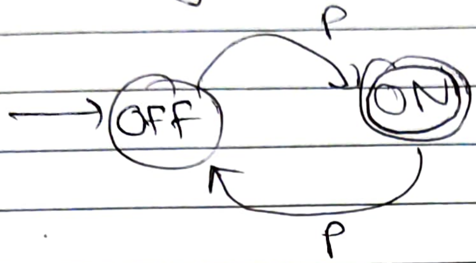
8th March 2023
Wednesday

## Class

Representing switch as an FA:




$L = \{ p^n \mid n \text{ is odd} \}$
$= \{ p, ppp, ppppp, \ldots \ldots \}$

## DFA

A DFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$,
where
$Q$ = finite set of states
$\Sigma \rightarrow$ alphabet
$\delta$ = Fn (Transition fn) that maps from
$Q \times \Sigma \rightarrow Q$
$q_0 \in Q$ initial (start state)
$F \in Q$ set of accepting (final) states

$\delta$ can be represented as Transition diagram or Transition table.

| | ~~OFF ON~~ p |
|---|---|
| → OFF | ON |
| * ON | OFF |

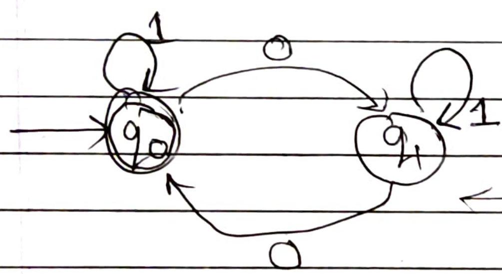
eg. Design a DFA which accepts all strings over $\{0,1\}$ with even no of 0's.

→

L = { all strings over $\{0,1\}$ with even no. of 0's }

$L = \{ \epsilon, 00, 010, 1, \ldots \}$

$L \neq \{ 0, \cancel{\epsilon}, 01, \ldots \}$

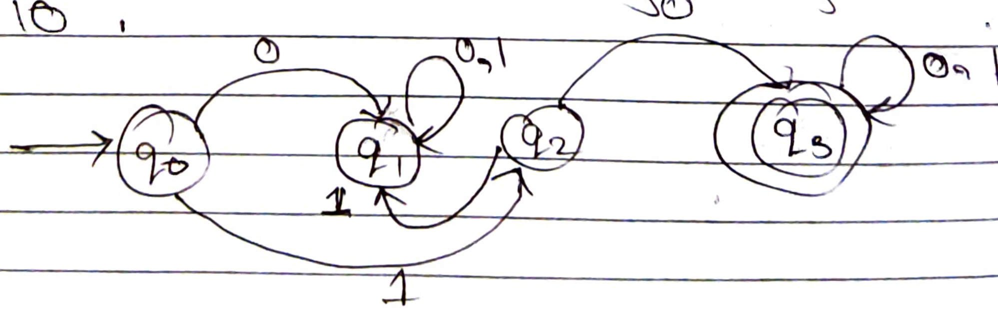


$M = (\{q_0, q_1\}, \{0,1\}, \delta, q_0, \{q_0\})$

$Q \quad \Sigma \quad \delta \quad q_0 \quad F$

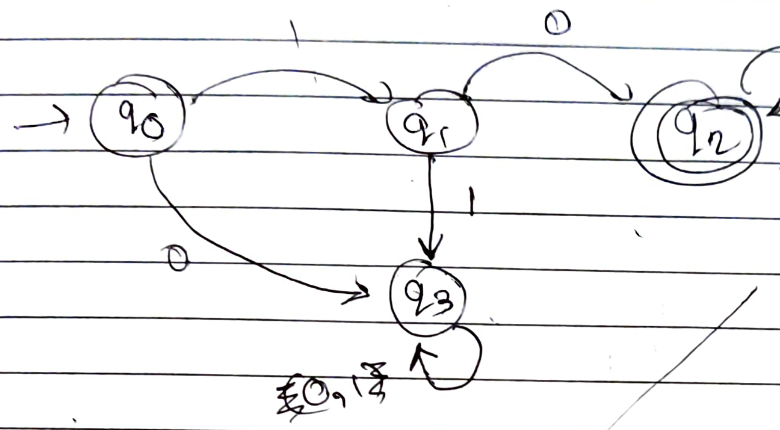
Q. Construct a DFA that accepts strings over $\{0,1\}$, where every string starts with 10.

→



$L = \{10, 101, 100, \ldots\ldots\}$

$L \neq \{0, 1, 00, 11, \ldots\}$



~~Note~~

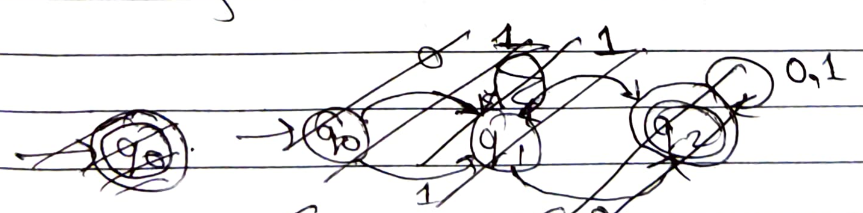
**Language.**

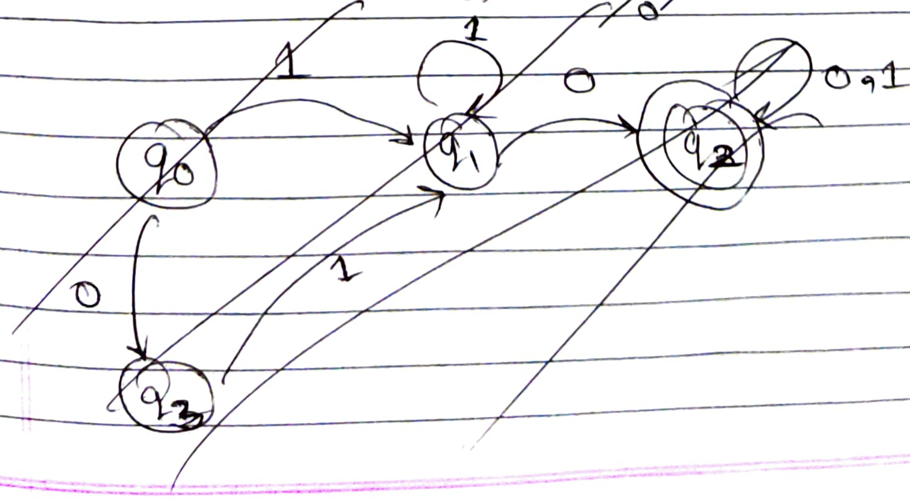
The language of DFA $M = \{Q, \Sigma, \delta, q_0, F\}$ is the set of all strings accepted by the DFA.

