Finite Automata & Formal languages

- john Martin
- peterlinz

(1) languages

(2) Models of computation (Abstract Machine)

(3) Computability

↓

Finite Automata (DFA or NFA)

push down Automata (PDA)

Turning machines.

Monika .P. B.E., M.Tech.,

- all tokens are recognized by DFA.
- executing perticular algorithm is computability

structural Representations

(1) Grammar — recognized by PDA.

(2) Regular expressions. - lang recognized by DFA or NFA

Central Concept of Theory

(1) **Alphabets** :- $\Sigma = (0, 1)$

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

(2) **strings** — Σ 1011

Σ abb

— Empty string represented by $\{\}$, but we do it by ϵ or λ

Central concepts of Automata Theory 22 Feb 05

- Alphabets — finite non-empty symbols. $\Sigma = \{0,1\}$ - Binary alphabet. $\Sigma = \{a,b,\ldots z\}$ - lower case letters.
- strings
- Languages

Monika .P. B.E., M.Tech.,

Alphabets: finite non-empty set of symbols. denoted by Σ defined as follows

(i) $\Sigma = \{0,1\}$ - binary alphabet

(ii) $\Sigma = \{a,b \ldots z\}$ - lower case letters.

(iii) set of all ASCII characters or printable characters.

strings: is a finite sequence of symbols choosen from some alphabet Σ

eg: 110111 is a string over the alphabet $\Sigma = \{0,1\}$

(ii) the string abb over the alphabet $\Sigma = \{a,b\}$

(i) **Empty string**: An empty string is the string with zero occurances of symbols It is denoted by ϵ (Epsilon)

$|\epsilon| = 0$

(2) <u>length of a string</u> :-

the ——"——"——"—— —— w denoted by $|w|$ is the <u>no</u> of symbols in a string

$w = |110111| = 6$

$w = |abb| = 3$.

Monika .P. B.E., M.Tech.,

(3) <u>Reverse of a string</u> :-

The Reverse of a string w is obtained by writting the symbols in the reverse order.

If w is a string, then the reverse of a string is denoted by w^R where <u>R</u> indicates reverse.

$w^R = a_n a_{n-1} \cdots a_1$

(4) <u>Concatenation of strings</u> :-

The ——"——"—— 2 ——"—— x & y is a string obtained by appending the symbols y to the right end of x

i.e., if $x = a_1 a_2 \cdots a_n$

$y = b_1 b_2 \cdots b_m$

then x & y i.e.,

$xy = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m$

powers of an alphabet :-

If Σ is an —"—, we can Express the set of all strings of a certain length from that alphabet using the exponential notation denoted by Σ^k

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

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

$$\Sigma^3 = \{000, 001 \dots\}$$

Languages :- A set of strings all of which are choosen from some Σ^* is called a Language OR

It is defined as a subset of Σ^*

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

$$\Sigma^* = \{\epsilon, a, b, ab, aabb, aaabbb, \dots\}$$

$*$ indicates zero or more occurances of symbols.

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

The set {a, ab, aabb} is the language on Σ

Kleene closure (star closure) :-

$L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$

positive closure : $L^+ = L^1 \cup L^2 \cup L^3 \cup -$

Defining the languages in the form of notation [Set - former]

Monika .P. B.E., M.Tech.,

It is common to discribe a language using a set former as {w / something about w}

eg: (1) {w / w consists of an equal no of 0's & 1's }

(2) {w / w is a binary integer that is prime }

Unit - 1 · Introduction to Finite Automata

Finite Automata

→ " is a recognizer

(1) DFA [Deterministic Finite Automata] or FA :-

A ___ " ___ " ___ " ___ is a 5 - tuple system

tuple - 5 different parts :

def^n :- A DFA is a 5-tuple system 23/Feb/05

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

where
- Q is a finite set of states
- Σ is a finite input alphabet
- δ is a transition func^n mapping from $Q \times \Sigma \longrightarrow Q$
- q_0 is a special state called start state, $q_0 \in Q$
- F is a set of Final states, $F \subseteq Q$

---

operation of a DFA :- (working)

(1) At the initial time it is assumed to be in the initial state q_0

(2) During each move of Automata, the i/p mechanism advances one position to the right. So each move consumes one i/p symbol

(3) when end of the string is reached. the string is accepted if the automata is in one of the final state otherwise the string is rejected.

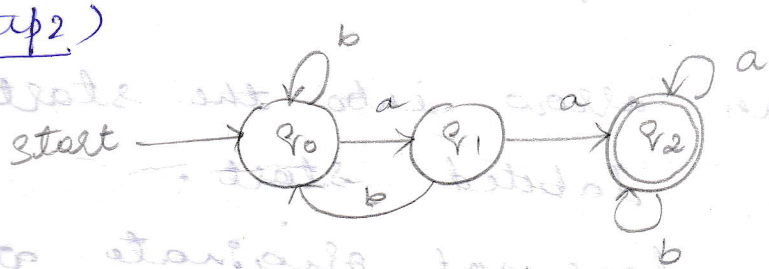
(4) the i/p mechanism can move only from left to right & reads exactly one symbol at a time

pattern recognition pbm

problem :- write a DFA which will recognize the set of all strings having a substring aa over the alphabet $\Sigma = (a, b)$

soln: **step 1)** the language $L = \{aa, abaab, abaa, aabb, \ldots\}$

step 2)



Monika .P. B.E., M.Tech.

Transition diagram

step 3) $\delta = Q \times \Sigma \rightarrow Q$

	δ	a	b
start →	q_0	q_1	q_0
	q_1	q_2	q_0
final → *	q_2	q_2	q_2

step 4 :- $M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$

Notations for DFA :-

Monika .P. B.E., M.Tech.,

(i) Transition diagrams :-

A ——"—————— for a DFA $M = (Q, \Sigma, \delta, q_0, F)$ is a graph defined as follows.

(a) for each state in Q, there is a node

(b) for each state q in Q & each i/p symbol a in Σ, let $\delta(q, a) = p$.

Then the transition diagram has an arc from node q to node p labeled a

(c) there is an arrow into the start state q_0 labeled start.

This arrow does not originate at any node

(d) Nodes corresponding to accepting states (those in F) are marked by double circles.

States not in F have a single circle.

(ii) <u>Transition table</u> :- It is a tabular representation of a func^n like δ, that takes 2 arguments & returns a value. The rows of the table correspond to the states & the coloumns correspond to the i/p.

abbaab

$$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_2$$

Extended Transition Function :-

If δ is our transition func^n, then the extended transition func^n is a func^n constructed from δ is called $\hat{\delta}$ (δ-hat).

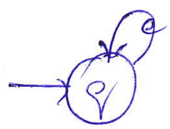
The extended transition func^n is a func^n that takes state $\underline{Q}$ & a string $\underline{w}$ & returns a state $\underline{p}$. where the Automata reaches when starting in the state Q & processing sequence of i/ps w.

we define $\hat{\delta}$ by the induction on the length of the i/p string as follows -

Monika .P. B.E., M.Tech.,

<u>Basis</u>:- $\hat{\delta}(q, \epsilon) = q$

i.e if we are in the state q & read no i/ps., then we still in state q.



<u>induction</u>:- suppose <u>w</u> is a string in the form xa i.e., a is the last symbol of <u>w</u>, & x is the string consisting of all but the last symbol.

for <u>eg</u>:- w = 1101 is broken into x = 110 & a = 1

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

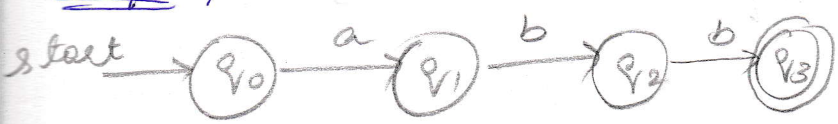
<u>problem</u>: (1) write a dfA to recognize the set of all strings ending in abb. over the alphabet $\Sigma = \{a, b\}$

pattern recognition pbm.

$L = \{abb, aababb, bb \cdots abb, aa \cdots abb, \cdots\}$

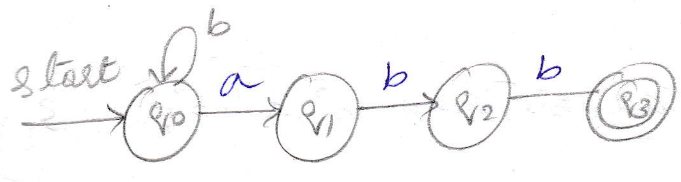
the minimum string that can be accepted by the machine is abb. It requires 4 states with q_0 as start state & q_3 as final state.

step 1 :-

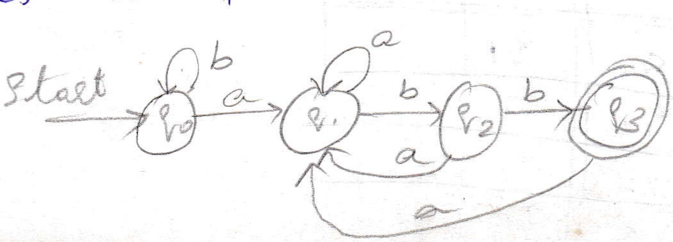


step 2 :- suppose if we have the i/p string bb -- abb, since the string ends with abb, so the entire string has to be accepted by the machine.

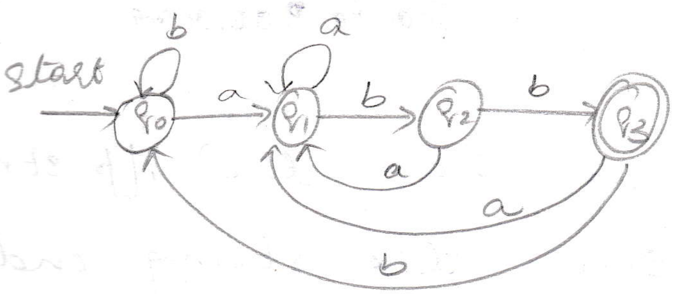
Hence any no of b's stays in state q_0 & machine enters into the final state if the string ends in abb.



step 3 :- If the machine is any one of the states q_1, q_2 & q_3. & if the current i/p symbol is a & if the next i/p symbol is bb, the entire string should be accepted.



step 4 :- In the state q_3 if the i/p symbol is b, then enter into the state q_0 so that if the next i/p is abb we enter into a final state q_3



Monika .P. B.E., M.Tech.

In the set notation, the line which accepted by dfa can be represented as

$L = \{(a+b)^* abb\}$

or

$L = \{(a+b)^n abb \mid n \geq 0\}$

$(a+b)^* = \{\epsilon, a, b, aa, bb, aaa, b^3, a^4, b^4, \ldots\}$

or $L = \{(a|b)^* abb\}$

or

$L = \{(a|b)^n abb \mid n \geq 0)$

Transition table

δ	Σ: a	b
start → q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_3
* q_3	q_1	q_0

Q

$M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_3\})$

② St the string bbabb is accepted.

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

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

$\hat{\delta}(q_0, bb) = \delta(\hat{\delta}(q_0, b), b) = \delta(q_0, b) = q_0$

$\hat{\delta}(q_0, bba) = \delta(\hat{\delta}(q_0, bb), a) = \delta(q_0, a) = q_1$

$\hat{\delta}(q_0, bbab) = \delta(\hat{\delta}(q_0, bba), b) = \delta(q_1, b) = q_2$

$\hat{\delta}(q_0, bbabb) = \delta(\hat{\delta}(q_0, bbab), b) = \delta(q_2, b) = q_3$

$\therefore$ since q_3 is final state, the string is accepted.

③ St the string aabab is rejected.

Monika .P. B.E., M.Tech,

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

$\hat{\delta}(q_0, a) = \delta(\hat{\delta}(q_0, \epsilon), a) = \delta(q_0, a) = q_1$

$\hat{\delta}(q_0, aa) = \delta(\hat{\delta}(q_0, a), a) = \delta(q_1, a) = q_1$

$\hat{\delta}(q_0, aab) = \delta(\hat{\delta}(q_0, aa), b) = \delta(q_1, b) = q_2$

$\hat{\delta}(q_0, aaba) = \delta(\hat{\delta}(q_0, aab), a) = \delta(q_2, a) = q_1$

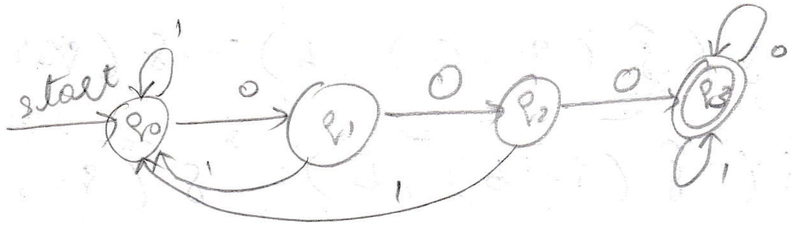
$\hat{\delta}(q_0, aabab) = \delta(\hat{\delta}(q_0, aaba), b) = \delta(q_1, b) = q_2$

ce q_2 is not final state, the string is rejected.

