DATE: PAGE:

Unit - 01

Automata Theory

It is the study of abstract computing devices or machines
Automaton = Abstract computing device

Notions

Alphabet = A finite, non-empty set of symbols
Denoted by Σ (Sigma)
Ex: Binary : $\Sigma = \{0, 1\}$
Digits : $\Sigma = \{0-9\}$
Alpha-numeric : $\Sigma = \{a-z, A-Z, 0-9\}$

String or Word = Finite sequence of Symbol or alphabet
Chosen from Σ
Empty String is ε (Epsilon)
length of string = Number of characters in Σ
Concentration can be done.

Powers of an alphabet

If $\Sigma = \{0, 1\}$
Null string = $\emptyset$
Empty string = Length is zero but there is some string i.e, ε
Σ^+ =
Σ^* = Including empty string

Applications of Finite Automata:

- Lexical analyzer of compiler
- Software for scanning large bodies of text for pattern finding
- ~~Des~~ Software for designing and checking in digital circuits.
- Software for verifying systems of all types that have finite no. of states.

Language:

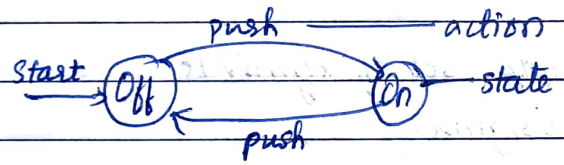
L is said to be language

L is set of all possible strings with some constraints

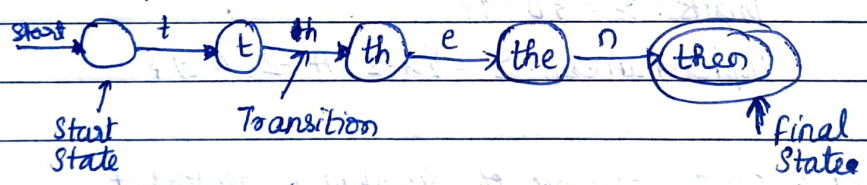
$\emptyset$ denotes empty language

Finite Automata : Example

1)



2)



Types: Deterministic FA

Non Deterministic FA

NFA with Epsilon [ε-NFA]

Σ^* denotes zero or more occurence

Σ^+ denotes one or more occurence

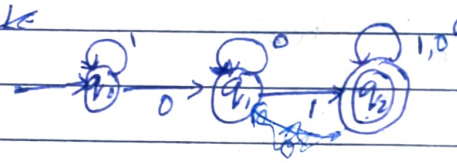
DFA

Consists of
- Q Finite set of states
- Σ Finite set of input symbols (alphabet)
- q_0 Start state
- F - Set of accepting states
- δ A transition function which is mapping between $Q \times \Sigma = Q$

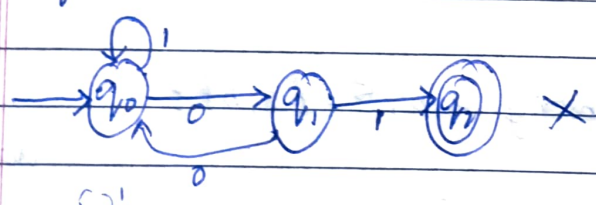
DFA is defined by 5 tuple

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

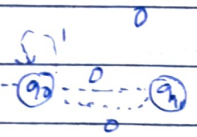
Ex: Build a DFA for $L = \{ w / w$ is binary string that contains 01 as substring $\}$



To build state diagram for this question, all the possible string consisting of 01 should taken and also it should reach final state



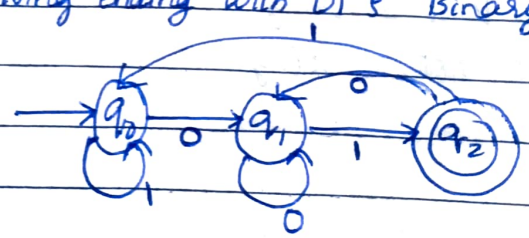
If q_1 to q_0 ~~there~~ their is transition of 0 then 001 is not satisfied.



001 is not reached till final state.