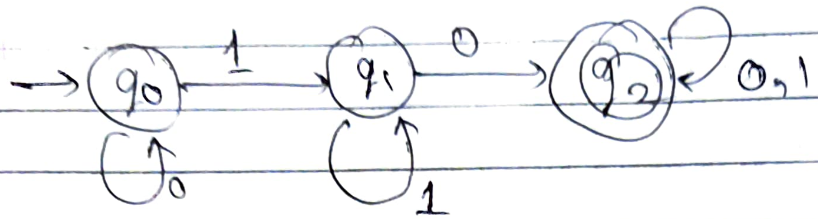
eg.1. Accept all the strings which contain substring 10.

eg.2. All strings which end with 10.

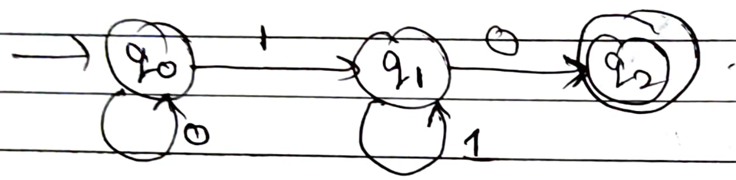
eg 1.







eg. All strings which end with 10.



9th March, 2023

Extended Transition function

$\hat{\delta} : Q \times \Sigma^* \to Q$

- Basis

  $\hat{\delta}(q, \varepsilon) = q$

- Induction

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

  w → string, a → symbol

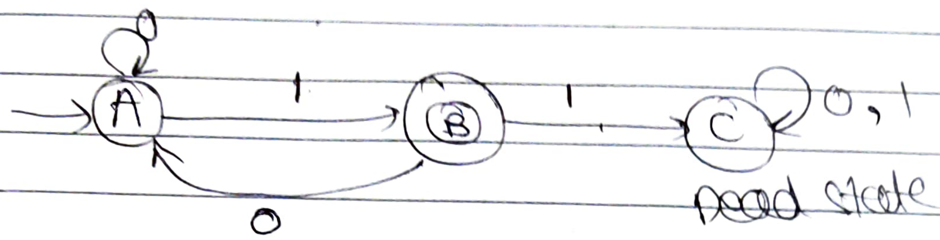
$w \in \Sigma^*$
$a \in \Sigma$

* A string $w$ is accepted by a DFA $M = (Q, \Sigma, \delta, q_0, F)$ if

~~$\hat{\delta}((q_0, w) \in F)$~~

$\hat{\delta}(q_0, w) \in F$

Q. Design a FA which ~~accepts~~ is ending with ~~a single~~ 1.




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

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

$$= \delta(\delta(\delta(\hat{\delta}(A,\epsilon),1),0),1)$$

$$= \delta(\delta(\delta(A,1),0),1)$$

$$= \delta(\delta(B,0),1)$$

$$= \delta(A,1)$$

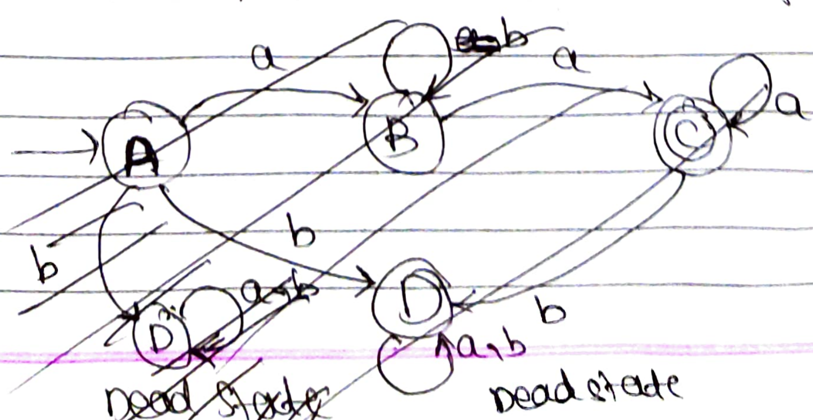
$$= B \in F$$

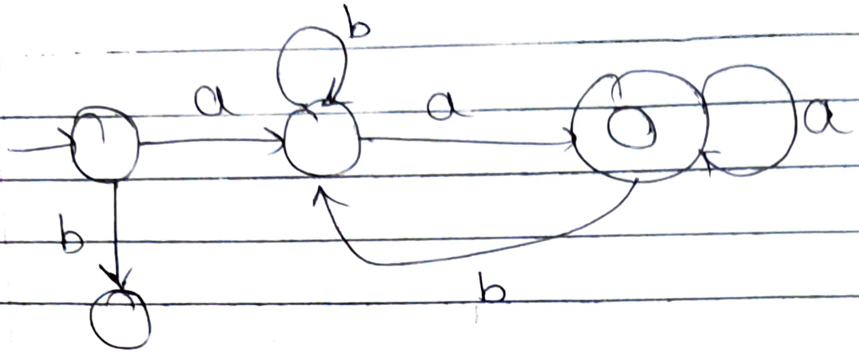
## Regular Languages.

A Language L is regular if it is accepted by some DFA.

