$. Eg. $(abc)^R = cba$

ii) if $|w| \geq 1$, then $\forall a \in \Sigma (\forall u \in \Sigma^* (w = ua))$, then

$$w^R = au^R$$

Relations

* Substrings : A string 't' is a substring of 's' iff 't' continiously occurs in string s.

* Proper Substring : A string 't' is a proper substring of 's' iff 't' is a substring of 's' and $t \neq s$.

Ex. = 'εaabbaε'

$$\underbrace{\varepsilon, a, aa, aab, aabba}_{\text{Proper Substring}}$$

* Prefix : A string 's' is a prefix of 't' iff there exists $\exists x \in \Sigma^* (t = sx)$

* Proper Prefix : A string 's' is a P.P of t, iff string is prefix of t and $s \neq t$

Ex. Consider "abba"

$$\underbrace{\varepsilon, a, ab, abb}_{\text{Proper Prefix}}, abba$$

* Suffix : A string 's' is a suffix of 't' iff there exists $\exists x \in \Sigma^* (t = xs)$.

* Proper Suffix : A string s is a P. Suffix of s iff string is suffix of t and $s \neq t$

Ex. Consider "abba"

$$\underbrace{\varepsilon, b, bb, bba}_{\text{Proper Suffix}}, abba$$

Page

Techniques for defining Languages

- By enumeration | defining property

Examples : (Enumeration)

Let $L = \{w \in \{a,b\}^* :$ all string begin with a $\}$

$L = \{a, ab, aab, aabb, \ldots\}$

Strings not in L are :

$\{b, ba, \epsilon, bbb, \ldots\}$

- By (defining property)

Let $L = \{w \in \{a\}^* : |w| \bmod 3 = 1\}$

$L = \{a, a^4, a^7, a^{10}, \ldots\}$

Strings not in L are :

$\{\epsilon, a^2, a^3, a^5, a^6, \ldots\}$

Functions on Languages : All set operations like Union, Intersection, Difference and Complement can be applied.

Concatenation of Languages

Ex. $L_1 = \{aa, ab\}$

$L_2 = \{xx, yy\}$

$L_1L_2 = \{aaxx, aayy, abxx, abyy\}$

Set Operations on Languages

- $L_1 = \{\epsilon, a^2, a^4, a^6, \ldots\}$ // Even no. of a's
- $L_2 = \{a^1, a^3, a^5, \ldots\}$ // Odd no. of a's

→ $L_1 \cup L_2 = \Sigma^*$ @ $\{a\}^*$ // Union Operation

→ $L_1 \cap L_2 = \phi$ @ $\{\}$ // Intersection Operation

→ $L_1 - L_2$ // Difference Operation

→ $\sim(L_1 - L_2)$ // Complement Operation.

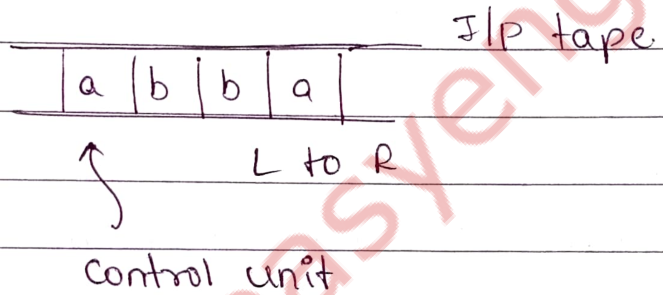
July '18 '19

Deterministic Finite State Machines (DFSM / FSM)

Jan 2019

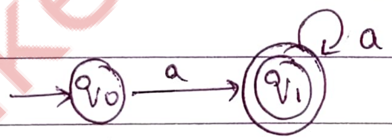
Definition : A FSM or DFSM, M is quintuple : $(Q, \Sigma, \delta, s, F)$

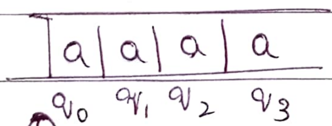
where,
- Q is finite set of states
- Σ is the input alphabet
- $s \in Q$ is the initial state
- F - End / final accepting state
- δ - It is the transition function it maps from $Q \times \Sigma$ to Q

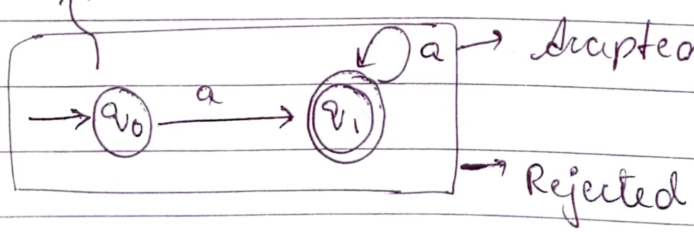


Ex. L_1 = Strings of one or more a's

$L_1 = \{a, aa, aaa, aaaa, \ldots\}$







Finite State ~~of~~ Machine has 3 Parts

i) Control Unit :

- Reading the contents of I/P ~~ty~~ tape from L to R and cell by cell

ii) Input Tape :

- It contains many rectangular cell of each cell is able to hold any only one symbol.

iii) Read head :

- Control unit is reading the contents of I/P tape from L to R and cell by cell.

The Logic or the strategy to recognize the lang. resides inside control unit.

Finally the machine rises either of the two messages (accepted @ rejected).

How to write transition diagram?

Steps are

1) Find the minimum string accepted, this decides the no. of states in the FSM, in most of the cases.

2) Then, take longer strings and make them accepted, while modifying the transitions

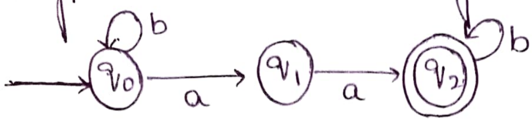
3) Check for the minimum strings that are not to be accepted, are really not accepted as per the transition diagram.

4) See that each state has transitions equal to the no. of alphabets present (Σ).

5) Two transition on the same alphabet do not go to different states.

Examples

1) From the given transition diagram of DFSM define all the tuples.



Sol →

$Q = \{q_0, q_1, q_2\}$ $M = \{Q, \Sigma, q_0, F, \delta\}$

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

$q_0 \rightarrow$ Initial State

$F \rightarrow q_2$ (Final State)

δ is shown in transition table

δ	a	b
q_0	q_1	q_0
q_1	q_2	-
q_2	-	q_2

2) Draw a DFSM to accept strings of 0's and 1's ending with string 011.

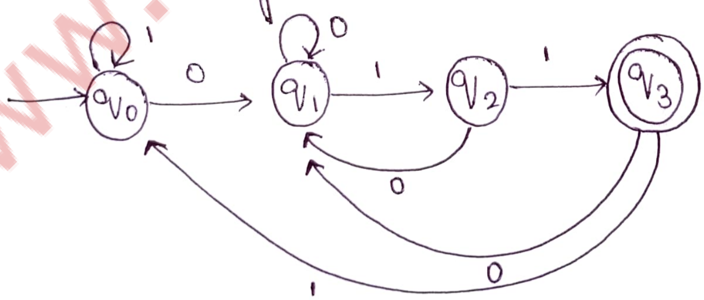
Sol → $\Sigma = \{0, 1\}$

$(0+1)^* 011 \rightarrow$ Regular Expression.

Minimum String → 011

\# As the string ends with 011.

Sample String - 0011, 1011, 10011.



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

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

$q_0 \rightarrow$ Initial State

$F \rightarrow q_3$ (Final State)

3) Draw a DFA to accept string of a's and b's having sub string aa.

Sol→ Regular Expression : $(a+b)^* aa (a+b)^*$

Minimum String : aa

Sample Strings : baab, abaa, aabb

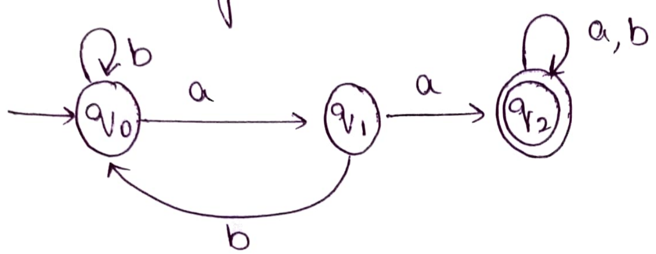
$M = \{Q, \Sigma, q_0, F, \delta\}$

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

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

$q_0 \rightarrow$ Initial State

$F \rightarrow q_2 \rightarrow$ Final State



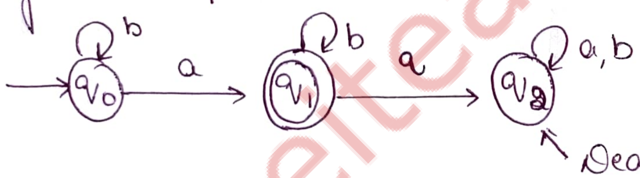
4) Obtain a DFA to accept strings of a's and b's having exactly one a.

Sol→ $\Sigma = \{a, b\}$

Minimum String : a

Sample String : a, bab, bba, abb.

Regular Expression : $b^* a b^*$



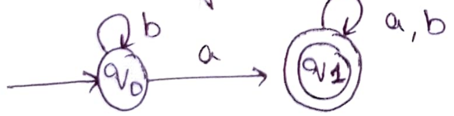
5) Obtain a DFA to accept strings of a's and b's having atleast one a.

Sol→ $\Sigma = \{a, b\}$

Regular Expression : $(a+b)^* a (a+b)^*$

Minimum String : a

Sample String : a, aa, aaa, abab.



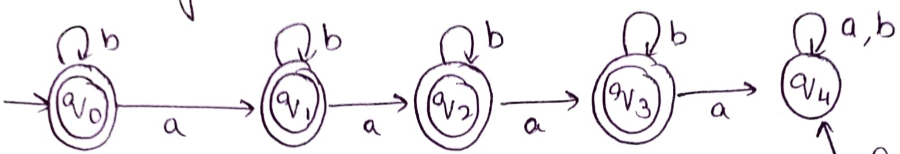
6) Obtain a DFA to accept strings a's and b's having not more than 3 a's.

Sol → $\Sigma = \{a, b\}$

\# Not more than 3 a's canbe either 0, @ 1, @ 2, @ 3 a's.

Min String : ϵ

Sample String : ϵ, a, aa, aaa, ba, aab



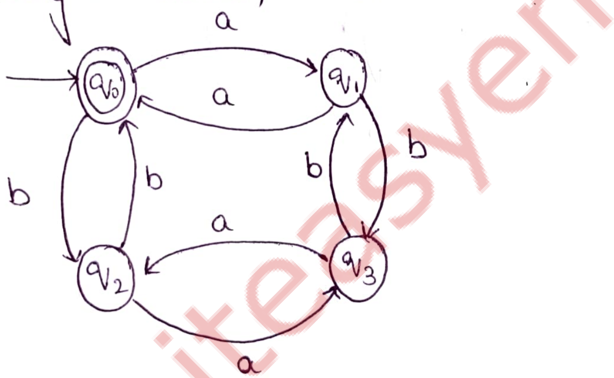
Jan 2019

7) Obtain a DFA to accept strings of a's and b's having even number of a's and b's.

Sol → $\Sigma = \{a, b\}$

Min String = ϵ, aa, bb

Sample String : abab, aabb, aaaa, abababab.



q_1 → Odd no. of a's, Even of b's

q_2 → Even no. of a's, Odd of b's

q_3 → Odd no. of a's, odd no. of b's

8) Construct a DFA to accept strings of 0's and 1's having i) 011 as the substring (ii) String starting with 011

iii) $L = \{ w \mid w \in (a+b)^* \mid\mid N_a(w) \bmod 3 = 0 \quad N_b(w) \bmod 2 = 0 \}$

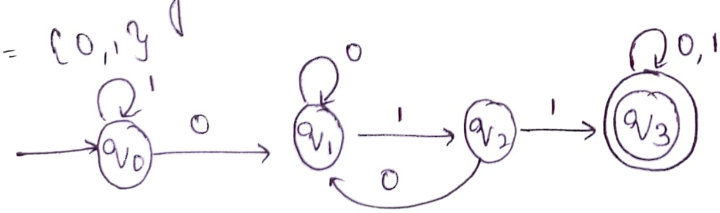
Sol Jan 2019

i) Having 011 as substring

\# $N_a(w) \bmod 3$ means a in multiples of 3 and respectively b in multiples of 2