(3) write a dfa to recognize all the strings having 3 consecutive zero's over the alphabet $\Sigma = \{0,1\}$

pattern recognition prob

$L = \{000, 111000111, 0001111 \ldots, 111 \ldots 000 \ldots\}$



Here $\Sigma = 0,1$
so I need 2 transition state for each

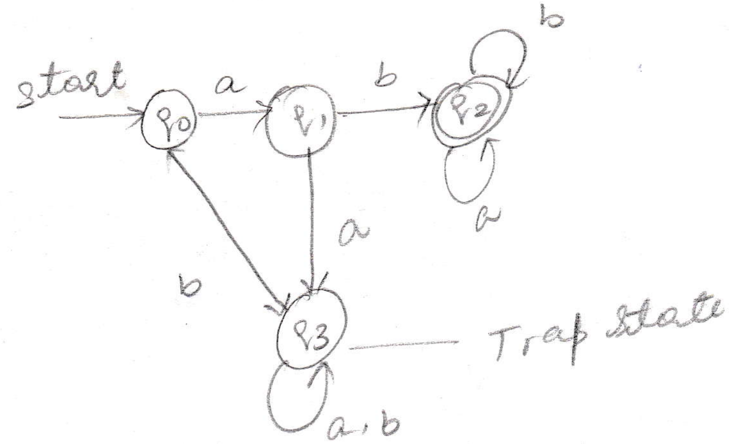
transition table

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

$M = (\{q_0, q_1, q_2, q_3\}, \{a,b\}, \delta, q_0, \{q_3\})$

(4) write a DFA to recognize all the strings having a prefix ab over the alphabet $\Sigma = \{a,b\}$.

$L = \{ab, abb, abaa, abb \ldots b, aba \ldots a\}$



Transition table

δ	a	b
q_0	q_1	q_3
q_1	q_3	q_2
q_2	q_2	q_2
q_3	q_3	q_3

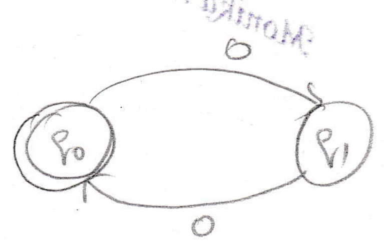
$M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_2\})$

---

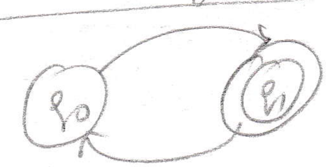
(5) write a DFA to recognize even no of 0's over the alphabet $\Sigma = \{0\}$.

Modulo K counter pbm

$L = \{\epsilon, 00, 0^4, 0^6$



for odd no's of zero's



---

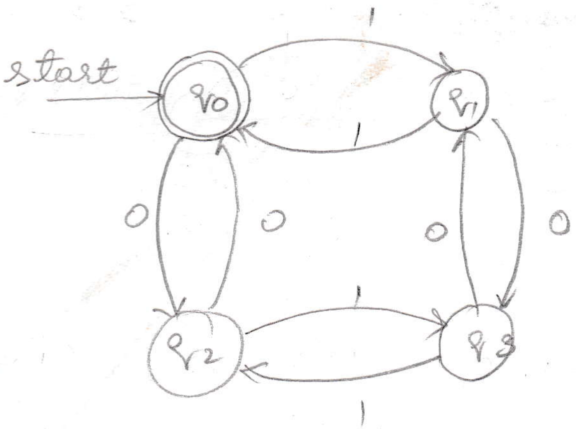
(6) write a DFA to recognize all the strings having even-no of 0's & even no of 1's over the alphabet $\Sigma = \{0, 1\}$

(ii) Even no of 0's & odd no of 1's

(iii) odd no of 0's & even no of 1's.

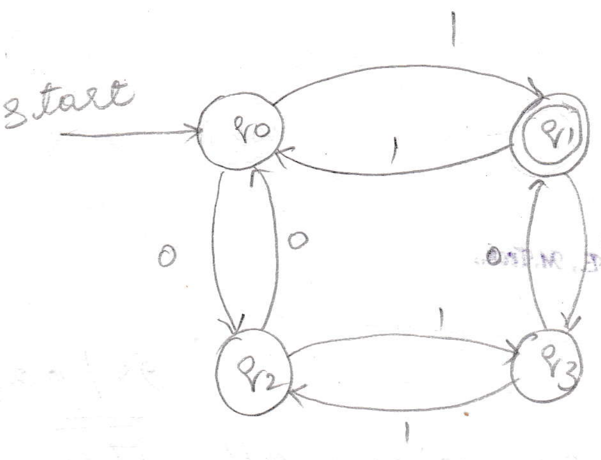
(iv) Both odd no of 0's & odd no of 1's

L = { ε, 0011, 1100, 0101, 1010 }



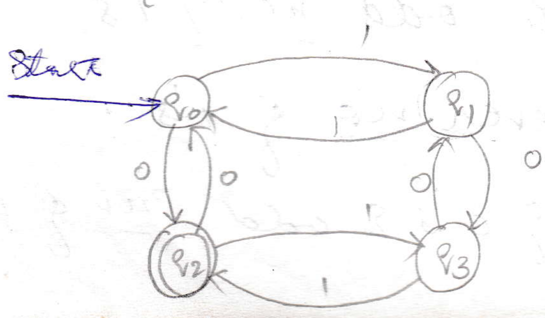
(ii) Even no of 0's & odd no of 1's

L = { [illegible], 001, 100, 010, ... }



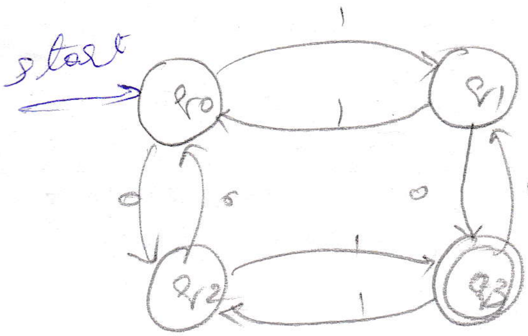
(iii) odd no of 0's & even no of 1's

L = { 110, 011, 101, ... }



(iv) odd no's of 0's & odd no's of 1's

L = {111000, 000111, 010101, 01010



St the string 101101 is accepted by the dfa where even no's of 0's & 1's.

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

$\hat{\delta}(q_0, 1) = \delta(\hat{\delta}(q_0, \epsilon), 1) = \delta(q_0, 1) = q_1$

$\hat{\delta}(q_0, 10) = \delta(\hat{\delta}(q_0, 1), 0) = \delta(q_1, 0) = q_3$

$\hat{\delta}(q_0, 101) = \delta(\hat{\delta}(q_0, 10), 1) = \delta(q_3, 1) = q_2$

$\hat{\delta}(q_0, 1011) = \delta(\hat{\delta}(q_0, 101), 1) = \delta(q_2, 1) = q_3$

$\hat{\delta}(q_0, 10110) = \delta(\hat{\delta}(q_0, 1011), 0) = \delta(q_3, 0) = q_1$

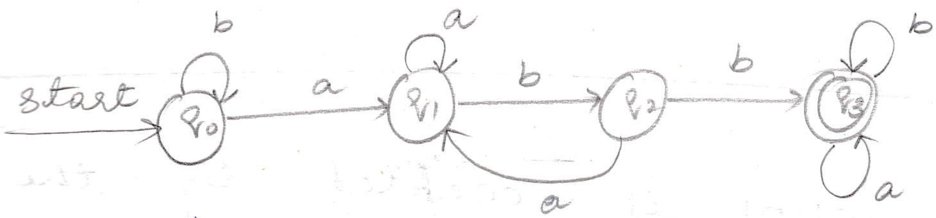
$\hat{\delta}(q_0, 101101) = \delta(\hat{\delta}(q_0, 10110), 1) = \delta(q_1, 1) = q_0$

q_0 is the final state so, the string is accepted

(2) write a dfa ~~to~~ recognize all the strings having a substring abb. over the alphabet $\Sigma = \{a, b\}$

Here abb can be prefix, suffix or infix.

$L = \{ abb, aabbb \ldots, aaa \ldots abb, aaabbb \ldots, \ldots \}$

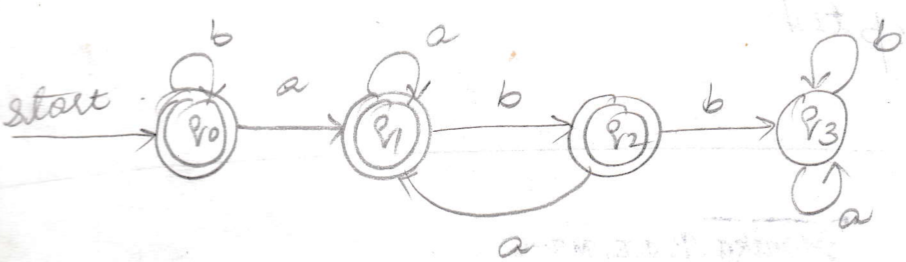


δ	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_3	q_3

(3) write a dfa, to recognize all the strings except those containing abb as a substring, over the alphabet $\Sigma = \{a, b\}$

$L = \{ abb, aabbb \ldots, aaa \ldots abb, aaabbb \ldots, \ldots \}$

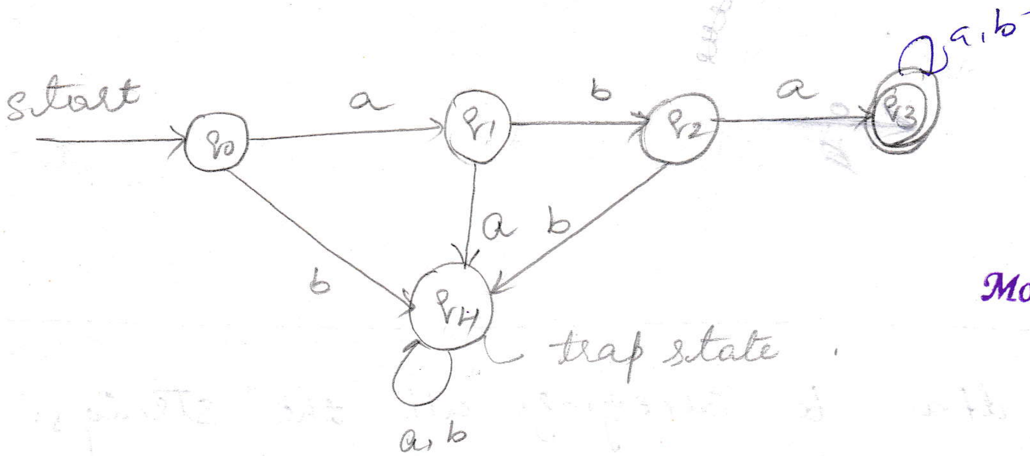
$\bar{L} = \Sigma^* - L$



4) write a dfa to recognize all the strings having a prefix aba. over the alphabet $\Sigma = \{a, b\}$

$L = \{ aba, abaaa\ldots, ababb\ldots, \ldots \}$.

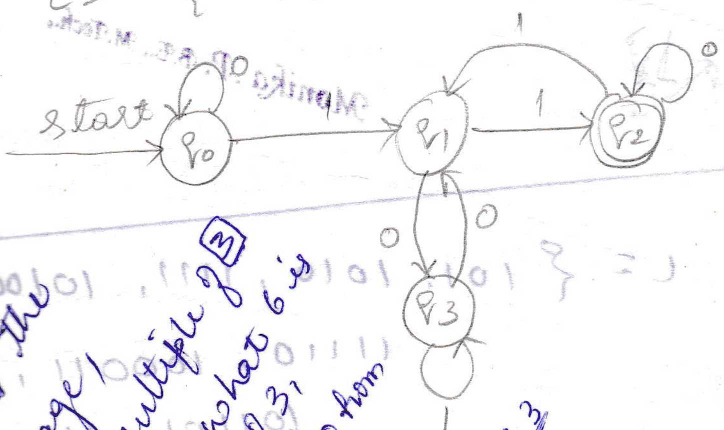
while constructing prefix we always have a trap state.