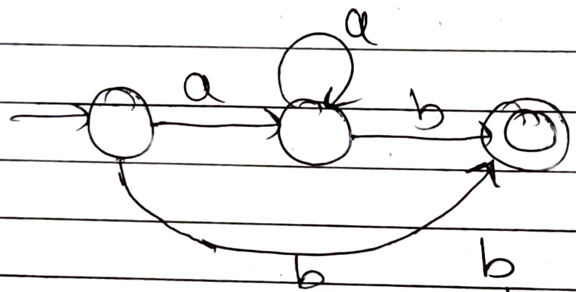
any no. of, any combination of

$L = \{awa \mid w \in \{a,b\}^*\}$ is regular?



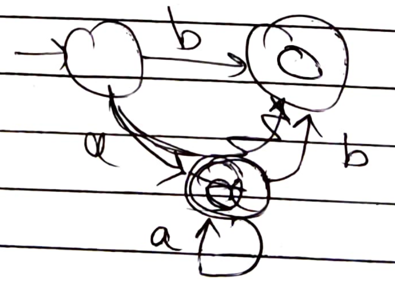


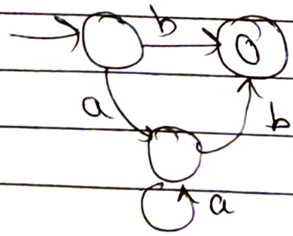
eg. $L = \{a^n b \mid n \geq 0\}$



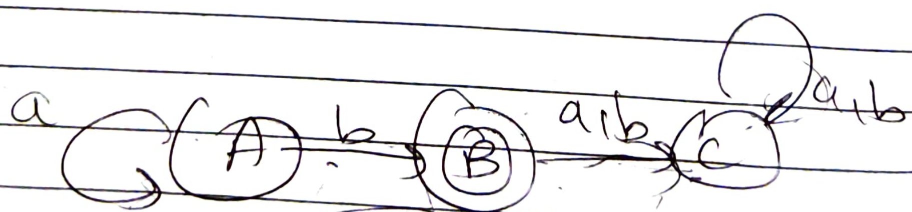
b
ab
aab
aaaaab

eg. ~~is $L^2$ regular?~~





eg.




$b^3$

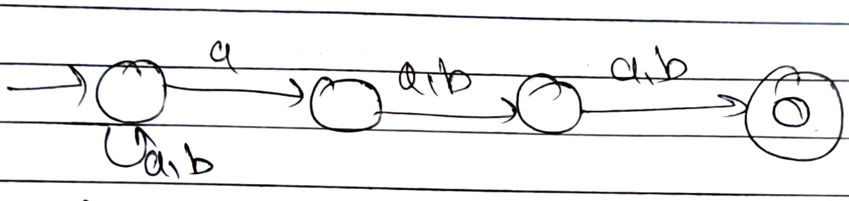
Finite control in DFA

# NFA

Non-Deterministic Finite Automata

$w \in \{a,b\}$ the third symbol from the end is a

$w \in \{a,b\}$




NFA

$\delta : Q \times \Sigma \to 2^Q$

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

NFA is a 5 tuple $M = (Q, \Sigma, \delta, q_0, F)$, where

$Q, \Sigma, \delta, q_0, F$ are.....

and $\delta$ is the transitn fn given by $f : Q \times \Sigma \to 2^Q$

$Q \to$ the finite set of states

$\Sigma \to$ finite input alphabet

$\delta \to$ transition function

$q_0 \to$ initial state

$F \to$ set of final states

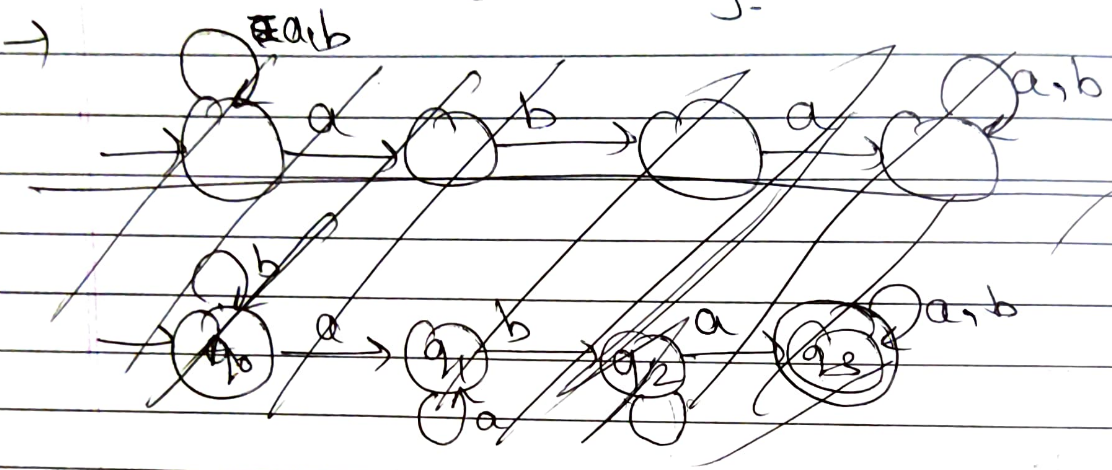
| | DFA | NFA |
|---|---|---|
| 1) $\delta$ | $Q \times \Sigma \to Q$ | $Q \times \Sigma \to 2^Q$ |
| 2) No. of transition on a symbol | one | zero or more |
| 3) Easier to Design | Difficult | Easy |
| 4) Finite control space requirement | More | Less |
| 5) No. of threads to recognize string of a Lang | One | Many |

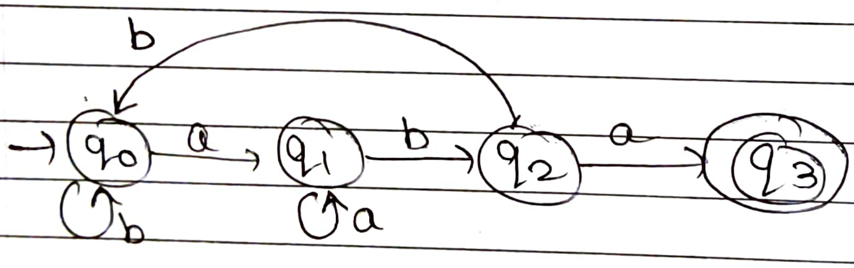
# TOC TUT



Q.1. Accept strings over {a,b} containing "aba" as a substring.