Min String : 011

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

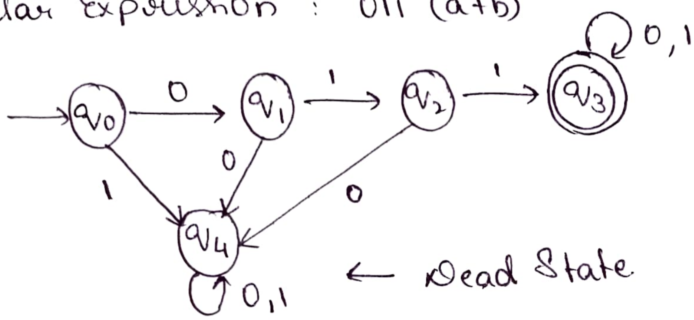


ii) Starting with 011

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

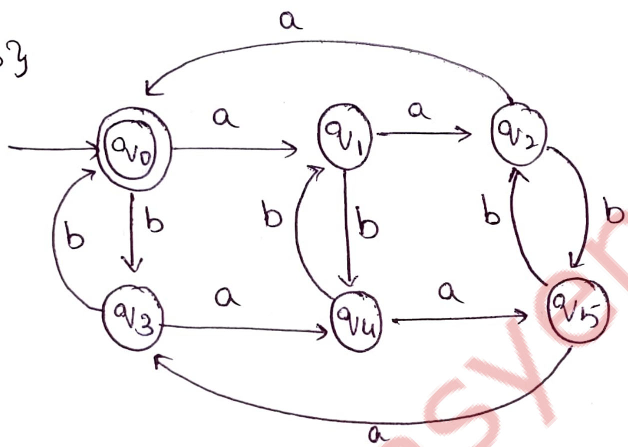
Min String : 011

Regular Expression : $011(a+b)^*$



iii) $L = \{w \mid w \in (a+b)^* \text{ and } N_a(w) \bmod 3 = 0,\ N_b(w) \bmod 2 = 0\}$

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

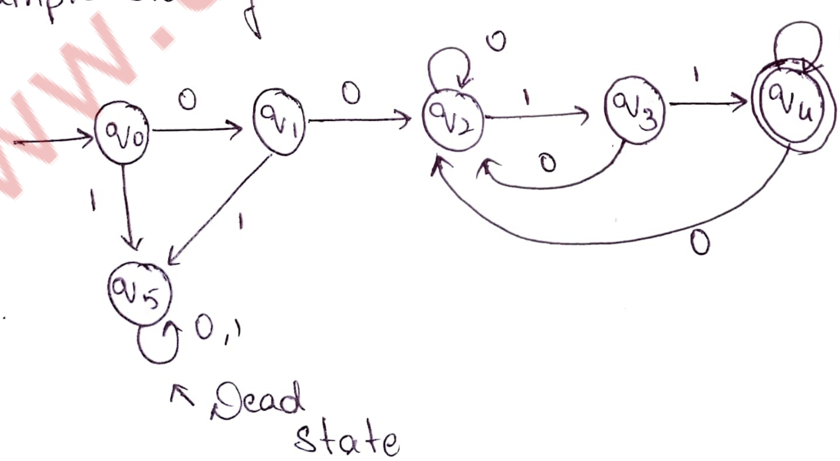


9) Obtain a DFA accepting strings of 0's and 1's starting with atleast 2 0's and ending with atleast 2 1's.

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

Min String : 0011

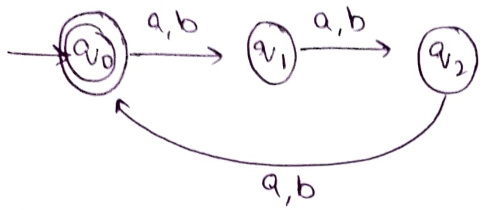
Sample String : 001010011 , 0010111



10) Obtain a DFSM to accept language

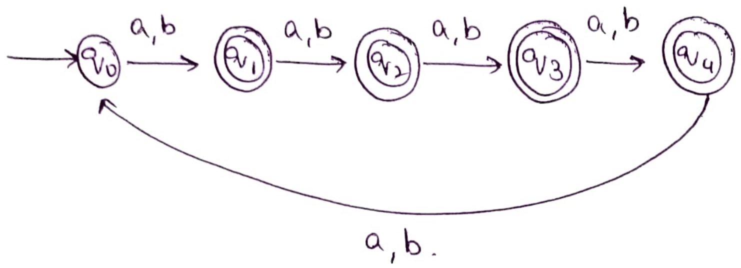
i) $L = \{w \mid |w| \bmod 3 = 0\}$ on $\Sigma = \{a,b\}$ (or) $L = \{w \mid |w| \equiv_3 0\}$ on $\Sigma = \{a,b\}$

Sol →



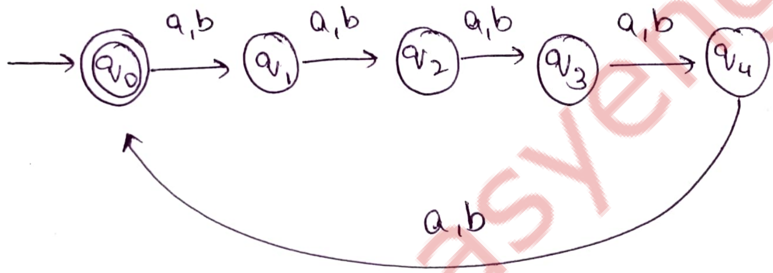
ii) $L = \{w \mid |w| \bmod 5 \neq 0\}$ on $\Sigma = \{a,b\}$

Sol →



iii) $L = \{w \mid (|w|) \bmod 5 = 0\}$ on $\Sigma = \{a,b\}$

Sol →



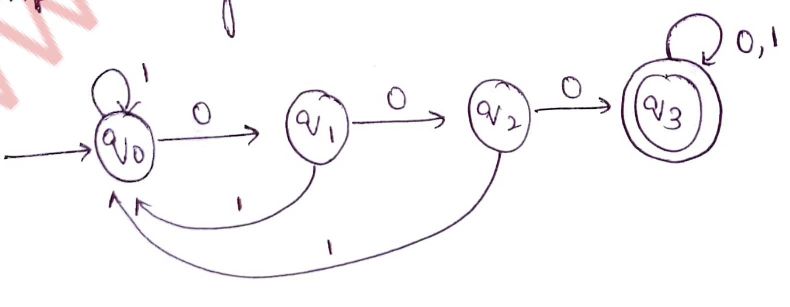
11) Draw a DFSM that accepts strings of 0's and 1's that contain 3 consecutive 0's as a substring.

Sol →

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

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

Sample String = $\{000, 0001, 100011, 010001100\}$

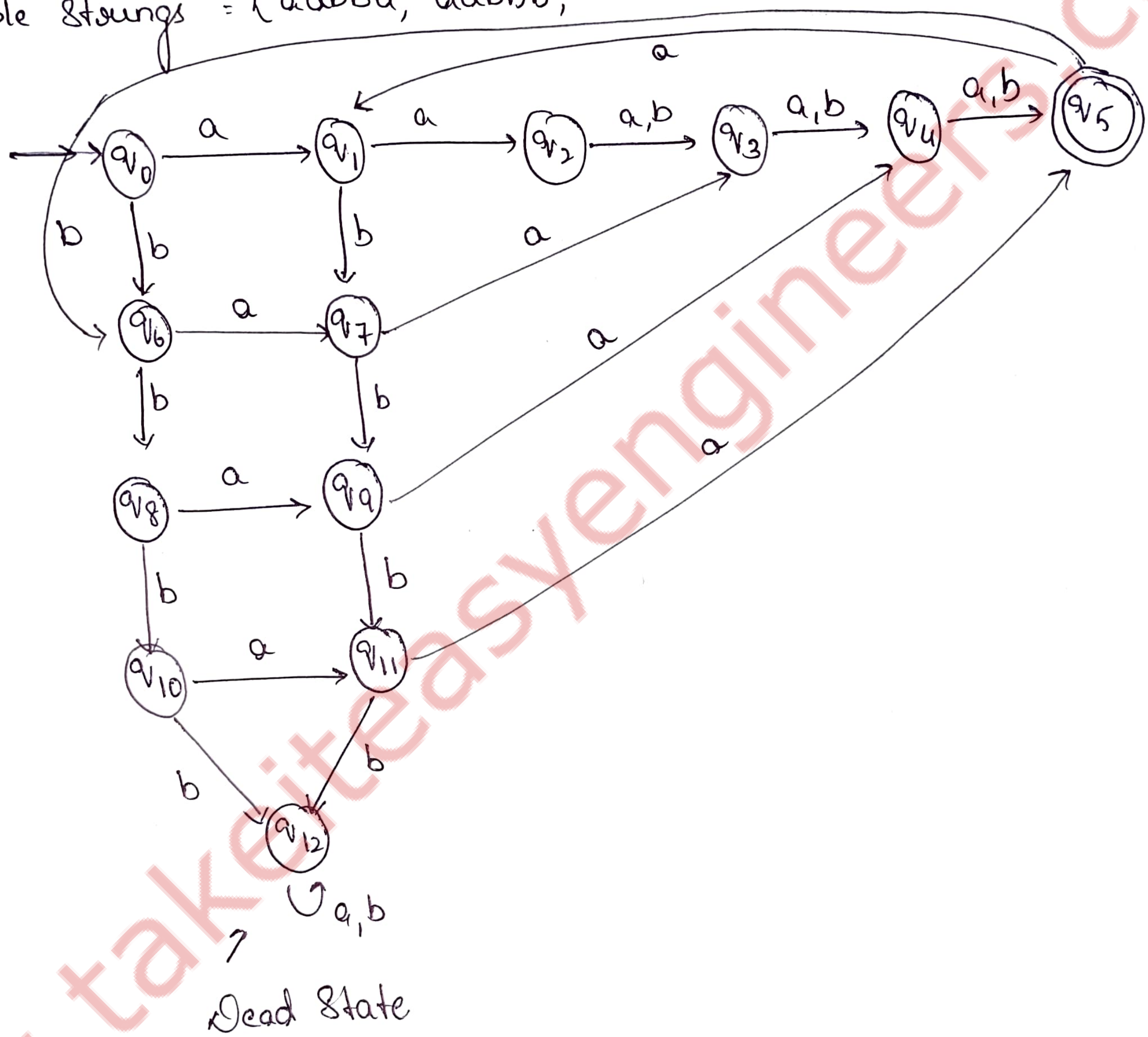


12) Construct a DFSM to accept strings of a's and b's such that each block of 5 consecutive symbols should have atleast 2 a's.

Sol → $\Sigma = \{a, b\}$

Min String = aaaaa

Sample Strings = {aabba, aabbb, abaab . . . }



Jan 2018

13) Draw a DFSM to accept strings of a's and b's ending with bab.

Sol → $\Sigma = \{a, b\}$

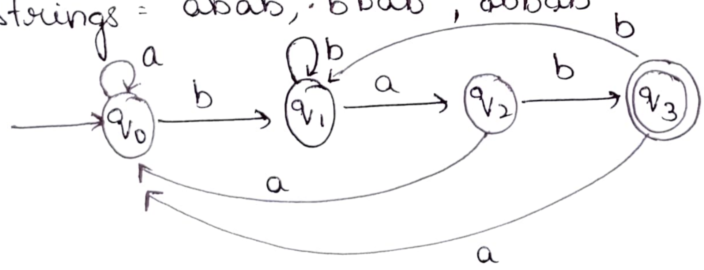
Min String = bab

Sample Strings = abab, bbab, abbab

$q_0 \rightarrow$ Initial State

$F \rightarrow q_3 \rightarrow$ Final State

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



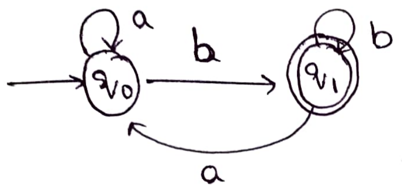
$M = (Q, \Sigma, q_0, F, \delta)$

14) Design a DFSM to accept strings of a's and b's where every string ends in b.

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

Min String : b

Sample strings : ab, abb, aabb, babb.



15) Design a DFSM to accept strings of 0's and 1's where it begin with 0 and end with 1.

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

Min String : 01

Sample String : 0011, 01001, 011001, ...

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

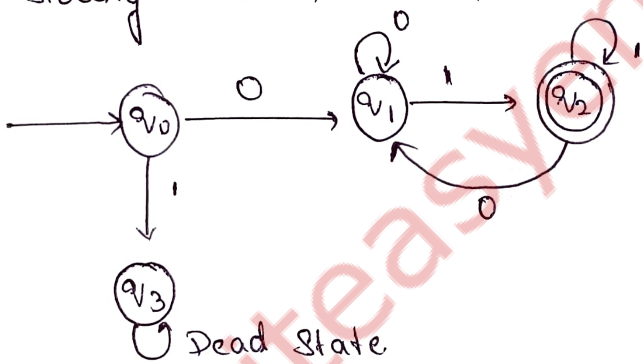
$Q = \{q_0, q_1, q_2, q_3\}$

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

q_0 - Initial

q_2 - Final/End state

$\delta = Q \times \Sigma \Rightarrow Q$

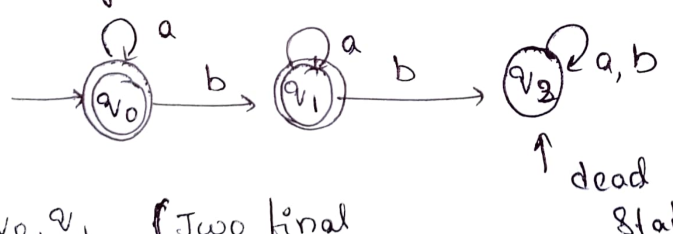


16) Design a DFSM that accepts strings of a's and b's that contain not more than 1 b

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

Min String = ε, b

Sample String = ab, aab, aba



F = q_0, q_1 (Two final states)

Non deterministic Finite State Machines (NFSM)

Definition : A NFSM , M is a Quintuple : $(Q, \Sigma, \Delta, S, A)$
where,

- $Q \rightarrow$ Finite set of states
- $\Sigma \rightarrow$ Input alphabets
- $S \rightarrow$ Initial state
- $F \rightarrow$ subset of K is the set of accepting states
- $\delta \rightarrow$ It is the transition function it maps from $(Q \times (\Sigma \cup \{\varepsilon\}))$ to Q

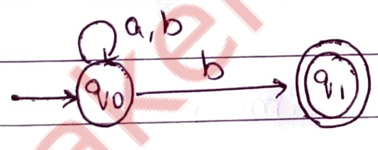
Example

i) Write a NFSM to accept the language
$L = \{ w \in \{a, b\}^* \mid |w| \text{ ends in } b\}$.

