CS & IT ENGINEERING

Theory of Computation

NFA

Lecture No.- 02

By- Venkat sir

Recap of Previous Lecture

Topic ① Emptyness problem → No final state

Topic ② finiteness problem → Loop (or) Cycle
→ unreachable state
→ state from which final state is not reachable.

③ → Equivalence prob

DFA

9^{th} input from end is a

2^9

DFA → {512}

NFA → 10 ($n+1$)

Topics to be Covered

- **Topic** Finite Automaton & Regular Languages.
- **Topic** Pushdown Automata & Context free Languages.
- **Topic** Turing Machine & Recursive Enumerable Languages.
- **Topic** Undecidability.

Topic : Deterministic Finite Automata

FORMAL DFA :

DFA is defined as

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

Q : Finite set of states

Σ : Input alphabet

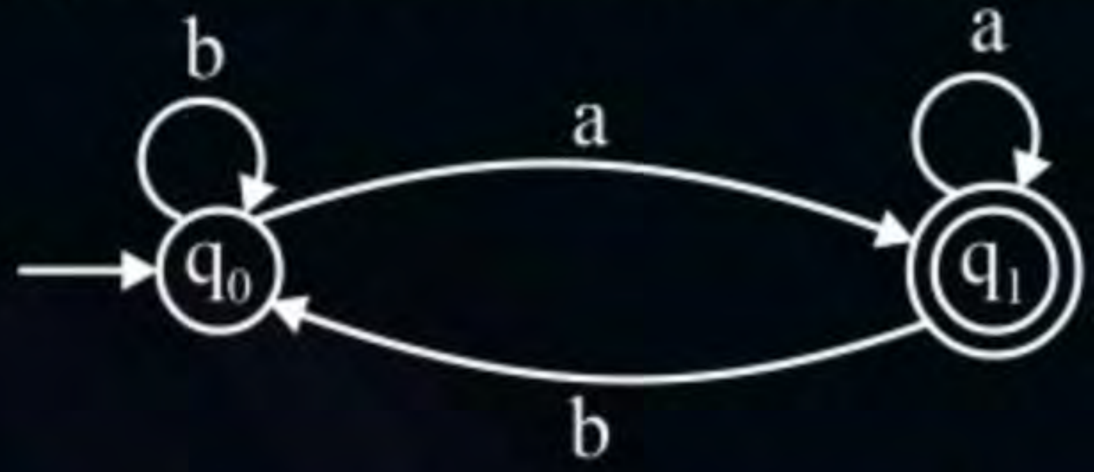
q_0 : Initial state

F : Set of final states

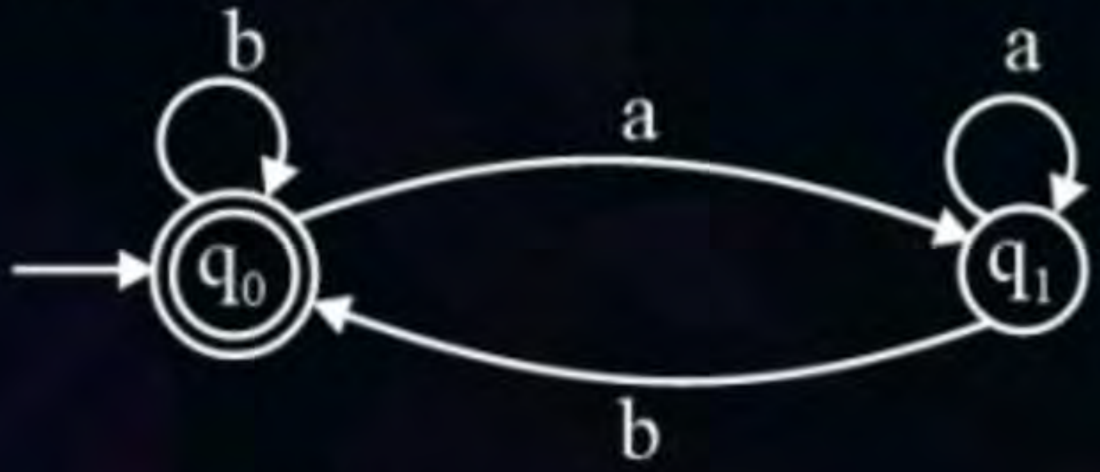
δ : Transition function $Q * \Sigma \rightarrow Q$

Topic : Complement of DFA

By interchanging final and non final states we can convert into complement DFA.



Set of all strings ending with a after complement

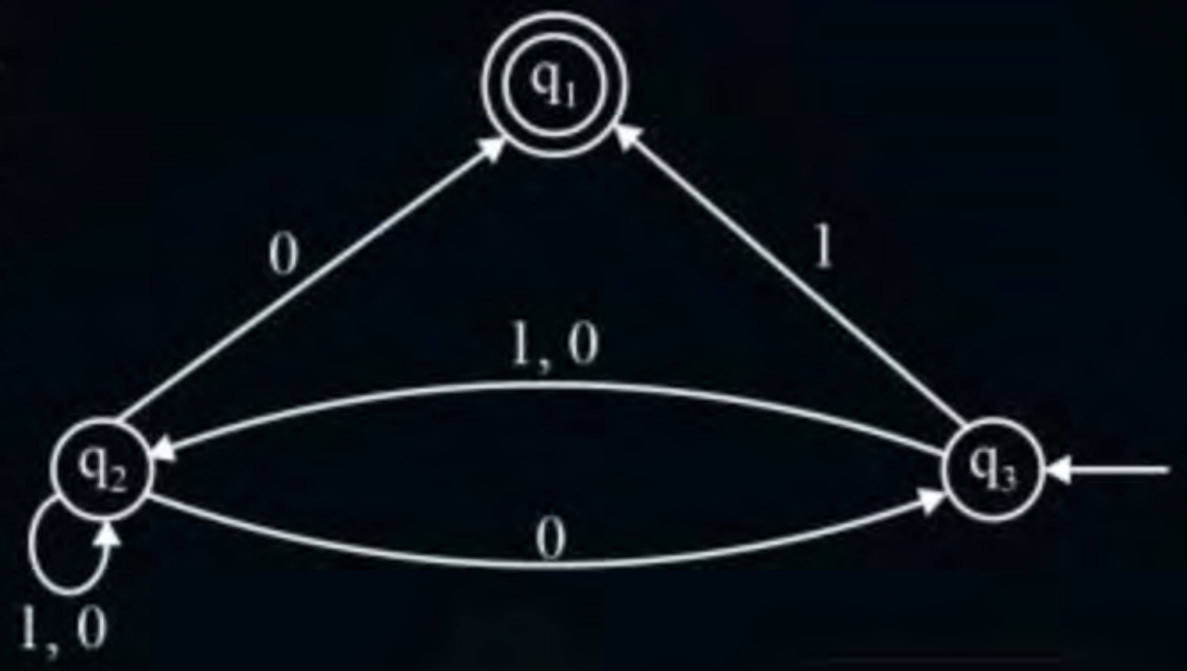


Set of all string .... Ending with

MCQ

#Q. Consider the NFA M shown below.

Let the language accepted by M be L. Let L_1 be the language accepted by the NFA M_1, obtained by changing the accepting state of M to a non-accepting state and by changing the non-accepting state of M to accepting states. Which of the following statements is true?