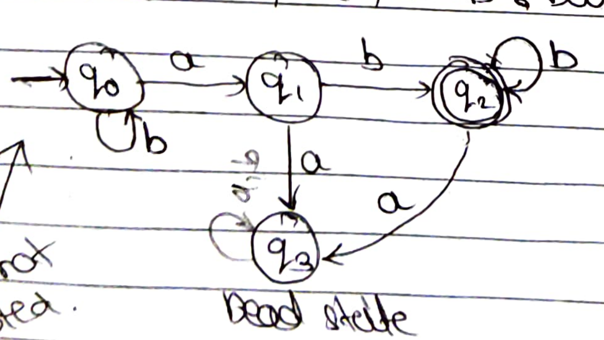
→





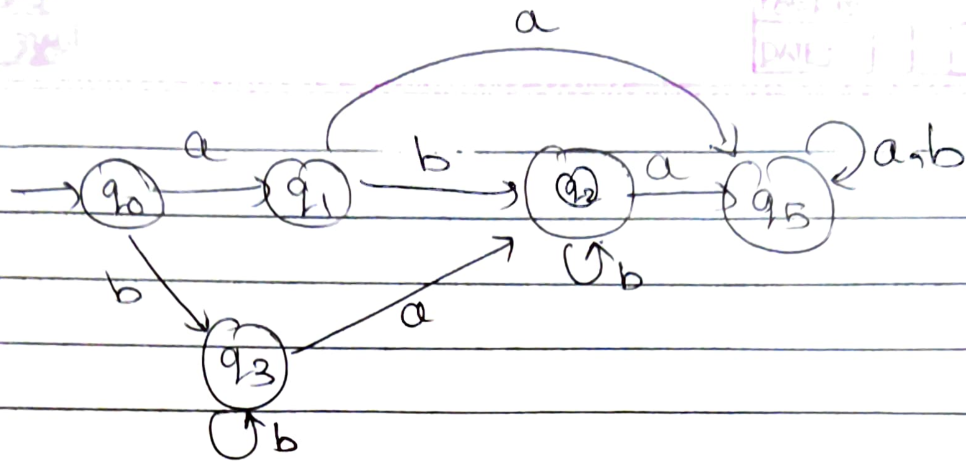

Q.2. Accept strings over {a,b} with exactly 1 a and atleast 1 b.

→ L = {ab, ba, abb, bab}




ba is not accepted.

So this is wrong.

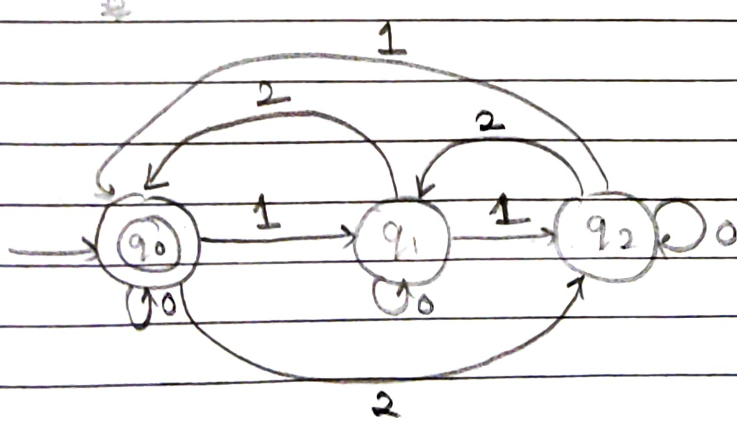




Q.3 $\Sigma = \{0,1,2\}$. DFA to find sum of input symbols and accept if the sum is divisible by 3.

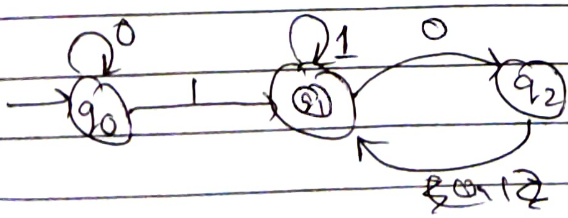
→

0 + 0 + 0 → / 3
0 + 1 + 2 → / 3
1 + 2 + 0 → / 3
1 + 1 + 1 → / 3
2 + 2 + 2 → / 3





Q4. Describe the language of DFA.



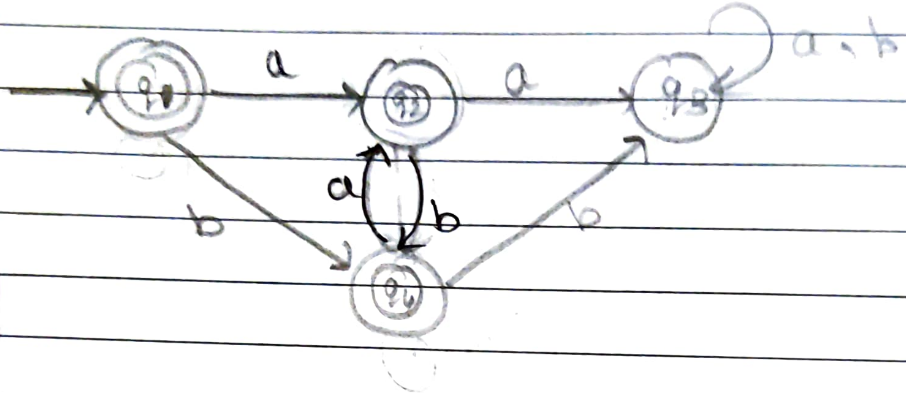


L = {01, 0001, 011111, 0100, 0101, ...

PAGE NO.:
DATE:

Q.5 ~~W∈A,B~~

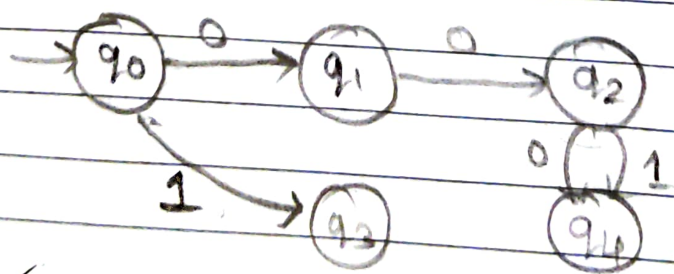
→ ~~w~~ ∈ w = {a,b} does not contain aa / bb.