Sol $\rightarrow$ $\Sigma = \{a, b\}$ Regular Expression : $(a+b)^* b$

Min String b.

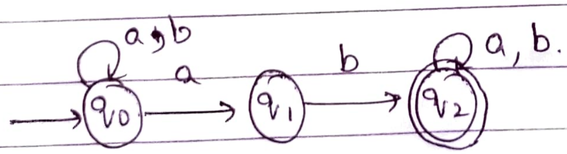
Sample String = $\{b, bb, bbb, aaab, ababb, \ldots\}$



ii) Write NFSM to recognize $L = \{ w \in \{a, b\}^* \mid |w| \text{ Contains } ab\}$

Sol $\rightarrow$ $\Sigma = \{a, b\}$ Regular Expression $(a+b)^* ab (a+b)^*$

Min String = a, b



Sample Strings : $\{aaba, aabbb, baaa, \ldots\}$

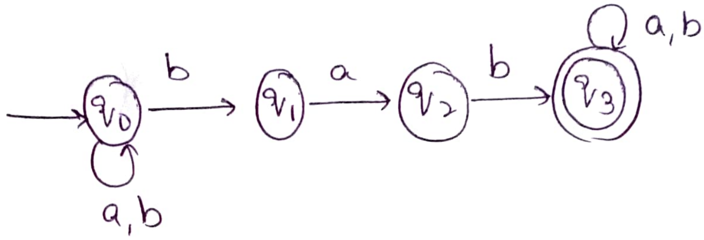
3) Design NFSM that accepts strings of a's and b's that contain substring bab.

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

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

$S = q_0$ (Initial)

$F = q_3$ (Final)



δ	a	b
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_1	q_2	-
q_2	-	b
$* \; q_3$	q_3	q_3

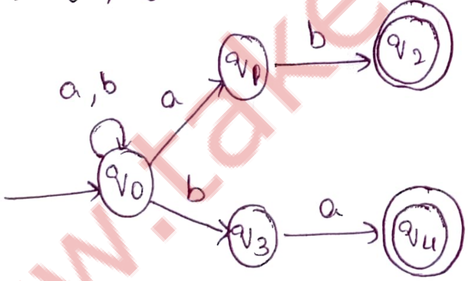
July 2019

4) Design a NFSM that accept the language

$L = \{w \in \{a,b\}^* \mid$ every w ends in ab @ ba$\}$

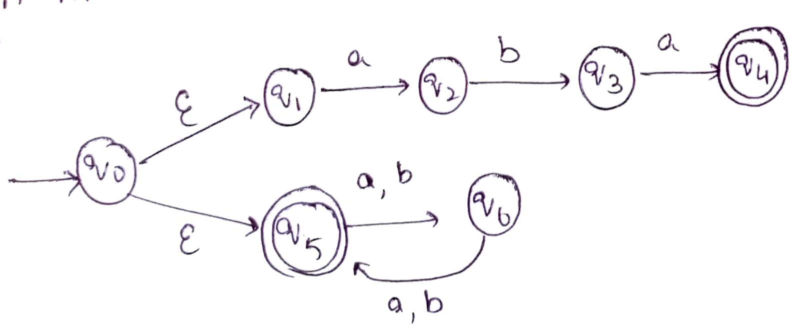
Sol, Regular expression : $(a+b)^* (ab + ba)$

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



5) Design NFSM. $L = \{w \in \{a,b\}^* \mid w = aba$ @ $|w|$ is even$\}$

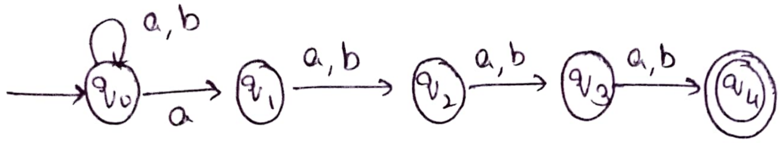
Sol,



6) Design a NFSM.

$L = \{w \in \{a,b\}^* \mid \forall$ string the 4^{th} character from right is $a\}$.

Sol,



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

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

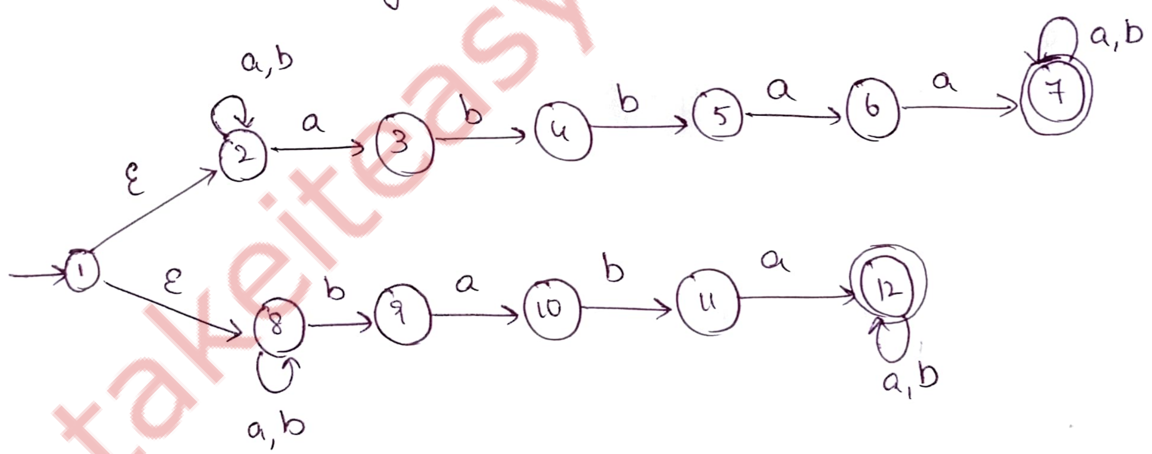
$S = q_0$ (Initial)

$F = q_4$ (final / End state)

Δ	a	b
q_0	$\{q_0, q_1\}$	q_0
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_4	q_4
q_4	-	-

7) Design a NFSM

$L = \{w \in \{a,b\}^* \ \forall x, y \in \{a,b\}^*, w = (xabbaay) \vee w = (xbabay)\}$

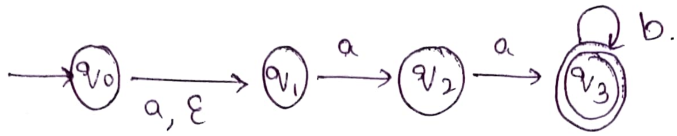


8) Write a NFSM to recognise the language.

$L = \{w \in \{a,b\}^* \mid w$ is made up of an optional a followed by aa, zero or more b's$\}$

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

Regular Expression. $= (a+\varepsilon)\, aa\, (b)^*$



Equivalence of NDFSM and DFSM.

We can say NFSM and DFSM are equivalent when the language accepted by both are same.

Epsilon closures of state 'q'

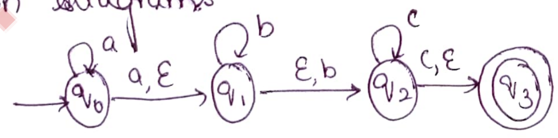
It is defined as a set of states that is reachable from q on Epsilon transition only.

$$eps(q) = \{p \in K : (q, w) \xrightarrow{*} (p, w)\}$$

Examples

1) Compute epsilon closures for all the states given in the transition diagrams

Sol, i)



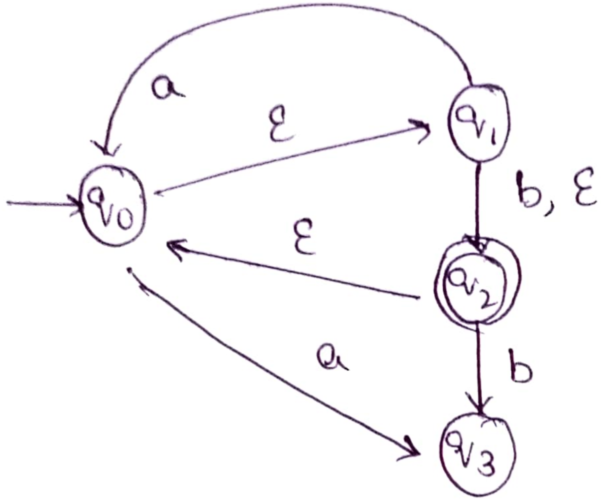
$Eps(q_0) = \{q_0, q_1, q_2, q_3\}$

$Eps(q_1) = \{q_1, q_2, q_3\}$

$Eps(q_2) = \{q_2, q_3\}$

$Eps(q_3) = \{q_3\}$

ii)



$Eps(q_0) = \{q_0, q_1, q_2\}$

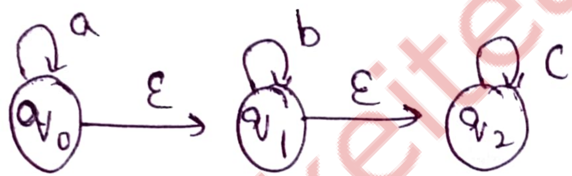
$Eps(q_1) = \{q_1, q_2, q_0\}$

$Eps(q_2) = \{q_0, q_1, q_2\}$

$Eps(q_3) = \{q_3\}$

NOTE : eps (any state) = { that state is always included }

iii)



$Eps(q_0) = \{q_0, q_1, q_2\}$

$Eps(q_1) = \{q_1, q_2\}$

$Eps(q_2) = \{q_2\}$.

Conversion of NDFSM to DFSM.

Let $M_N = (Q_N, \Sigma_N, \delta_N, q_0, A_N)$ be an NDFSM and accepts the language $L(M_N)$. There should be an equivalent DFSM $M_D = (Q_D, \Sigma_D, \delta_D, q_0, A_D)$ such that $L(M_D) = L(M_N)$. The procedure to convert an NDFSM to its equivalent DFSM is shown below.

Step 1

The state i.e. start state of NDFSM M_N is the start state of DFSM M_D. So add q_0 (which is the start state of NDFSM) to Q_D and find the transitions from this state. The way to obtain different transitions is shown in Step 2.

Step 2

For each state $[q_i, q_j \ldots q_k]$ in Q_D, the transitions for each input symbol in Σ can be obtained as below

1) $\delta_D([q_i, q_j \ldots q_k], a) = \delta_N(q_i, a) \cup \delta_N(q_j, a) \cup \ldots \cup \delta_N(q_k, a)$
$= [q_l, q_m, \ldots q_n]$ say.

2) Add the state $[q_l, q_m \ldots q_n]$ to Q_D if its not already present in Q_D.

3) Add the transition from $[q_i, q_j \ldots q_k]$ to $[q_l, q_m, \ldots q_n]$ on the input symbol a iff the state $[q_l, q_m, \ldots q_n]$ is added to Q_D in b4 step.

Step 3

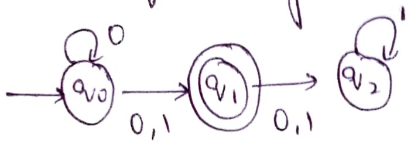
The state $[q_a, q_b \ldots q_c] \in Q_D$ is the final state, if atleast one of the state in $[q_a, q_b \ldots q_c \in A_N]$ i.e. atleast one component in $[q_a, q_b \ldots q_c]$ should be final state of NDFSM.

Step 4

If epsilon (ϵ) is accepted by NDFSM, then start state q_0 of DFSM is made the final state.

Jan 2019

1) Convert the following NDFSM to DFSM



Step 1

q_0 is start of DFSM

So, $Q_D = \{[q_0]\}$

Step 2

Find the new states from each state in Q_D.