5) write a dfa to recognize all the strings when interpretted as a binary no is a multiple of 3. over the alphabet $\Sigma = \{0, 1\}$. The numbers must always begin with 1.

gible by k. pbms

$L = \{ 11, 110, 1001, 1100, 1111, \ldots \}$



understand the language, it is multiple of $\boxed{3}$ means, what 6 is multiple of 3, do u get 0 from 3 no.

u get 6 from 3

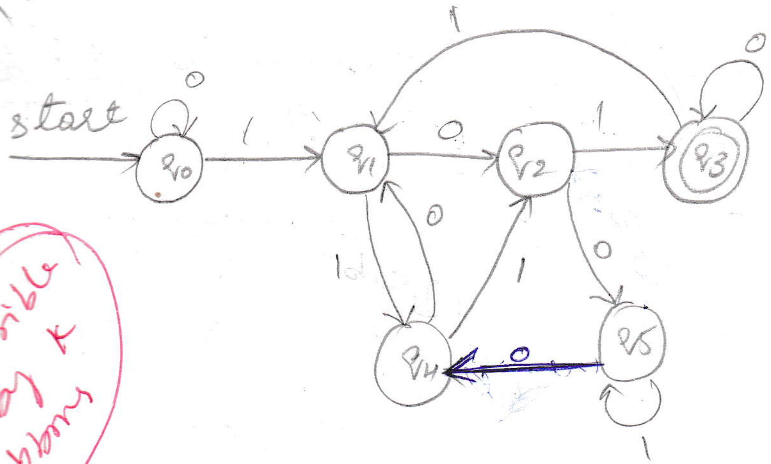
see u may think even 0 is a multiple of 3 & be careful, don't make 1st state q_0 as Final state if u make so, then u will accept ϵ but ϵ doesn't come into picture here.

⑥ Multiple of 5, always beginning with 1, if begins with 0, it should not be accepted

101101 = 45

L = { 101, 1010, 1111, 10100, 11001, 11110, 100010, 101000, }

10111 = 23



for Multiple of 3 we had q_0 to q_3

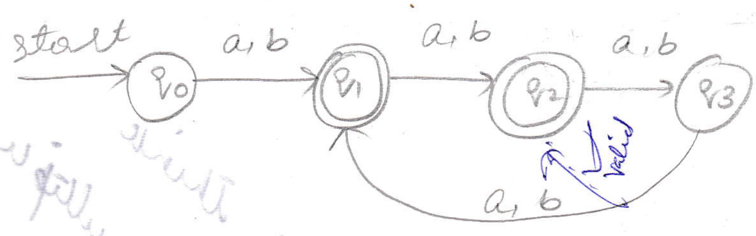
If we say multiple of 5 we will have q_0 to q_5.

Divisible by k problems

---

⑦ write a dfa to recognize all the strings given in the language $L = w \mid |w| \bmod 3 \neq 0$ over the alphabet $\Sigma = \{a, b\}$.

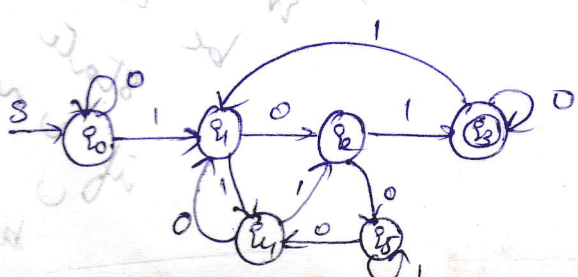
L = { ∈, a, b, aa, bb, ab, ba, ... }



Monika .P. B.E., M.Tech.,

Ans for Multiples of 5

L = { 101, 1010, 1111, 10100, 11001, 11110, 100011, 101000, 101101, 110010 }



100 - 1100100

200 - 11001000

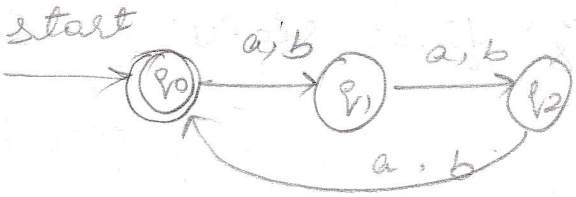
250 - 11111010

300 - 111110100

write a DFA for

(1) $L = \{ \omega \mid |\omega| \bmod 3 = 0 \}$ over $\Sigma = (a, b)$

$L = \{ \epsilon, aaa, bbb, aba, \ldots \}$



transition table

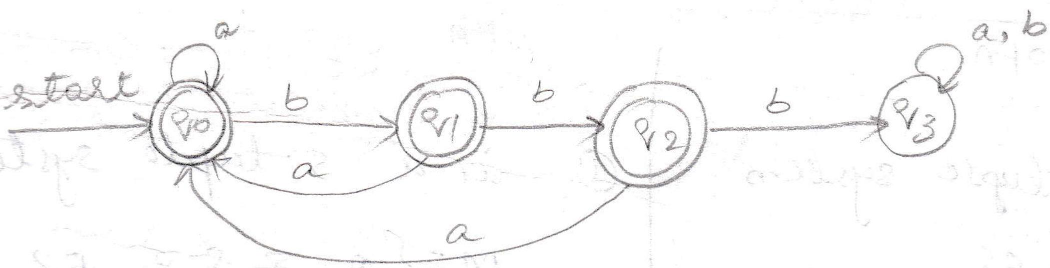
δ	a	b
q_0	q_1	q_1
q_1	q_2	q_2
q_2	q_0	q_0

$M = (\{ q_0, q_1, q_2 \}, \{a, b\}, \delta, \{q_0\}, q_0)$

---

(2) DFA to recognize all the strings having atmost 2 consecutive b's over the alphabet $\Sigma = \{a, b\}$.

$L = \{ \epsilon, b, bb, abb, \ldots \}$



Monika .P. B.E., M.Tech.,

Non-deterministic Finite Automata (NFA)

An NFA is a 5 tuple or Quintuple system $M = (Q, \Sigma, \delta, q_0, F)$

where Q is non Empty finite set of states.

Σ is non Empty, finite set of i/p alphabet

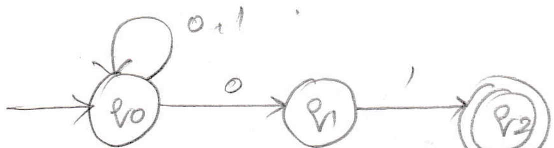
δ is a transition func^n which maps from

$\delta : Q \times \Sigma \Rightarrow 2^Q$.

$q_0 \in Q$ called start state.

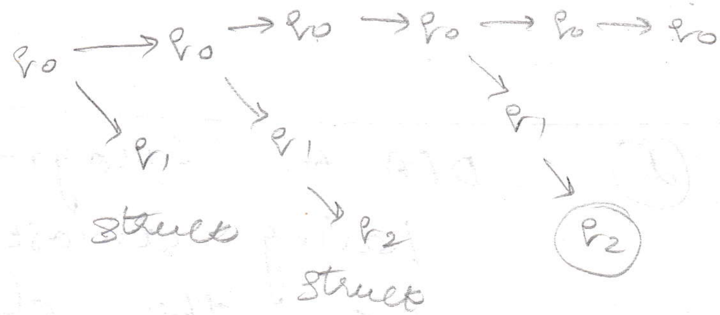
$F \subseteq Q$ is a set of final states. $\epsilon = \{\epsilon\}$

$\phi = \{\}$



$\omega = 00101$.

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



difference b/w NFA & DFA

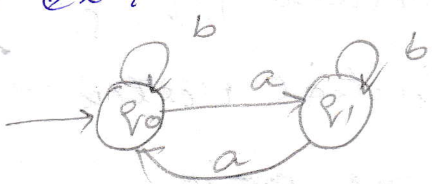
Monika .P. B.E., M.Tech.,

DFA	NFA → an Efficient Mechanism to describe some Complicated languages Concisely
① is a 5 tuple system defined as $M = \{Q, \Sigma, \delta, q_0, F\}$ $\delta : Q \times \Sigma \longrightarrow Q$	① is a 5 tuple system $M = \{Q, \Sigma, \delta, q_0, F\}$ But $\delta : Q \times (\Sigma \cup \epsilon) \Rightarrow 2^Q$
② there can be zero or 1 transition on an i/p symbol	② there can be zero, 1 or more Transitions on the same i/p symbol.

(3) no ϵ transitions Exists.

(4) It is difficult to construct but implementation is Easier

(5) ex :-



(6) Cannot guess about its i/p.

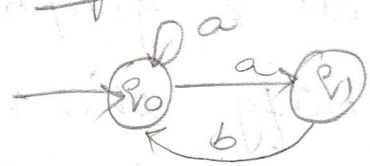
(7) Not very powerful.

(8) At a time, can be in only one state

(3) ϵ transitions can exist.

(4) It is Easy to construct & difficult to implement.

(5) eg :



(6) Have ability to guess something about its i/p.

(7) More powerful

(8) Has power to be in more than 1 state at one.

Equivalence b/w NFA & DFA

Theorem: Let L be the lang accepted by NFA, $M_n = \{Q_n, \Sigma, \delta_n, q_0, F_n\}$ then there exist a DFA

$M_D = (Q_D, \Sigma, \delta_D, q_0, F_D)$ such that

$L = \mathcal{L}(M_D)$.

proof: Given M_n we use the procedure / Method NFA to DFA to construct the transition graph G_D for M_D

Every vertex must have exactly length of Σ $|\Sigma|$ out going edges each labeled with diff elements of Σ.

2 Methods for NFA to DFA
1. Lazy Evaluation
2. Subset

During the constr some of the edges may be missing but the procedure continues until it is complete

Method 1 – Lazy Evaluation Method

procedure NFA to DFA

(1) Create a graph G_D with vertex $\{q_0\}$. Identify this vertex as a initial vertex

(2) Repeat the foll steps until no more edges are left.
take any vertex $\{q_i, q_j, \ldots q_k\}$ of G_D that has no outgoing edges for some a in Σ.

Monika .P. B.E., M.Tech.,

(a) compute $\hat{\delta}(q_i, a), \hat{\delta}(q_j, a) \ldots \hat{\delta}(q_k, a)$ Then form the union of all these $\hat{\delta}$ yielding the set $\{q_l, q_m, \ldots q_n\}$

(b) create a vertex for G_D labeled $\{q_l, q_m, \ldots q_n\}$ if it does not already exist.

(c) Add to G_D an edge from $\{q_i, q_j, \ldots q_k\}$ to $\{q_l, q_m, \ldots q_n\}$

q_1 label it with a

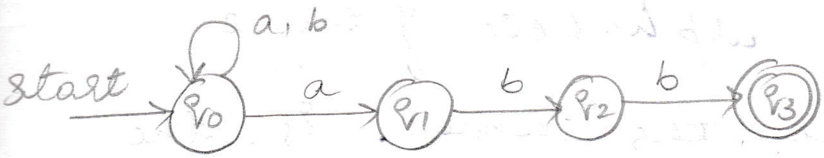
(3) Every state of G_D whose label contains q_m belongs to F_n is identified as a final vertex

(4) If M_N accepts ϵ, vertex q_0 in G_D is also made a final vertex.

problem

① Convert the foll NFA into an equivalent DFA.

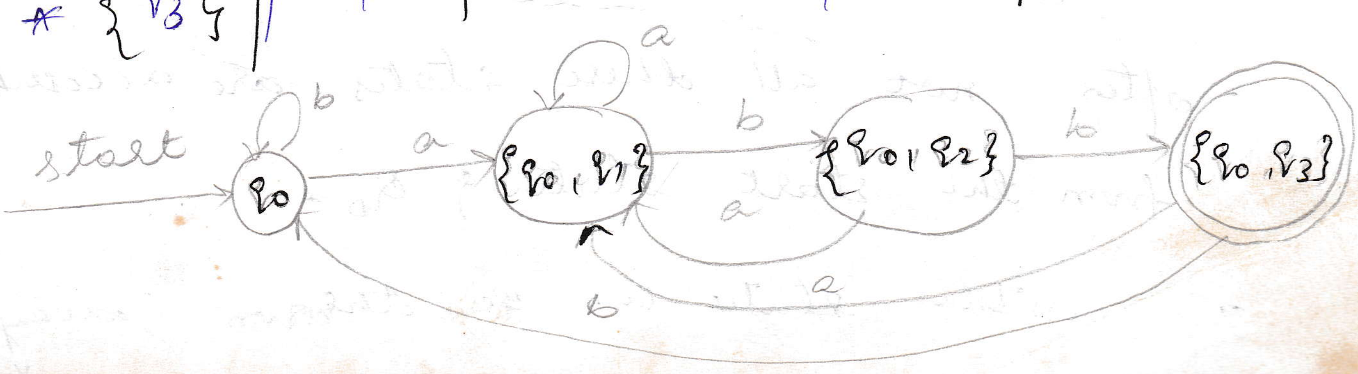
Lazy Evaluation Method



$L = \{(a|b)^* abb\}$

δ_{NFA}	a	b
start → $\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	ϕ	$\{q_2\}$
$\{q_2\}$	ϕ	$\{q_3\}$
* $\{q_3\}$	ϕ	ϕ

δ_{DFA}	a	b
→ $\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_3\}$
* $\{q_0, q_3\}$	$\{q_0, q_1\}$	$\{q_0\}$



procedure for NFA to DFA using subset construction :

method 2

- the subset ——— starts from an NFA

$N = \{Q_n, \Sigma, \delta_n, q_0, F_N\}$.