Ex: $L = \{ w \mid w$ is string ending with 01 $\}$ Binary

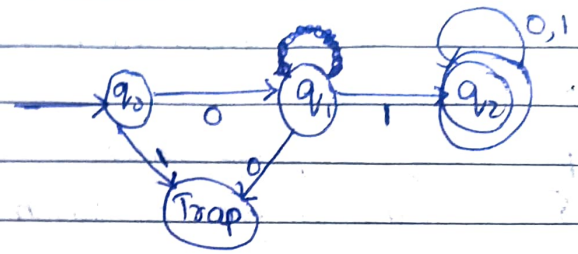
111<u>01</u>
000<u>01</u>
101<u>01</u>
001<u>01</u>
⋮



Is 100 accepted? :- No bcoz on applying 100 to DFA it won't reach final state so it is rejeted.

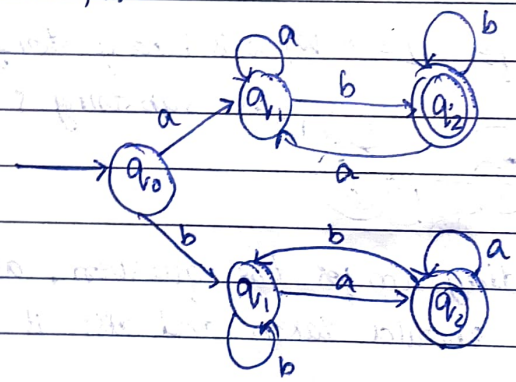
Ex: L = {w | Binary string where 01 should be the beginning of all strings}

01 1111
01 0101
01 000
01 0110

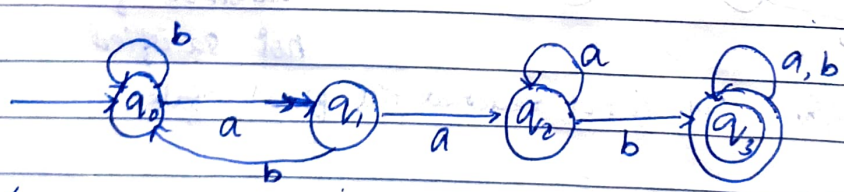


Ex: L = {w | w where leftmost symbol differs from rightmost symbol}
$\Sigma = \{a, b\}$

a — b
b — a



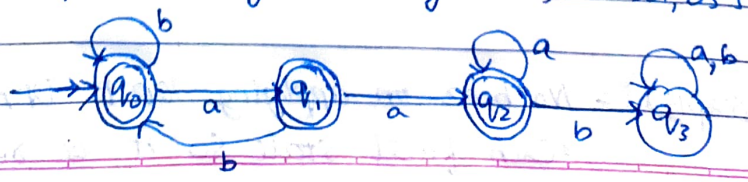
Ex: L = { ~~except~~ containing substring aab, $\Sigma = \{a, b\}$ }



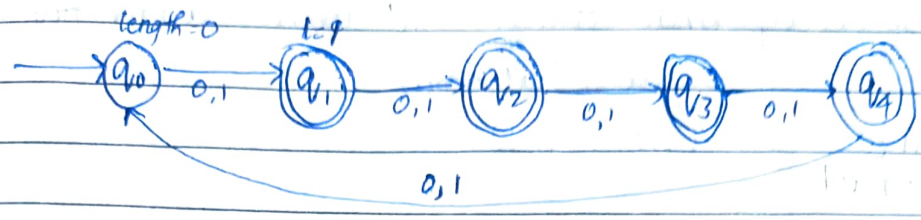
aabb ✓
aaab ✓
baab ✓
aaaab ✓
babbaabbba ✓

~~Here, aab is used in string~~
~~aab itself is not a string~~

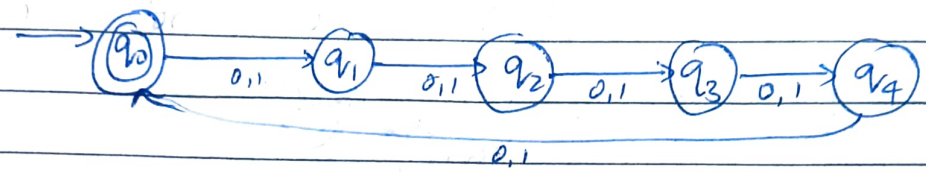
Ex: L = { Except containing substring aab, $\Sigma = \{a, b\}$ }