Consider $[q_0]$:

when $a=0$, $\delta_D([q_0],0) = \delta_N([q_0,0])$

$= [q_0, q_1]$

when $a=1$, $\delta_D([q_0],1) = \delta_N([q_0],1)$

$= [q_1]$

So, Now,

$Q_D = \{[q_0],[q_0,q_1],[q_1]\}$

Now consider $[q_0, q_1]$.

when $a=0$, $\delta_D([q_0,q_1],0) = \delta_N([q_0,q_1],0)$

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

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

$= [q_0,q_1,q_2]$

when $a=1$, $\delta_D([q_0,q_1],1] = \delta_N([q_0,q_1],1)$

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

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

$= [q_1,q_2]$

So Now,

$Q_D = \{[q_0],[q_0,q_1],[q_1],[q_0,q_1,q_2],[q_1,q_2]\}$

Consider state $[q_1]$

when $a = 0$, $\delta_D([q_1], 0) = \delta_N([q_1], 0)$

$= [q_2]$

when $a = 1$, $\delta_D([q_1], 1) = \delta_N([q_1], 1)$

$= [q_2]$

So now,

$Q_D = \{[q_0], [q_0, q_1], [q_1], [q_0, q_1, q_2], [q_1, q_2], [q_2]\}$

consider state $[q_0, q_1, q_2]$

when $a = 0$, $\delta_D([q_0, q_1, q_2], 0) = \delta_N([q_0, q_1, q_2], 0)$

$= \delta_N(q_0, 0) \cup \delta_N(q_1, 0) \cup \delta_N(q_2, 0)$

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

$= [q_0, q_1, q_2]$

when $a = 1$, $\delta_D([q_0, q_1, q_2], 1) = \delta_N([q_0, q_1, q_2], 1)$

$= \delta_N(q_0, 1) \cup \delta_N(q_1, 1) \cup \delta_N(q_2, 1)$

$= \{q_1\} \cup \{q_2\} \cup \{\phi\}$

$= [q_1, q_2]$

So, the states obtained are not new states. $\therefore$ Q_D remains same.

consider state $[q_1, q_2]$

when $a = 0$, $\delta_D([q_1, q_2], 0) = \delta_N([q_1, q_2], 0)$

$= \delta_N(q_1, 0) \cup \delta_N(q_2, 0)$

$= \{q_2\} \cup \{\phi\}$

$= [q_2]$

when $a = 1$, $\delta_D([q_1, q_2], 1) = \delta_N([q_1, q_2], 1)$

$= \delta_N(q_1, 1) \cup \delta_N(q_2, 1)$

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

$= [q_2]$

Since, the states obtained are not new, Q_D remains same.

consider state $[q_2]$

when $a=0$, $\delta_D([q_2],0) = \delta_N([q_2],0)$
$= \{\phi\}$

when $a=1$, $\delta_D([q_2],1) = \delta_N([q_2],1)$
$= [q_2]$

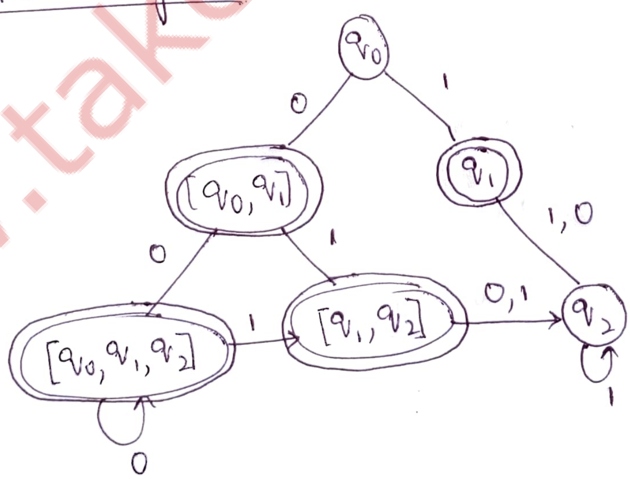
Since, the states obtained are not new states,
Q_D remains same.

$$Q_D = \{[q_0],[q_1,q_2],[q_2],[q_0 q_1 q_2][q_0,q_1][q_1]\}$$

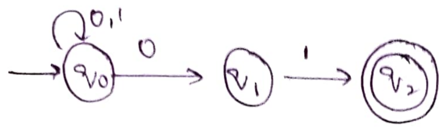
Transition Table

δ	0	1
q_0	$[q_0,q_1]$	$[q_1]$
$[q_0,q_1]$	$[q_0,q_1,q_2]$	$[q_1,q_2]$
$[q_1]$	$[q_2]$	$[q_2]$
$[q_0,q_1,q_2]$	$[q_0,q_1,q_2]$	$[q_1,q_2]$
$[q_1,q_2]$	$[q_2]$	$[q_2]$
$[q_2]$	ϕ	$[q_2]$

Transition Diagram



2) Convert NFSM into DFSM



Step1

$[q_0]$ is starting state of DFSM.

$\delta_D = [q_0]$.

Step2

consider q_0.

when $a=0$, $\delta_D([q_0],0) = \delta_N([q_0],0)$

$= [q_0,q_1]$

when $a=1$, $\delta_D([q_0],1) = \delta_N([q_0],1)$

$= [q_0]$.

Now, $Q_D = \{[q_0],[q_0,q_1]\}$

Consider $[q_0,q_1]$

when $a=0$, $\delta_D([q_0,q_1],0) = \delta_N([q_0,q_1],0)$

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

$= [q_0,q_1] \cup [\phi]$

$= [q_0,q_1]$

when $a=1$, $\delta_D([q_0,q_1],1) = \delta_N([q_0,q_1],1)$

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

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

$= [q_0,q_2]$

$Q_D = \{[q_0],[q_0,q_1],[q_0,q_2]\}$

Consider $[q_0, q_2]$

when $a=0$, $\delta_D([q_0, q_2], 0) = \delta_N([q_0, q_2], 0)$

$= \delta_N(q_0, 0) \cup \delta_D(q_2, 0)$

$= [q_0, q_1]$

when $a=1$, $\delta_N([q_0, q_2], 1) = \delta_N([q_0, q_2], 1)$

$= \delta_N(q_0, 1) \cup \delta_N(q_2, 1)$

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

$= [q_0]$

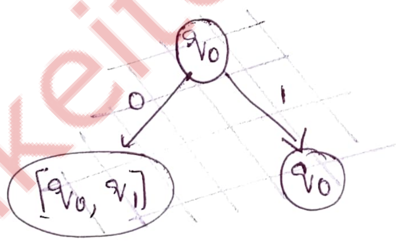
Since, obtained states are not new,

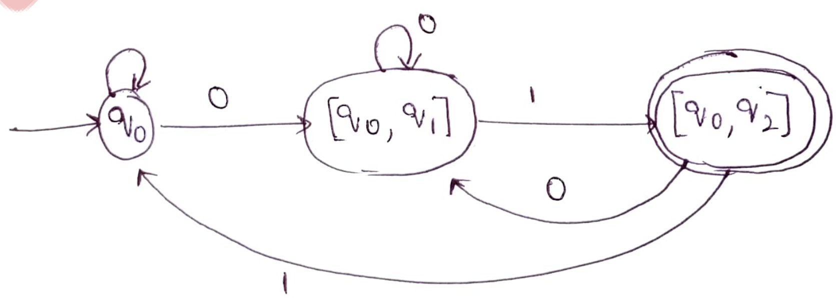
Q_D remains same. $\Rightarrow Q_D = \{[q_0], [q_0, q_1], [q_0, q_2]\}$

Transition Table.

	δ	0	1
$\rightarrow$	$[q_0]$	$[q_0, q_1]$	$[q_0]$
	$[q_0, q_1]$	$[q_0, q_1]$	$[q_0, q_2]$
$*$	$[q_0, q_2]$	$[q_0, q_1]$	$[q_0]$

Transition Diagram.



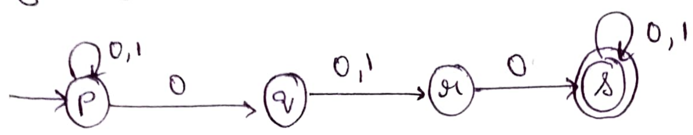


Convert NFSM to DFSM

3)

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

Sol →



Step 1

$Q_D = [p]$ is starting state

Step 2

consider [p]

when $a=0$, $\delta_D[p,0] = \delta_N[p,0]$

$= [p].[q]$

$= [p,q]$

when $a=1$, $\delta_D[p,1] = \delta_N[p,1]$

$= [p]$.

Now,

$Q_D = \{[p],[p,q]\}$

consider [p, q]

when $a=0$, $\delta_D([p,q],0) = \delta_N([p,q],0)$

$= \delta_N(p,0) \cup \delta_N(q,0)$

$= [p,q] \cup \{r\}$

$= [p,q,r]$

when $a=1$, $\delta_D([p,q],1) = \delta_N([p,q],1)$

$= \delta_N(p,1) \cup \delta_N(q,1)$

$= \{p\} \cup \{r\}$

$= [p,r]$

Now,

$Q_D = \{[p],[p,q],[p,q,r],[p,r]\}$

Consider [p,r]

when $a=0$, $\delta_D([p,r],0) = \delta_N([p,r],0)$

$= \delta_N(p,0) \cup \delta_N(r,0)$

$= \{p,q\} \cup \{r\}$

$= [p,q,s]$

when $a=1$, $\delta_D([p,r],1) = \delta_N([p,r],1)$

$= \delta_N(p,1) \cup \delta_N(r,1)$

$= \{p\} \cup \{\phi\}$

$= [p]$

$Q_D = \{[p],[p,q],[p,q,r],[p,r],[p,q,s]\}$

Consider [p,q,r].

when $a=0$, $\delta_D[(p,q,r),0] = \delta_N[(p,q,r),0)$

$= \delta_N(p,0) \cup \delta_N(q,0) \cup \delta_N(r,0)$

$= \{p,q\} \cup \{r\} \cup \{s\}$

$= [p,q,r,s]$

when $a=1$, $\delta_D[(p,q,r],1) = \delta_N([p,q,r],1)$

$= \delta_N(p,1) \cup \delta_N(q,1) \cup \delta_N(r,1)$

$= \{p\} \cup \{r\} \cup \{\phi\}$

$= [p,r]$

$Q_D = \{[p],[p,q],[p,q,r],[p,r],[p,q,s],[p,q,r,s]\}$

consider [p,q,r,s]

when $a=0$, $\delta_D[[p,q,r,s],0] = \delta_N[\{p,q,r,s],0]$

$= \delta_N(p,0) \cup \delta_N(q,0) \cup \delta_N(r,0) \cup \delta_N(s,0)$

$= \{p,q\} \cup \{r\} \cup \{s\} \cup \{s\}$

$= [p,q,r,s]$

when $a=1$, $\delta_D[[p,q,r,s],1] = \delta_N[(p,q,r,s),1]$

$= \delta_N(p,1) \cup \delta(q,1) \cup \delta(r,1) \cup \delta(s,1)$

$= \{p\} \cup \{r\} \cup \{\phi\} \cup \{s\}$

$= [p,r,s]$

$Q_D = \{[p],[p,q],[p,q,r],[p,r],[p,q,s],[p,q,r,s],[p,r,s]\}$

Consider $[p,q,s]$

when $a=0$, $\delta_D[[p,q,s],0] = \delta_N[[p,q,s],0]$

$= \delta_N(p,0) \cup \delta_N(q,0) \cup \delta_N(s,0)$

$= \{p,q\} \cup \{r\} \cup \{s\}$

$= [p,q,r,s]$

when $a=1$, $\delta_D[[p,q,s],1] = \delta_N[[p,q,s],1]$

$= \delta_N(p,1) \cup \delta_N(q,1) \cup \delta(s,1)$

$= [p,r,s]$

Consider $[p,r,s]$

when $a=0$, $\delta_D[[p,r,s],0] = \delta_N[[p,r,s],0]$

$= \delta_N(p,0) \cup \delta_N(r,0) \cup \delta_N(s,0)$

$= \{p,q\} \cup \{s\} \cup \{s\}$

$= [p,q,s]$

when $a=1$, $\delta_D[[p,r,s],1] = \delta_N[[p,r,s],0]$

$= \delta_N(p,0) \cup \delta_N(r,0) \cup \delta_N(s,0)$

$= [p] \cup [s]$

$= [p,s]$

Consider $[p,s]$

when $a=0$, $\delta_D[[p,s],0] = \delta_N[[p,s],0]$

$= \delta_N(p,0) \cup \delta_N(s,0)$

$= \{p,q\} \cup \{s\}$

$= [q,p,s]$

when $a=1$, $\delta_D[[p,s],1] = \delta_N[[p,s],1]$

$= \delta_N(p,1) \cup \delta_N(s,1)$

$= \{p\} \cup \{s\}$

$= [p,s]$

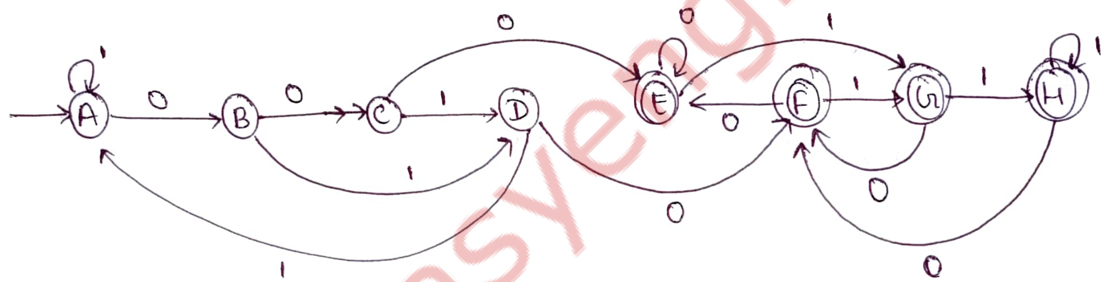
$\therefore$ we arrive at

$Q_D = \{[p],[p,q],[p,q,r],[p,r],[p,q,r,s],[p,q,s],[p,r,s],[p,s]\}$.