Its goal is the description of a DFA

$D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$

such that $L[D] = L[N]$

step 1 :- The i/p alphabets of the 2 automata are the same. & the start state of D is the set containing only the start state of N.

step 2 : Q_D is the set of subsets of Q_n ;

i.e., Q_D is the power set of Q_N

i.e, if $\left[\begin{array}{l} Q_N = n \text{ states} \\ Q_D = 2^n \text{ states} \end{array}\right]$.

often not all these states are accessible from the start state of Q_D.

Inaccessible states can be thrown away

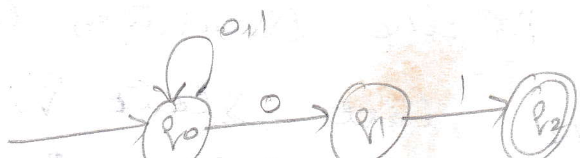
So effectively the no of states of D may be much smaller than 2^n.

step 3: F_D is the set of subsets S of Q_N such that $S \cap F_N \neq \phi$. i.e.,

F_D is all sets of NFA states that include at least one accepting state of N.

step4: for each set $S \subseteq Q_N$ & for each i/p symbol a in Σ $\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$

① Convert into DFA for the following NFA.



$N = 3$ states

$D = 2^3$ states

The subset construction produces a DFA with $2^3 = 8$ states corresponding to the 3 states of NFA.

The transition table for the 8 different states is as follows:

Monika .P. B.E., M.Tech.,

Subset construction Method

δ	0	1
ϕ	ϕ	ϕ
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	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	$\{q_2\}$
$* \{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

By renaming the states of 8 DFA as A, B, C, D, E, F, G, H

δ	a	b
A	A	A
$\rightarrow$B	E	B
C	A	D
*D	A	A
E	E	F
*F	E	B
*G	A	D
*H	E	F

States reachable from B (Start state are B, E, F, re all can b Elimi

of the 8 states in the transition table starting from the start state q_0, we can only reach states $\{q_0\}$, $\{q_0, q_1\}$ & $\{q_0, q_2\}$.

The other 5 states are inaccessible from the start state & may as well not be there.

We can avoid the exponential

..., of constructing transition table

entries

for every subset of states if we perform lazy evaluation on the subset as follows

Basis: W.K.T the single set consisting only of NFA start state is accessible

$\therefore Q_D = \{q_0\}$. write δNFA

Induction: step 1: Compute $\delta_D(S, a)$ for all accessible states. Since q_0 is the start state we find that

$\delta_D(\{q_0\}, 0) = \delta_N(q_0, 0)$

$= \{q_0, q_1\}$

$|||^{ly}$ $\delta_D(\{q_0\}, 1) = \delta_N(q_0, 1)$

$= \{q_0\}$

Monika .P. B.E., M.Tech.,

$\therefore$

δ DFA	0	1
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$

step 2 : One of the 2 sets we computed is old, $\{q_0\}$. However the other set $\{q_0, q_1\}$ is new & its transitions must be computed

$$\delta_D(\{q_0, q_1\}, 0) = \delta_N(q_0, 0) \cup \delta_N(q_1, 0)$$
$$= \{q_0, q_1\} \cup \phi$$
$$= \{q_0, q_1\}.$$

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

δ_{DFA}	0	1
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

Now compute the new state $\{q_0, q_2\}$

$$\delta_D(\{q_0, q_2\}, 0) = \delta_N(q_0, 0) \cup \delta_N(q_2, 0)$$
$$= \{q_0, q_1\} \cup \phi$$
$$= \{q_0, q_1\}$$

$$\delta_D(\{q_0, q_2\}, 1) = \delta_N(q_0, 1) \cup \delta_N(q_2, 1)$$

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

$$= \{q_0\}$$

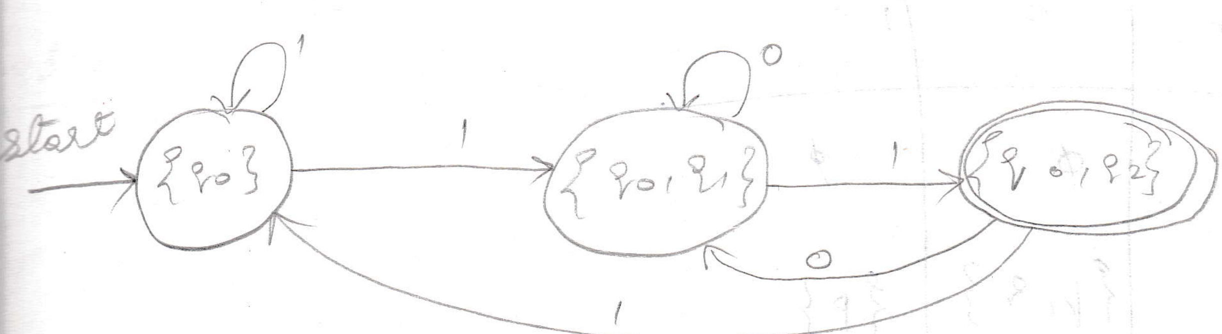
$$\therefore \delta_D(\{q_0, q_2\}, 1) = \{q_0\}.$$

Since from the transition table there are no new states $\therefore$ we stop computing.

$\therefore$ the final transition table & DFA is

δ_{DFA}	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 Evaluation Method



Minimized DFA

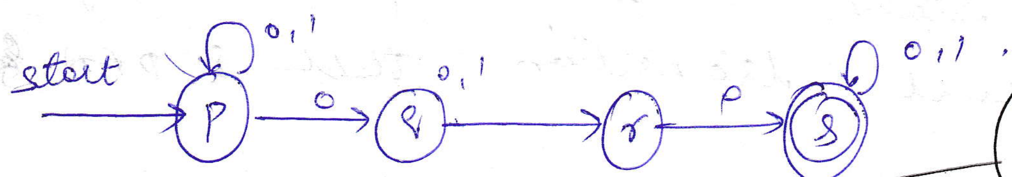
Convert to a DFA the following NFA

①

δ_{NFA}	0	1
$\rightarrow p$	$\{p, q\}$	$\{p\}$
q	$\{r\}$	$\{r\}$
r	$\{s\}$	ϕ
$* s$	$\{s\}$	$\{s\}$

Monika .P. B.E., M.Tech.

The given NFA is :



Subset construction Method

step 1 : The subset construction produces a DFA with $2^4 = 16$ states corresponding to the 4 states of NFA. The trans" table for 16 different states

δ_{NFA}	0	1
ϕ	ϕ	ϕ
$\rightarrow \{p\}$	$\{p, q\}$	$\{p\}$
$\{q\}$	$\{r\}$	$\{r\}$

	{r}	{s}	ϕ
+	{s}	{s}	{s}
✓	{p, q}	{p, q, r}	{p, r}
✓	{p, r}	{p, q, s}	{p}
✓	{p, s}	{p, q, s}	{p, s}
	{q, r}	{r, s}	{r}
	{q, s}	{r, s}	{r, s}
+	{r, s}	{s}	{s}
✓	{p, r, s}	{p, q, s}	{p, s}
✓	{p, q, r}	{p, q, r, s}	{p, r}
+	{q, r, s}	{r, s}	{r, s}
✓	{p, q, s}	{p, q, r, s}	{p, r, s}
✓	{p, q, r, s}	{p, q, r, s}	{p, r, s}

step 2 from start state accessible states are

{p}, {p, q}, {p, q, r}, {p, r}, {p, q, s}, {p, s}, {p, q, r, s}, {p, r, s} step 2 Basic: all know that single set consisting only of NFA start state is accessible

$\therefore Q_0 = \{p\} \rightarrow$ start

Lazy Evaluation Method

Induction

step 1: compute $\delta_D(S, a)$ for all accessible set

since, p is the start state

$\delta_D(\{p\}, 0) = \delta_N(\{p\}, 0)$

$= \{p, q\}$

$\delta_D(\{p\}, 1) = \delta_N(\{p\}, 1)$

$= \{p\}$

δ_{NFA}	0	1
$\{p\}$	$\{p, q\}$	$\{p\}$

A new state is found

step 2 compute transitions of $\{p, q\}$

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

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

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

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

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

$= \{p, r\}$

TT

δ_{DFA}	0	1
$\{p\}$	$\{p, q\}$	$\{p\}$
$\{p, q\}$	$\{p, q, r\}$	$\{p, r\}$

Step 3: compute transitions of $\{p, q, r\}$ & $\{p, r\}$

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

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

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

$$\delta_N(\{p, r\}, 1) = \{p\}$$

Transition table

δ DFA	0	1
$\{p\}$	$\{p, q\}$	$\{p\}$
$\{p, q\}$	$\{p, q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r, s\}$	$\{p, r\}$
$\{p, r\}$	$\{p, q, s\}$	$\{p\}$

step 4 compute transitions for $\{p,q,r,s\}$ & $\{p,q,s\}$

$\delta_D(\{p,q,r,s\},0) =$

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

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

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

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

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

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

Monika .P. B.E., M.Tech.,

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

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

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

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

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

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

δ_{DFA}	0	1
$\{p\}$	$\{p, q\}$	$\{p\}$
$\{p, q\}$	$\{p, q, r\}$	$\{p, r\}$
$\{p, q, r\}$	$\{p, q, r, s\}$	$\{p, r\}$
$\{p, r\}$	$\{p, q, s\}$	$\{p\}$
$\{p, q, r, s\}$	$\{p, q, r, s\}$	$\{p, r, s\}$
$\{p, q, s\}$	$\{p, q, r, s\}$	$\{p, r, s\}$

