CS & IT ENGINEERING

Theory of Computation

DFA, NFA

Lecture No.- 04

By- Venkat sir

Recap of Previous Lecture

Topic

Topic

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

min DFA

DFA Construction

length of the string

exactly $n \Rightarrow (n+2)$ states

atleast $n \Rightarrow (n+1)$ states

atmost $n \Rightarrow (n+2)$

Div by $n \Rightarrow n$

a's div by 2 (and) # b's div by 3

{2 * 3 = 6 states}

Topic : Expressive Power

Number of languages accepted by particular automata is knowns as expressive power.

(TM > LBA > PDA > FA)

1. Expressive power of NFA and DFA same. Hence every NFA is converted into DFA.
2. Expressive power of NPDA is more than DPDA. Hence conversion not possible
3. Expressive power of DTM and NTM is same.

Topic : DFA

m, n

NOTE:

Number of States of DFA on length conditions

(i) Then in the given condition on length if one number divide other number then number of states of minimal DFA for "and" automata is LCM of given condition. LCM(m,n)

(ii) Number of states of minimal DFA for "OR" automata is GCD of given condition. GCD(m,n)

(iii) In the given length condition one number not divide other number then

→ If GCD of given condition is 1 then number of states of 'and' automata OR automata is multiplication of given condition.

Topic : DFA

as div by 2 (and) as div by 4 = 8

1. Length of the string divisible by 2 (and) divisible by 4 ⇒ LCM(2,4) = 4
2. Length of string divisible by 2 (OR) divisible by 4. ⇒ GCD(2,4) = 2
3. Length of string divisible by 3 (and) divisible by 4 ⇒ LCM(3,4) = 12
4. Length of string divisible by 3 (OR) divisibly by 4 ⇒ LCM(3,4) = 12
5. Length of string divisible by 6 (OR) divisibly by 8 ⇒ LCM(m,n)
6. Number of a's divisible by 6 (and) number of divisible by 8. ⇒ LCM(6,8) = 24 (as 6s × 8s = 48)
7. length of the string div by 4 and div by 6 ⇒ LCM(4,6) = 12
8. " " " div by 3 (OR) div by 6 ⇒ GCD(3,6) = 3

(Q) Construct min DFA over $\Sigma = \{a, b\}$ where 2nd input Symbol is from right end. (R.H.S)

