NFA / NDFA

Non- Deterministic FA

DATE: / /
PAGE:

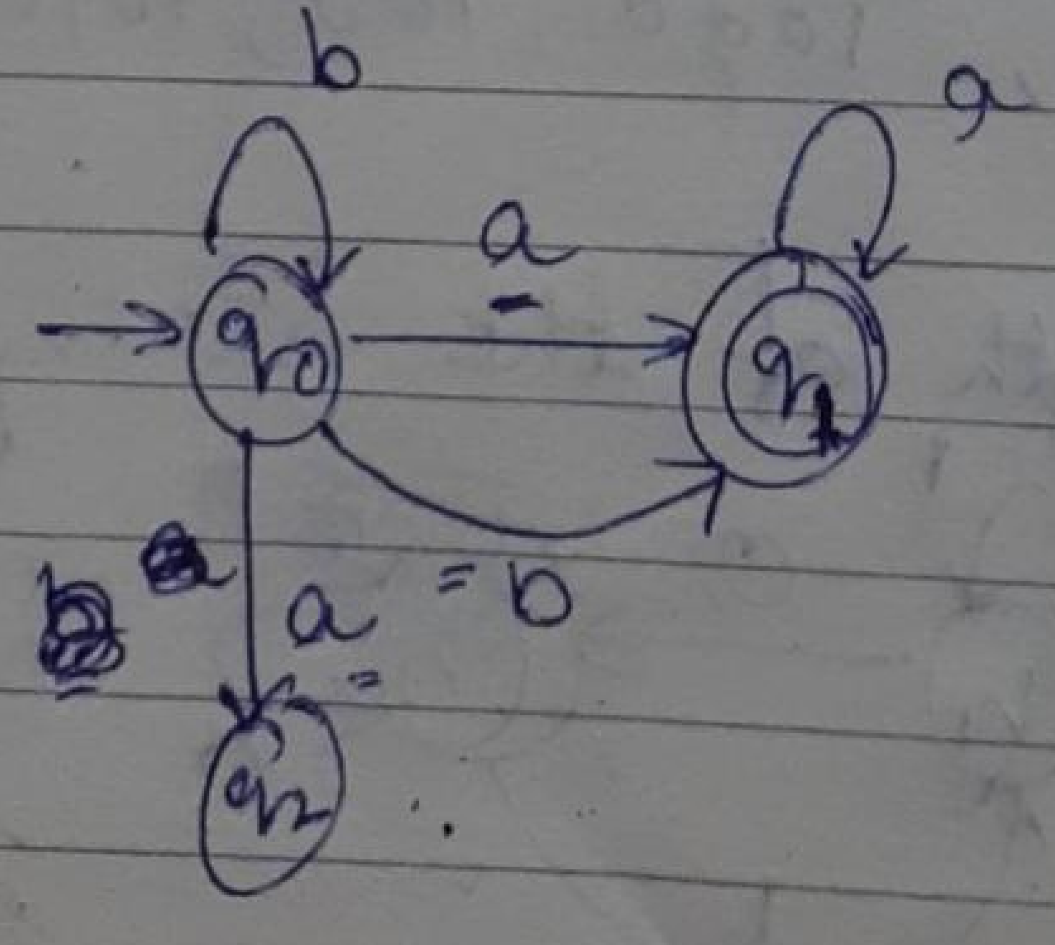
→ The FA are called NFA when there exist many paths to specific i/p from the current state to the next state.

→ It is easy to construct NFA that DFA from a given regular language

→ Every NFA is not DFA, But each NFA can be transmitted to DFA.

→ NFA is defined in the same way as DFA, but with two exceptions.

- It contains multiple next state
- It contains ∈ transitions.



⇒ Formal Definition of NFA

NFA also have 5 tuples same as DFA, but with different transition fn:

$$\delta : Q \times \Sigma \; 2^{Q}$$

δ ↓ Transition fn.