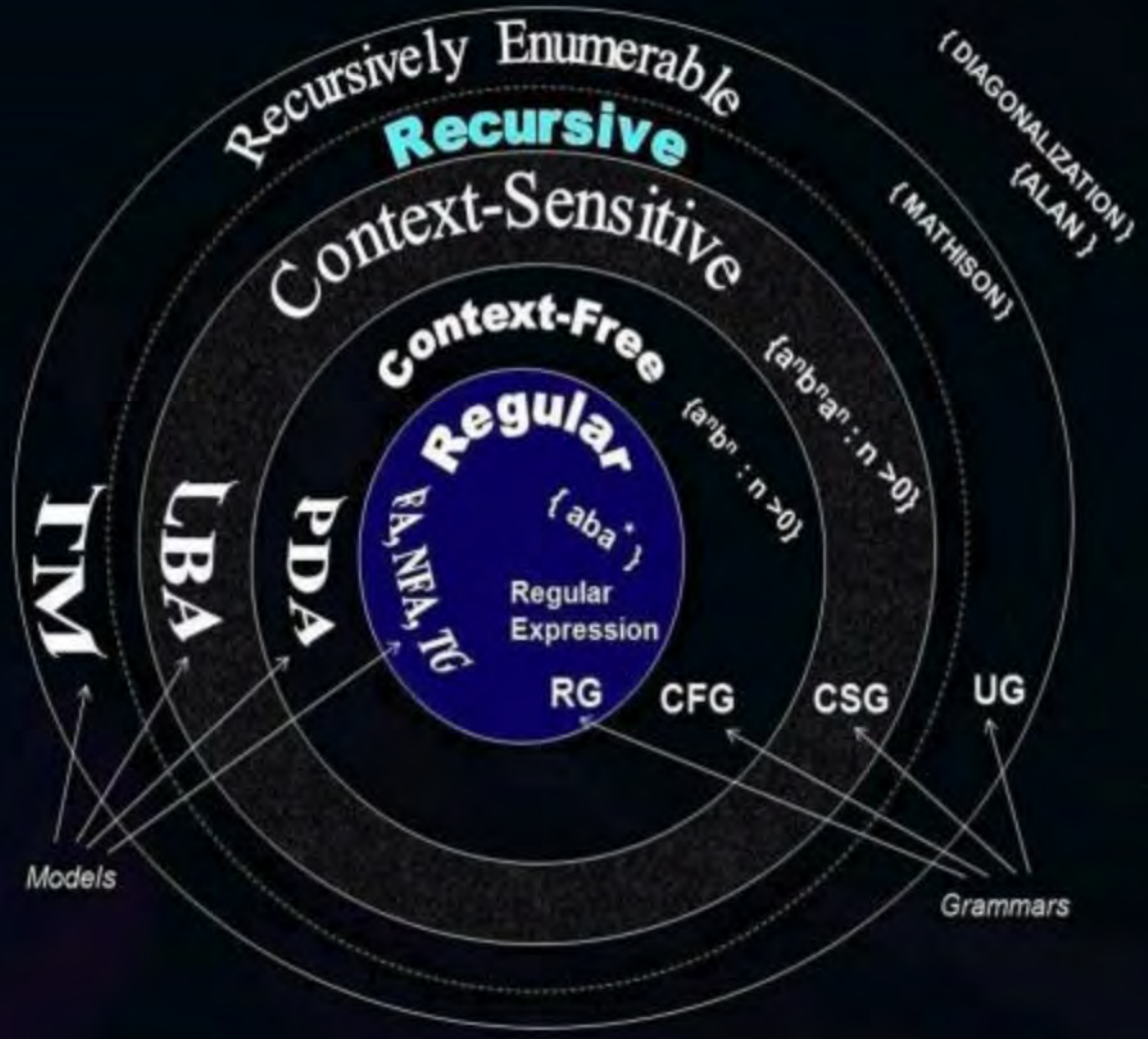
A $L_1 = \{0, 1\}^* - L$

B $L_1 = \{0, 1\}^*$

C $L_1 \subseteq L$

D $L_1 = L$

Topic : Theory of Computation



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.

MCQ

#Q. Let D_f, D_p are number of languages accepted by DFA and DPDA respectively.
Let N_f, N_P are number of languages accepted NFA and NPDA respectively.
Which of the following is true.

A. $N_f = D_f$
$N_p = D_p$

B. $N_f \supset D_f$
$N_p \supset D_p$

C. $N_f = D_f$
$N_p \subset D_p$

D. None

MCQ

#Q. In which of the cases stated below the following statement is false?
"Every nondeterministic machine M_1 there exists an equivalent deterministic machine M_2 recognizing the same language"

A. M_1 is non deterministic FA

B. M_1 is non deterministic turing machine

C. M_1 Is non deterministic PDA

D. None

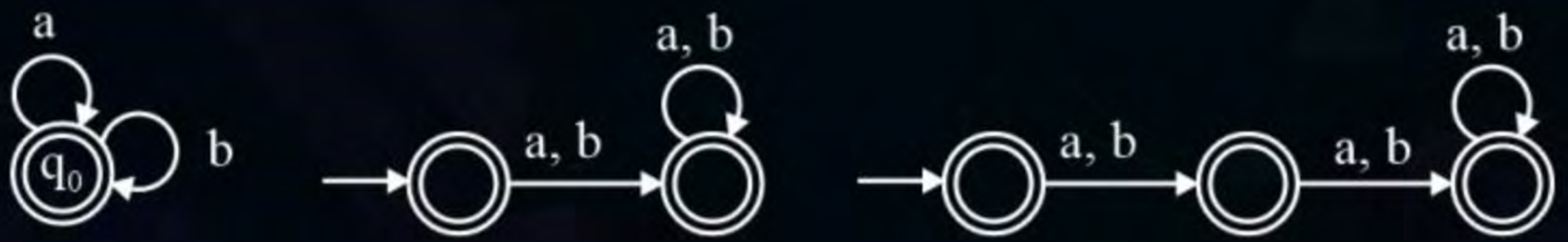
Topic : DFA Construction

Construct minimalstate DFA that accerpts all strings os 0's and 1's where each string ending with 00.

Topic : Minimization of DFA

→ For a given regular language even though many DFA exist but minimal state DFA is unique.

Ex: Complete Language: Σ^*



Topic : Minimization Algorithm

1. State equivalence algorithm
2. Table filling algorithm

Equivalent States:

Two states q_0, q_1 are said to be equivalent both $\delta(q_0, x)$ and $\delta(q_2, x)$, $\forall\, x \in \Sigma^*$ should result either final state or non final state.