Q.6 Zeroes & $1^s$ are equal such that ~~such that~~ each prefix have atmost 1 more zero than 1's ~~and~~ or atmost 1 more 1 ~~than~~ than 0's

L = { ∈ , 01 , 10,

~~Q.7 String of length 3 over {0,1} with atleast~~
~~→ 2 zeroes each diff. is atmost 1.~~




~~L = { 000 , 001 , 100 , 010 }~~

10th March

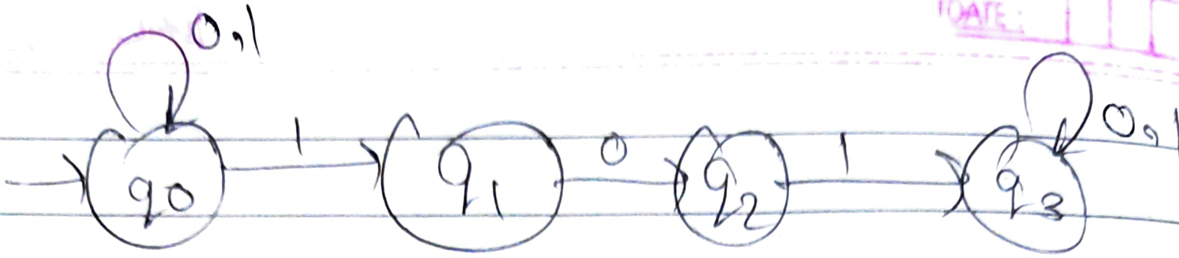
## $\hat{\delta}$ of NFA :-

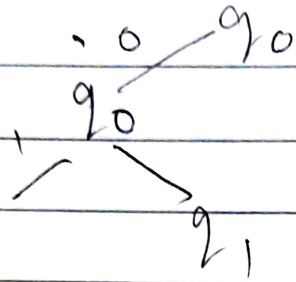
basic : $\hat{\delta}(q, \epsilon) = \{q\}$

induction : $\hat{\delta}(q, w) = \{r_1, r_2, \dots, r_m\}$
where $w = xa$ $a \in \Sigma$.
and $\hat{\delta}(q, x) = \{p_1, p_2, \dots p_k\}$
then $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \dots r_m\}$

The language accepted by NFA
$M = (Q, \Sigma, \delta, q_0, F)$ is

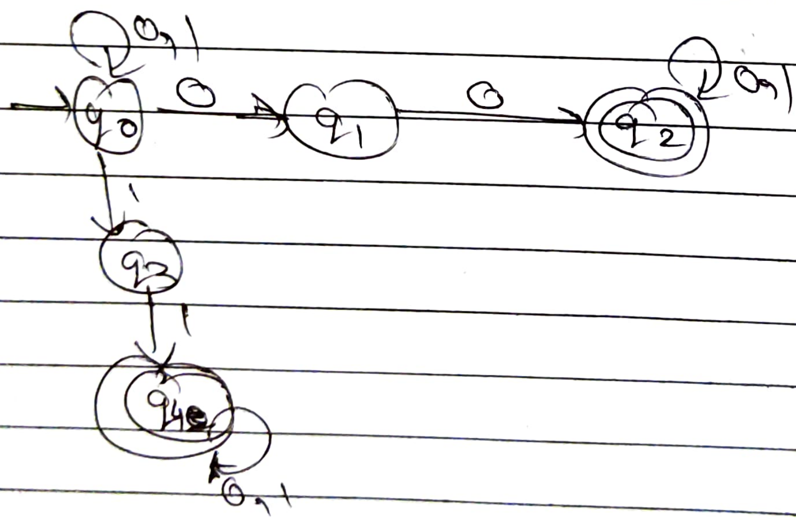
$L(M) = \{w \mid \hat{\delta}(q_0, w)$ contains a state in $F\}$




Design an NFA to accept ~~with~~ either two consecutive zeroes or two consecutive 1s. Check whether 01001 is accepted by your NFA.

→

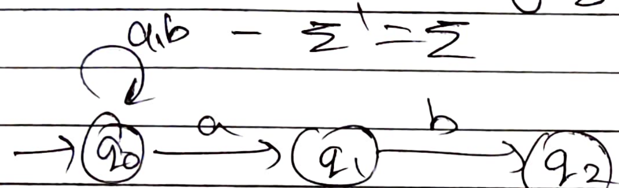



$\hat{\delta}(q_0,$

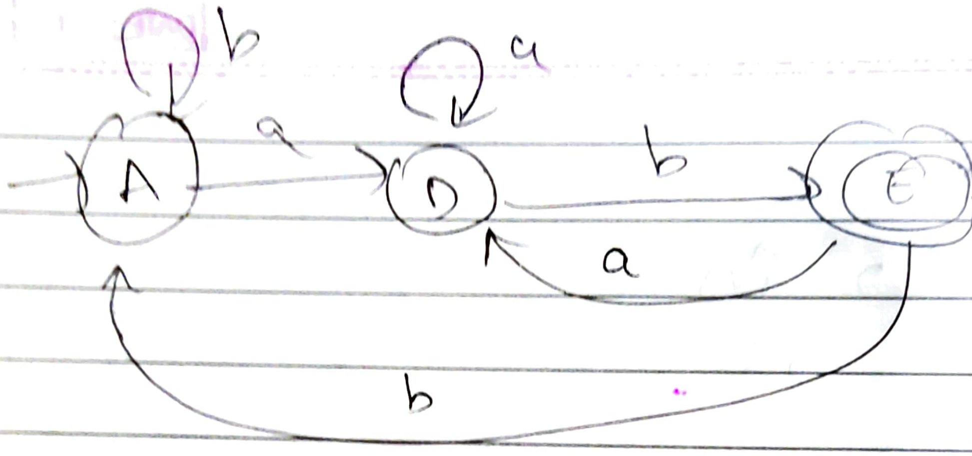
# Subset construction

Given NFA, $M_{NFA} = (Q, \Sigma, \delta, q_0, F)$
equivalent FA $M_{DF} = (Q', \Sigma', \delta', q_0', F')$
where $- Q' = 2^Q$
$- q_0' = [q_0]$
$-$ $F_1'$ is the set of states of $Q'$ containing a member of F.
$- \delta'([q_1, q_2 \ldots q_i], a)$
$= \delta(q_1, a) \cup \delta(q_2, a) \cup \ldots$
$\cup \delta(q_i, a)$
$- \Sigma' = \Sigma$

Eg.