2^Q ↓ Multiple states

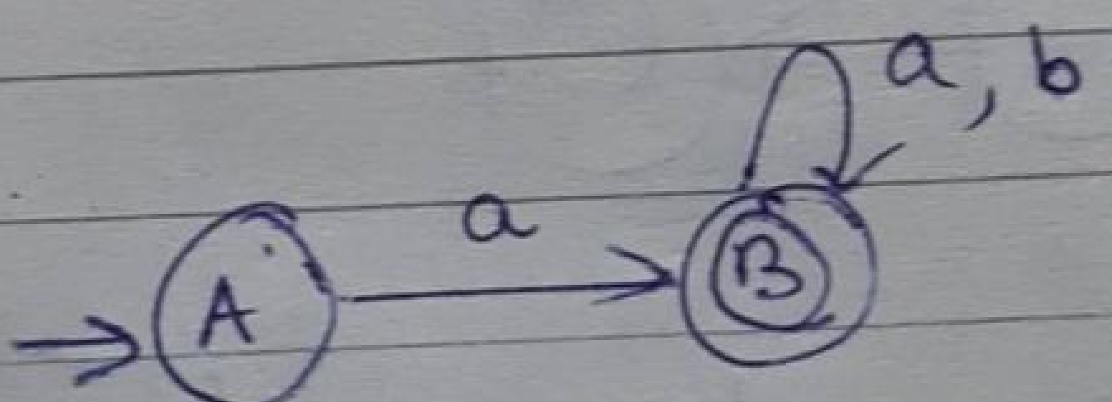
DFA

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

Example:

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

Ex. 1 $L_1 = \{$ starts with 'a' $\}$



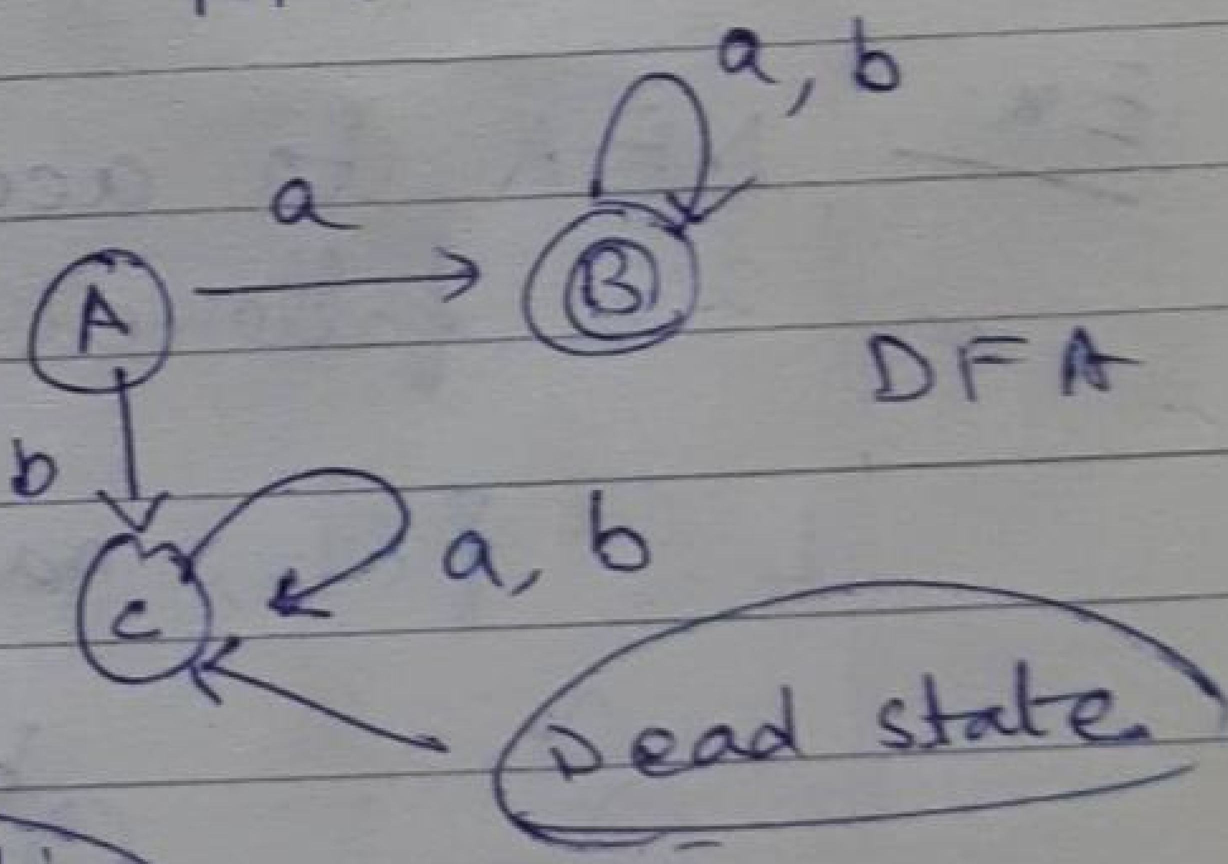
check for ab

A → B.

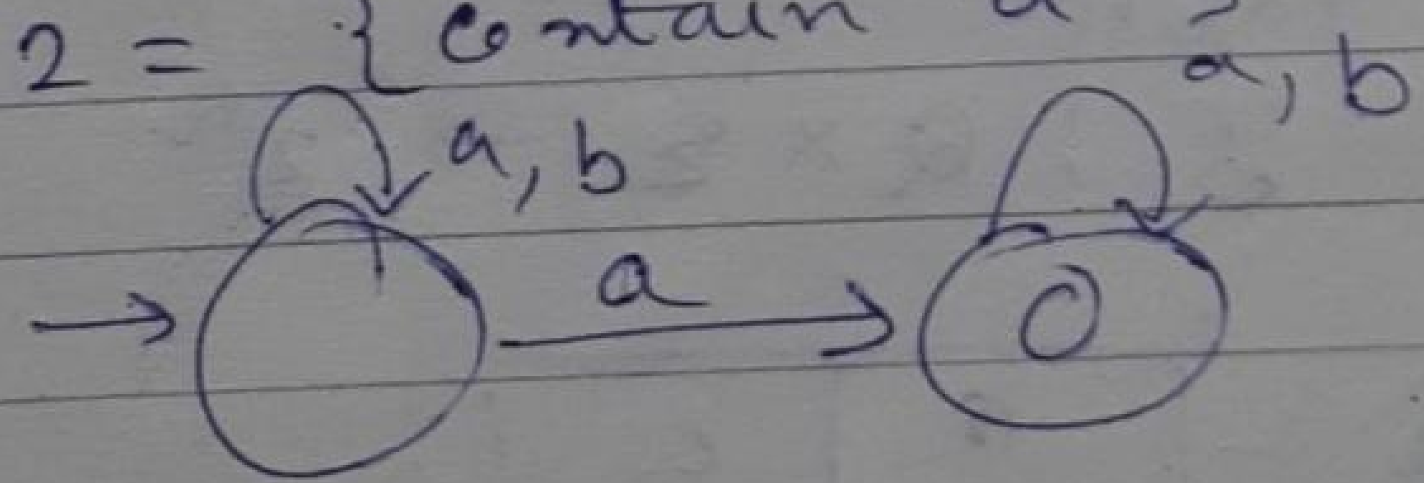
check for b

No input for b.