$\delta(q_1, x)$

F NF

$\delta(q_2, x)$

Topic : Procedure of minimization

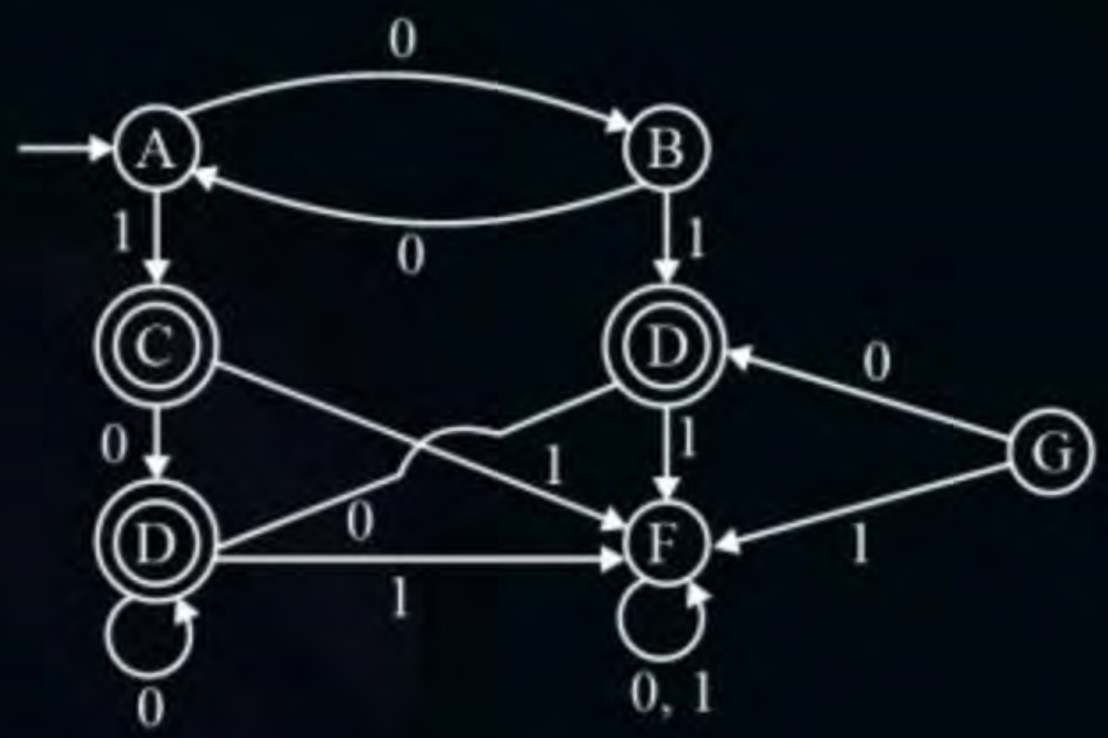
1. Elimination inaccessible states.

<u>**inaccessible state:**</u>

Any State which is not reachable from dead state is inaccessible state.

2. Apply algorithm steps
3. Merge single group into one state
4. Construct new minimized DFA

1. Reduce states of following DFA



Step-1: Elimination inaccessible state.

Note: Dead state is different from inaccessible state.

Topic : Procedure of minimization

Step:2

State	0	1
A	B	C
B	A	D
F	F	F
C	E	F
D	E	F
E	E	F

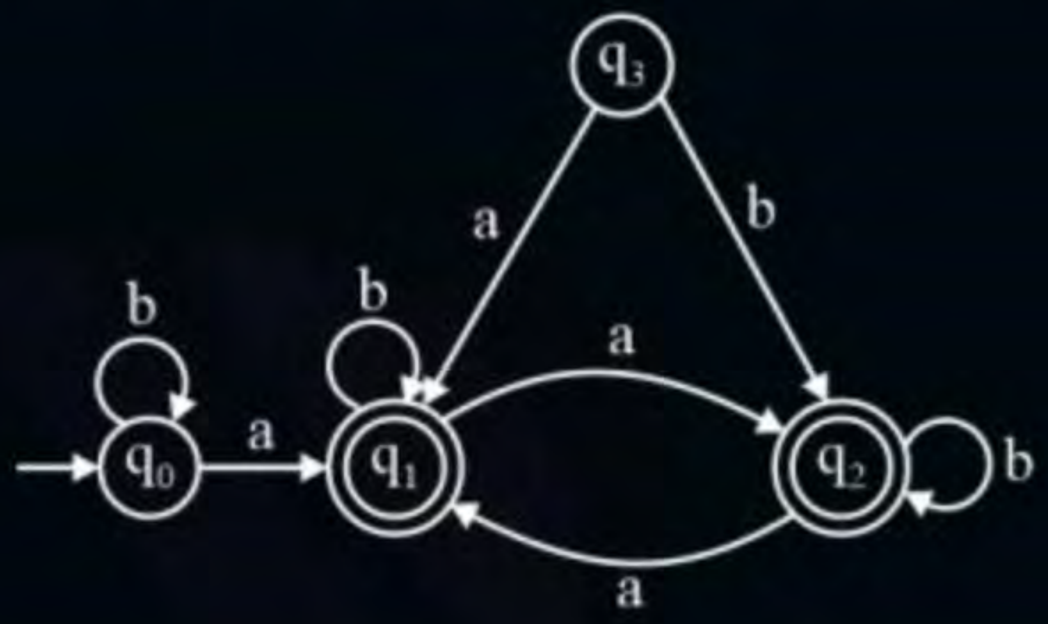
Algorithm:

1. {A, B, F} {C, D, E}
2.
3.

Minimized DFA

Topic : Procedure of minimization

Consider the following Finite State Automation



Topic : Procedure of minimization

Step 1: Eliminate q_3

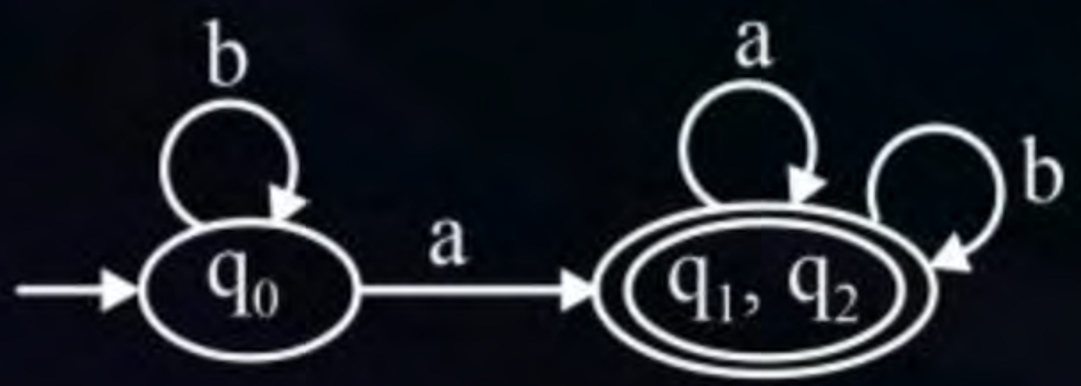
Step 2:

	a	b
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_1	q_2

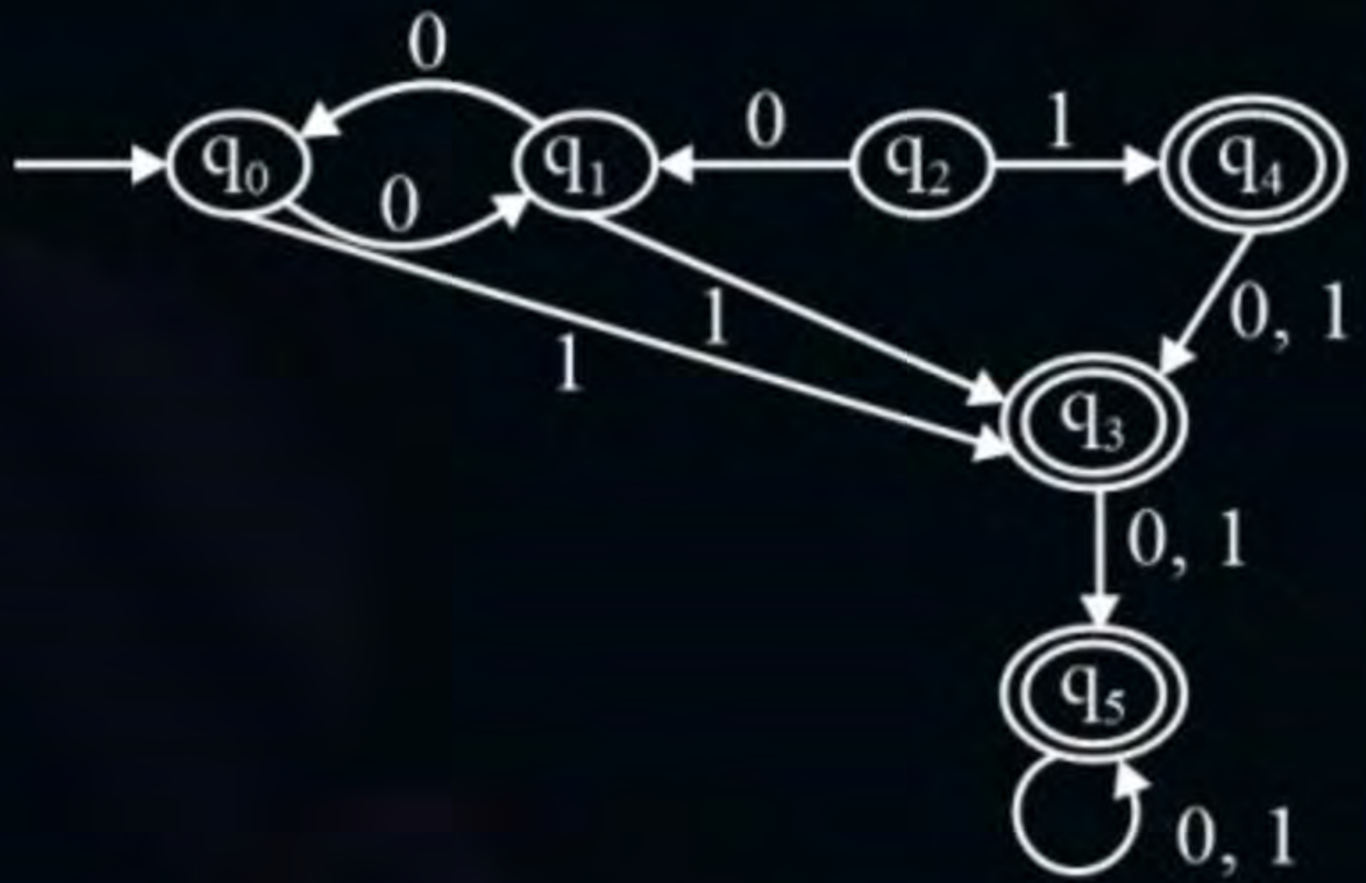
Algorithm step

1. $\{q_0\}\ \{q_1, q_2\}$
2. $\{q_0\}\ \{q_1, q_2\}$

Minimum DFA



Topic : Procedure of minimization



Minimize given DFA

Topic : Procedure

Step 1: Eliminate

Step 2:

	a	b
q_1	q_1	q_3
q_2	q_0	q_3
q_3	q_5	q_5
q_5	q_5	q_5

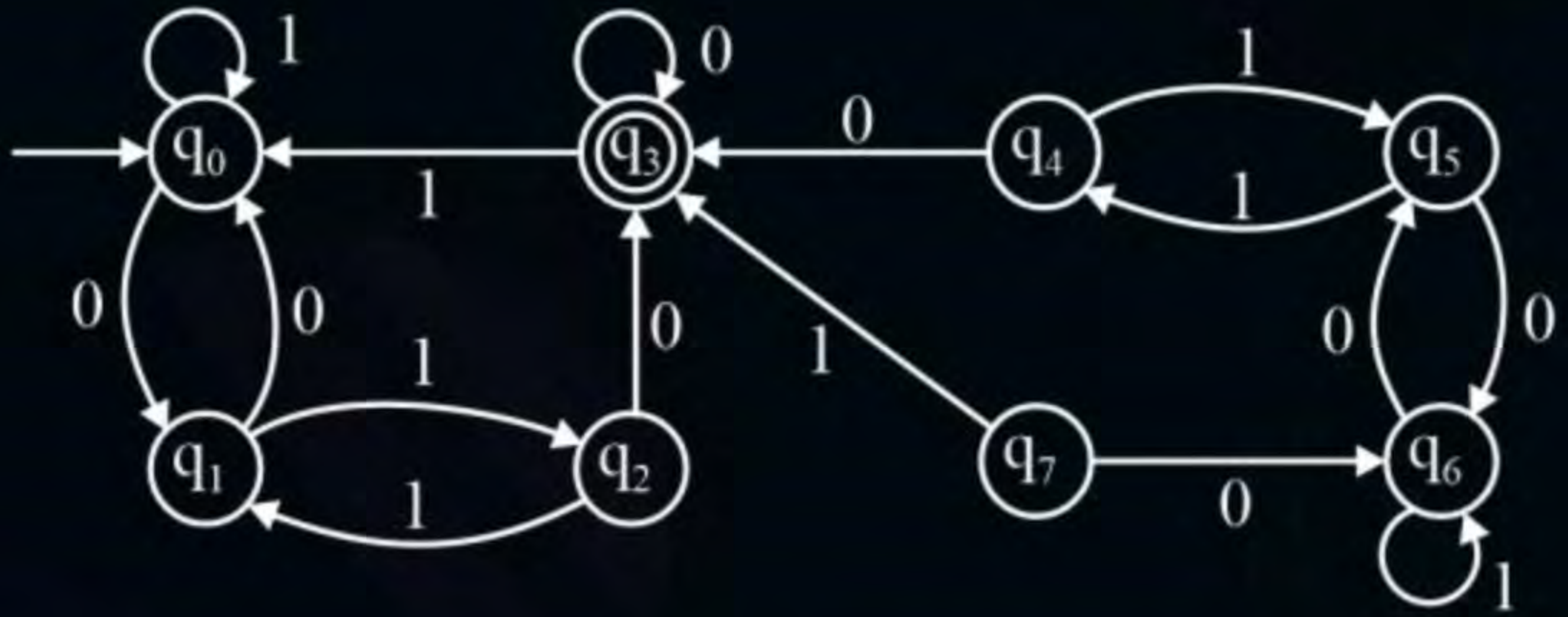
Algorithm

1. $\{q_0, q_1\}$ $\{q_3, q_5\}$

2.

Minimum DFA

Topic : Procedure of minimization



How many inaccessible states present in given DFA

[MSQ]

#Q. Consider the 5-state DFA M accepting the language L(M) ⊂ subset $(0 + 1)^*$ shown below. For any string $w \in (0 + 1)^*$ let $n_0(w)$ be the number of 0's in w and $n_1(w)$ be the number of 1's in w.