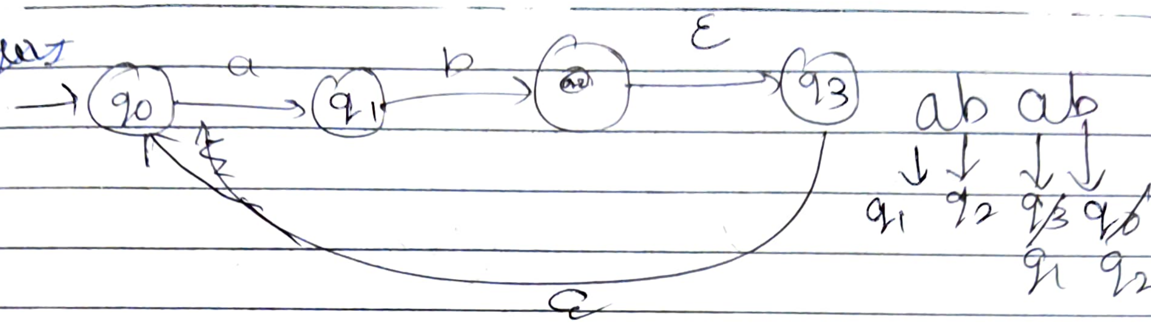



| | | a | b |
|---|---|---|---|
| $[q_0]$ | A | $[q_0\ q_1]$ | $[q_0]$ |
| $[q_1]$ | B | $\phi$ | $[q_2]$ |
| $[q_2]$ | C | $\phi$ | $\phi$ |
| $[q_0\ q_1]$ | D | $[q_0\ q_1]$ | $[q_0\ q_2]$ |
| $[q_0\ q_2]$ | E | $[q_0 q_1]$ | $[q_0]$ |
| $[q_1\ q_2]$ | F | $[\ \phi$ | $[q_2]$ |
| $[q_0\ q_1\ q_2]$ | G | $[q_0\ q_1]$ | $[q_0\ q_2]$ |
| $\phi$ | H | $\phi$ | $\phi$ |

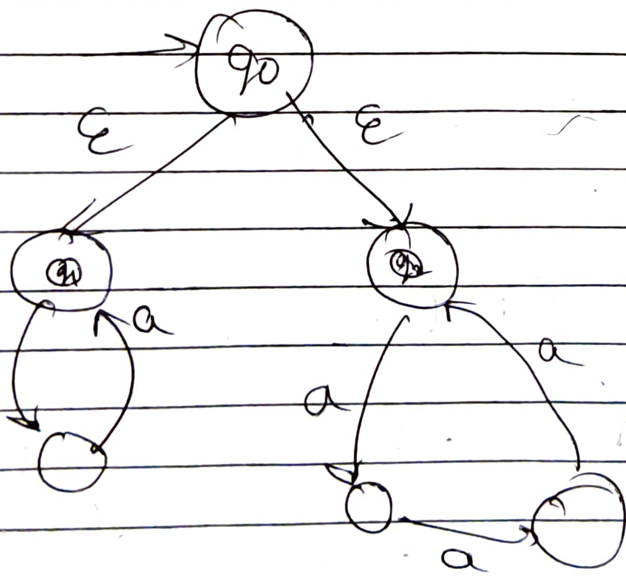



Rest are Dead

15th March, 2023




If a transition on ε is encountered, ~~withou~~ without consuming the next input symbol, the finite control creates multiple threads one following each of the ε transitions and one staying at the current state.

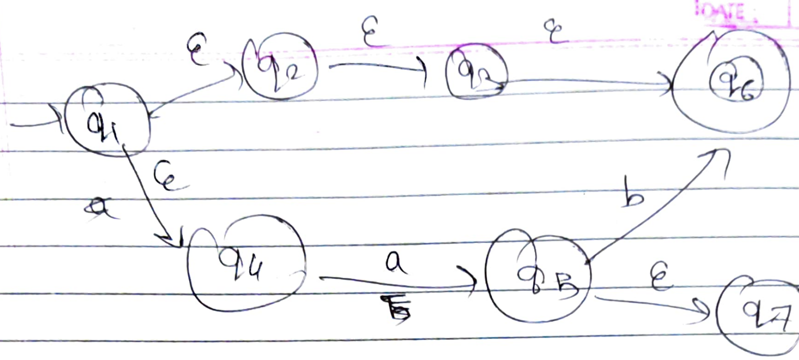



Let q be a state, the ε-closure of q is the set of all states that are reachable by following the ~~ep~~ ε transitions

ε-closure (q0) = {q0, q1, q2}

//

PAGE NO.
DATE:




ε-closure $(q_1) = \{q_1, q_2, q_4, q_3, q_6\}$
$(q_2) = \{q_2, q_3, q_6\}$
$(q_3) = \{q_3, q_6\}$
$(q_4) = \{q_4\}$
$(q_5) = \{q_5, q_7\}$
$(q_6) = \{q_6\}$
$(q_7) = \{q_7\}$

* Extended Transition Function

$\hat{\delta}(q, \epsilon) = \epsilon\text{-closure}(q)$

let $w = xa$

$\hat{\delta}(q, x) = \{p_1, p_2, \ldots p_k\}$

let $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \ldots r_m\}$

$\hat{\delta}(q, w) = \bigcup_{j=1}^{m} \epsilon\text{-closure}(r_j)$

- $w \in L$ accepted by ε-NFA

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

$\hat{\delta}(q_0, w)$ contains one of the final states.