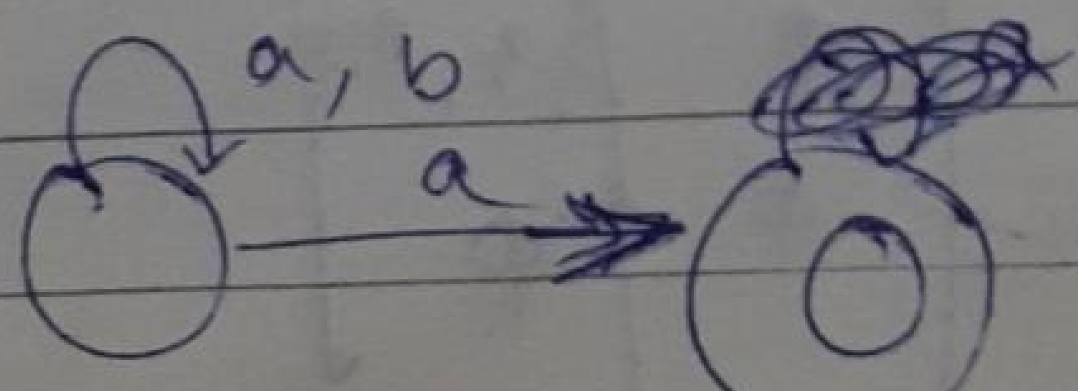
It is Dead configuration



Ex. 2 $L2 = \{$ contain 'a' $\}$



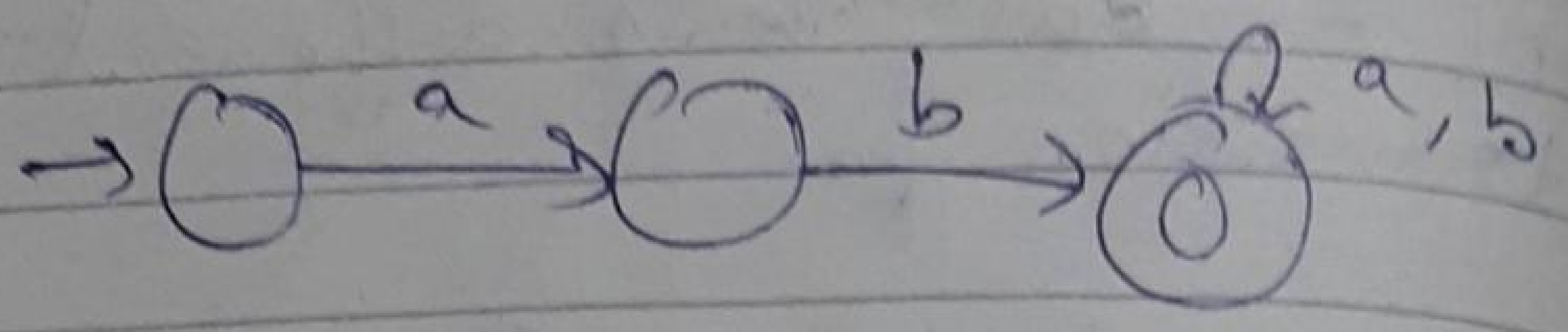
$L3 = \{$ Ends with 'a' $\}$



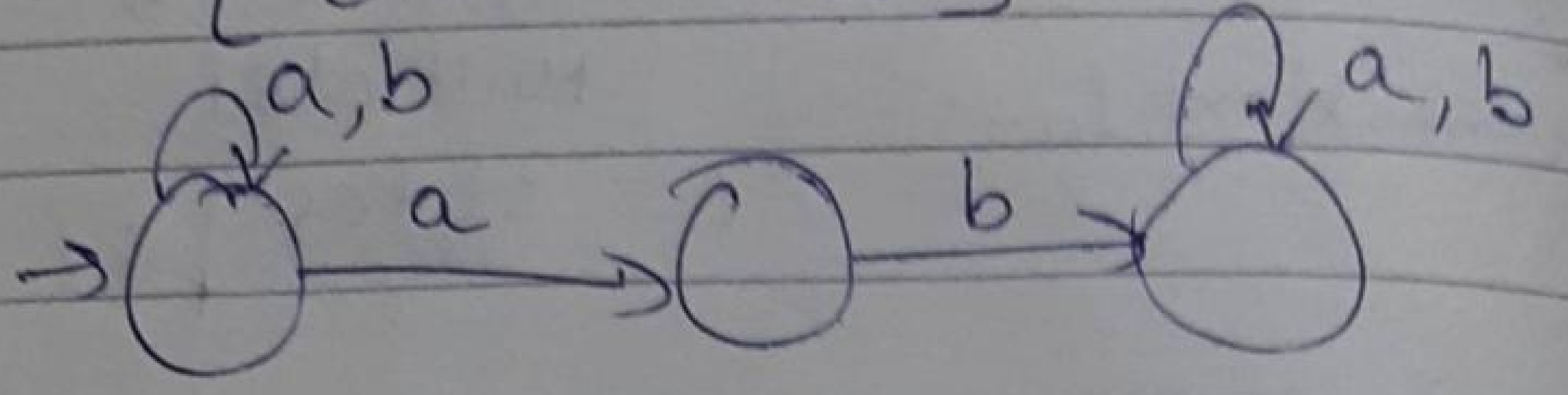
Scanned by TapScanner

DATE: / /
PAGE:

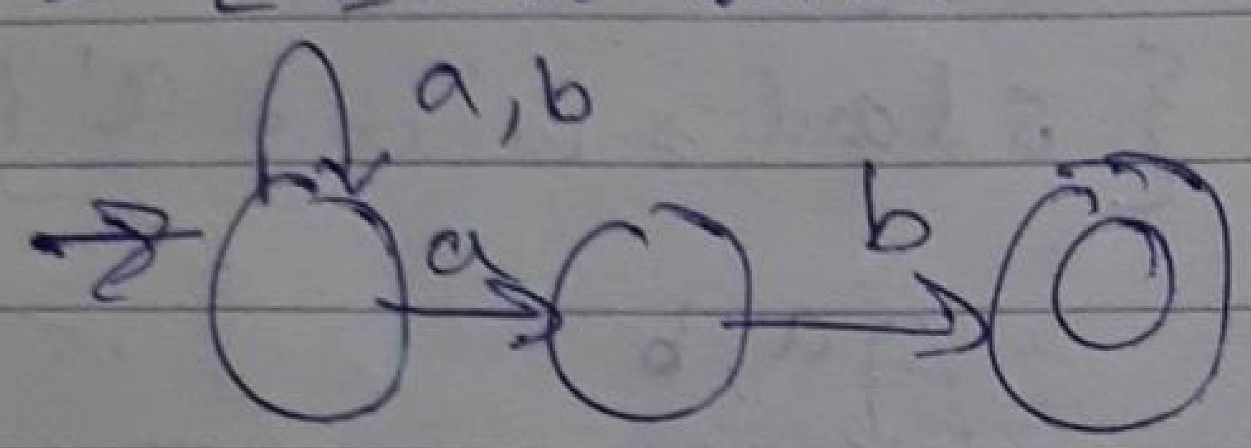
L4 = { starts with 'ab' }



L5 = { contains 'ab' }

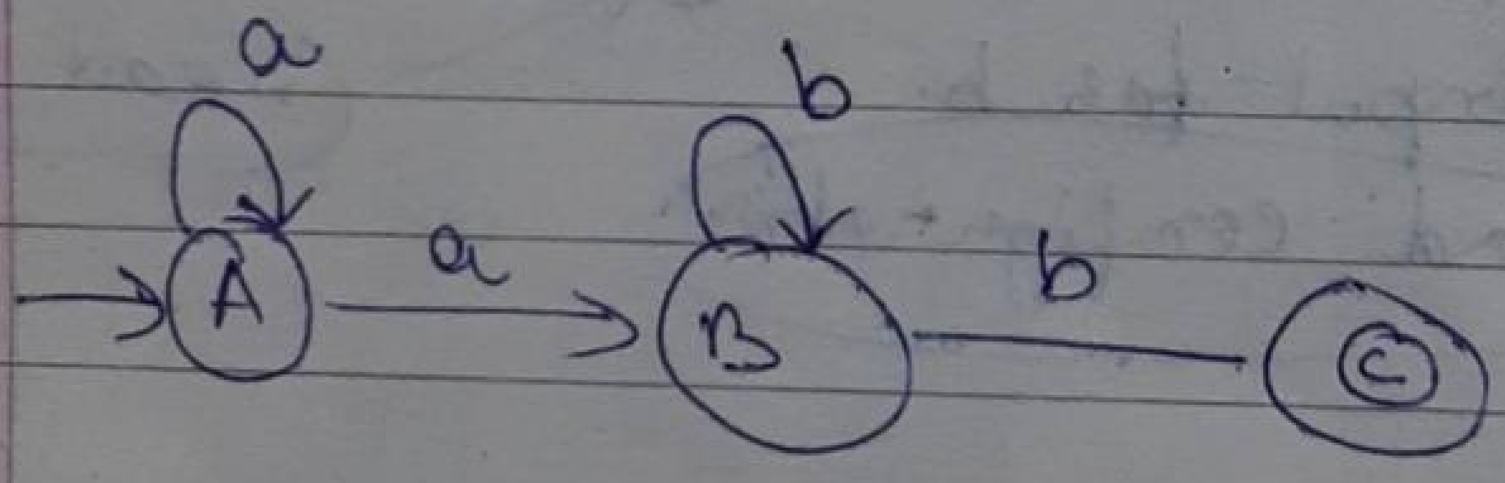


L6 = { Ends with 'ab' }



Ex NFA to accept the strings of all a's followed by all b's.

L = {ab, aaab, aaaabbbb, abbb....



Transition Table.

$\delta : Q \times \Sigma \rightarrow 2^{Q}$

$Q \backslash \Sigma$	a	b
q_0	$\{q_0, q_1\}$	ϕ
q_1	ϕ	$\{q_1, q_2\}$
q_2	ϕ	ϕ

DATE: / /
PAGE:

Definition

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

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

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

$\delta = Q \times \Sigma \rightarrow 2^Q$

$q_0 = q_0$

$F = \{q_2\}$

String verification

check abb

$\delta(q_0, abb) \rightarrow q_1$

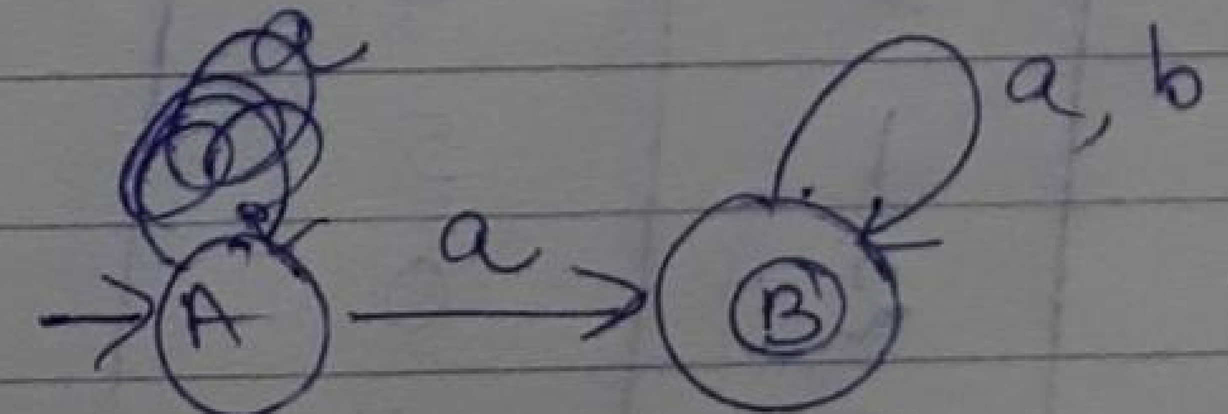
$\delta(q_1, bb) \rightarrow q_1$