(a, b) $\frac{a}{2}$ $\frac{a/b}{1}$

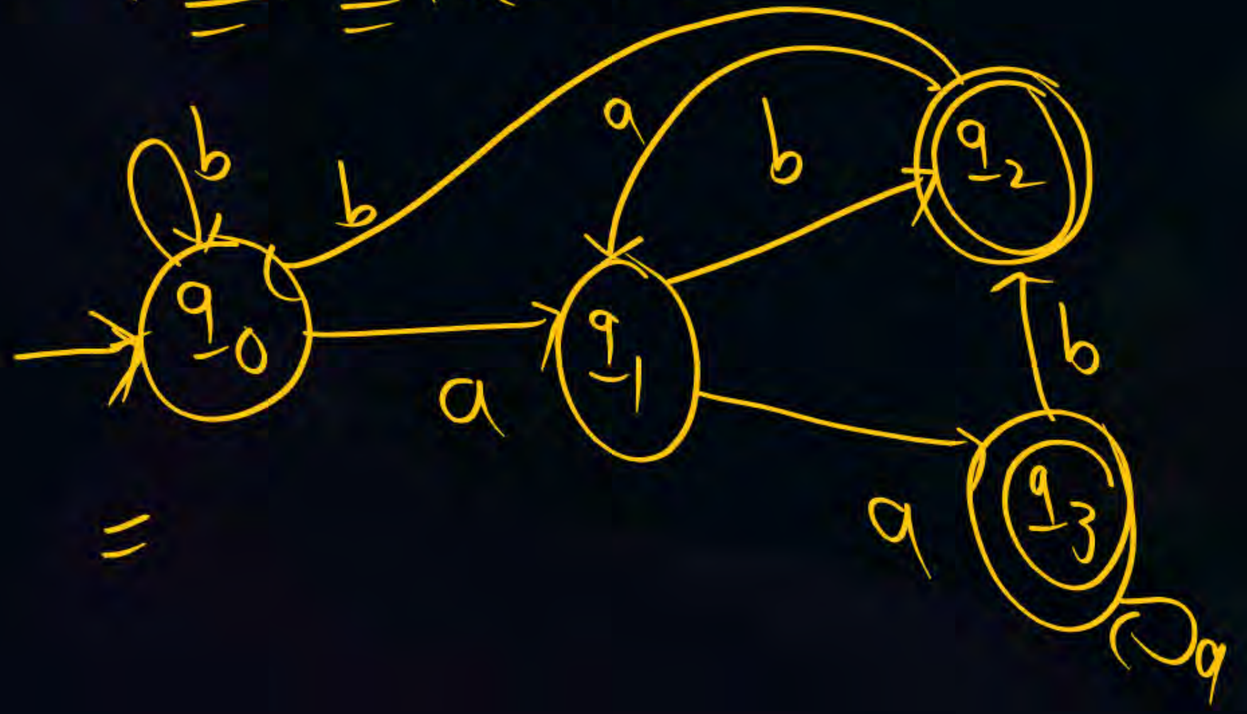
$L = \{aa, ab, aab, bab \cdots\}$

(a, b) $\frac{a\ a}{a\ b}$

aaa/ab

ab/ab

abbba



=

(Q) Construct min DFA over $\Sigma = \{a, b\}$ where 3rd input Symbol is a from right end. (R.H.S).

(a^*b) — $\frac{a}{3}$ $\frac{a|b}{2}$ $\frac{a|b}{1}$

(a^*b)