Ex: $L = \{w : |w| \bmod 5 != 0,\ w \in (0,1)\}$



Ex: $L = \{w : |w| \bmod 5 == 0$

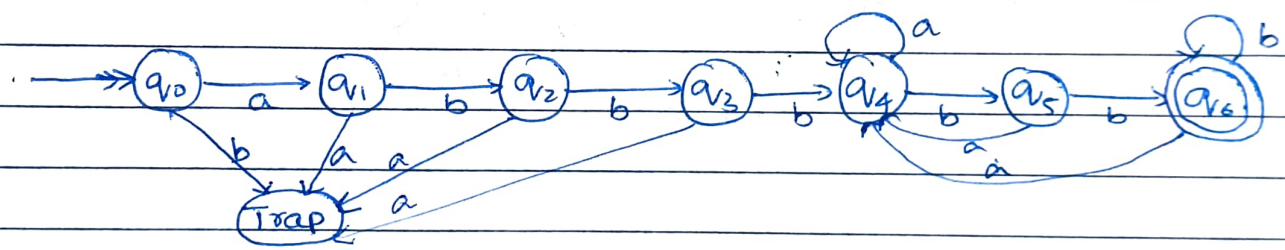


Ex: $L = \{ab^3 wb^2 : w \in \{a,b\}^*\}$

abbbbb : minimum string

* = zero or more occurence

Here w, can be present or may not.



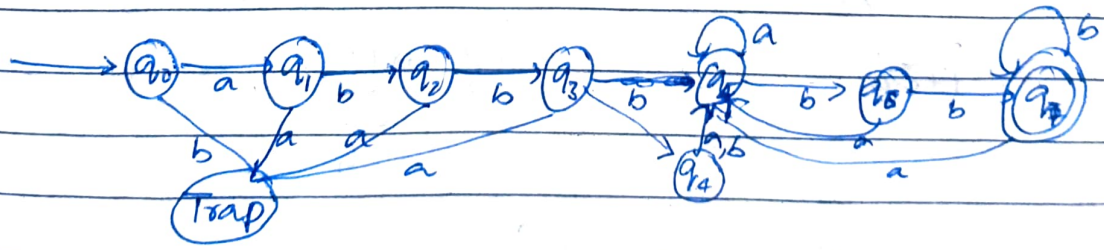
Ex: $L = \{ab^3 wb^2 : w \in \{a,b\}^+\}$

abbbabb or abbbbbb

Minimum string

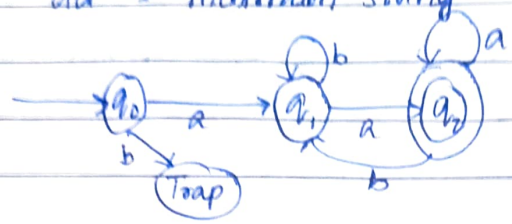
+ = one or more occurence

Here w, should have one or more occurence

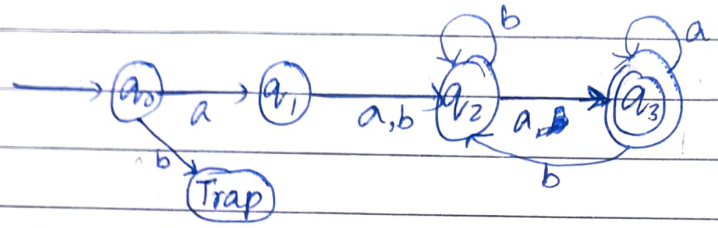


Ex: $L = \{ awa \mid \Sigma = \{a,b\}^* \}$

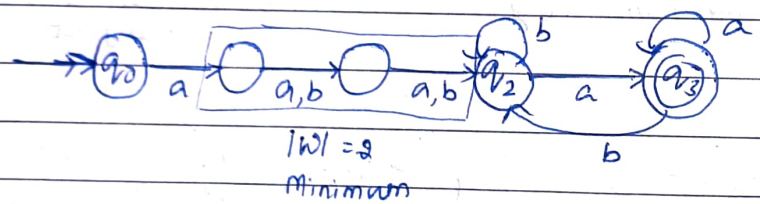
aa = minimum string



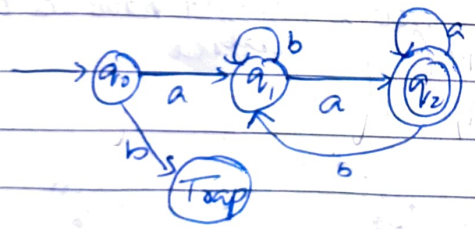
Ex: $L = \{ awa \mid \Sigma = \{a,b\}^+ \}$