Transition Table

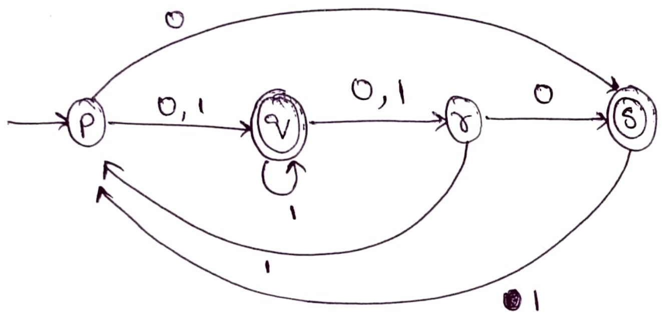
	δ	0	1
→	Let A = [p]	B [p,q]	A [p]
	B = [p,q]	C [p,q,r]	D [p,r]
	C = [p,q,r]	E [p,q,r,s]	D [p,r]
	D = [p,r]	F [p,q,s]	A [p]
*	E = [p,q,r,s]	E [p,q,r,s]	G [p,r,s]
*	F = [p,q,s]	E [p,q,r,s]	G [p,r,s]
*	G = [p,r,s]	F [p,q,s]	H [p,s]
*	H = [p,s]	F [p,q,s]	H [p,s].

Transition Diagram



4) Convert the following NDFSM into DFSM.

δ	0	1
$\rightarrow p$	$\{q, s\}$	q
$* q$	$\{r\}$	$\{q, r\}$
r	s	$\{p\}$
$* s$	$\emptyset$	$\{p\}$



<u>Step 1</u>

Consider p is initial state of DFSM.

$Q_D = \{p\}$

<u>Step 2</u>

Consider p

when $a = 0$, $\delta_D(p, 0) = \delta_N(p, 0)$
$= [q, s]$

when $a = 1$, $\delta_D(p, 1) = \delta_N(p, 1)$
$= [q]$

Now,

$Q_D = \{[p], [q], [q, s]\}$

Consider [q]

when $a = 0$, $\delta_D(q, 0) = \delta_N(q, 0)$
$= [r]$

when $a = 1$, $\delta_D(q, 1) = \delta_N(q, 1)$
$= [q, r]$

Now,

$Q_D = \{[p], [q], [q, s], [r], [q, r]\}$

Consider [q, s]

when $a=0$, $\delta_D[[q,s],0] = \delta_N[[q,s],0]$

$= \delta_N\{(q,0) \cup (s,0)\}$

$= [r]$

when $a=1$, $\delta_D[[q,s],1] = \delta_N[[q,s],1]$

$= \delta_N\{(q,1) \cup (s,1)\}$

$= [q,r,p]$

$Q_D = \{[p],[q],[q,s],[r],[q,r],[q,r,p]\}$

Consider [r]

when $a=0$, $\delta_D(r,0) = \delta_N(r,0)$

$= [s]$

when $a=1$, $\delta_D(r,1) = \delta_N(r,1)$

$= [p]$

Consider [q, r]

when $a=0$, $\delta_D([q,r],0) = \delta_N([q,r],0)$

$= \delta_N\{(q,0) \cup (r,0)\}$

$= [r,s]$

when $a=1$, $\delta_D([q,r],1) = \delta_N[(q,r),1]$

$= \delta_N\{(q,1) \cup (r,1)\}$

$= [q,r,p]$

$Q_D = \{[p],[q],[q,s],[r],[q,r],[q,r,p],[s],[r,s]\}$

Consider [q, r, P]

when $a=0$, $\delta_D[(q,r,P),0] = \delta_N[[q,r,P],0]$

$= \delta_N(q,0) \cup \delta_N(r,0)\ \delta_N(P,0)$

$= [r,s,q]$

when $a=1$, $\delta_D[[q,r,P],1] = \delta_N[[q,r,P],1]$

$= \delta_N\{(q,1) \cup (r,1) \cup (P,1)\}$

$= [q,r,P]$

$Q_D = \{[p],[q],[q,s],[r],[q,r],[q,r,P],[s],[r,s],[q,r,s]\}$

Consider [S]

when $a=0$, $\delta_D(S,0) = \delta_N([S],0)$

$= \{\phi\}$

when $a=1$, $\delta_D[[S],1] = \delta_N[[S],1]$

$= [P]$

Consider [q,r,s]

when $a=0$, $\delta_D[[q,r,s],0] = \delta_N[[q,r,s],0]$

$= \delta_N\{(q,0) \cup (r,0) \cup (s,0)]$

$= [r,s]$

when $a=1$, $\delta_D[[q,r,s],1] = \delta_N[[q,r,s],1]$

$= \delta_N\{(q,1) \cup (r,1) \cup (s,1)]$

$= [q,r,p] = [p,q,r]$

Consider [r,s]

when $a=0$, $\delta_D[(r,s),0] = \delta_N[[r,s],0]$

$= \delta_N(r,0) \cup \delta_N(s,0)$

$= \{s, \phi\}$

$= [s]$

when $a=1$, $\delta_D[[r,s],1] = \delta_N[[r,s],1]$

$= \delta_N\{[r,1] \cup [s,1]\}$

$= \{p\} \cup \{p\}$

$= [p]$

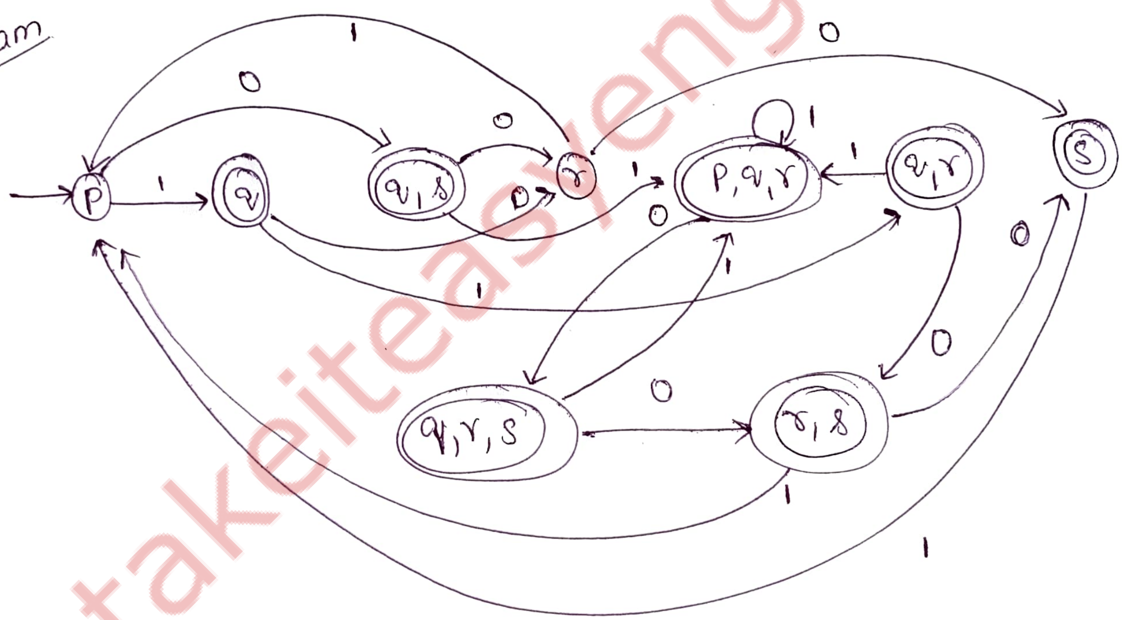
Therefore, we have

$Q_D = [[p],[q],[q,r],[r],[q,s],[q,r,p],[s],[r,s],[q,r,s]]$

Transition Table

	δ	0	1
→	[p]	[q, s]	[q]
*	[q]	[r]	[q, r]
*	[q, s]	[r]	[p, q, r]
	[r]	[s]	[p]
*	[p, q, r]	[q, r, s]	[p, q, r]
*	[q, r]	[r, s]	[q, r, p]
*	[s]	∅	[p]
*	[q, r, s]	[r, s]	[p, q, r]
*	[r, s]	[s]	[p]

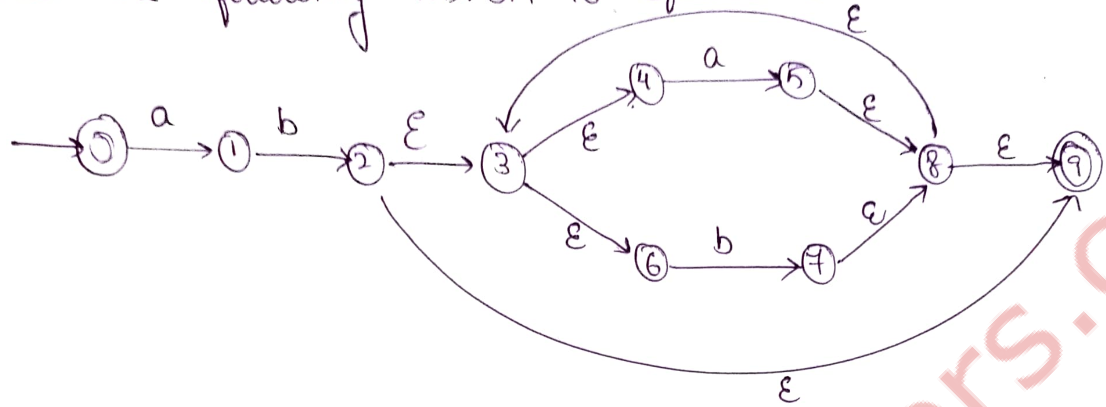
Transition Diagram



Jan 2018

ε - NDFSM to DFSM.

1) Convert the following NDFSM to equivalent DFSM



Sol: Starting State: ε Closure [0] = [0].

Q_D = [0]

Consider Q_D = [0]. initial State.

On input symbol 'a' and 'b'

when a : $\delta_D\{[0], a\}$ = ε-Closure(δ_N[0, a])

= ε-Closure (1)

= [1]

when b; δ_D[[0], b] = ε-Closure (δ_N[0, b])

= ε-Closure {ϕ]

= ϕ

Now,

Q_D = {[0], [1]}

consider [1]

On input symbols 'a' and 'b'

when a : δ_D[[1], a] = ε-Closure (δ_N(1, a))

= ε-Closure [ϕ]

= ϕ

when b : δ_D[1, b] = ε-Closure (δ_N (1, b))

= ε-Closure [2]

= [2, 3, 4, 6, 9]

Now,

Q_D = {[0], [1], [2, 3, 4, 6, 9]}

Consider [2,3,4,6,9]

On inputs 'a' and 'b'

when a : $\delta_D([2,3,4,6,9], a) = \varepsilon\text{-Closure}[\delta_N[2,3,4,6,9], a]$

$= \varepsilon\text{-Closure}[\delta_N\{(2,a) \cup (3,a) \cup (4,a) \cup (6,a) \cup (9,a)\}]$

$= \varepsilon\text{-Closure}[\phi \cup \phi \cup 5 \cup \phi \cup \phi]$

$= \varepsilon\text{-Closure}[5]$

$= [3,4,5,6,8,9]$

when b : $\delta_D[(2,3,4,6,9), b] = \varepsilon\text{-Closure}[\delta_N[2,3,4,6,9], b]$

$= \varepsilon\text{-Closure}[\delta_N\{(2,b) \cup (3,b) \cup (4,b) \cup (6,b) \cup (9,b)\}]$

$= \varepsilon\text{-Closure}[7]$

$= [7,8,9,3,4,6] = [3,4,6,7,8,9]$

$Q_D = \{[0], [1], [2,3,4,6,9], [3,4,5,6,8,9], [3,4,6,7,8,9]\}$

Consider [3,4,5,6,8,9]

when b : $\delta_D[[3,4,5,6,8,9], b] = \varepsilon\text{-Closure}[\delta_N([3,4,5,6,8,9], b)]$

$= \varepsilon\text{-Closure}[\delta_N\{(3,b) \cup (4,b) \cup (5,b) \cup (6,b) \cup (8,b) \cup (9,b)\}]$

$= \varepsilon\text{-Closure}[7]$

$= [3,4,6,7,8,9]$

when a : $\delta_D[[3,4,5,6,8,9], a] = \varepsilon\text{-Closure}[\delta_N([3,4,5,6,8,9], a)]$

$= \varepsilon\text{-Closure}[\delta_N\{(3,a) \cup (4,a) \cup (5,a) \cup (6,a) \cup (8,a) \cup (9,a)\}]$