aaa

aab

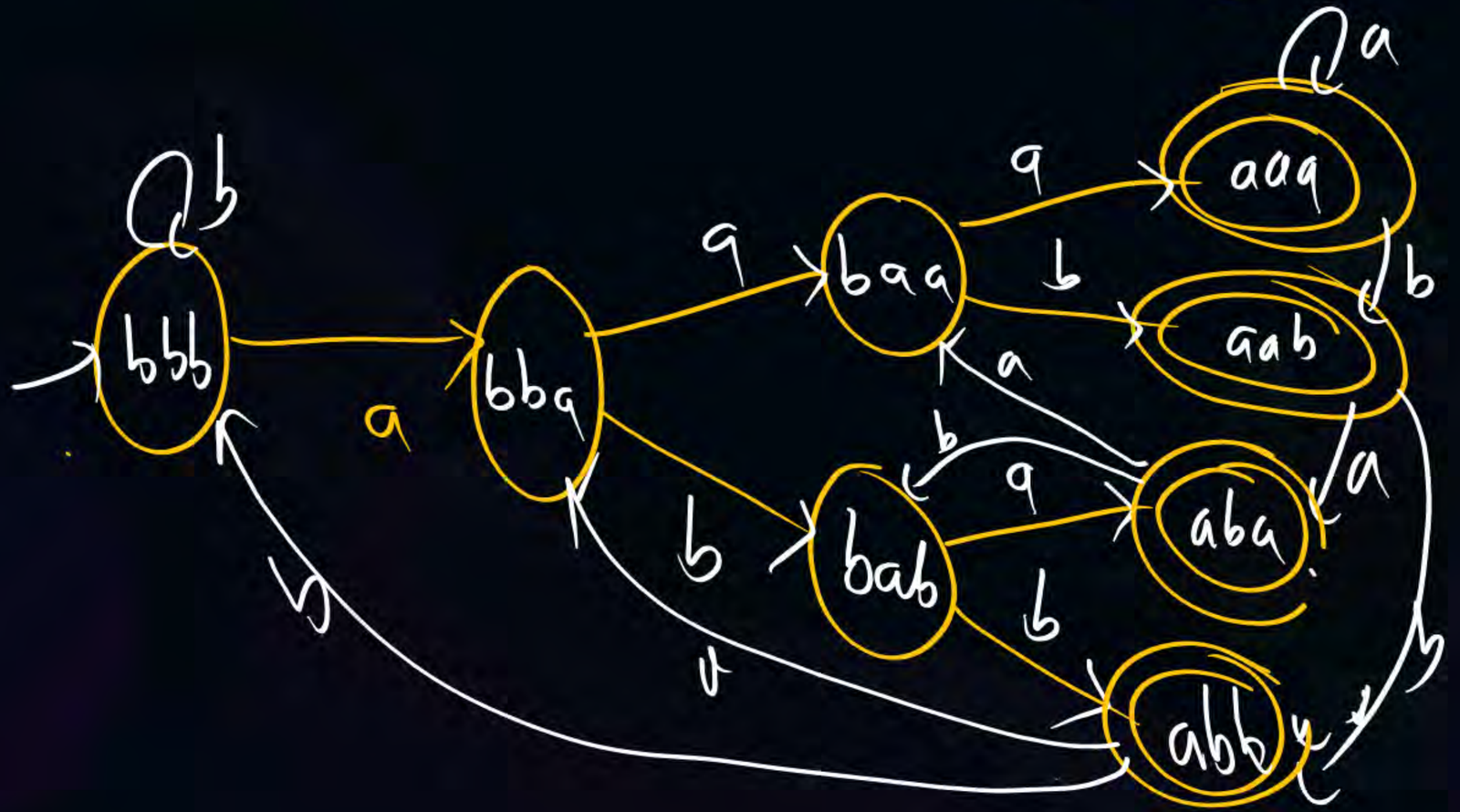
aba

abb

(Q) Construct min DFA over $\Sigma = \{a, b\}$ where 3rd input symbol is a from right end. (R.H.S).

$(a|b)\,\frac{a}{3}\;\frac{a|b}{2}\;\frac{a|b}{1}$

$= 2^3$



aaaa
aaab
aaba
aabb
aaba
abaa
abab

(Q) Construct min DFA over $\Sigma = \{a, b\}$ where n^{th} input symbol is a from right end. (R.H.S)

$2 \to 4$

$3 \to 8$

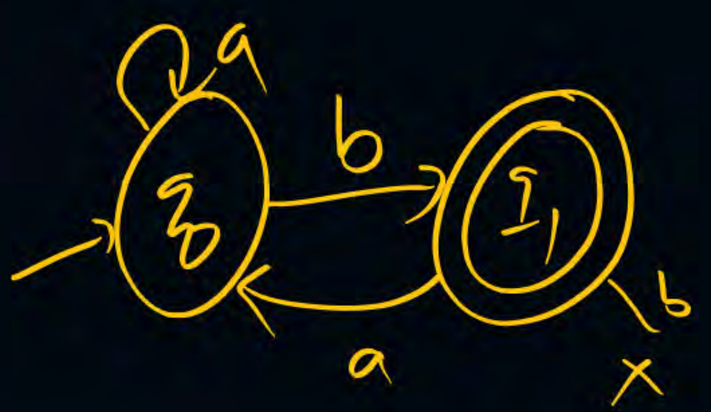
$\vdots$

$n^{th} \to$ (2) States

8^{th} input a from end.

$2^8 \Rightarrow$ 256 states

NFA 9 ! min



NFA

DFA

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

DFA, NFA

Topic : Non-Deterministic Finite Automaton (NFA)

In NFA from the given state on the given input symbol there may be 0 number of transition (or) 1 transition (or) more than one transition exist.