Step 5: compute transitions for $\{p, r, s\}$

$$\delta_D(\{p, r, s\}, 0) = \{p, q\} \cup \{s\} \cup \{s\}$$
$$= \{p, q, s\}$$

$$\delta_D(\{p, r, s\}, 1) = \{p\} \cup \{\phi\} \cup \{s\}$$
$$= \{p, s\}$$

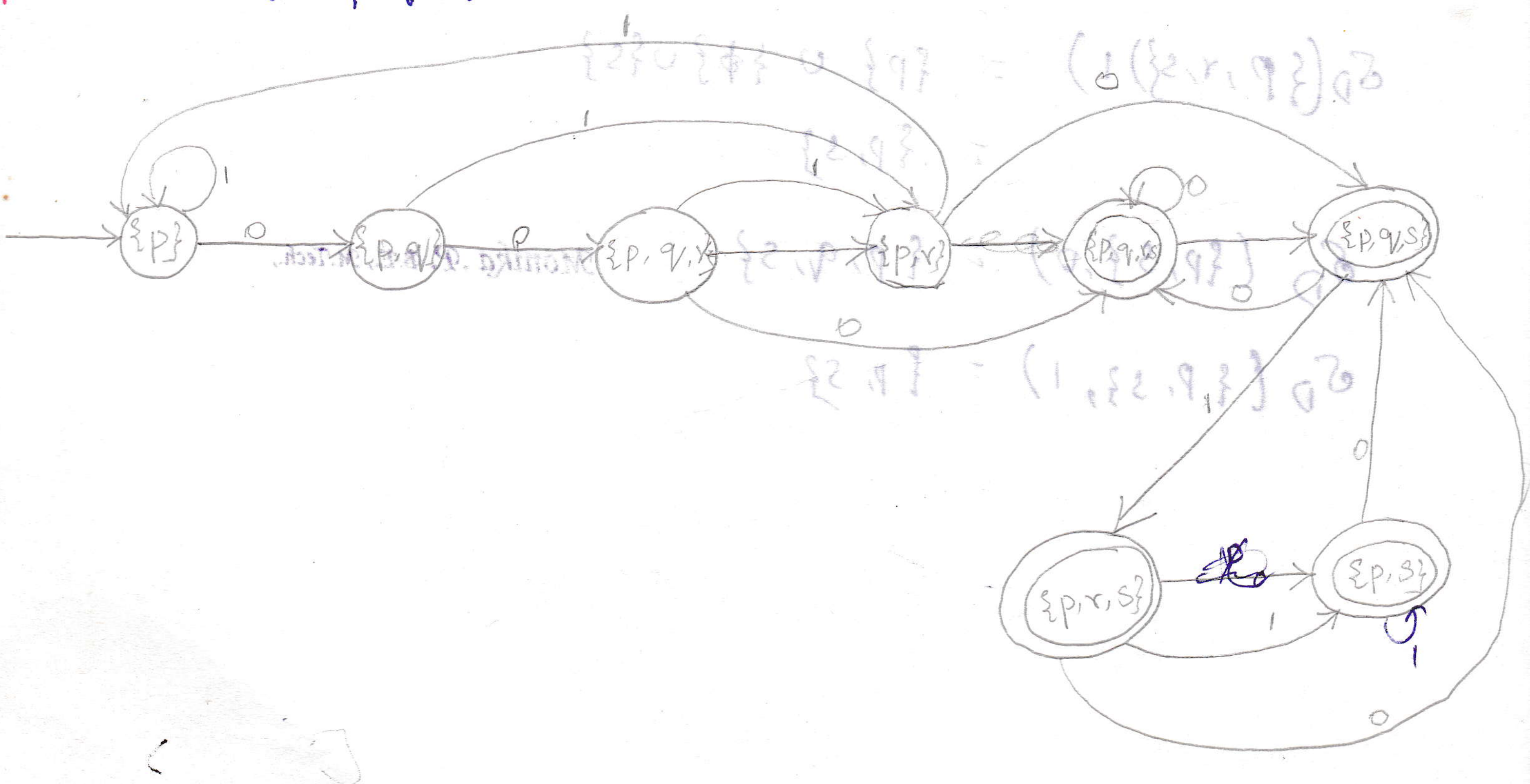
$$\delta_D(\{p, s\}, 0) = \{p, q, s\}$$

$$\delta_D(\{p, s\}, 1) = \{p, s\}$$

∴ Final transition table

δ_{DFA}	0	1
{p}	{p, q}	{p}
{p, q}	{p, q, r}	{p, r}
{p, q, r}	{p, q, r, s}	{p, r}
{p, r}	{p, q, s}	{p}
* {p, q, r, s}	{p, q, r, s}	{p, r, s}
* {p, q, s}	{p, q, r, s}	{p, r, s}
* {p, r, s}	{p, q, s}	{p, s}
* {p, s}	{p, q, s}	{p, s}

Solution via lazy Evaluation Method

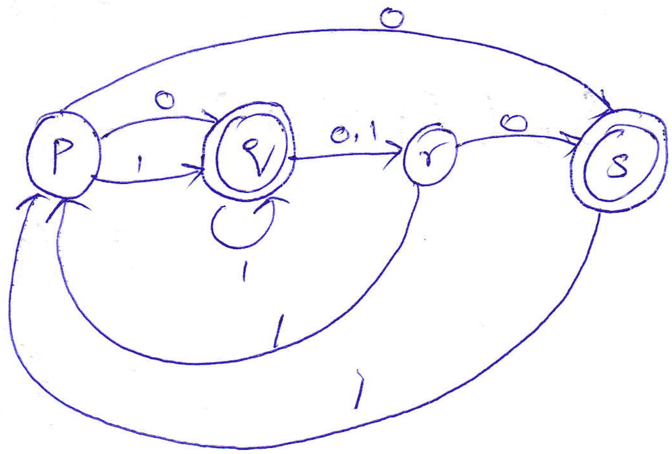


(2) convert to a DFA, the following NFA

Transition table

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

Transition diagram



NFA diagram

If we Apply subset construction algorithm

The subset construction produces a DFA with $2^4 = 16$ states corresponding to the 4 states of NFA. The transition table for 16 different states to be constructed, hence, let us solve using lazy evaluation method.

Mathematical Induction

Basis: start is common for both DFA & NFA

— start state for a DFA is,

δ	0	1
p	{q,s}	{q}

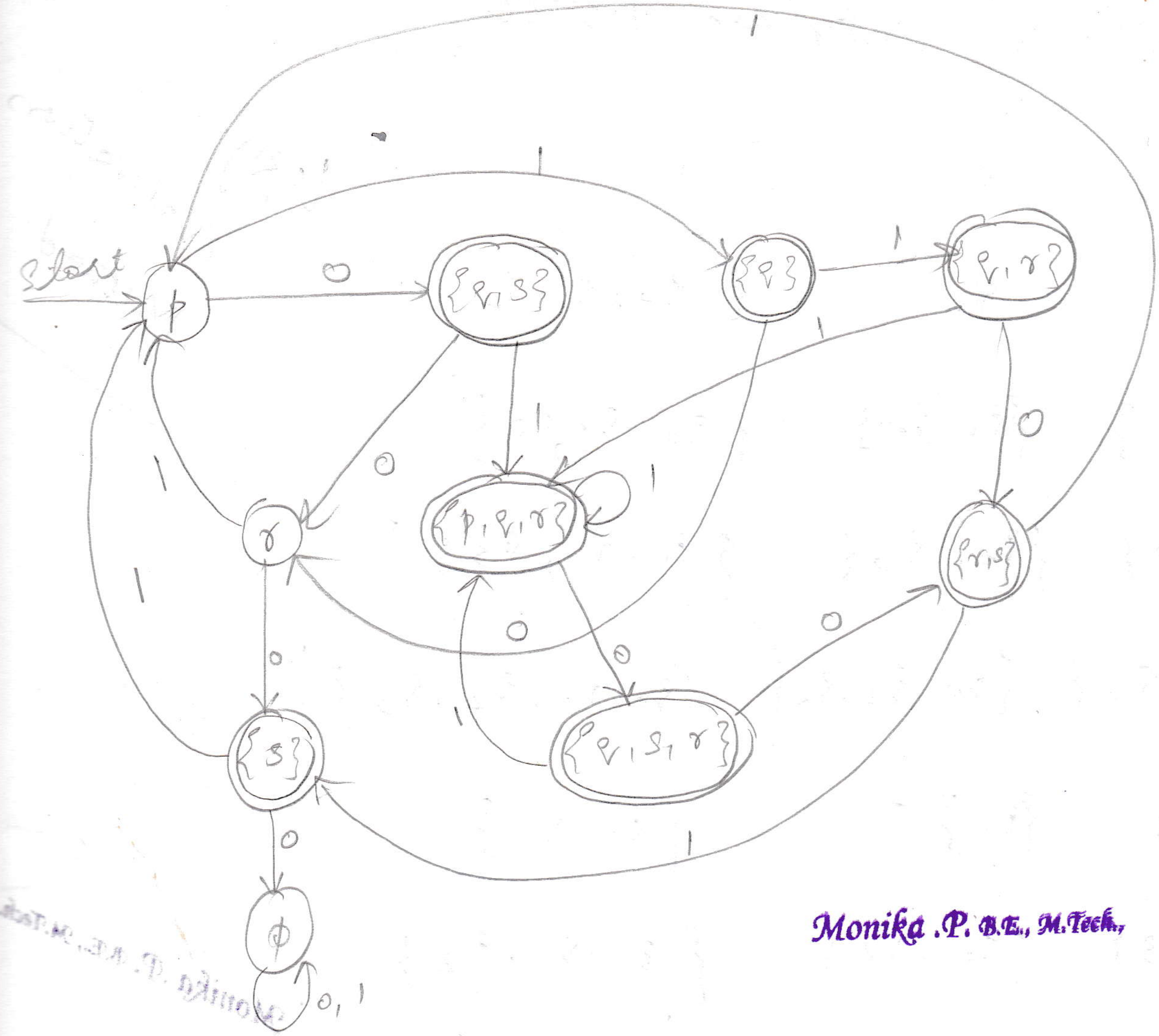
Monika .P. B.E., M.Tech.,

Induction:

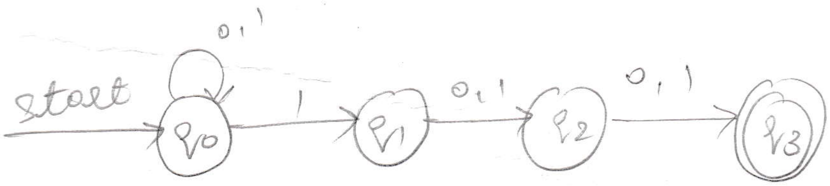
$$\delta_D(\{q,s\}, 0) = \delta_N(q,0) \cup \delta_N(s,0)$$
$$= r \cup \phi$$
$$= \{r\}$$

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

δ	0	1
start → p	$\{q,s\}$	$\{q\}$
* $\{q,s\}$	$\boxed{\{r \cup \phi\} = \{r\}}$	$\boxed{\{q,r\} \cup p = \{p,q,r\}}$
* $\{q\}$	$\{r\}$	$\{q,r\}$
$\{r\}$	$\{s\}$	$\{p\}$
* $\{p,q,r\}$	$\{q,s,r\}$	$\boxed{\{q \cup q,r \cup p\} = \{p,q,r\}}$
* $\{q,r\}$	$\{r,s\}$	$\boxed{\{q,r \cup p\} = \{p,q,r\}}$
* $\{s\}$	ϕ	$\{p\}$
* $\{q,r,s\}$	$\{r,s\}$	$\boxed{\{q,r \cup p \cup p\} = \{p,q,r\}}$
* $\{r,s\}$	$\{s\}$	$\boxed{\{p \cup p\} = \{p\}}$



(3) Convert the following NFA into a DFA



3rd symbol from the right end is 1

111
101
100

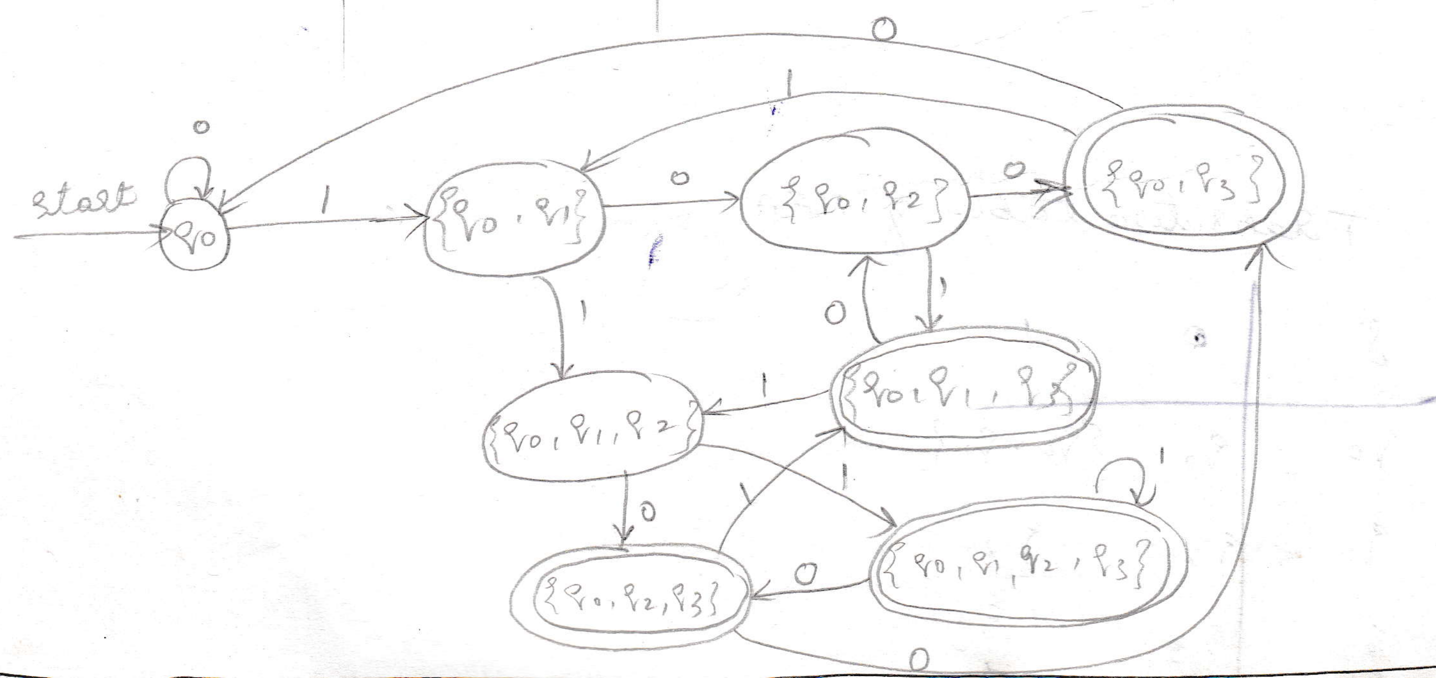
Transition table of NFA

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

$2^4 = 16$ states are possible

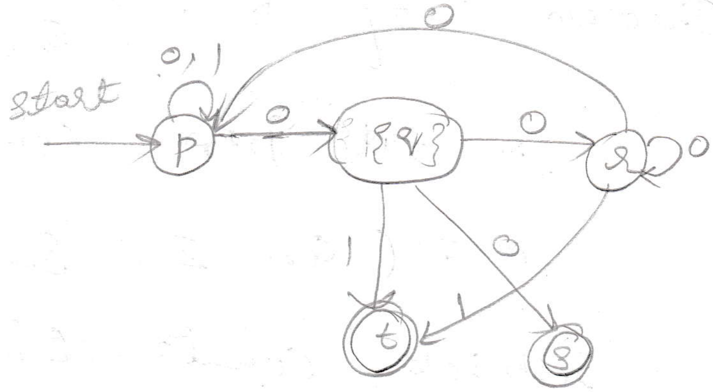
Lazy Evaluation Method

δ	0	1
start $\rightarrow$ q_0	q_0	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
* $\{q_0, q_3\}$	$\{q_0\}$	$\{q_0, q_1\}$
* $\{q_0, q_1, q_3\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
* $\{q_0, q_2, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_3\}$
* $\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$