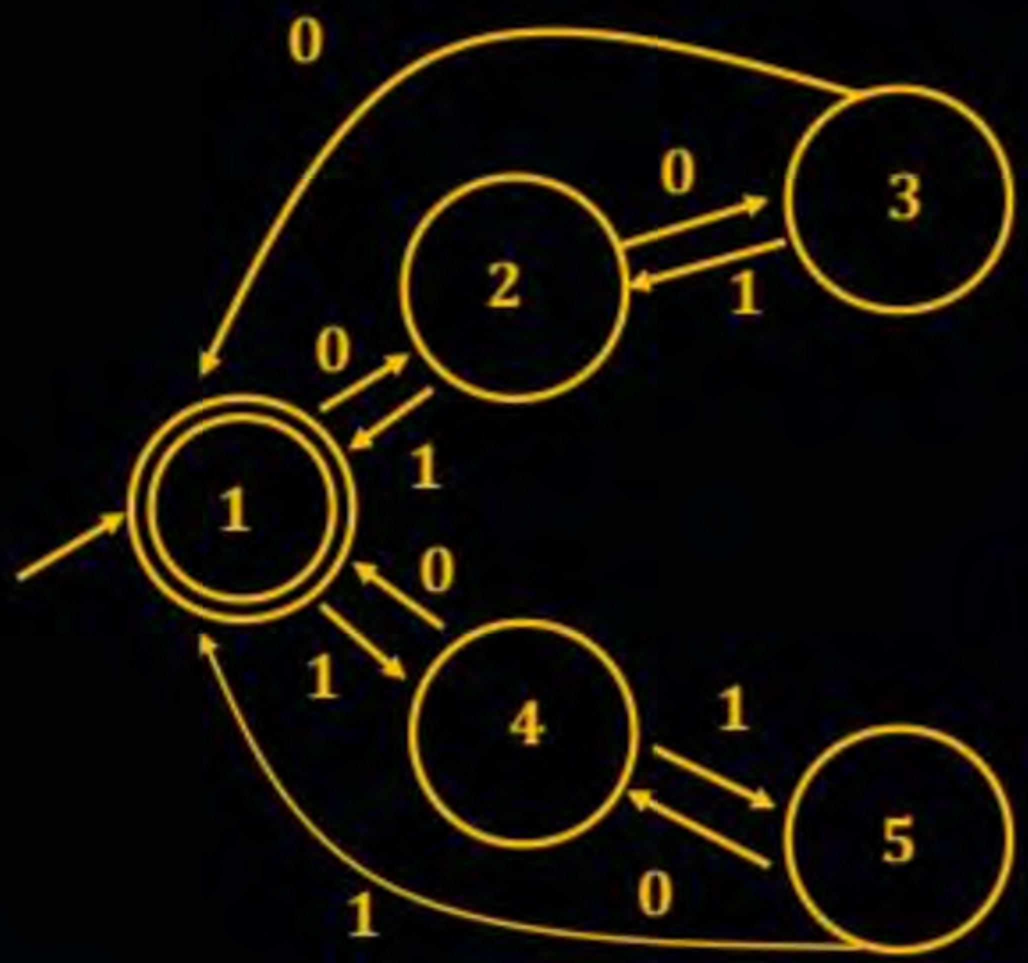
Which of the following statements is/are FALSE? **[GATE-CS-shift-I-24: 2M]**

(A) States 2 and 4 are distinguishable in M

(B) States 2 and 5 are distinguishable in M

(C) Any string w with $n_0(w) = n_1(w)$ is in L(M)

(D) States 3 and 4 are distinguishable in M



Topic : DFA

#Q. Construct the minimal DFA that accept all binary no divisible by

Topic : DFA

Construct the minimal DFA that accept all strings of a's and b's where

1. Each string ending with b.
2. Each string start with a and end with b.
3. Each string starting and ending with different symbol.
4. Each string starting and ending with same symbol.

Topic : DFA

Construct the minimal DFA that accept all string a's and b's where

1. Length of string exactly 4.
2. Number of a's length of string atleast 4.
3. Length of string atmost 4.
4. Length of string divisible by 4.
5. Number of a's exactly 5.
6. Number of b's exactly 2.
7. Number of a's divisible by 3.
8. Number of b's not divisible by 4.
9. Length of the string even.

Topic : DFA

#Q. Length of string exactly 4.

Topic : DFA

NOTE:

→ Minimal DFA that accept exactly N length string requires (N + 2) states includes dead state.

→ Minimal DFA that accept atleast N length string requires (N + 1) states.

→ Minimal DFA that accept atmost N length string requires (N + 2) states includes dead states.

→ The minimal DFA that accept length of the string divisible by N then requires N states.

Topic : DFA

Construct a minimal DFA that accept all string a's and b's. where number of a's divisible by 2 and number of b's divisible by 3.

Topic : DFA

How many number of state are there with minimum DFA for the following state.

a) Number of a’s divisible by 2 and number of b’s not divisible by 3.

b) Number of a’s divisible by 2 and number of b’s atleast 3.

c) Number of a’s atleast 2 and number of b’s atleast 3.

d) Number of a’s exactly 2 and number b’s atleast 2.

e) Number of b’s atmost 3 and number b’s exactly 3.

f) Number of a’s not divisible by 2 or number of b’s exactly 3.

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.

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

(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.

(iv) The given condition on length one number not divides other and GCD of given condition is not equal to 1 then number of states of 'and' automata, number of states of 'OR' automata is LCM of given condition.

Find the number of stage of minimal DFA for the following matrix.

(Length of the string divisible by 3 or divisible by 6)

GCD (3, 6) = 3

Length of the string di is by 4 and di by 6

(v) LCM (4, 6) = 12

(vi) Number of a's divisible by 4 AND number of b's divisible by 6.

$4 \times 6 = 24$

Topic : DFA

Find the number of stage of minimal DFA for the following matrix.

Length of the string divisible by 3 or divisible by 6

GCD (3, 6) = 3

Length of the string di is by 4 and di by 6

LCM (4, 6) = 12

Number of a's divisible by 4 AND number of b's divisible by 6.

$4 \times 6 = 24$

Topic : DFA

1. Length of the string divisible by 2 and divisible by 1
2. Length of string divisible by 2 OR divisible by 4.
3. Length of string divisible by 3 divisible by 4
4. Length of string divisible by 3 OR divisibly by 4
5. Length of string divisible by 6 OR divisibly by 8
6. Number of a's divisible by 6 and number of divisible by 8.

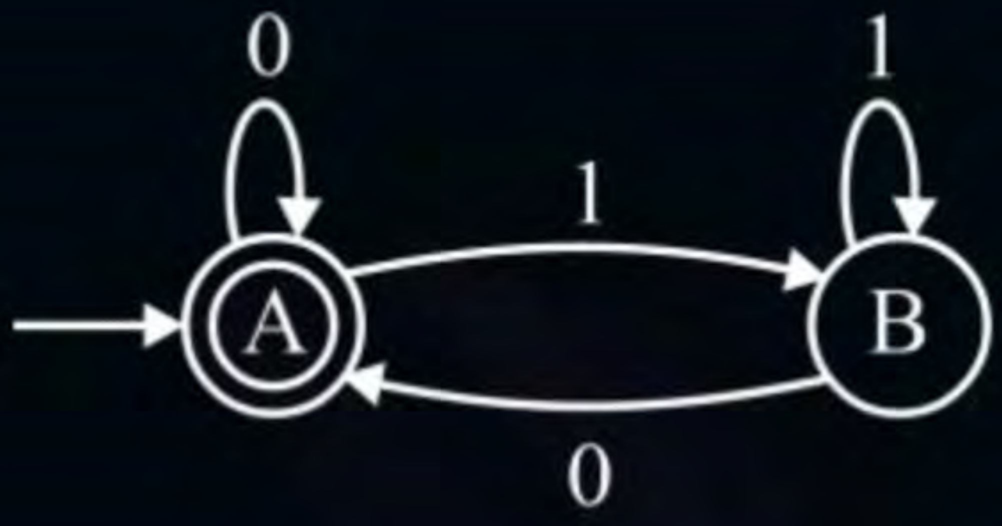
Topic : Decision Properties of Finite Automata