$= \varepsilon\text{-Closure}[5]$

$= [3,4,5,6,8,9]$

Consider [3, 4, 6, 7, 8, 9]

On a:

$\delta_D[[3,4,6,7,8,9], a] = \varepsilon\text{-Closure}[\delta_N([3,4,6,7,8,9], a)]$

$= \varepsilon\text{-Closure}[\delta_N\{(3,a) \cup (4,a) \cup (6,a) \cup (7,a) \cup (8,a) \cup (9,a)\}]$

$= \varepsilon\text{-Closure}[5]$

$= [3,4,5,6,8,9]$

On b:

$\delta_D[[3,4,6,7,8,9], b] = \varepsilon\text{-Closure}[\delta_N[[3,4,6,7,8,9], b]]$

$= \varepsilon\text{-Closure}[\delta_N\{(3,b) \cup (4,b) \cup (6,b) \cup (7,b) \cup (8,b) \cup (9,b)\}]$

$= \varepsilon\text{-Closure}[7]$

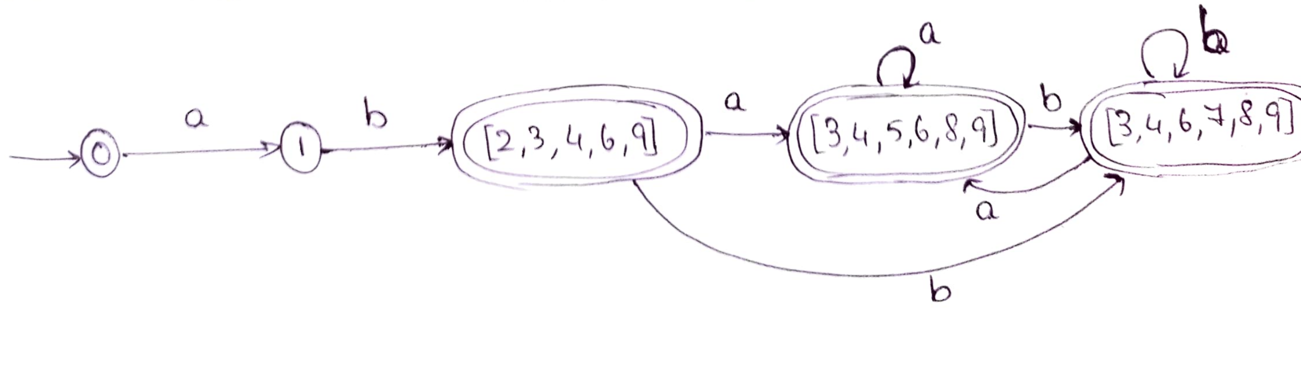
$= [3,4,6,7,8,9]$

Now, we have

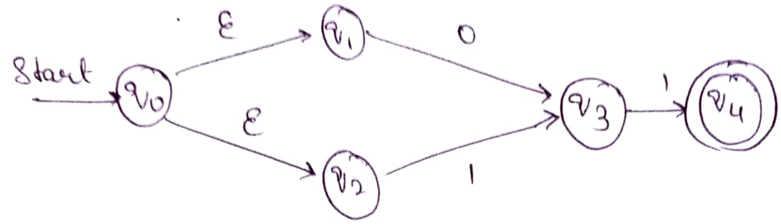
$Q_D = \{[0], [1], [2,3,4,6,9], [3,4,5,6,8,9], [3,4,6,7,8,9]\}$

Table

	δ	a	b
→	[0]	[1]	-
	[1]	-	[2,3,4,6,9]
*	[2,3,4,6,9]	[3,4,5,6,8,9]	[3,4,6,7,8,9]
*	[3,4,5,6,8,9]	[3,4,5,6,8,9]	[3,4,6,7,8,9]
*	[3,4,6,7,8,9]	[3,4,6,5,8,9]	[3,4,6,7,8,9]



2) Convert the ε NDFSM into Equivalent DFSM.



Sol → Let us obtain ε closures of each other.

ε - Closure $(q_0) = [q_0, q_1, q_2]$ — ①

ε-Closure $(q_1) = [q_1]$

ε- Closure $[q_2] = [q_2]$

ε- Closure $[q_3] = [q_3]$

ε- Closure $[q_4] = [q_4]$

\# Consider $[q_0, q_1, q_2]$: $Q_D = [q_0, q_1, q_2]$

On input symbol '0' and '1'

when 0, $\delta_D[[q_0, q_1, q_2], 0] = \varepsilon\text{-Closure}[\delta_N([q_0, q_1, q_2], 0)]$

$= \varepsilon\text{-Closure}[\delta_N\{(q_0, 0) \cup (q_1, 0) \cup (q_2, 0)\}]$

$= \varepsilon\text{-Closure}[\phi \cup q_3 \cup \phi]$

$= \varepsilon\text{-Closure}[q_3] = [q_3]$

when 1, $\delta_D[(q_0, q_1, q_2), 1] = \varepsilon\text{-Closure}[\delta_N([q_0, q_1, q_2], 1)]$

$= \varepsilon\text{-Closure}[\delta_N\{(q_0, 1) \cup (q_1, 1) \cup (q_2, 1)\}]$

$= \varepsilon\text{-Closure}[\phi \cup \phi \cup q_3]$

$= [q_3]$

$Q_D = \{[q_0, q_1, q_2], [q_3]\}$.

\# consider $[q_3]$

when input symbol is 0 : $\delta_D([q_3], 0) = \varepsilon\text{-Closure}(\delta_D([q_3], 0))$

$= \varepsilon\text{-Closure}(\phi)$

$= \phi$

When input symbol is 1

$$\delta_D([q_3],1) = \varepsilon\text{-Closure}(\delta_N([q_3],1))$$
$$= \varepsilon\text{-Closure}(q_4)$$
$$= [q_4]$$

$Q_D = \{[q_0, q_1, q_2], [q_3], [q_4]\}$.

Consider $[q_4]$

when input symbol is 0 : $\delta_D([q_4],0) = \varepsilon\text{-Closure}[(\delta_D(q_4),0)]$
$= \varepsilon\text{-Closure}\{\phi\}$
$= \phi$.

when input symbol is 1 : $\delta_D[[q_4],1] = \varepsilon\text{-Closure}[\delta_D[q_4],1]$
$= \varepsilon\text{-Closure}(\phi)$
$= \phi$.

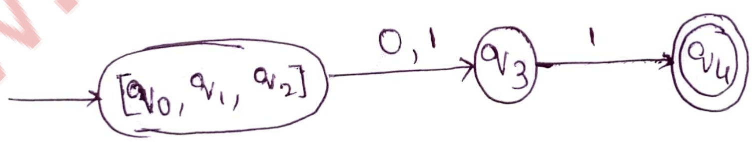
We have.

$Q_D = \{[q_0, q_1, q_2], [q_3], [q_4]\}$

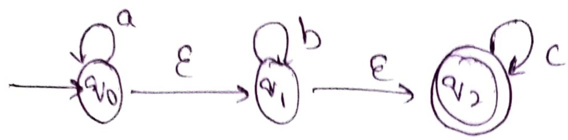
Transition Table.

	δ	0	1
$\rightarrow$	$[q_0, q_1, q_2]$	q_3	q_3
	$[q_3]$	-	q_4
$*$	$[q_4]$	-	-

Transition Diagram of Obtained DFSM.



3) Convert ε NFSM into equivalent DFSM.



Sol →

ε- Closure $(q_0) = [q_0, q_1, q_2]$

ε- Closure $(q_1) = [q_1, q_2]$

ε- Closure $(q_2) = [q_2]$

Starting with ε- Closure $(q_0) = [q_0, q_1, q_2]$

$Q_D = [q_0, q_1, q_2]$

H consider $[q_0, q_1, q_2]$

On input signal a

$\delta_D[[q_0, q_1, q_2], a] = \varepsilon\text{- Closure}(\delta_N([q_0, q_1, q_2], a))$

$= \varepsilon\text{- Closure}(\delta_N\{(q_0, a) \cup (q_1, a) \cup (q_2, a)\})$

$= \varepsilon\text{- Closure}(q_0 \cup \phi \cup \phi)$

$= \varepsilon\text{- Closure}(q_0) = [q_0, q_1, q_2]$

On iput signal b

$\delta_D[[q_0, q_1, q_2], b] = \varepsilon\text{- Closure}(\delta_N([q_0, q_1, q_2], b))$

$= \varepsilon\text{- Closure}(\delta_N\{(q_0, b) \cup (q_1, b) \cup (q_2, b)\})$

$= \varepsilon\text{- Closure}(\phi \cup q_1 \cup \phi)$

$= \varepsilon\text{- Closure}(q_1) = \varepsilon[q_1] = [q_1, q_2]$

On inpout signal c

$\delta_D[[q_0, q_1, q_2], c] = \varepsilon\text{- Closure}(\delta_N([q_0, q_1, q_2], c])$

$= \varepsilon\text{- Closure}(\delta_N\{(q_0, c) \cup (q_1, c) \cup (q_2, c)\})$

$= \varepsilon\text{- Closure}(\phi \cup \phi \cup q_2)$

$= \varepsilon\text{- Closure}(q_2) = [q_2]$

Now,

$Q_D = \{[q_0, q_1, q_2], [q_1, q_2], [q_2]\}$

Now Consider $[q_1, q_2]$

when on input a

$$\delta_D[[q_1, q_2], a] = \varepsilon\text{-Closure}((\delta_N(q_1, q_2), a))$$
$$= \varepsilon\text{-Closure}(\delta_N\{(q_1, a) \cup (q_2, a)\})$$
$$= \varepsilon\text{-Closure}(\phi \cup \phi) = [\phi]$$

when on input b

$$\delta_D[[q_1, q_2], b] = \varepsilon\text{-Closure}((\delta_N([q_1, q_2], b))$$
$$= \varepsilon\text{-Closure}(\delta_N\{(q_1, b) \cup (q_2, b)\})$$
$$= \varepsilon\text{-Closure}(q_1 \cup \phi)$$
$$= \varepsilon\text{-Closure}(q_1) = [q_1, q_2]$$

when on input c

$$\delta_D[(q_1, q_2), c] = \varepsilon\text{-Closure}(\delta_N((q_1, q_2), c))$$
$$= \varepsilon\text{-Closure}(\delta_N\{(q_1, c) \cup (q_2, c)\})$$
$$= \varepsilon\text{-Closure}(\phi \cup q_2)$$
$$= [q_2]$$

Since no new states are obtained Q_D remains same.

Consider state $[q_2]$

On input symbol a

$$\delta_D([q_2, a]) = \varepsilon\text{-Closure}(\delta_N[[q_2], a])$$
$$= \varepsilon\text{-Closure}(\phi)$$
$$= [\phi]$$

on b.

$$\delta_D[[q_2], b] = \varepsilon\text{-Closure}(\delta_N[[q_2], b])$$
$$= \varepsilon\text{-Closure}(\phi)$$
$$= \phi$$

On input symbol c.

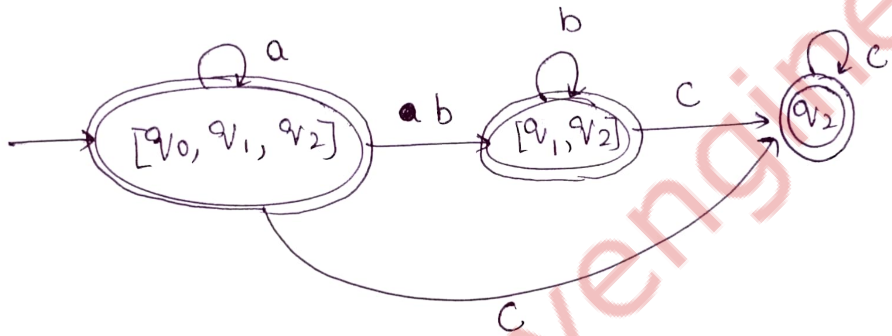
$$\delta_D([q_2], c) = \varepsilon\text{-Closure}(\delta_N(q_2, c))$$
$$= \varepsilon\text{-Closure}(q_2)$$
$$= [q_2]$$

$$Q_D = \{[q_0, q_1, q_2], [q_1, q_2], [q_2]\}$$

Transition Table

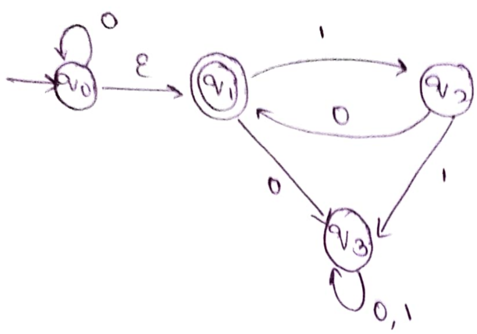
	δ	a	b	c
→*	[q_0, q_1, q_2]	[q_0, q_1, q_2]	[q_1, q_2]	[q_2]
*	[q_1, q_2]	-	[q_1, q_2]	[q_2]
*	[q_2]	-	-	[q_2]

Transition DFSM diagram



Jan 2020

4) Convert ϵ-NFSM to DFSM.