NFA is formally define as:-

NFA $\{Q, \Sigma, q_0, F, \delta\}$

δ: DFA: $Q \times \Sigma \to Q$

δ: NFA: $Q \times \Sigma \to 2^Q$ (or) powerset (Q)

Q	-	Finite number of states (set of state)
Σ	-	Input alphabet
q_0	-	initial state → one
F	-	Set of final states
δ	-	transition function

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

$\{q_0, q_1\}$

powerset

$\delta(q_0, a) = \{\}\ \{q_0\}\ \{q_1\}\ \{q_0, q_1\} = 4$

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

1 2 3 4 5 6 7 8

powerset 2^3

2^n

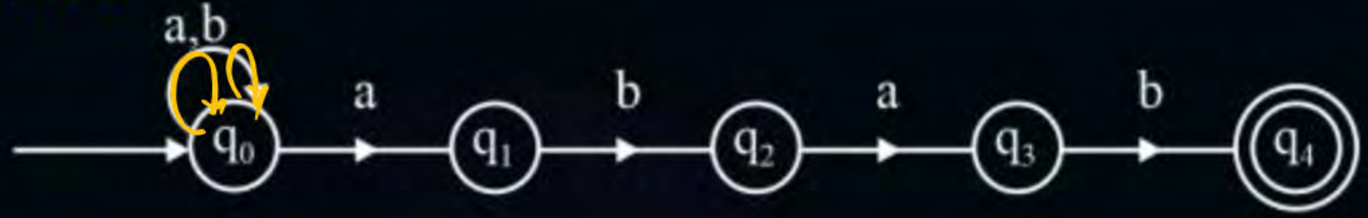
Topic : Non-Deterministic Finite Automaton

- Construction of NFA is easy than DFA.
- Minimization not possible for NFA
- Complementation not possible for NFA
- NFA from the given state on the given input string multiple state possibility may be exist.
- Language recognization is easy in DFA compare to NFA.
- In NFA, for valid string also automata may halt in non-final state.
- In NFA for the valid even though multiple non-final transition exist for one final state transition should exit.
- All DFA are NFA but all NFA need not be DFA.

Topic : NFA

Construct the NFA that accept all strings of a's & b's where every string ending with abab.

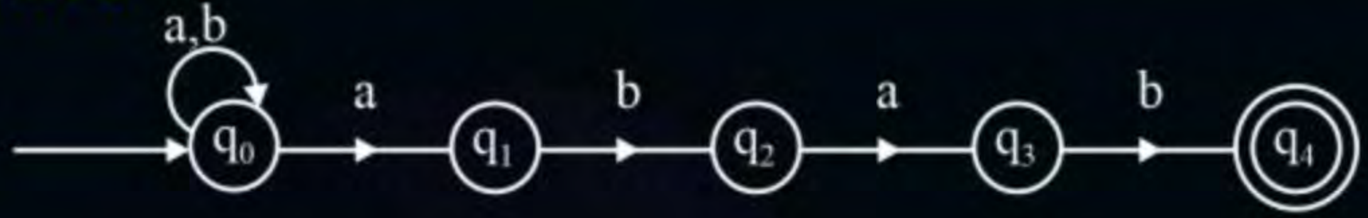
NFA:



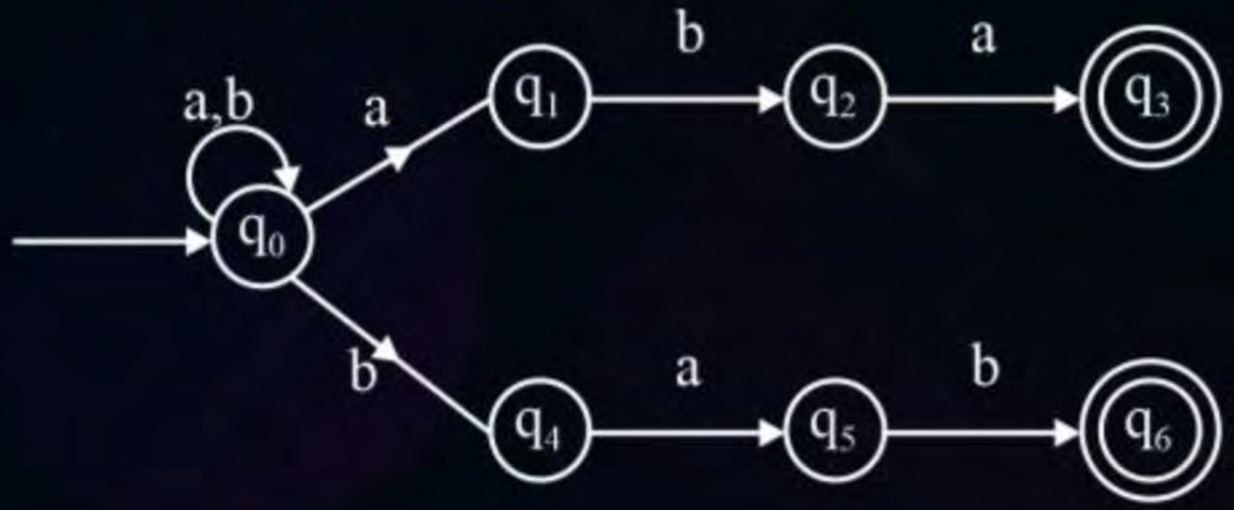
Topic : NFA

Construct the NFA that accept all strings of a's & b where every string ending with abab.

NFA:

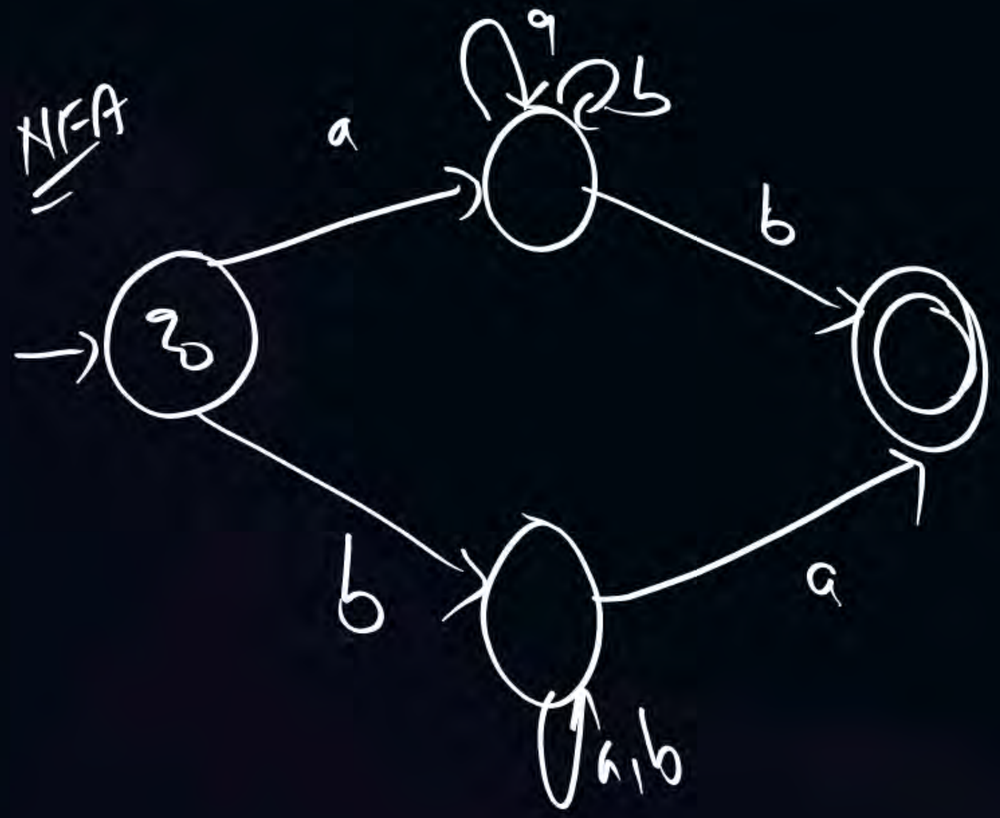


Each string ending with aba or bab



(Q) each string starting and ending with different symbol over $\Sigma = \{a, b\}$