Ex: $L = \{ awa \mid \Sigma = \{a,b\}, |w| \geq 2 \}$



Transition Table



δ	a	b
$\rightarrow q_0$ (start state)	q_1	–
q_1	q_2	q_1
$*q_2$ (final state)	q_2	q_1

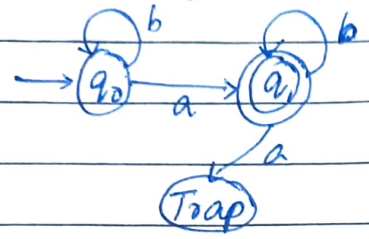
Σ

Q

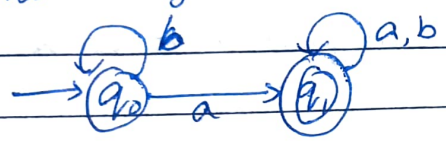
$\delta : Q \times \Sigma \rightarrow Q$

Ex: For $\Sigma = \{a, b\}$, construct DFA that accept the sets consisting of

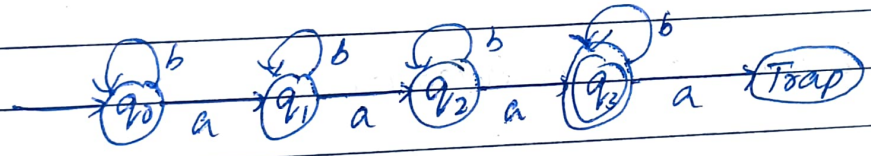
(a) all strings with exactly one a



(b) all strings with atleast one a



* (c) all strings with no more than 3 a

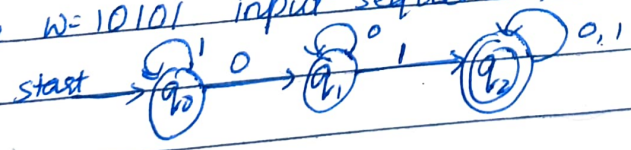


Extended Transition function δ^*

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

To check if string is rejected or accepted, δ^* is used.

Ex: Use $w = 10101$ input sequence, Generate δ^* to δ(Path)



$\hat{\delta}(q0, \varepsilon) = q0$

$\hat{\delta}(q0, 1) = \delta(\hat{\delta}(q0, \varepsilon), 1) = \delta(q0, 1) = q0$

$\hat{\delta}(q0, 10) = \delta(\hat{\delta}(q0, 1), 0) = \delta(q0, 0) = q1$

$\hat{\delta}(q0, 101) = \delta(\hat{\delta}(q0, 10), 1) = \delta(q1, 1) = q2$

$\hat{\delta}(q0, 1010) = \delta(\hat{\delta}(q0, 101), 0) = \delta(q2, 0) = q2$

$\hat{\delta}(q0, 10101) = \delta(\hat{\delta}(q0, 1010), 1) = \delta(q2, 1) = \underline{\underline{q2}}$

Accepted.

Let $M = (Q, \Sigma, q_0, A, \delta)$ be finite automata. Then Extended Transition function is defined as

$$\delta^* = Q \times \Sigma^* \longrightarrow Q$$

Language of DFA

DFA accepts string w if there is a path from q_0 to an accepting (or final) state that is labeled by w

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

$L(A)$ = all strings that lead to an accepting state from q_0

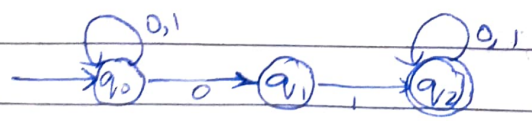
NFA = Non Deterministic Finite Automata

NFA has 5 tuple = $\{Q, \Sigma, q_0, F, \delta\}$

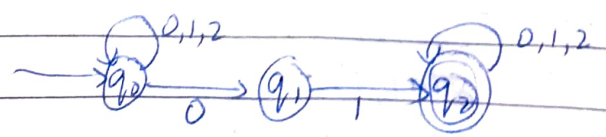
$$\delta = Q \times \Sigma \longrightarrow 2^Q \quad \text{[Subset of } Q\text{]}$$

There exist more than one possible transition from one state on the same input symbol.