Sol Starting state: $\epsilon\text{-closure}(q_0) = [q_0, q_1]$

$Q_D = [q_0, q_1]$

\# Consider $[q_0, q_1]$

when input is 0

$$\delta_D([q_0,q_1],0) = \epsilon\text{-closure}(\delta_N([q_0,q_1],0))$$
$$= \epsilon\text{-closure}(\delta_N\{(q_0,0) \cup (q_1,0)\})$$
$$= \epsilon\text{-closure}(q_0 \cup q_3)$$
$$= [q_0, q_1, q_3]$$

when input is 1

$$\delta_D[[q_0,q_1],1] = \epsilon\text{-closure}(\delta_N[(q_0,q_1),1])$$
$$= \epsilon\text{-closure}(\delta_N\{(q_0,1) \cup (q_1,1)\})$$
$$= \epsilon\text{-closure}(\phi \cup q_2)$$
$$= [q_2]$$

\# Consider $[q_2]$

when input is 0, $\delta_D(q_2,0) = \epsilon\text{-closure}(\delta_N(q_2,0))$

$$= \epsilon\text{-closure}(q_1)$$
$$= [q_1]$$

when input is 1, $\delta_D(q_2,1) = \epsilon\text{-closure}(\delta_N(q_2,1))$

$$= \epsilon\text{-closure}(q_3)$$
$$= [q_3]$$

\# Consider $[q_1]$

when input is 0, $\delta_D(q_1,0) = \epsilon\text{-closure}(\delta_N(q_1,0))$

$$= \epsilon\text{-closure}(q_3)$$
$$= [q_3]$$

when input is 1, $\delta_D(q_1,1) = \epsilon\text{-closure}(\delta_N(q_1,1))$

$$= \epsilon\text{-closure}(q_2)$$
$$= [q_2]$$

Consider q_3

when input is 0 : $\delta_D(q_3, 0) = \epsilon\text{-closure}(\delta_N(q_3, 0))$
$= \epsilon\text{-closure}(q_3)$
$= [q_3]$

when input is 1 : $\delta_D(q_3, 1) = \epsilon\text{-closure}(\delta_N(q_3, 1))$
$= \epsilon\text{-closure}(q_3)$
$= [q_3]$

consider $[q_0 q_1 q_3]$

when input is 0

$\delta_D[(q_0, q_1, q_3), 0] = \epsilon\text{-closure}(\delta_D((q_0, q_1, q_3), 0))$
$= \epsilon\text{-closure}(\delta_D\{(q_0, 0) \cup (q_1, 0) \cup (q_3, 0)\})$
$= \epsilon\text{-closure}(q_0 \cup q_3 \cup q_3)$
$= \epsilon\text{-closure}(q_0 \cup q_3) = \epsilon\text{-closure}(q_0, q_3)$
$= [q_0, q_1, q_3]$

when input is 1

$\delta_D[(q_0, q_1, q_3), 1] = \epsilon\text{-closure}[\delta_D((q_0, q_1, q_3), 1)]$
$= \epsilon\text{-closure}(\delta_D\{(q_0, 1) \cup (q_1, 1) \cup (q_3, 1)\})$
$= \epsilon\text{-closure}(\phi \cup q_2 \cup q_3)$
$= [q_2, q_3]$

consider $[q_2, q_3]$

when input is 0

$\delta_D[(q_2, q_3), 0] = \epsilon\text{-closure}[\delta_D((q_2, q_3), 0)]$
$= \epsilon\text{-closure}[\delta_D\{(q_2, 0) \cup (q_3, 0)\}]$
$= \epsilon\text{-closure}[q_1 \cup q_3]$
$= [q_1, q_3]$

when input is 1

$\delta_D[(q_2, q_3), 1] = \epsilon\text{-closure}(\delta_D((q_2, q_3), 1))$
$= \epsilon\text{-closure}(\delta_D\{(q_2, 1) \cup (q_3, 1)\})$
$= \epsilon\text{-closure}(q_3 \cup q_3)$
$= [q_3]$

consider $[q_1, q_3]$

when on 0

$\delta_D[(q_1, q_3), 0] = \epsilon\text{-closure}(\delta_D((q_1, q_3), 0))$
$= \epsilon\text{-closure}(\delta_D\{(q_1, 0) \cup (q_3, 0)\})$
$= \epsilon\text{-closure}(q_3 \cup q_3)$
$= [q_3,]$

when 0,1

$\delta_D[(q_1, q_3), 1] = \varepsilon\text{-Closure}(\delta_D((q_1, q_3), 1))$

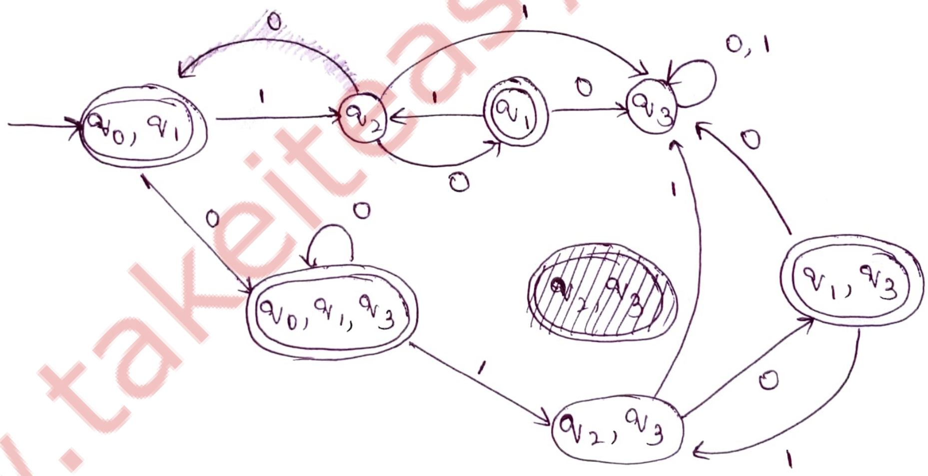
$= \varepsilon\text{-Closure}(\delta_D\{(q_1, 1) \cup (q_3, 1)\})$

$= \varepsilon\text{-Closure}(q_2 \cup q_3)$

$= [q_2, q_3].$

$Q_D = \{[q_0, q_1], [q_0, q_1, q_3], [q_2], [q_1], [q_3], [q_2, q_3], [q_1, q_3]\}$

	δ	0	1
$\rightarrow *$	$[q_0, q_1]$	$[q_0, q_1, q_3]$	$[q_2]$
	$[q_2]$	$[q_1]$	$[q_3]$
$*$	$[q_1]$	$[q_3]$	$[q_2]$
$*$	$[q_0, q_1, q_3]$	$[q_0, q_1, q_3]$	$[q_2, q_3]$
	$[q_3]$	$[q_3]$	$[q_3]$
	$[q_2, q_3]$	$[q_1, q_3]$	$[q_3]$
$*$	$[q_1, q_3]$	$[q_3]$	$[q_2, q_3]$



DFA Minimization using Myphill - Nerode Theorem

Input - DFA

Output - Minimized DFA

Step 1 : Draw a table for all pairs of states (Q_i, Q_j) not necessary connected directly [All are unmarked initially].

Step 2 : Consider every state pair (Q_i, Q_j) in DFA where $Q_i \in F$ and $Q_j \notin F$ & vice versa and mark them. [Here F is final states]

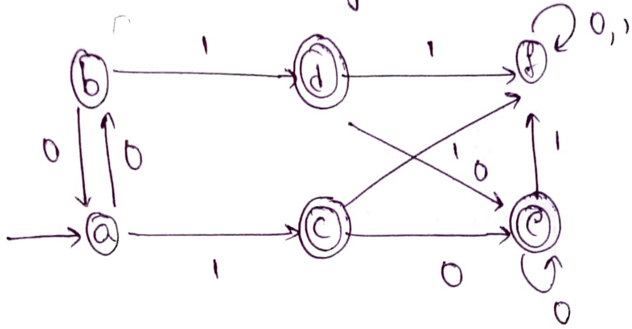
Step 3 : Repeat this step until we cannot mark anymore states -

If there is an unmarked pair (Q_i, Q_j) mark it if the pair $\{\delta(Q_i, A), \delta(Q_j, A)\}$ is marked for some input alphabet A.

Step 4 : Combine all the unmarked (Q_i, Q_j) and make them a single state in reduced DFA.

Examples

1) Minimize the following DFA



Step 1 : We write a table for all pair states. and mark the state pairs.

b					
c	✓	✓			
d	✓	✓			
e	✓	✓			
f	✓	✓	✓	✓	✓
	a	b	c	d	e

Step 2

Now, we will try to mark the state pairs with red colour checked mark. If input 1 to state (a, f) it will go to (c, f). (c, f) is already marked hence we will make pair (a, f).

δ	0	1
(a, b)	(b, a)	(c, d)
✓ (a, f)	(b, f)	(c, f)
(c, d)	(e, e)	(f, f)
(c, e)	(e, e)	(f, f)
✓ (b, f)	(a, f)	(d, f)
(d, e)	(e, e)	(f, f)

Now, After step 2, we have got state combinations {a,b}, {c,d}, {c,e}, {d,e} that are unmarked.

We can recombine {(c,d), (c,e), (d,e)} into (c,d,e)

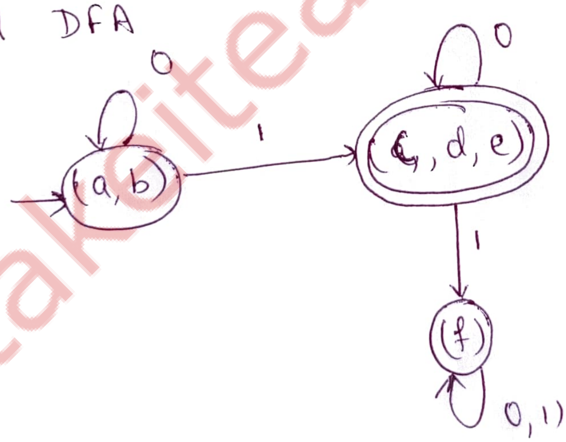
Hence we got two combined states as (a,b) and (c,d,e).

So, the final minimized DFA will contain three states
{f}, {a,b}, {c,d,e}

Transition Table

	δ	0	1
→	{a,b}	{a,b}	{c,d,e}
*	{c,d,e}	{c,d,e}	{f}
*	{f}	{f}	{f}

Minimized DFA



Q) Minimize the following DFSM.

Sol,

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

Step 1

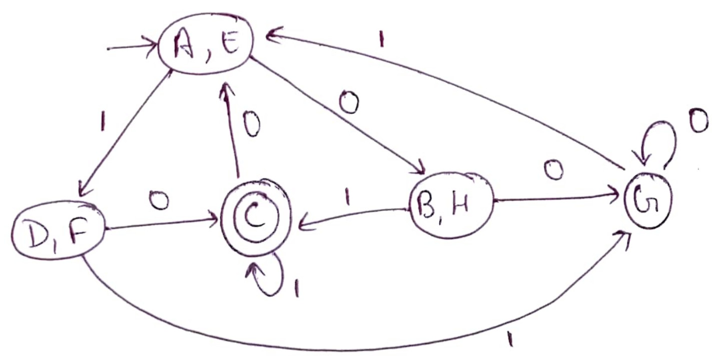
B	✓						
C	✓	✓					
D	✓	✓	✓				
E		✓	✓	✓			
F	✓	✓	✓		✓		
G	✓	✓	✓	✓	✓	✓	
H	✓		✓	✓	✓	✓	✓
	A	B	C	D	E	F	G.

→ (A,E), (B,H), (D,F) are the indistinguishable states

and

→ C and G are distinguishable states [i.e. completely filled states]

Minimized DFSM. [Check Next Page Transition Table]



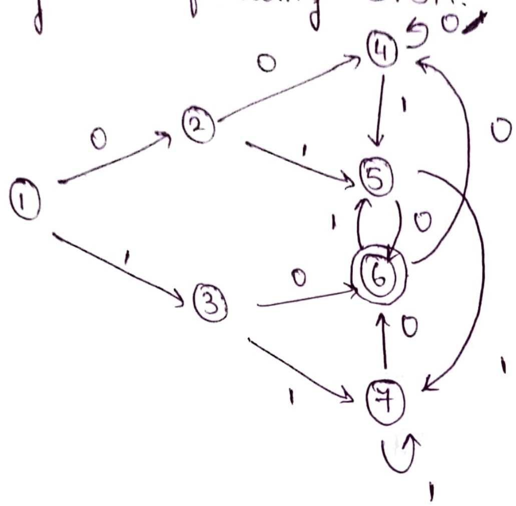
Step 2

(p, q)	(r, s)	(r, s)
δ	0	1
✓ (A, B)	(B, G)	(F, C)
✓ (A, D)	(B, C)	(F, G)
(A, E)	(B, H)	(F, F)
✓ (A, F)	(B, C)	(F, G)
✓ (A, G)	(B, G)	(F, E)
✓ (A, H)	(B, G)	(F, C)
✓ (B, D)	(G, C)	(C, G)
✓ (B, E)	(G, H)	(C, F)
✓ (B, F)	(G, G)	(C, G)
✓ (B, G)	(G, G)	(C, E)
(B, H)	(G, G)	(C, C)
✓ (D, E)	(C, H)	(G, F)
(D, F)	(C, C)	(G, G)
✓ (D, G)	(C, G)	(G, E)
✓ (D, H)	(C, G)	(G, C)
✓ (E, F)	(H, C)	(F, G)
✓ (E, G)	(H, G)	(F, E)
✓ (E, H)	(H, G)	(F, C)
✓ (F, G)	(C, G)	(G, E)
✓ (F, H)	(C, G)	(G, C)
✓ (G, H)	(G, G)	(E, C)

Transition Table of Minimized DFSM.