NFA to DFA

δ_{NFA}	0	1
$\rightarrow p$	$\{p, q\}$	$\{p\}$
q	$\{r, s\}$	$\{t\}$
r	$\{p, r\}$	$\{t\}$
$* s$	ϕ	ϕ
$* t$	ϕ	ϕ

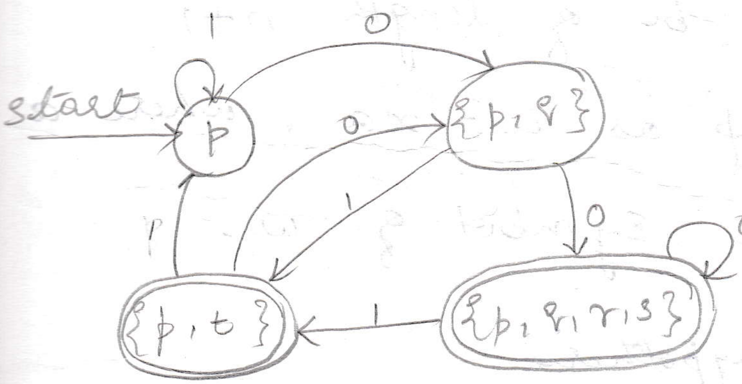


Inductive step

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

Lazy Evaluation Method

Monika .P. B.E., M.Tech.,



Theorem : If $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ is a DFA constructed from NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ by the subset construction, then $L(D) = L(N)$

proof :- we have to proove by induction on length of w $|w|$ is,

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$$

Monika .P. B.E., M.Tech

Basic :- let $|w| = 0$ i.e., $w = \epsilon$.

By the Basic defn of $\hat{\delta}$ for DFA & NFA both $\hat{\delta}_D(\{q_0\}, \epsilon)$ & $\hat{\delta}_N(q_0, \epsilon) = \{q_0\}$

Induction :- let w be of length $n+1$.

Break w up as $w = xa$, where a is a final symbol of w.

By inductive hypothesis

$$\hat{\delta}_D(\{q_0\}, x) = \hat{\delta}_N(q_0, x)$$

Let both these sets of N's states be $\{p_1, p_2, p_3, \ldots p_k\}$

- The inductive part of the defn of $\hat{\delta}$ for NFA's

is $\hat{\delta}_N(q_0, w) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$. — ①   a - final symbol.

The subset construction on the other hand tells us that $\delta_D(\{p_1, p_2, \ldots p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$ — ②

using eqn ② & the fact that

$\hat{\delta}_D(\{q_0\}, x) = \{p_1, p_2, \ldots, p_k\}$ in the inductive form of the defn of $\hat{\delta}$ for DFA is

$$\hat{\delta}_D(\{q_0\}, w) = \delta_D(\hat{\delta}_D(\{q_0\}, x), a)$$

$$= \delta_D(\{p_1, p_2, \ldots p_k\}, a)$$

$$= \bigcup_{i=1}^{k} \delta_N(p_i, a) \quad — ③$$

from the eqn ① & eqn ③ it show that $\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$ ∴

$\boxed{L(D) = L(N)}$

Unit - 2   Finite Automata & Regular Expressions

NFA with ϵ

An NFA-ϵ or ϵ-NFA is a 5 tuple system [or Quintuple system]

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

where Q is finite set of states

Σ is finite set of i/p Alphabet

δ is the transition func maped from

$\delta : Q \times (\Sigma \cup \epsilon) = 2^Q$.

$q_0 \in Q$ called start state

$F \subseteq Q \rightarrow$ subset of Q called final state / Set of final states

---

Epsilon - closure :

Monika .P. B.E., M.Tech.,

in formally : we write ϵ-close of a state q by foll all transitions out of q that are labelled ϵ.

defn : we define the ϵ-closure ϵ-close(q) recursively as follows

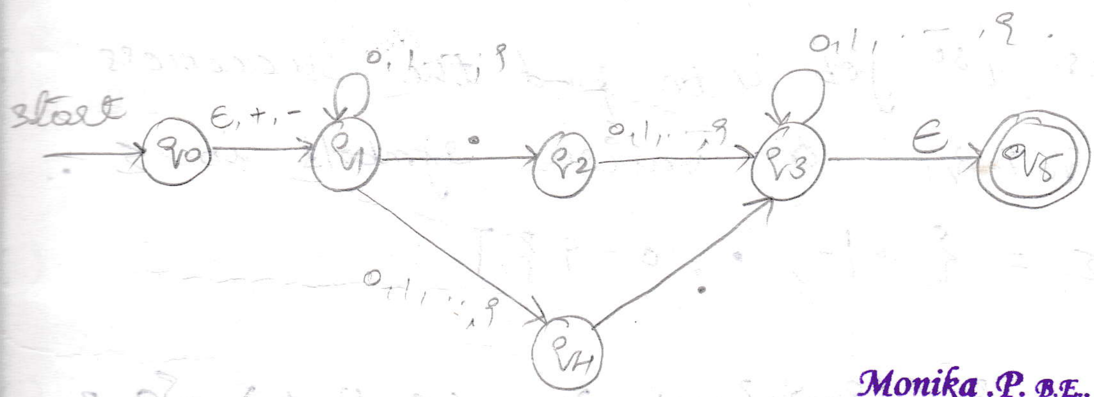
Basis : state q is in ECLOSE(q)

i.e., (Self state)

Induction :- If state p is in ECLOSE(q), & there is a transition from state p to state r labeled ϵ, then r is in ECLOSE(q).

---

eg: construct an ENFA that accepts decimal nos. Consisting of (i) an optional + or - sign.
(ii) A string of digits
(iii) A decimal point &
(iv) Another string of digits. either this string of digits or the string in (2) Question case empty, But atleast one of the 2 strings of digits must be non-empty.



Monika .P. B.E., M.Tech.,

PTO

δ	ϵ	$+\vert-$	$0,1,\cdots,9$	$\cdot$
$\xrightarrow{start} q_0$	$\{q_1\}$	$\{q_1\}$	ϕ	ϕ
q_1	ϕ	ϕ	$\{q_1, q_4\}$	$\{q_2\}$
q_2	ϕ	ϕ	$\{q_3\}$	ϕ
q_3	q_5	ϕ	$\{q_3\}$	ϕ
q_4	ϕ	ϕ	ϕ	$\{q_3\}$
$* q_5$	ϕ	ϕ	ϕ	ϕ.

Since the start state of ϵ-NFA is q_0.

$\therefore$ the start state of DFA is ECLOSE(q_0)

Monika .P. B.E., M.Tech.,

$ECLOSE(q_0) = \{q_0, q_1\}$.

step 1:- our 1st job is to find the successors of q_0 & q_1 on various symbols in Σ.

$[\Sigma = \{+|-, \cdot, 0-9\}]$

compute ① $\delta_D(\{q_0, q_1\}, +|-) = (q_1 \cup \phi) = \{q_1\}$.

$= ECLOSE(q_1) = \{q_1\}$.

② Compute $\delta_D(\{q_0, q_1\}, \cdot) = \{\phi \cup q_2\} = \{q_2\}$

$= ECLOSE(q_2) = \{q_2\}$

(3) compute $\delta_D(\{q_0, q_1\}, 0\text{-}9) = \{q_1, q_4\}$

$= ECLOSE(\{q_1, q_4\})$

$= ECLOSE(q_1) \cup ECLOSE(q_4)$

$= \{q_1 \cup q_4\}$.

step 2

① $\delta_D(\{q_1\}, +|-) = \phi$

$= ECLOSE(\phi) = \phi$

② $\delta_D(\{q_1\}, \cdot) = \{q_2\}$

$= ECLOSE(q_2) = q_2$

③ $\delta_D(\{q_1\}, 0\text{-}9) = \{q_1, q_4\}$

$= ECLOSE(q_1, q_4)$

$= (q_1, q_4)$

step 3

Monika .P. B.E., M.Tech,

① $\delta_D(\{q_2\}, +|-) = \phi$

$= ECLOSE(\phi) = \phi$.

② $\delta_D(\{q_2\}, \cdot) = \phi$

$= ECLOSE(\phi) = \phi$

③ $\delta_D(\{q_2\}, 0\text{-}9) = \{q_3\}$

$= ECLOSE(q_3) = \{q_3, q_5\}$

step 4

① $\delta_D(\{q_1 q_4\}, +|-) = \phi = ECLOSE(\phi) = \phi$

$\delta_D(\{q_1, q_4\}, \cdot) = \{q_2, q_3\} = ECLOSE(q_2, q_3) = \{q_2, q_3, q_5\}$

$\delta_D(\{q_1 q_4\}, 0\text{-}9) = \{q_1 q_4\} = ECLOSE(q_1, q_4) = \{q_1, q_4\}$

step 5

① $\delta(\{q_3, q_5\}, +|-) = \phi$
$= ECLOSE(\phi) = \phi$

② $\delta(\{q_3, q_5\}, \cdot) = \phi = ECLOSE(\phi) = \phi$

③ $\delta(\{q_3, q_5\}, 0-9) = \{q_3\} = ECLOSE(q_3) = \{q_3, q_5\}$

step 6

① $\delta(\{q_2, q_3, q_5\}, +|-) = \phi$
$= ECLOSE(\phi) = \phi$

② $\delta(\{q_2, q_3, q_5\}, \cdot) = \phi = ECLOSE(\phi) = \phi$

③ $\delta(\{q_2, q_3, q_5\}, 0-9) = \{q_3\} = ECLOSE(q_3) = \{q_3, q_5\}$

δ_{DFA}	+|-	$\cdot$	0-9
$\rightarrow \{q_0, q_1\}$	$\{q_1\}$	$\{q_2\}$	$\{q_1, q_4\}$
$\{q_1\}$	ϕ	$\{q_2\}$	$\{q_1, q_4\}$
$\{q_2\}$	ϕ	ϕ	$\{q_3, q_5\}$
$\{q_1, q_4\}$	ϕ	$\{q_2, q_3, q_5\}$	$\{q_1, q_4\}$
$* \{q_3, q_5\}$	ϕ	ϕ	$\{q_3, q_5\}$
$* \{q_2, q_3, q_5\}$	ϕ	ϕ	$\{q_3, q_5\}$