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

$\hat{\delta}(q_0, a) = \epsilon\text{-closure}(\delta(\hat{\delta}(q_0, \epsilon), a))$.

$= \cancel{\delta(q_0}$

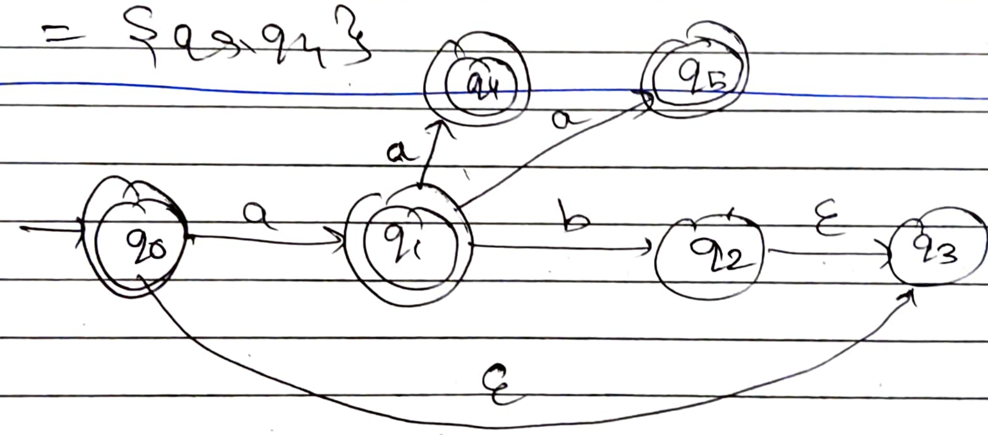
$= \epsilon\text{-closure}(\underset{\phi}{\delta(q_0, a)} \cup \underset{\phi}{\delta(q_1, a)} \cup \underset{q_3}{\delta(q_2, a)})$

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

$= \{q_3, q_4\}$

---

Q.




$\epsilon\text{-closure}(q_0) = \{q_0\}$

$\cancel{\delta(q_0, aba)}$

$\hat{\delta}(q_0, a) = \epsilon\text{-clos}(\delta(\hat{\delta}(q_0, \epsilon), a))$

$= \epsilon\text{-clos}(\delta(q_0, a)) = \epsilon\text{-clo}(q_1)$

$= \{q_1\}$

$\hat{\delta}(q_0, ab)$

$= \epsilon\text{-closer}(\delta(\hat{\delta}(q_0, a), b))$

$= \epsilon\text{-clo}(\delta(q_1, b))$

$= \epsilon\text{-clo}(q_2)$

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

PAGE NO.
DATE:

$\hat{\delta}(q_0, aba)$

$= \epsilon\text{-clo}(\delta(\hat{\delta}(q_0, ab), a))$

$= \epsilon\text{-clo}(\delta(q_2, q_3, q_0), a)$
$\phi \quad \phi$

$= \epsilon\,\delta((q_2, a) \cup (q_3, a) \cup (q_0, a))$

$= \epsilon\text{ clos}(a_1)$

$= \{a_1\}$ which was acc.
$\therefore$ aba is acc.

## * ε-NFA to DFA

Let the ε-NFA be $M_{NFA} = (Q, \Sigma, \delta, q_0, F)$
$M_{DFA} = \{Q', \Sigma', \delta', q_0', F'\}$

where,
$Q' = 2^Q$
$q_0' = \epsilon\text{-closure}(q_0)$
$F'$ = Set of states of $q'$ containing a member of F.

$\delta'$

$s \in Q'$ be $\{p_1, p_2, \ldots p_k\}$

(i) Compute $\bigcup_{i=1}^{k} \hat{\delta}(p_i, a) \rightarrow \{r_1, r_2, \ldots r_m$

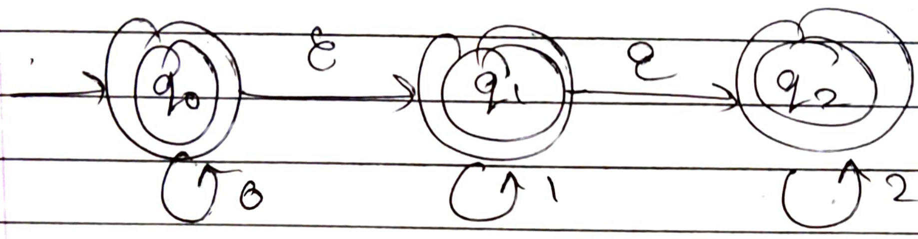
(ii) $\delta'(s, a) \rightarrow \bigcup_{j=1}^{m} \epsilon\text{closure}(r_j)$

16th March, 2023

DFA : $Q \times \Sigma \rightarrow Q$

NFA : $Q \times \Sigma \rightarrow 2^Q$

$\varepsilon$-NFA : $Q \times (\Sigma \cup \varepsilon) \rightarrow 2^Q$




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

$\varepsilon\text{-clos}(q_1) = \{q_1, q_2\}$

$\varepsilon\text{-clos}(q_2) = \{q_2\}$

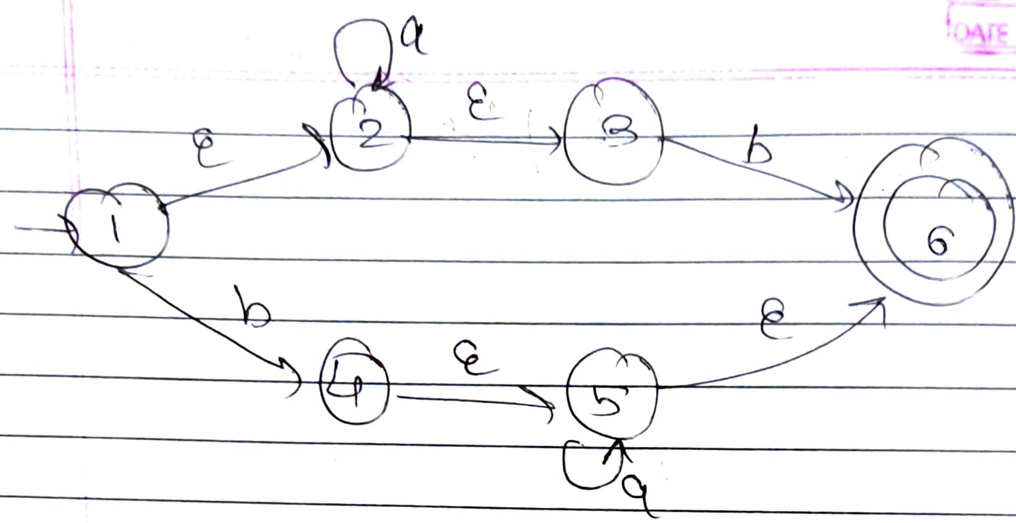
- $\delta(\{q_0, q_1, q_2\}, 0)$

$= \varepsilon\text{-clos}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0))$