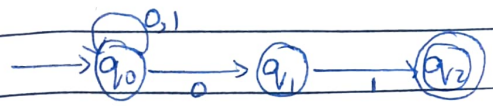
Eg: $L = \{ w \mid$ string should contain $01 \}$



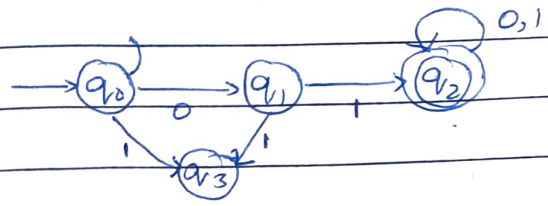
Eg: $L = \{ w \mid$ string should contain 01 and $\Sigma = \{0, 1, 2\}$



Ex: $L = \{ w \mid w \text{ ends in } 01 \}$

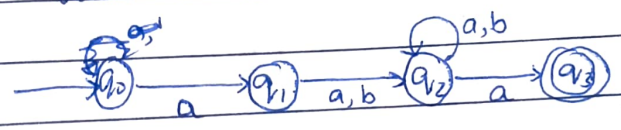


Ex: $L = \{ w \mid w \text{ begins with } 01 \}$



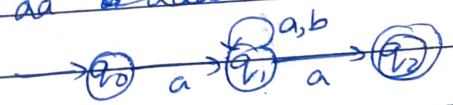
Ex: $L = \{ aWa \mid \Sigma = \{a,b\}^+ \}$

aaa aba

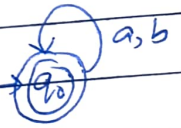


Ex: $L = \{ aWa \mid \Sigma = \{a,b\}^* \}$

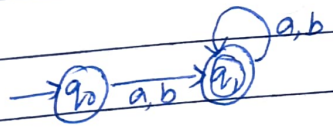
aa ~~aaa~~ 0



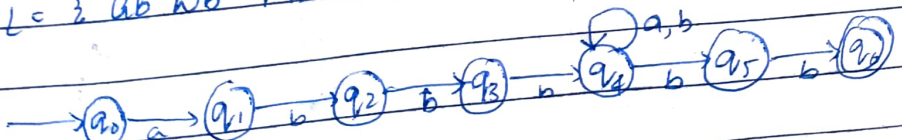
Ex: $(a+b)^*$



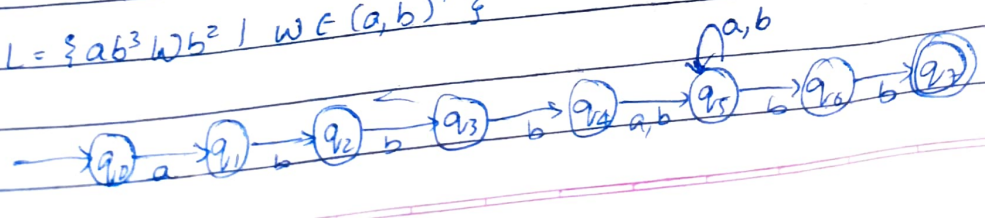
Ex: $(a+b)^+$



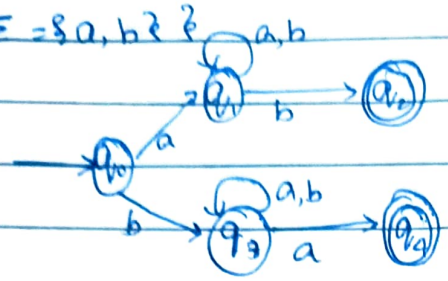
Ex: $L = \{ ab^3Wb^2 \mid W \in (a,b)^* \}$ abbbbb