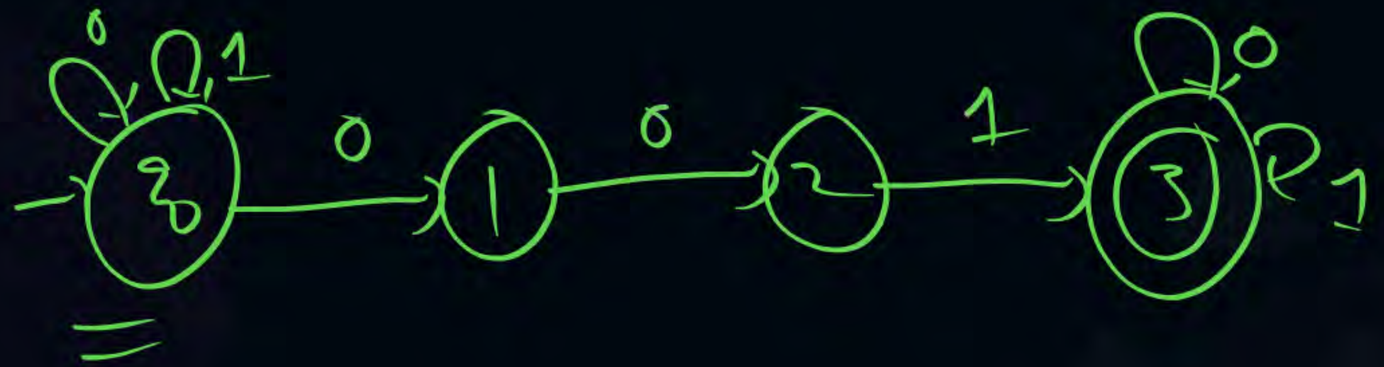
$a(a/b)^*b \;//\; b(a/b)^*a$



(Q) each string having 001 Substring over $\Sigma = \{0, 1\}$

$(0/1)^*\,001\,(0/1)^*$

NFA

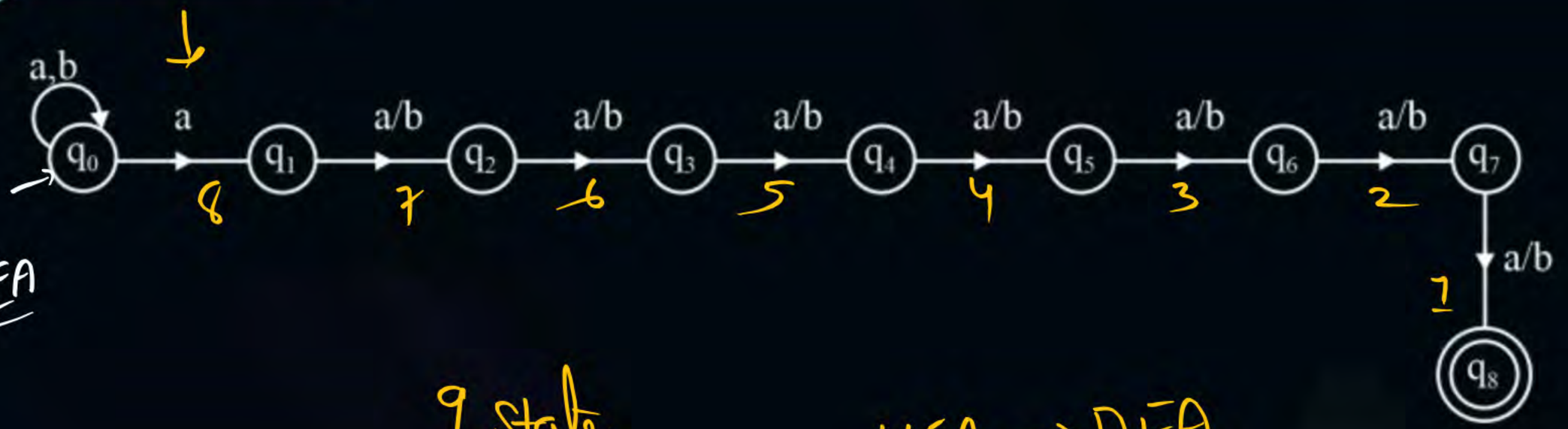


(Q) 8th input symbol is a from end. (R.H.S)

DFA

$2^8 = 256$

Topic : NFA



NFA

9 States

NFA ⟹ DFA

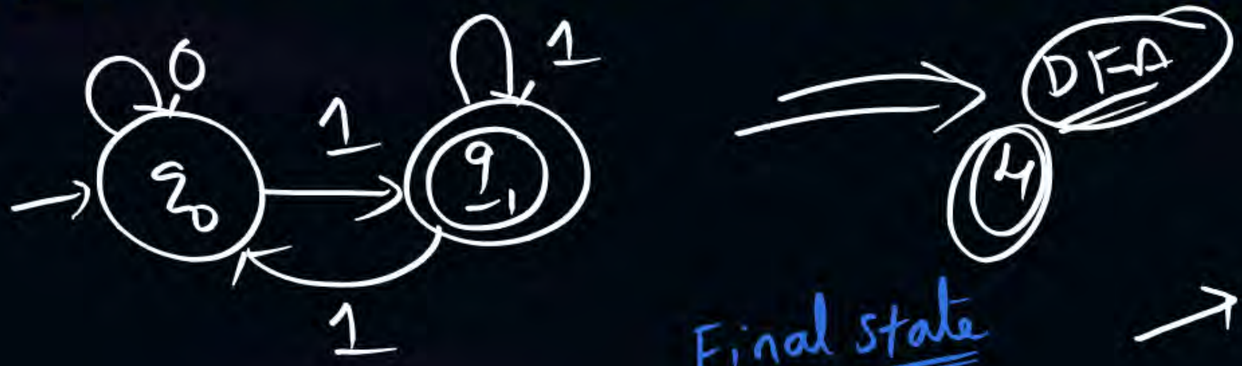
8th input symbol is a while reading from right side

Topic : NFA