$\delta(q_1, b) \rightarrow q_2$

$\delta(q_2, \epsilon) \rightarrow$ Accepted

Ex. NFA to accept string that start with 'a', $\Sigma = \{a, b\}$

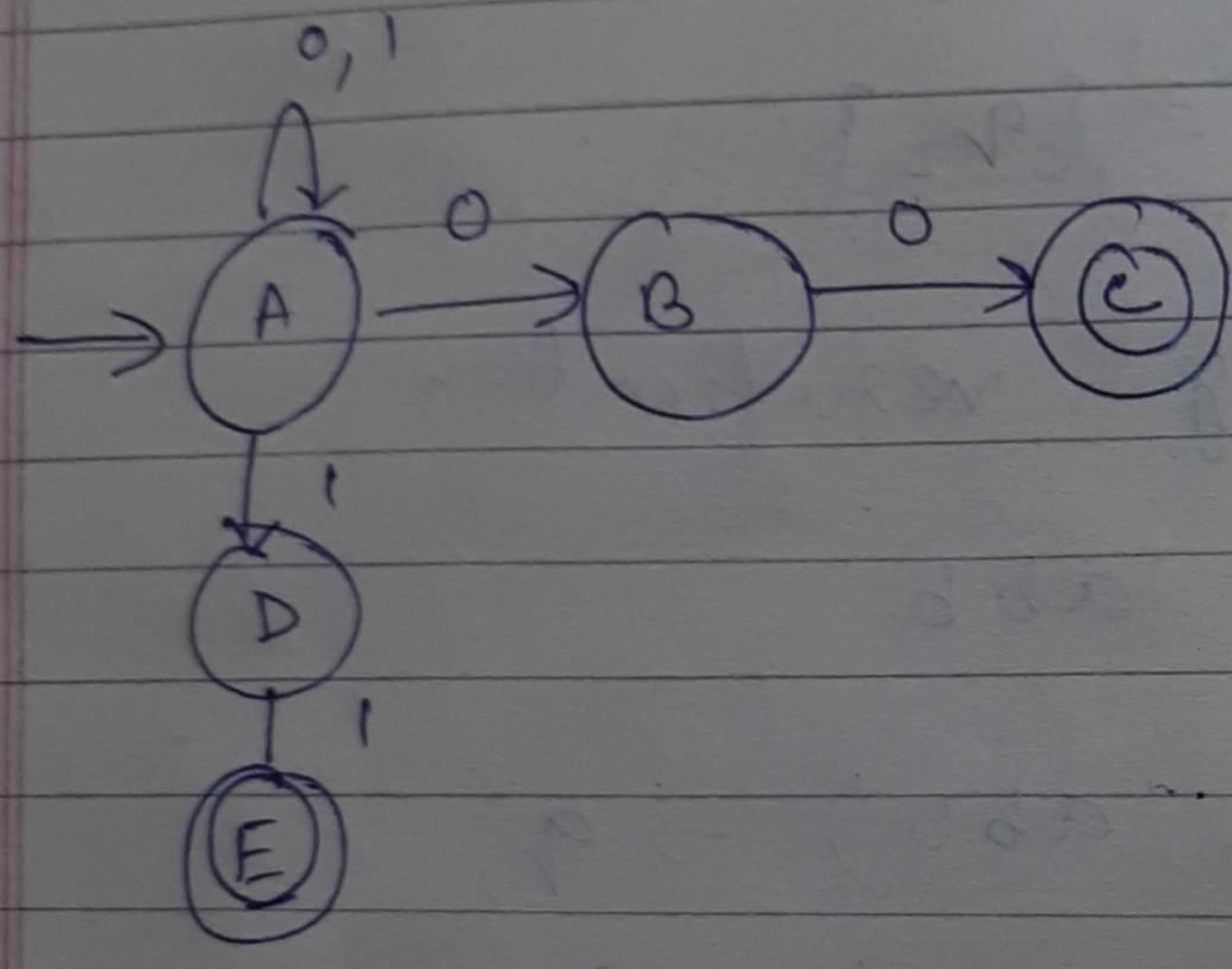
$L = \{a, aa, aaaa, ab, abaab, aabaa \ldots\}$



DATE: / /
PAGE:

Ex. NFA contai any no. of 0's & 1's followed by either '00' or '11'

L = {00, 11, 01011, 01000, 0100011, 1100, . . . }



Scanned by TapScanner