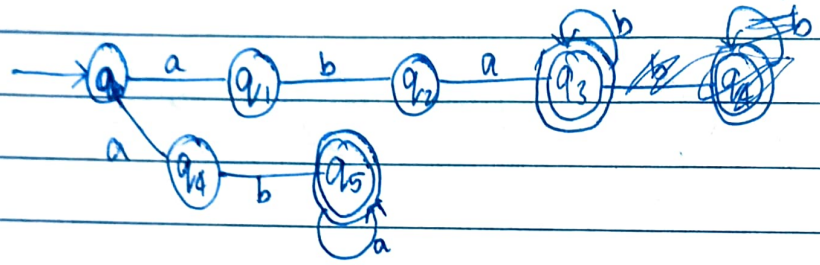
Ex: $L = \{ ab^3Wb^2 \mid W \in (a,b)^+ \}$



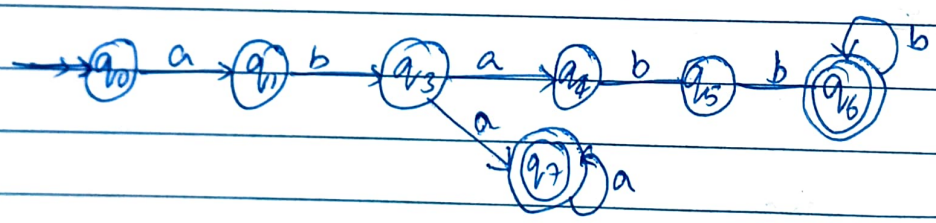
Ex: NFA for $L = \{w \mid$ where leftmost differ from rightmost symbol, $\Sigma = \{a, b\}\}$



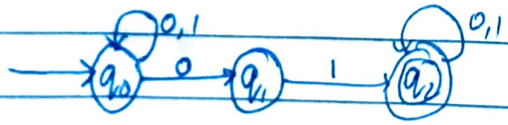
Ex: $L = \{w \mid w$ is $abab^n$ or $aba^n \mid n >= 0\}$



Ex: $L = \{w \mid w$ is $abab^n$ or $aba^n \mid n >= 2\}$



Extension of δ to NFA Paths



Ex: 1010

$\hat{\delta}(q0, \varepsilon) = q0$

$\hat{\delta}(q0, 1) = \delta(q0, 1) = q0$

$\hat{\delta}(q0, 10) = \delta(q0, 0) = \{q0, q1\}$

$\hat{\delta}(q0, 101) = \delta(q0, 1) \cup \delta(q1, 1) = \{q0, q2\}$

$\hat{\delta}(q0, 1010) = \delta(q0, 0) \cup \delta(q2, 0) = \{q0, q1, q2\}$

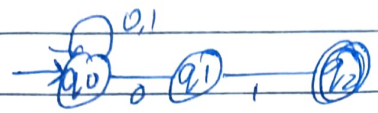
Accepted

Since q2 (final state) is reach

NFA To DFA

Ex:

NFA



	0	1
$\rightarrow q_0$	q_0, q_1	q_0
q_1	$\emptyset$	q_2
$* q_2$	$\emptyset$	$\emptyset$

Steps:

Enumerate all possible subsets

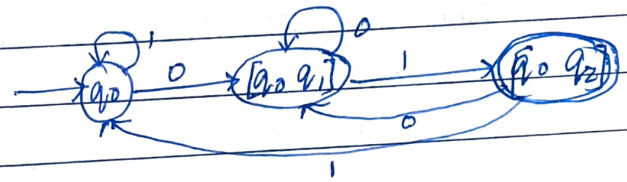
Determine transitions

Retain only those states reachable from q_0 (starting state)

	0	1
q_0	q_0, q_1	q_0
q_1	$\emptyset$	q_2
q_2	$\emptyset$	$\emptyset$
$[q_0, q_1]$	q_0, q_1	q_0, q_2
$[q_0, q_2]$	$q_0\ q_1$	q_0
$[q_1, q_2]$	$\emptyset$	q_2
$[q_0\ q_1, q_2]$	$q_0\ q_1$	$q_0\ q_2$

$\longrightarrow$

	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



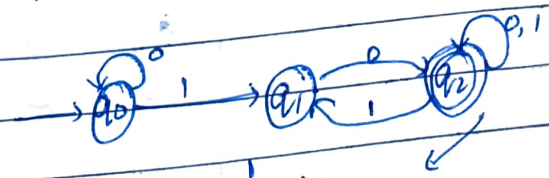
Whichever states contains final state in NFA will be final state of DFA.