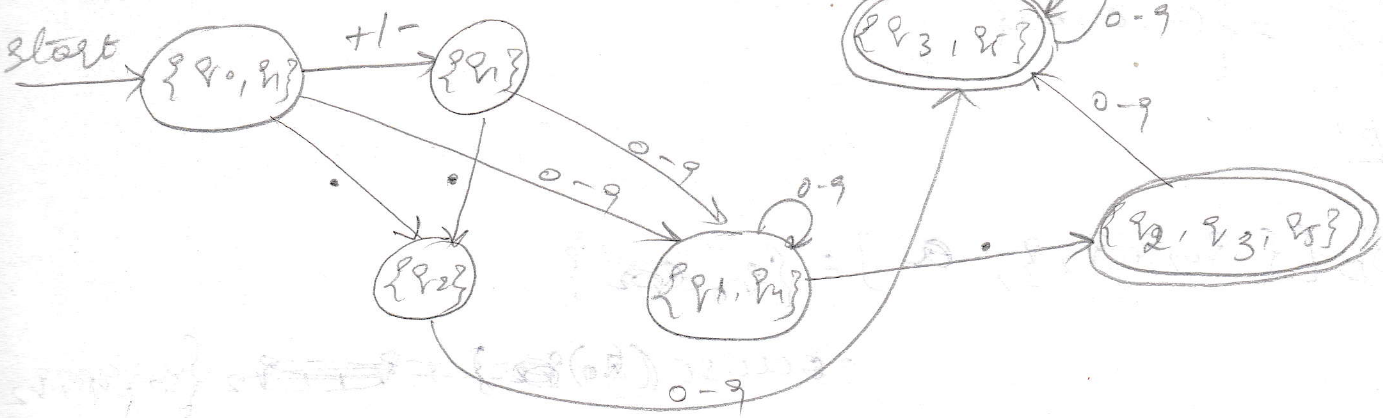
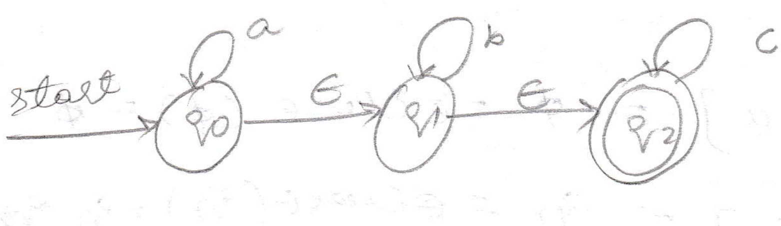


Fig: DFA

---

problem:

① convert the foll ε-NFA into a DFA.



set of strings consisting of zero or more a's followed by zero or more b's followed by zero or more c's

δ	ϵ	a	b	c
start → q_0	q_1	q_0	ϕ	ϕ
q_1	q_2	ϕ	q_1	ϕ
* q_2	ϕ	ϕ	ϕ	q_2

since start state of ε-NFA is q_0,

∴ the start state of DFA is Eclose(q_0).

$$\text{ECLOSE}(q_0) = \{q_0, q_1, q_2\}.$$

$[\Sigma = \{\epsilon, a, b, c\}]$

Step 1

① $\delta_D[\{q_0, q_1, q_2\}, a] = \{q_0\}$

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

② $\delta_D[\{q_0, q_1, q_2\}, b] = \{q_1\} = ECLOSE(q_1) = q_2, q_1$

$\delta_D[\{q_0, q_1, q_2\}, c] = \{q_2\} = ECLOSE(q_2) = q_2$

step2

① $\delta_D[\{q_1, q_2\}, a] = \phi = ECLOSE(\phi) = \phi$

② $\delta_D[\{q_1, q_2\}, b] = q_1 = ECLOSE(q_1) = q_1, q_2$

③ $\delta_D[\{q_1, q_2\}, c] = q_2 = ECLOSE(q_2) = q_2.$

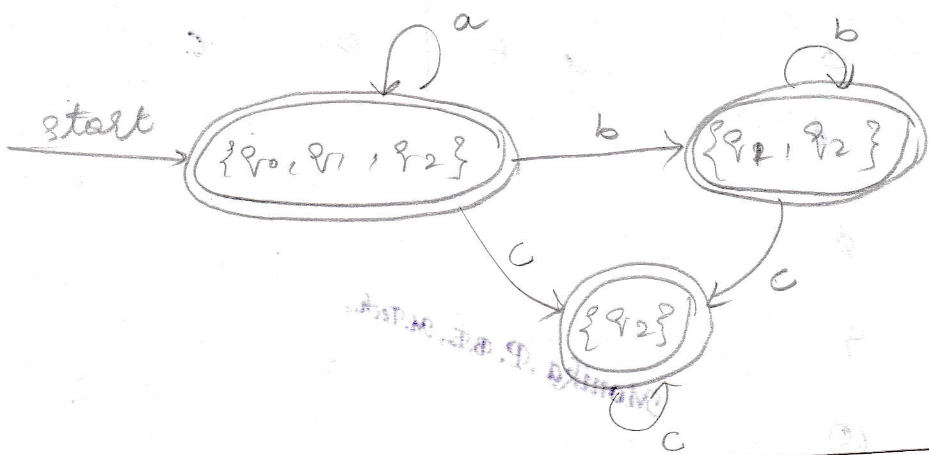
step 3

① $\delta_D[\{q_2\}, a] = \phi = ECLOSE(\phi) = \phi$

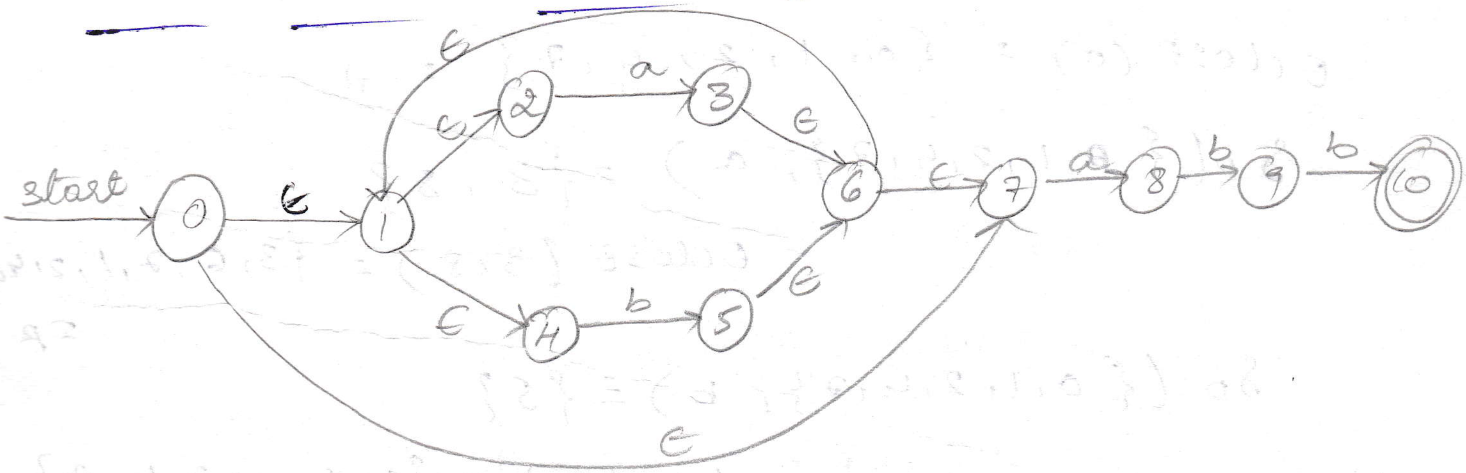
② $\delta_D[\{q_2\}, b] = \phi = Eclose(\phi) = \phi.$

③ $\delta_D[\{q_2, c] = q_2 = ECLOSE(q_2) = q_2$

δ_{DFA}	a	b	c
start → $\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_2, q_1\}$	$\{q_2\}$
* $\{q_1, q_2\}$	ϕ	$\{q_1, q_2\}$	q_2
* $\{q_2\}$	ϕ	ϕ	q_2



Convert ε-NFA to DFA for the following diagram



PTO

δ	ϵ	a	b
$\rightarrow$ 0	1, 7	ϕ	ϕ
1	2, 4	ϕ	ϕ
2	ϕ	3	ϕ
3	6	ϕ	ϕ
4	ϕ	ϕ	5
5	6	ϕ	ϕ
6	7, 1	ϕ	ϕ
7	ϕ	8	ϕ
8	ϕ	ϕ	9
9	ϕ	ϕ	10
* 10	ϕ	ϕ	ϕ

δ_{DFA}	a	b
A	B	C
B	B	D
C	B	C
D	B	E
* E	B	C

$$\text{Eclose}(0) = \{0, 1, 2, 4, 7\} = A$$

$$\delta_D(\{0, 1, 2, 4, 7\}, a) = \{3, 8\}$$

$$\text{Eclose}\{3, 8) = \{3, 6, 7, 1, 2, 4, 8\} = B$$

$$\delta_D(\{0, 1, 2, 4, 7\}, b) = \{5\}$$

$$\text{Eclose}(5) = \{5, 6, 1, 2, 4, 7\} = C$$

step 2

$$\delta_D(\{3, 6, 7, 1, 2, 4, 8\}, a) = \{8, 3\}$$

$$\text{Eclose}(8, 3) = \{8, 3, 6, 7, 1, 2, 4\} = B$$

$\delta_D(\{1,2,3,4,6,7,8\}, b) = \{5,9\}$

$\text{Eclose}(5,9) = \{5,6,1,2,4,7,9\} = D$

step ③

$\delta_D(\{1,2,4,5,6,7\}, a) = \{3,8\}$

$\text{Eclose}(3,8) = B$

$\delta_D(\{1,2,4,5,6,7\}, b) = \{5\}$

$\text{ECLOSE}(5) = C$

step ④

$\delta_D(\{1,2,4,5,6,7,9\}, a) = \{3,8\}$

$\text{Eclose}(3,8) = B$

$\delta_D(\{1,2,4,5,6,7,9\}, b) = \{5,10\}$

$\text{ECLOSE}(5,10) = \{1,2,4,5,6,7,10\}$
$= E$

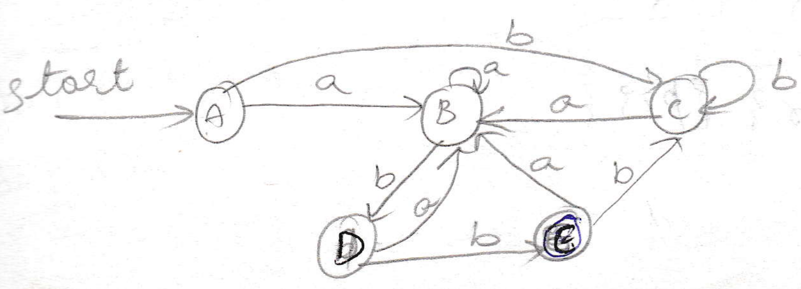
step ⑤

$\delta_D(\{1,2,4,5,6,7,10\}, a) = (3,8)$

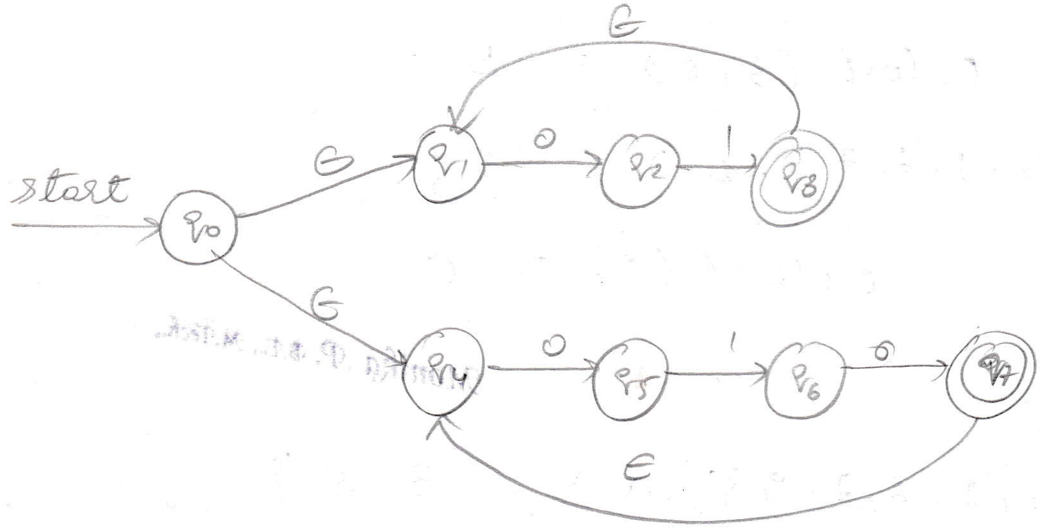
$\text{Eclose}(3,8) = B$

$\delta_D(\{1,2,4,5,6,7,10\}, b) = \{5\}$

$\text{Eclose}(5) = C$



- Design ε-NFA for the following language
- the set of strings that consists of either 01 repeated one or more times or 010 repeated one or more times.