δ	0	1
→ (A, E)	(B, H)	(D, F)
(B, H)	[G]	[C]
(D, F)	[C]	[G]
☆ [C]	(A, E)	[C]
[G]	(G)	(A, E)

3) Minimize the following DFSM.



Sol →

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

Pairs marked with black(✓) are those of which excatly one element is in F (Final State)

The pairs marked in (✓) are those marked on second pass. Eg. (2,5) is one of these since $(2,5) \xrightarrow{0} (4,6)$ and pair (4,6) is already marked on pass 1 (✓)

Step 2

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

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

(1,2) (1,4) (2,4) (3,5) (3,7) (5,7) are the indistinguishable states. and {6} is the distinguishable state

We can combin

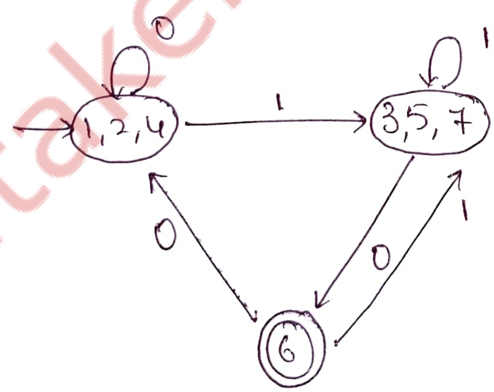
i) (1,2) (1,4) (2,4) = {1, 2, 4}
ii) (3,5) (3,7) (5,7) = {3, 5, 7}

} Indistinguishable Pairs

Transition Table of Minimized DFSM

	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}

Transition Diagram of Minimized DFSM



Jan 2018

4) Minimize the following DFSM.

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

Sol →

B	✓						
C	✓	✓					
D	✓	✓	✓				
E	✓	✓		✓			
F	✓		✓	✓	✓		
G		✓	✓	✓	✓	✓	
H	✓	✓	✓	✓	✓	✓	✓
	A	B	C	D	E	F	G.

(A,G), (B,F), (C,E) are indistinguishable pairs

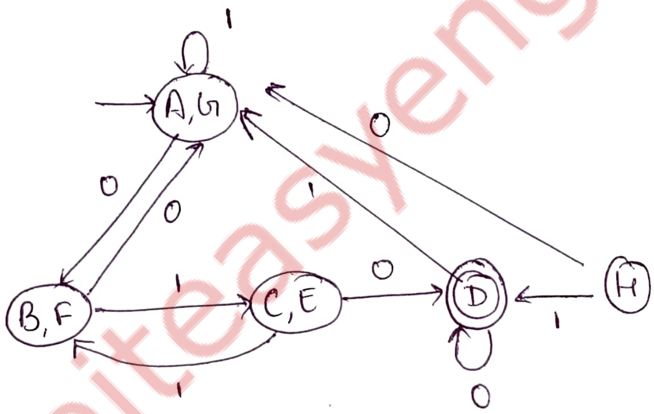
(D) and (H) are the distinguishable states.

δ (p,q)	0 (r,s)	1 (r,s)
(A,B)	(B,A)	(A,C)
(A,C)	(B,D)	(A,B)
(A,E)	(B,D)	(A,F)
(A,F)	(B,G)	(A,E)
(A,G)	(B,F)	(A,G)
(A,H)	(B,G)	(A,D)
(B,C)	(A,D)	(C,B)
(B,E)	(A,D)	(C,F)
(B,F)	(A,G)	(C,E)
(B,G)	(A,F)	(C,G)
(B,H)	(A,G)	(C,D)
(C,E)	(D,D)	(B,F)
(C,F)	(D,G)	(B,E)
(C,G)	(D,F)	(B,G)
(C,H)	(D,G)	(B,D)
(E,F)	(D,G)	(F,E)
(E,G)	(D,F)	(F,G)
(E,H)	(D,G)	(F,D)
(F,G)	(G,F)	(E,G)
(F,H)	(G,G)	(E,D)
(G,H)	(F,G)	(G,D)

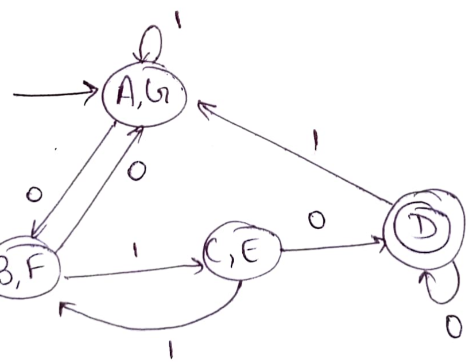
Transition Table

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

Minimized DFSM Transition Diagram



Now, since H is not reachable state to starting state (A,G)
H is removed.



Transducer : A device that converts variations in a physical quantity, such as pressure @ brightness, into an electrical signal @ vice versa.

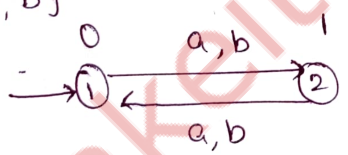
Moore Machine : A Moore Machine, M is a seven tuple: $(K, \Sigma, O, \delta, D, s, A)$ where

→ K is finite set of states
→ Σ is input alphabet
→ O is the output alphabet
→ $s \in K$ is the start state
→ A subset of K is the set of accepting states (not imp)
→ δ is the transition function it maps from $K \times \Sigma$ to K
→ D is the Display @ output function from $K \to (O)^*$

Example

In moore machine each state is associated with a output.
Suppose we want Moore machine to output '0' when the length of input string is even, otherwise output '1'.

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



Transition Table

Transition Function.	Input = a	Input = b	Output
1	2	2	0
2	1	1	1

Mealy Machine (Transducer)

A mealy machine, M is a six tuple
(K, Σ, O, S, s, A) where

$K \rightarrow$ finite set of states

$\Sigma \rightarrow$ Input alphabet

$O \rightarrow$ Output Alphabet

$s \in K$ is start state

A subset of K is the set of accepting states

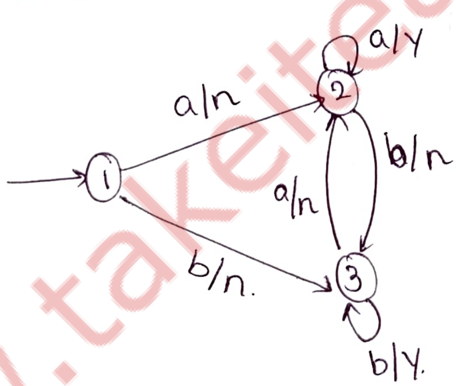
δ is the transition function it maps from $(K \times \Sigma)$ to $(K \times O^*)$

NOTE : Output is associated with each input.

Example

$L = \{w \in \{a,b\}^* \ w$ ends in aa @ bb$\}$

Requirements are when input ends aa it should have Y and when ends bb should have 'n'. output.



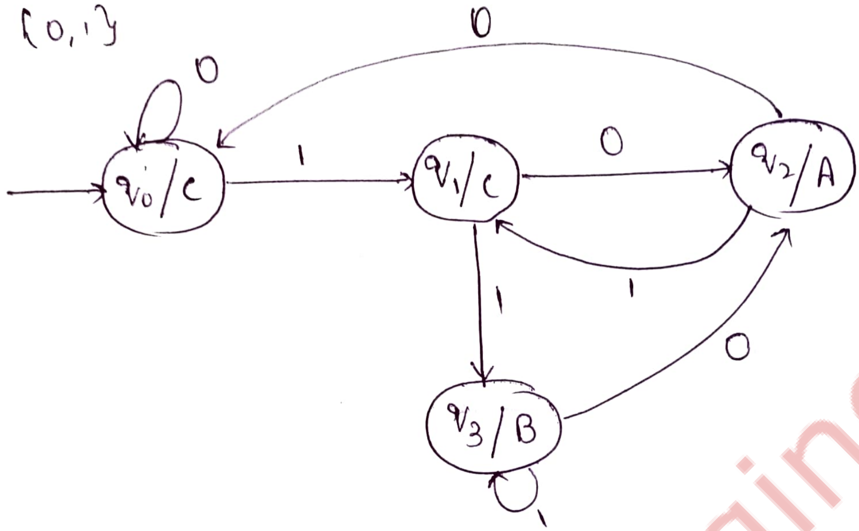
Transition Table

Trans function	a	O/P	b	O/P
1	2	n	3	n
2	2	Y	3	n
3	2	n	3	Y

Examples (Moore Machine)

1) Construct a Moore Machine that prints 'A' 'B' and 'C' depending on input that ends with '10', '11' @ other respectively

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

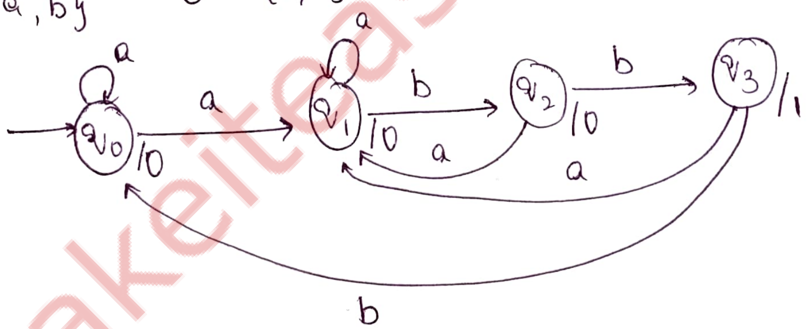


A, B, C are the outputs

2) Draw a moore machine that counts the occurrence of sequence 'abb' in any input string over {a,b}

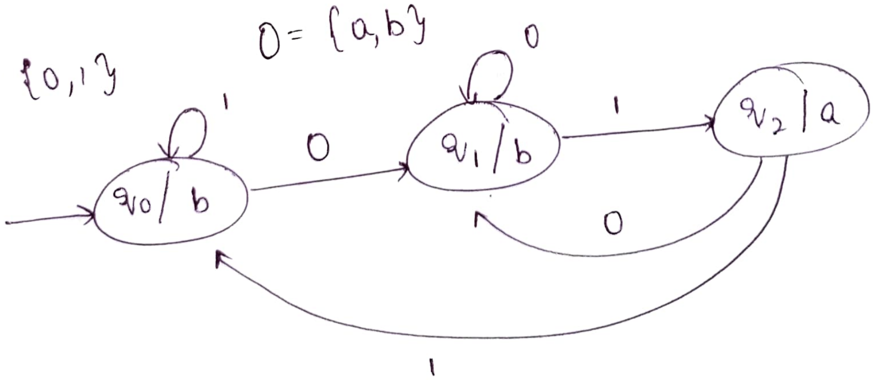
Sol, $\Sigma = \{a, b\}$ $O = \{0, 1\}$

0 and 1 are the outputs.



3) Construct a Moore Machine that prints 'a' whenever O/p sequence is '01' is encountered in input binary string.

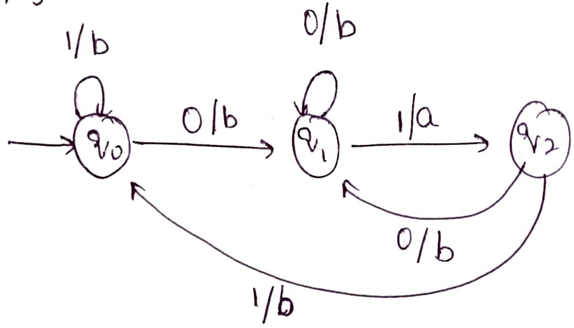
Ans $\Sigma = \{0, 1\}$ $O = \{a, b\}$



Examples (MEALY Machine)

1) Design a mealy machine that prints 'a' whenever the sequence '01' is encountered in binary input string.

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

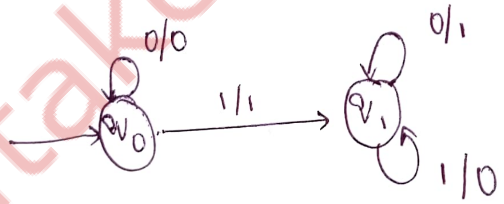


2) Design a mealy machine that takes input and produces 2's complement of that number as output. Assume the string is read from LSB to MSB and carry is discarded.

Sol. 2's Complement = 1's complement + 1

Ex

```
   10100            11100
→  01011         →  00011
+      1         +      1
---------        ---------
   001100           00100
```



3) Construct a mealy machine that produces 1's complement of any binary input string.

Sol. Input String : 00101

Output String : 11010