~~code~~

INTRODUCTION To TOC ①

⇒ L A G

language

automata

grammar

⇒ symbols → {a, b, c, 0, 1, 2 ....}

alphabet → Σ (a, b)

→ finite set of symbol

string → sequence of alphabets

Language → collection of strings

AUTOMATA ②

$\Rightarrow$ languge → finite
→ infinite

$\Rightarrow$ $\Sigma = \{a, b\}$
$L_1 = 2$
$\{ab, ba, aa, bb\}$ } finite languguage

$\Rightarrow$ $\Sigma = \{a, b\}$
$L_2 = \{$string with atleast 1 a$\}$
$L_2 = \{a, aa, aaa \ldots, ab, abb \ldots\}$ } ∞

$\Rightarrow$ to check whether string belongs to language we use automata

automata → FA (finite automata)
PDA (Pushdown automata)
TM (turing machine)

KLEENE CLOSURE

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

$\Sigma^0 = \{$set of all strings of length 0$\}$

$\Sigma^1 = \{$ —— " —— 1 $\}$

$\Sigma^2 = \{$ —— " —— 2 $\}$

$\vdots$

$\Sigma^* = \Sigma^0 + \Sigma^1 + \Sigma^2 \ldots \Sigma^n$ (kleene closure)

GRAMMAR

⇒ grammar = $\{V, T, P, S\}$

$V \rightarrow$ Variable

$T \rightarrow$ terminal

$P \rightarrow$ Production rule

$S \rightarrow$ Start symbol

⇒ You can check whether string is part of language by using (grammar || automata)

⇒ e.g

$S \longrightarrow aSb \mid \varepsilon$

minimum $\longrightarrow \varepsilon \longrightarrow ab \longrightarrow aaSbb$
this goes on

⇒ generate grammar for

$L = \{ n_a(\omega) = n_b(\omega) \}$

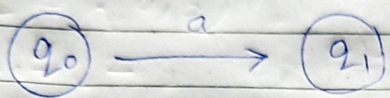
$S \longrightarrow SS \mid aSb \mid bSa \mid \varepsilon$

Page No.
Date / /

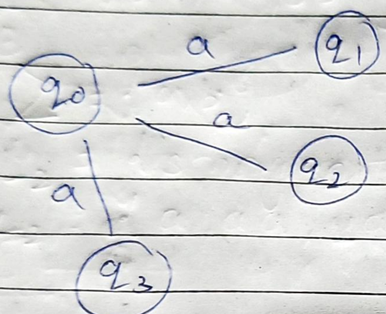
✱ DFA

⟹ Deterministic finite Automata.

⟹ non-multiple output for single input



} DFA



} NFA

⟹ DFA $(Q, \Sigma, \delta, q_0, F)$

Q → set of finite states

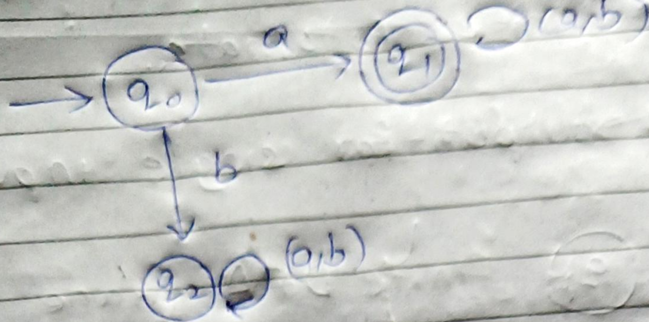
Σ → inputs

δ → transition table

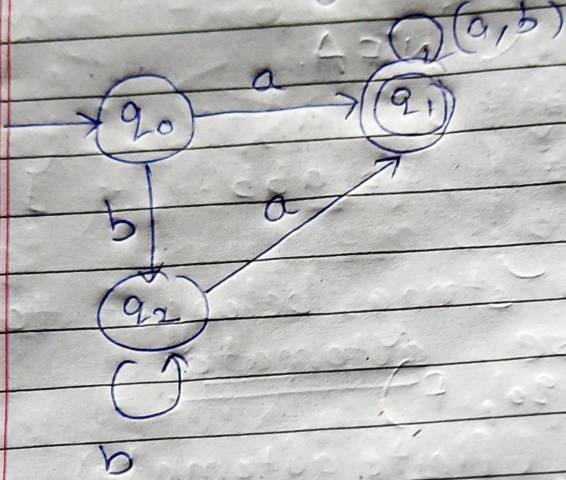
q_0 → initial state

F → final state $(F \subseteq Q)$

⇒ DFA are string start with a



⇒ containing 'a'



⇒ ending with 'b'