- Equivalence Problem

- Finiteness Problem

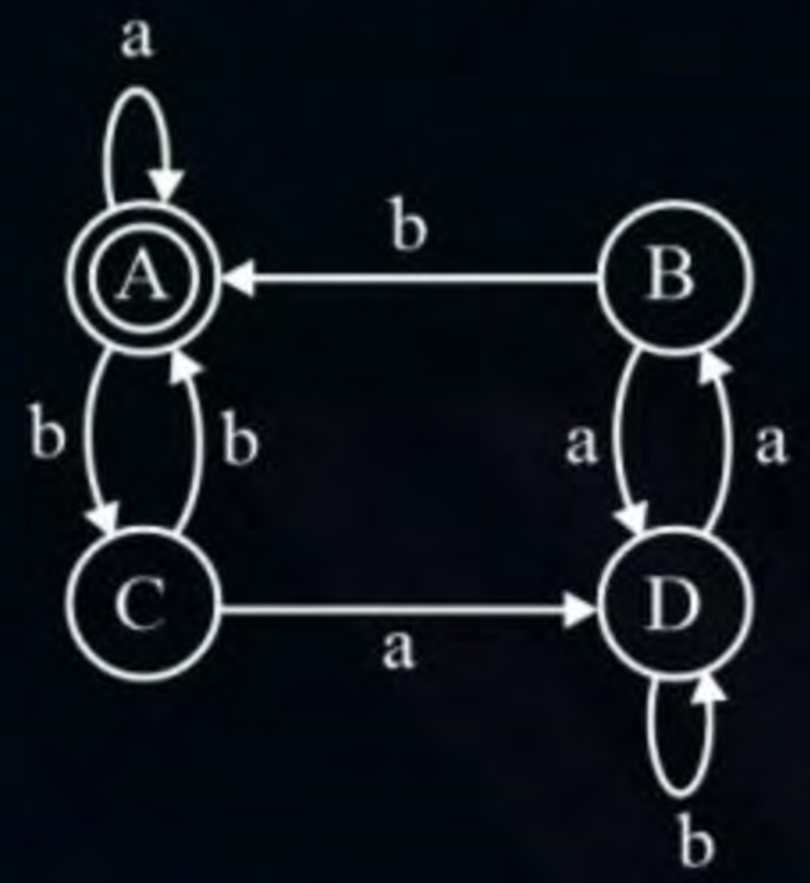
- Emptiness Problem

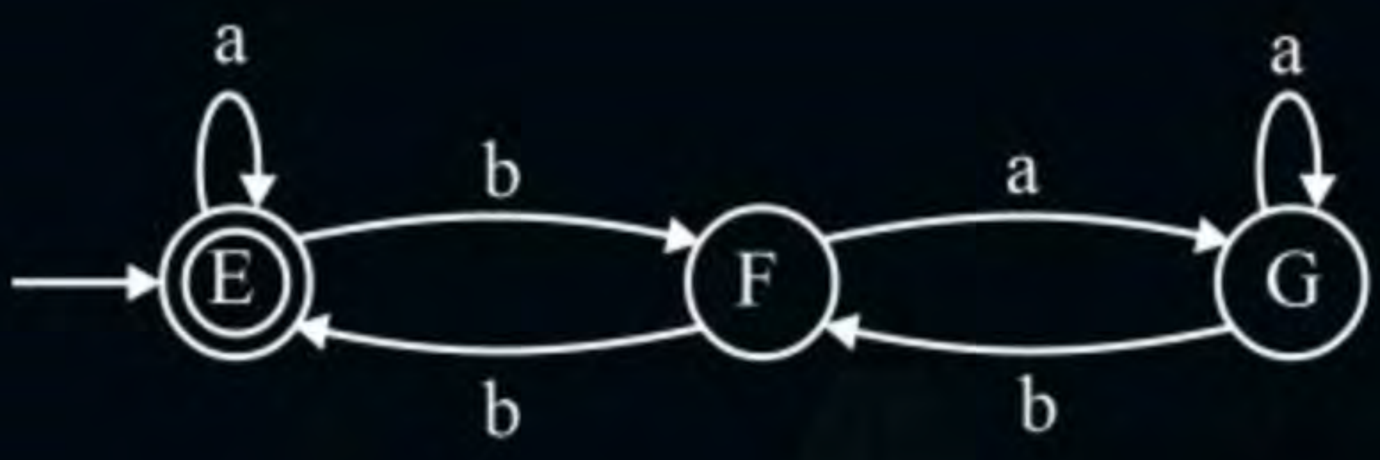
#Q. Verify following two automata accepts same language or not



	0	1

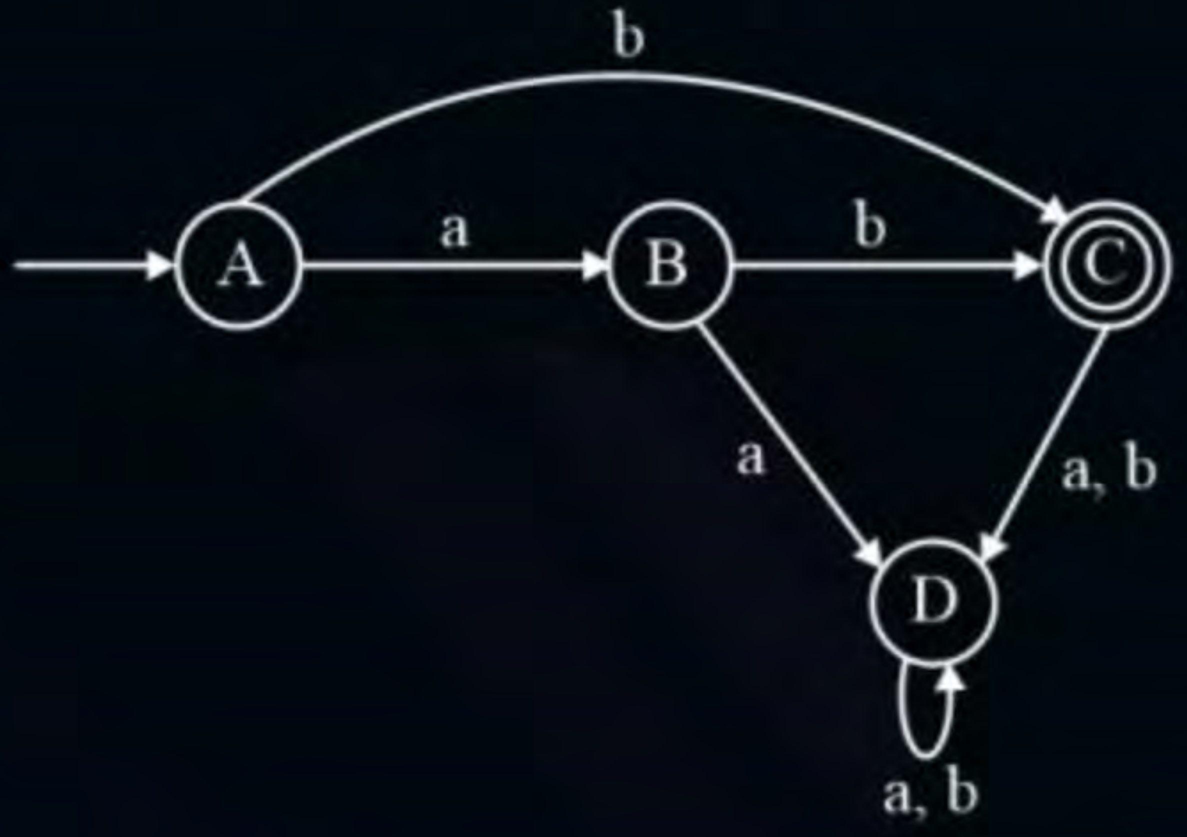
#Q. Verify following two automata accepts same language or not





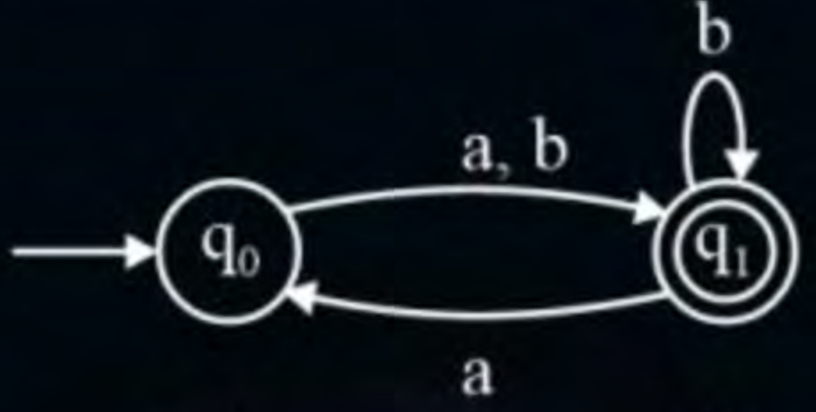
	a	b
(A, E)		