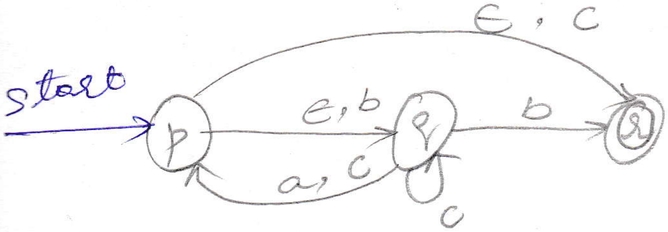
---

(1)

	δ	ε	a	b	c
→	p	φ	{p}	{q}	{r}
	q	{p}	{q}	{r}	φ
*	r	{q}	{r}	φ	{p}.

(2)

	δ	ε	a	b	c
→	p	{q,r}	φ	{q}	{r}
	q	φ	{p}	{r}	{p,q}
*	r	φ	φ	φ	φ



δ_{DFA}	a	b	c
$\rightarrow * \{p, q, r\}$	$\{p, q, r\}$	$\{q, r\}$	$\{p, q, r\}$
$* \{q, r\}$	$\{p, q, r\}$	$\{r\}$	$\{p, q, r\}$
$* \{r\}$	ϕ	ϕ	ϕ

$$E\text{CLOSE}(p) = (p, q, r)$$

step 1

$$\delta_D(\{p, q, r\}, a) = \{p\} = E\text{CLOSE}(p) = \{p, q, r\}$$

$$\delta_D(\{p, q, r\}, b) = \{q, r\} = E\text{CLOSE}(q, r) = (q, r)$$

$$\delta_D(\{p, q, r\}, c) = \{p, q, r\}$$
$$E\text{CLOSE}(p, q, r) = (p, q, r)$$

step 2

$$\delta_D(\{q, r\}, a) = \{p\}$$
$$E\text{CLOSE}(p) = \{p, q, r\}$$

$$\delta_D(\{q, r\}, b) = \{r\} = E\text{CLOSE}(r) = \{r\}$$

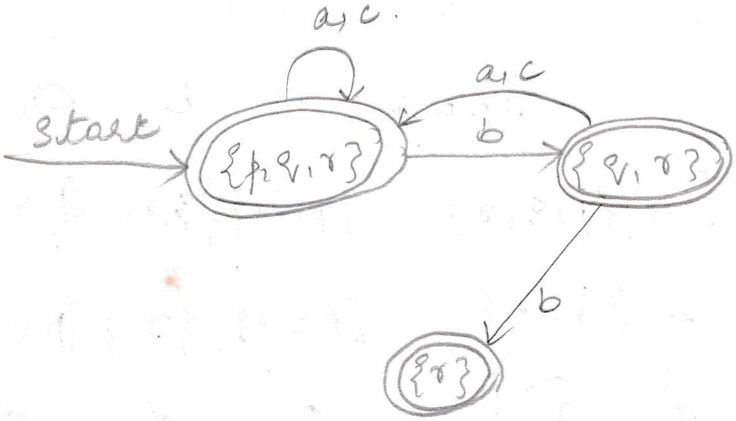
$$\delta_D(\{q, r\}, c) = \{p, q\} = E\text{CLOSE}(p, q) = (p, q, r)$$

step 3

$$\delta_D(\{r\}, a) = \phi = E\text{CLOSE}(\phi) = \phi.$$

$$\delta_D(\{r\}, b) = \phi = E\text{CLOSE}(\phi) = \phi.$$

$$\delta_D(\{r\}, c) = \phi = E\text{CLOSE}(\phi) = \phi.$$

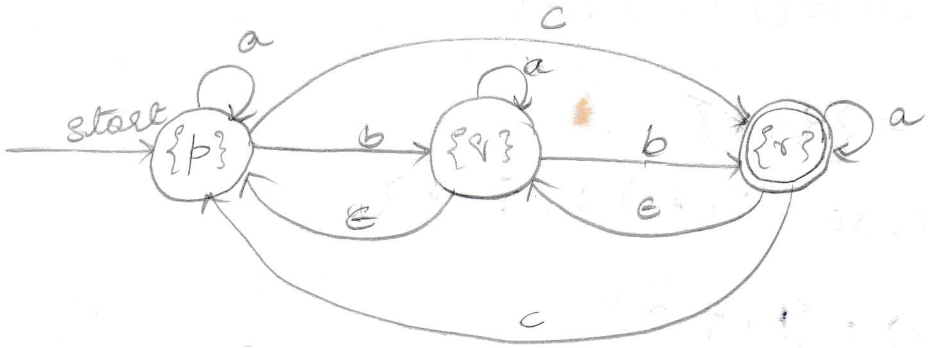


DFA

Monika .P. B.E., M.Tech.

---

(1)



ECLOSE (P) = p

δ_{DFA}	a	b	c
start → {p}	p	{p, q}	{p, q, r}
{p, q}	{p, q}	{p, q, r}	{p, q, r}
* {p, q, r}	{p, q, r}	{p, q, r}	{p, q, r}

<u>step 1</u>

$\delta_D(\{p\}, a) = \{p\} = ECLOSE(p) = p$

$\delta_D(\{p\}, b) = \{q\} = ECLOSE(q) = \{q, p\}$

$\delta_D(\{p\}, c) = \{r\} = ECLOSE(r) = \{r, q, p\}$

<u>step 2</u>

$\delta_D(\{p, q\}, a) = \{p, q\} = ECLOSE(p, q) = \{p, q\}$

$\delta_D(\{p, q\}, b) = \{q, r\} = ECLOSE(q, r) = \{p, q, r\}$

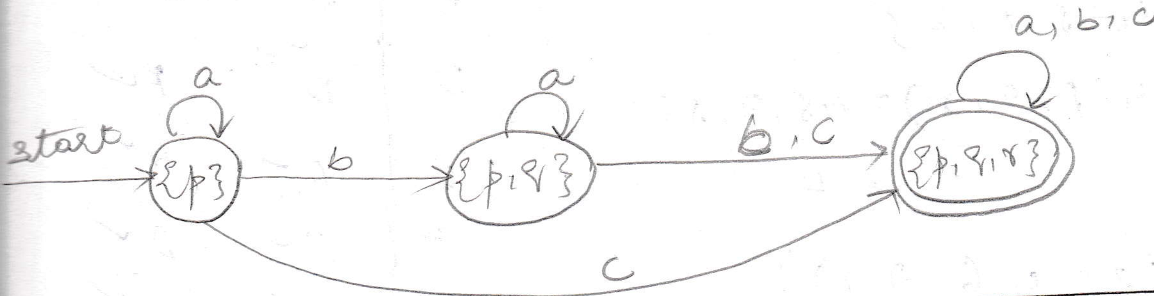
$\delta_D(\{p, q\}, c) = \{r\} = ECLOSE(r) = \{p, q, r\}$

step 3

$\delta_D(\{p,q,r\},a) = \{p,q,r\} = \text{ECLOSE}(p,q,r) = \{p,q,r\}$

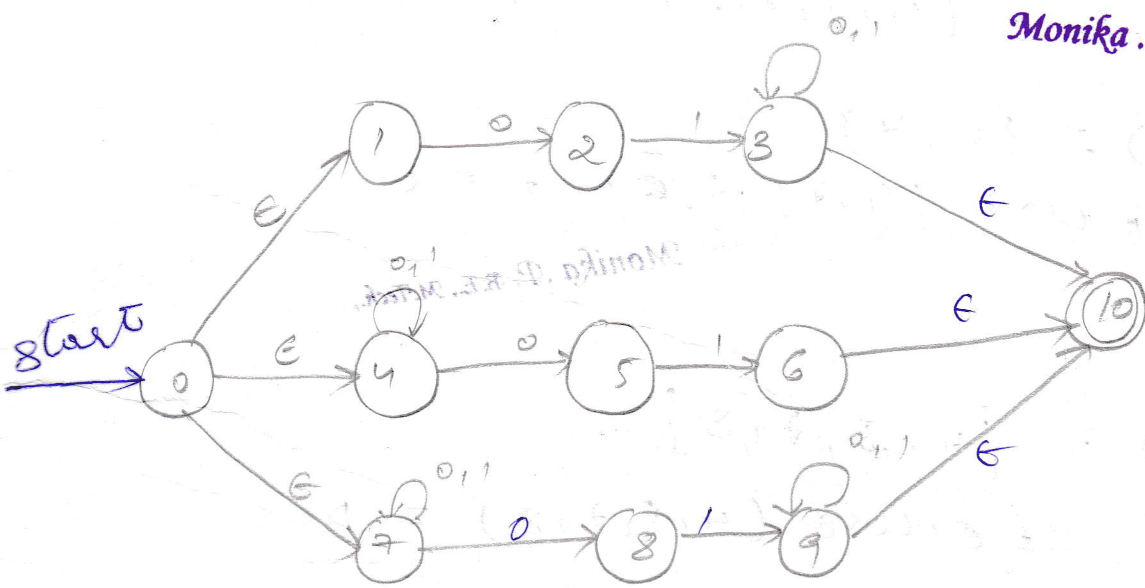
$\delta_D(\{p,q,r\},b) = \{q,r\} = \text{ECLOSE}(q,r) = \{p,q,r\}$

$\delta_D(\{p,q,r\},c) = \{p,r\} = \text{ECLOSE}(p,r) = \{p,q,r\}$



write an ε-NFA to recognize all the strings. that begin or end or both with 01.

Monika .P. B.E., M.Tech.



$\text{ECLOSE}(0) = 0,1,4,7 = A$

$\delta_D(A,0) = (2,4,5,7,8)$

$\text{ECLOSE}(2,4,5,7,8)$

$= 2,4,5,7,8 = B$

$\delta_D(A,1) = \{4,7\}$

$\epsilon close(4,7) = \{4,7\} = C$

step 2

$\delta_D(B,0) = \{4,5,7,8\}$

$\epsilon close(4,5,7,8) = 4,5,7,8$

$= D$

$\delta_D(B,1) = \{3,4,6,7,9\}$

$\epsilon close(B,1) = \{3,10,4,6,7,9\}$

$= E$

step 3

$\delta_D(C,0) = \{4,5,7,8\}$

$\epsilon close(4,5,7,8) = 4,5,7,8 = D$

$\delta_D(C,1) = \{4,7\}$

$\epsilon close\{4,7\} = 4,7 = C$

Monika .P. B.E., M.Tech.,

step 4

$\delta_D(D,0) = \{4,5,7,8\}$

$= \epsilon close(4,5,7,8) = D.$

$\delta_D(D,1) = \{4,6,7,9\}$

$\epsilon close(4,6,7,9) = \{4,6,10,7,9\}$

$= F$

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

step 5

$\delta_D(E,0) = \{3,4,5,7,8,9\}$

ECLOSE $(3,4,5,7,8,9) = \{3,10,4,5,7,8,9\}$

$= G.$

$\delta_D(E,1) = \{3,4,7,9\}$

$=$ ECLOSE $(3,4,7,9) = \{3,10,4,7,9\}$

$= H$

step 6

$\delta_D(F,0) = \{4,5,7,8,9\}$

$=$ ECLOSE $(4,5,7,8,9)$

$= \{4,5,7,8,9,10\} = I.$

$\delta_D(F,1) = \{4,7,9\}$

$=$ ECLOSE $(4,7,9) = (4,7,9,10)$

$= J$

Monika .P. B.E., M.Tech.,

step 7

$\delta_D(G,0) = \{3,4,5,7,8,9\}$

ECLOSE $(3,4,5,7,8,9) = \{3,10,4,5,7,8,9\}$

$= G.$

$\delta_D(G,1) = \{3,4,6,7,9\}$

ECLOSE $(G,1) = E.$

$\delta_D(H,0) = (3,4,5,7,8,9)$

$= ECLOSE(H,0) = G$

$\delta_D(H,1) = \{3, 4,7,9\}$

$ECLOSE(3,4,7,9) = \{3,10,4,7,9\}$

$= H$

Step 8

Monika .P. B.E., M.Tech.

$\delta_D(I,0) = \{4,5,7,8,9\}$

$ECLOSE(I,0) = I$

$\delta_D(I,1) = (4,6,7,9)$

$= ECLOSE(I,1) = F$

Step 9

$\delta_D(J,0) = (4,5,7,8,9)$

$= ECLOSE(J,0) = I$

$\delta_D(J,1) = (4,7,9)$

$ECLOSE(4,7,9)$

$= (4,7,9,10) = J$

δ_{DFA}	0	1
A	B	C
B	D	E
C	D	C
D	D	F
* E	G	H
* F	I	J
* G	G	E
* H	G	H
* I	I	F
* J	I	J

Monika .P. B.E., M.Tech.,