Note:- The NFA that accept strings of a's & b's where n^{th} input symbol is a while reading a strings from w right hand side requires (n+1) states.

(Q) Construct DFA for following NFA ? [Subset Construction Alg]

NFA



Final state

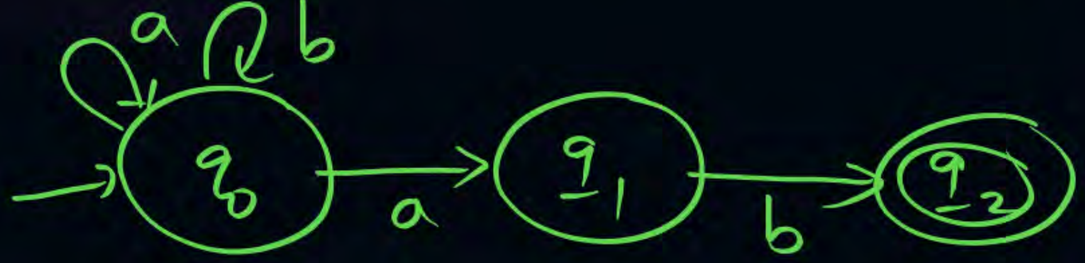
	0	1
→ q_0	q_0	q_1
q_1	$q_{D.S}$	q_0q_1
$q_{D.S}$	$q_{D.S}$	$q_{D.S}$
q_0q_1	q_0	q_0q_1

NFA

	0	1
→ q_0	q_0	q_1
q_1		q_0q_1

(Q) Construct DFA for given NFA?