$= \varepsilon\text{-clos}(\{q\})$

- $\delta(\{q_0, q_1, q_2\})$

$= \varepsilon\text{-clos}(\delta(q_0, 1) \cup \delta($




ε-closure(1) = {1, 2, 3}
ε-clo (2) = {2, 3}
ε-clo (3) = {3}
ε-clo (4) = {4, 5, 6}
ε-clo (5) = {5, 6}
ε-clo (6) = {6}

| | a | b |
|---|---|---|
| ~~→ [1 2 3]~~ | ~~[2 3]~~ | ~~[4 5 6]~~ |
| ~~→ [2 3]~~ | ~~[2 3]~~ | ~~[6]~~ |
| ~~[3]~~ | | |
| ~~[4, 5, 6]~~ | | |
| ~~[5, 6]~~ | | |
| ~~[6]~~ | | |
| ~~φ~~ | | |

| | a | b |
|---|---|---|
| [1 2 3] | [2 3] | [4 5 6] |
| [2 3] | [2 3] | [6] |
| [4 5 6] | [5 6] | φ |
| [6] | φ | φ |
| [5 6] | [5 6] | φ |
| φ | φ | φ |

# DFA minimization

We define 2 states p and q as equivalent if and only if:

$$p \equiv q \longleftrightarrow (\forall w \in \Sigma^*, \hat{\delta}(p,w) \in F$$

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

~~2 states are distinct if~~

$$p \equiv q \longleftrightarrow (\forall w \in \Sigma^*, \hat{\delta}(p,w) \notin F \text{ and}$$

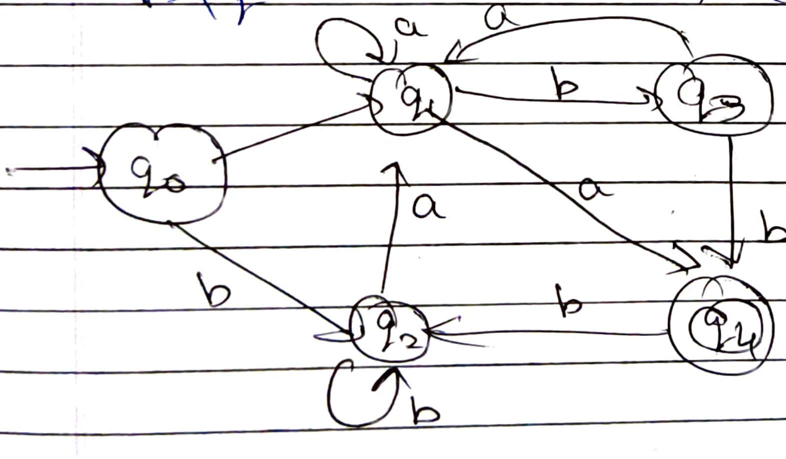
$\hat{\delta}(q,w) \notin F$.

2 states p and q are distinct iff:

$$p \not\equiv q \longleftrightarrow (\forall w \in \Sigma^*, \hat{\delta}(p,w) \in F \text{ and } \hat{\delta}(q,w) \notin F$$

or

$$p \not\equiv q \longleftrightarrow (\forall w \in \Sigma^*, \hat{\delta}(p,w) \notin F \text{ and } \hat{\delta}(q,w) \in F$$




| | | | | |
|---|---|---|---|---|
| $q_1$ | | | | |
| $q_2$ | | | | |
| $q_3$ | | | | |
| $q_4$ | X | X | X | X |
| | $q_0$ | $q_1$ | $q_2$ | $q_3$ |

10th March, 2023
Friday

PAGE NO.
DATE:

**Step 1** : Create lower triangle table, initially blank.

**Step 2** : For every pair of state $p, q$, if $p$ is Final & $q$ is not final, then mark $p\,q$.

**Step 3** : Loop until no change for iteration.
For each $(p, a)$ & input symbol $a$.
let $r = \delta(p, a)$ & $s = \delta(q, a)$

Let $\theta$ be unmarked

| | | | | |
|---|---|---|---|---|
| $q_1$ | X | | | |
| $q_2$ | | X | | |
| $q_3$ | X | X | X | |
| $q_4$ | X | X | X | X |
| | $q_0$ | $q_1$ | $q_2$ | $q_3$ |

$(p, a)$    $a$    $(r, s)$

$r = \delta(p, a)$      $s = \delta(q, a)$

* if $r, s$ is already marked, then mark $p, q$

$(q_0, q_1) \xrightarrow{a} (q_1, q_1)$
$\xrightarrow{b} (q_2, q_3)$

$(q_0, q_2) \xrightarrow{a} (q_1, q_1)$
$\xrightarrow{b} (q_2, q_2)$

$$(q_0, q_3) \xrightarrow{a} (q_1, q_1)$$
$$\xrightarrow{b} (q_2, q_4)$$

$$(q_1, q_2) \xrightarrow{a} (q_1, q_1)$$
$$\xrightarrow{b} (q_3, q_2)$$

$$(q_1, q_3) \xrightarrow{a} (q_1, q_1)$$
$$\xrightarrow{b} (q_3, q_4)$$
$$(q_2, q_3) \xrightarrow{a} (q_1, q_1)$$
$$\xrightarrow{b} (q_2, q_4).$$