To avoid enumerating all of power set, we use lazy creation

	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

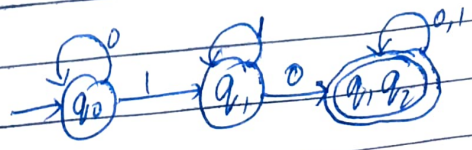
Determining the transitions which are required by seeing transition of one state.

Ex:



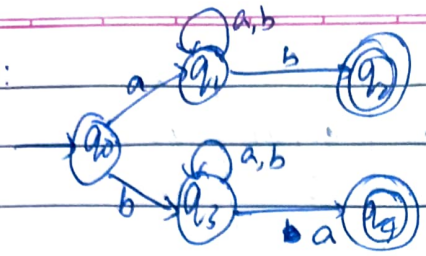
	0	1
$\rightarrow q_0$	q_0	q_1
q_1	$q_1\ q_2$	q_1
$* q_1, q_2$	$q_1\ q_2$	q_1, q_2

$\longrightarrow$

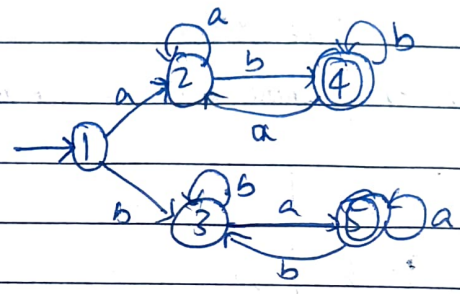


DATE: PAGE:

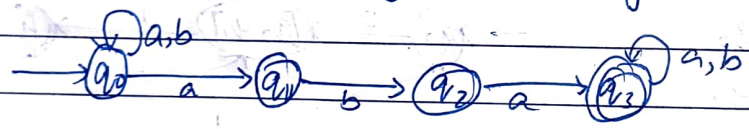
Ex 3:



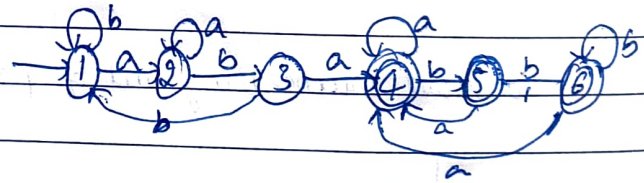
	a	b
1 → q_0	q_1	q_3
2 q_1	q_1	q_1 q_2
3 q_3	q_3 q_4	q_3
4 * $[q_1 q_2]$	q_1	q_1 q_2
5 * $[q_3 q_4]$	q_3 q_4	q_3



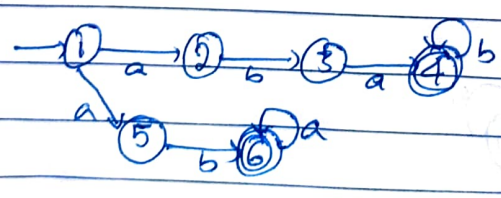
Ex 4: NFA to accept strings containing aba



	a	b
1 → q_0	$[q_0, q_1]$	q_0
2 $[q_0 q_1]$	$[q_0 q_1]$	$[q_0, q_2]$
3 $[q_0 q_2]$	$[q_0, q_1, q_3]$	$[q_0]$
4 * $[q_0 q_1 q_3]$	$[q_0 q_1 q_3]$	$[q_0 q_2 q_2]$
5 * $[q_0 q_2 q_2]$	$[q_0 q_1 q_3]$	$[q_0 q_3]$
6 * $[q_0 q_3]$	$[q_0 q_3 q_1]$	$[q_0 q_3]$



Ex 5:



	a	b
→ 1	2,5	∅
2,5	∅	3,6
*3,6	4,6	∅
*4,6	5	4
*6	6	∅
*4	∅	4