NFA



NFA

	a	b
→ q_0	$\{q_0, q_1\}$	q_0
q_1	—	q_2
q_2	—	—

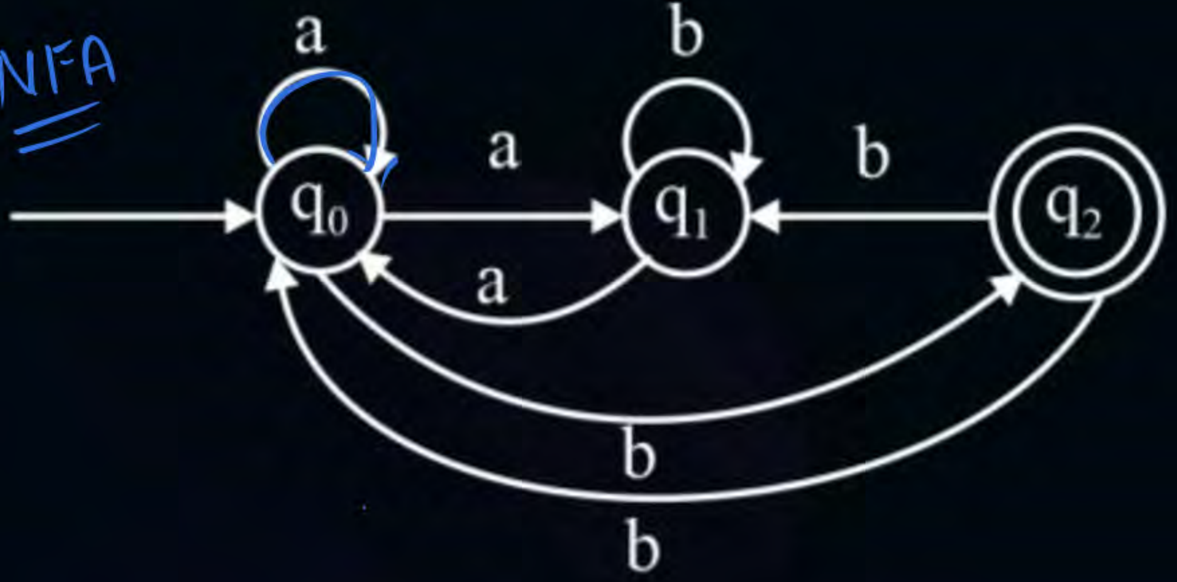
③ DFA

	a	b
→ q_0	q_0q_1	q_0
q_0q_1	q_0q_1	q_0q_2
q_0q_2	q_0q_1	q_0

Topic : NFA to DFA Conversion

Consider the following NFA what is the equivalent DFA for given NFA?

NFA



DFA ? 5 States

	a	b
→ q_0	q_0q_1	q_2
q_0q_1		
q_2		

NFA ⇒ DFA [Subset Construction]

2 ⟶ 4

3 ⟶ 3

3 ⟶ 5

NFA

Subset Const.

DFA

(1)

	a	b
→ q_0	q_0	q_0
q_1	q_1	q_1
q_2	q_2	q_2

	a	b
→ q_0	q_0	q_0

[MCQ] GATE 2025 [1 Mark]

#Q. A regular language L is accepted by a non-deterministic finite automaton (NFA) with n states. Which of the following statement(s) is/are FALSE?

NFA → n

→ DFA

1 ... 2^n

A T L may have an accepting NFA with $< n$ states.

B T L may have an accepting DFA with $< n$ states.

C T There exists a DFA with $\leq 2^n$ states that accepts L

D F Every DFA that accepts L has $> 2^n$ states.

THANK - YOU