#Q. Which of the following automata accepts infinite Language

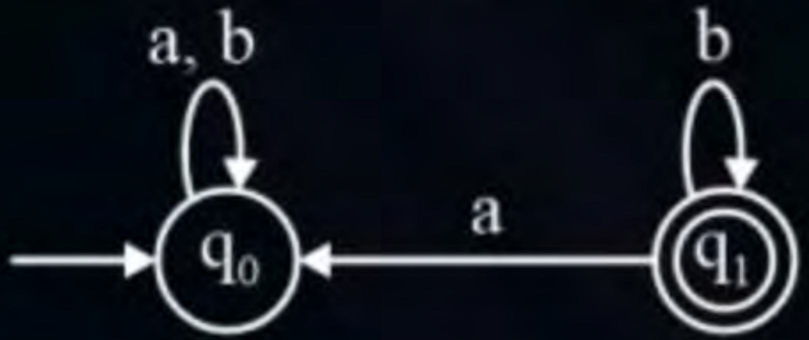


#Q. Which of the following automata accepts Empty Language.

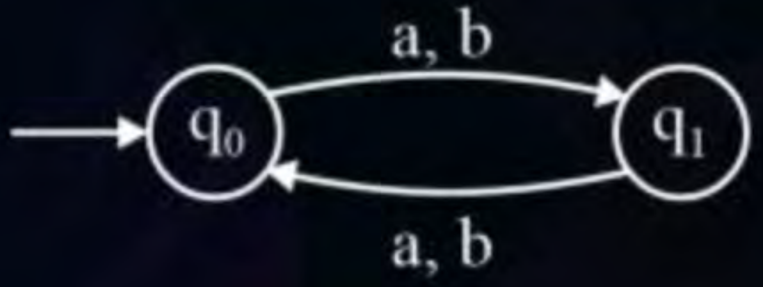
1.



2.



3.



Topic : Non-Deterministic Finite Automaton

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.

DFA

$Q \times \Sigma \to 1$

NFA ✓

NFA ✓

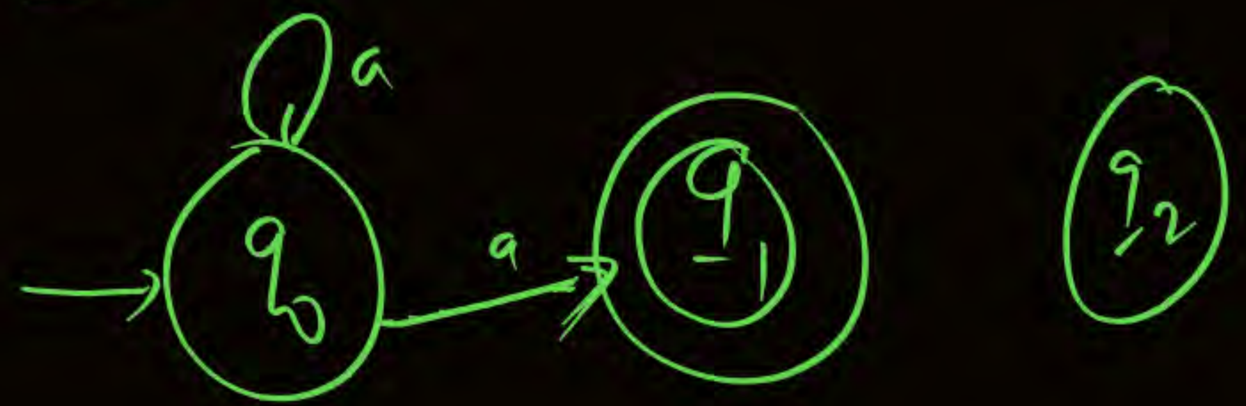
DFA → NFA

NFA

$Q \times \Sigma \to 0, 1, 2, 3 \ldots 2^Q$ (or) power set(Q)

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

$(q_0, b) = 8$



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

$\{q_0, q_1\} = \{\ \}\ \{q_0\}\ \{q_1\}\ \{q_0, q_1\} =$ power set

Every DFA is NFA

But every NFA need not be a DFA.

NFA is formally define as:- $\boxed{\{Q, \Sigma, q_0, F, \delta\}}$

Q	-	Finite number of states (set of state)
Σ	-	Input alphabet
q_0	-	initial state → only one
F	-	Set of final states → any no. of final states.
δ	-	transition function

$$\boxed{Q \times \Sigma \rightarrow 2^Q}$$

DFA $\rightarrow Q \times \Sigma \rightarrow Q$.

NFA $\rightarrow Q \times \Sigma \rightarrow 2^{Q}$.

Topic : Non-Deterministic Finite Automaton

L $\Sigma^* - L$

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

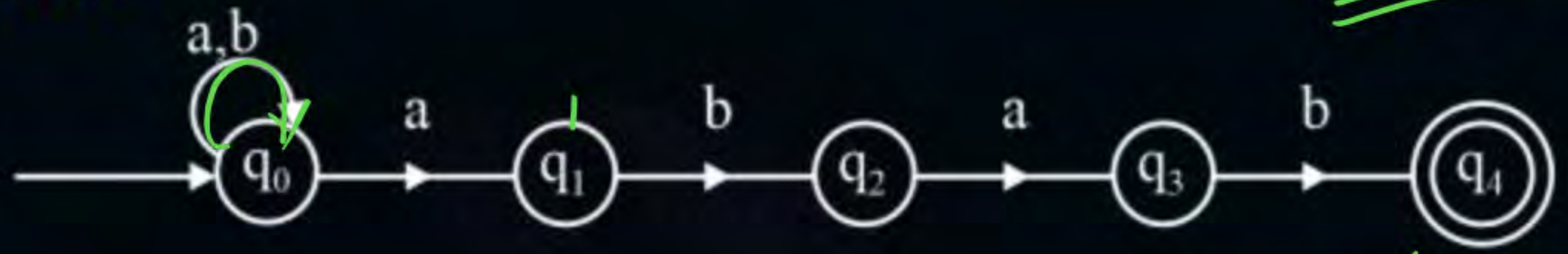
Deterministic → exact nature

Non deterministic → not " "

Topic : NFA

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

NFA:



NFA

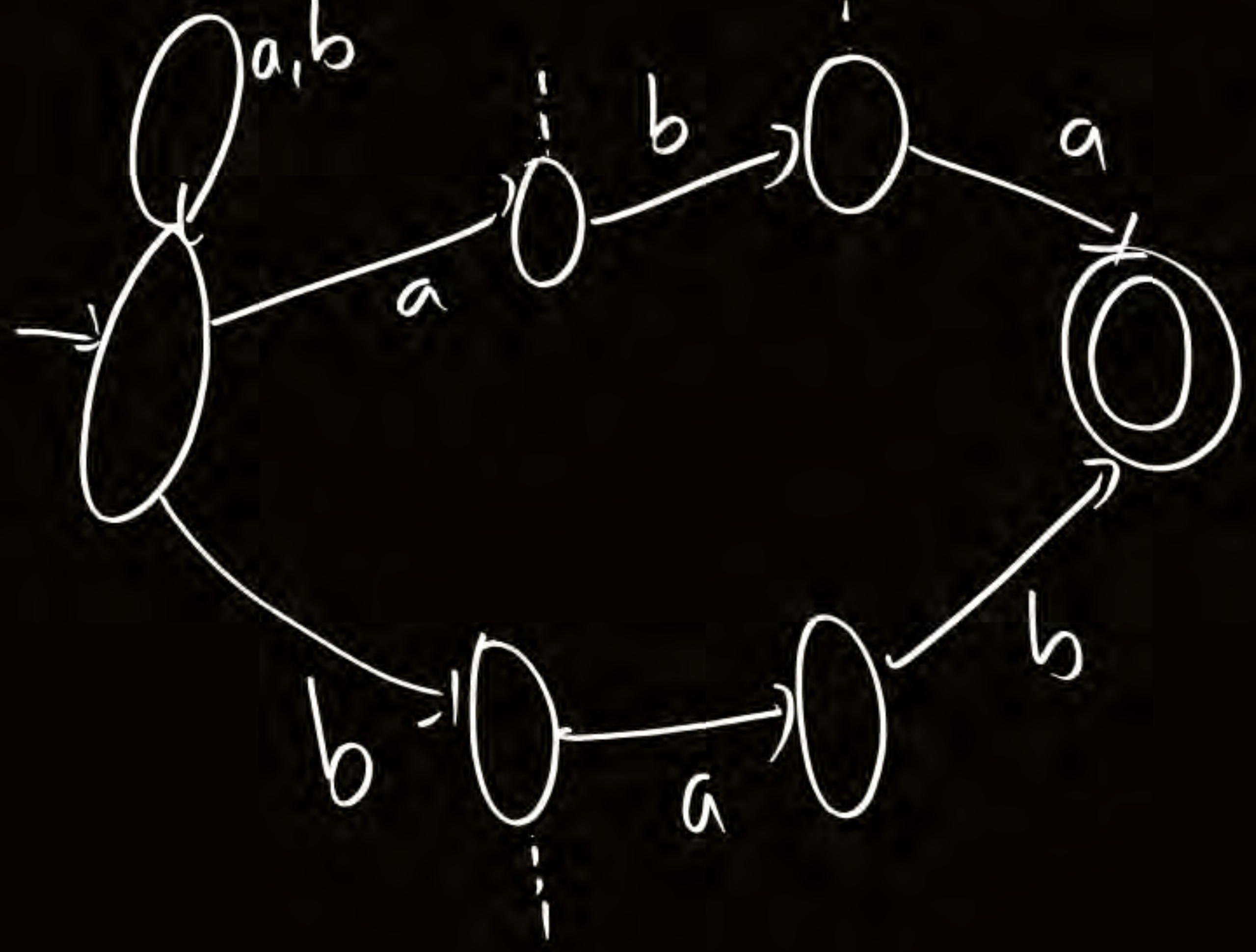
$\begin{pmatrix} a \cdot S \\ b \cdot S \end{pmatrix}$ abab

a·S
b·S aba (or) bab

every string is ending with

aba (or) bab

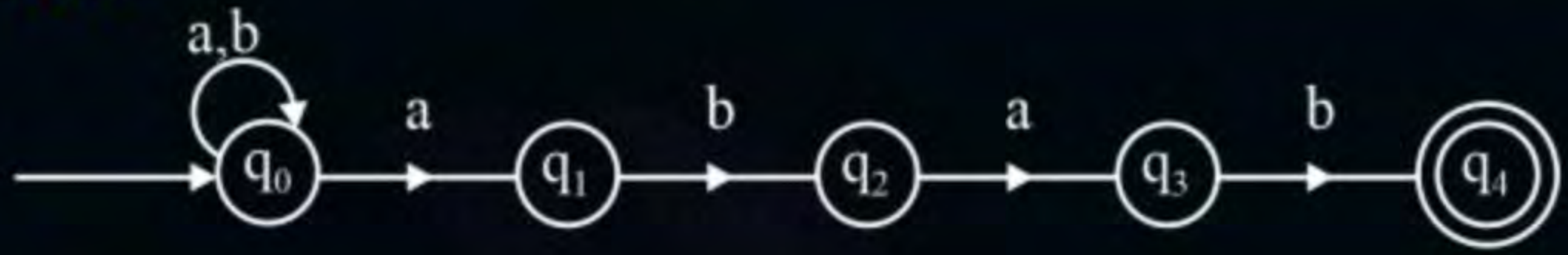
NFA?



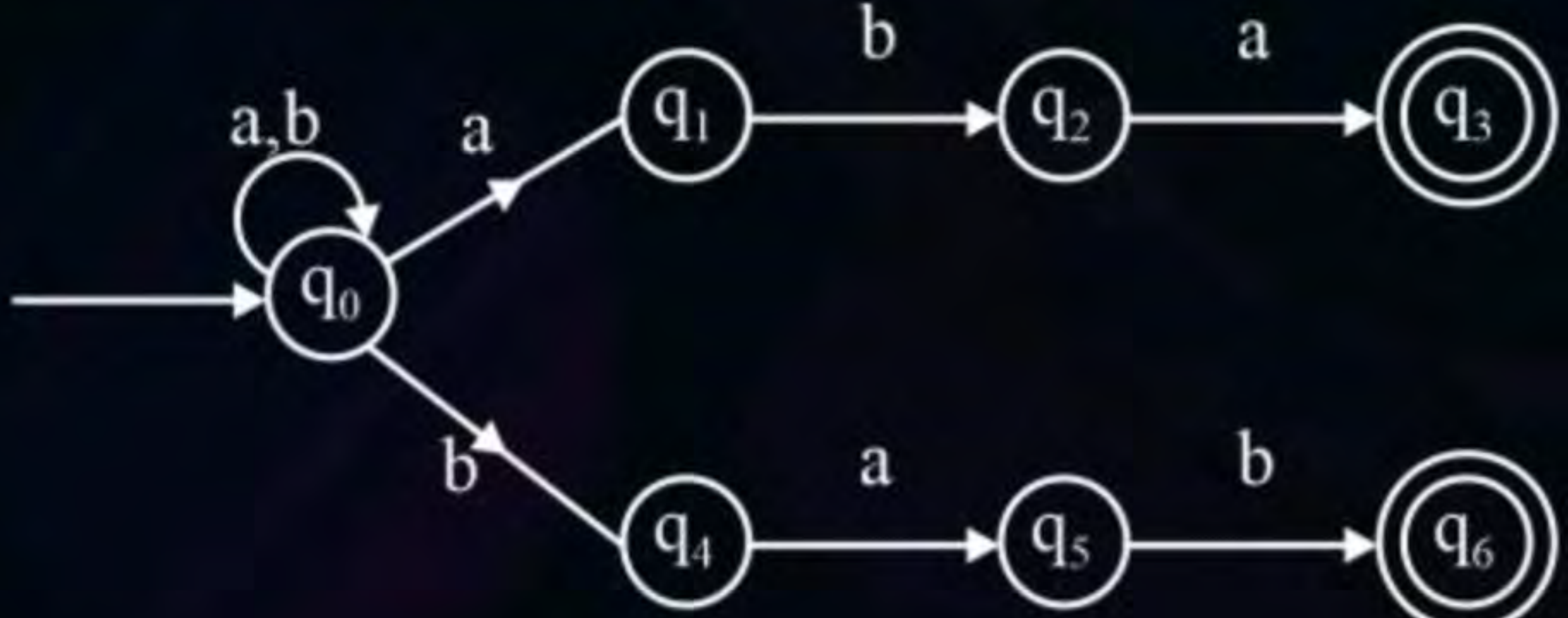
Topic : NFA

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

NFA:

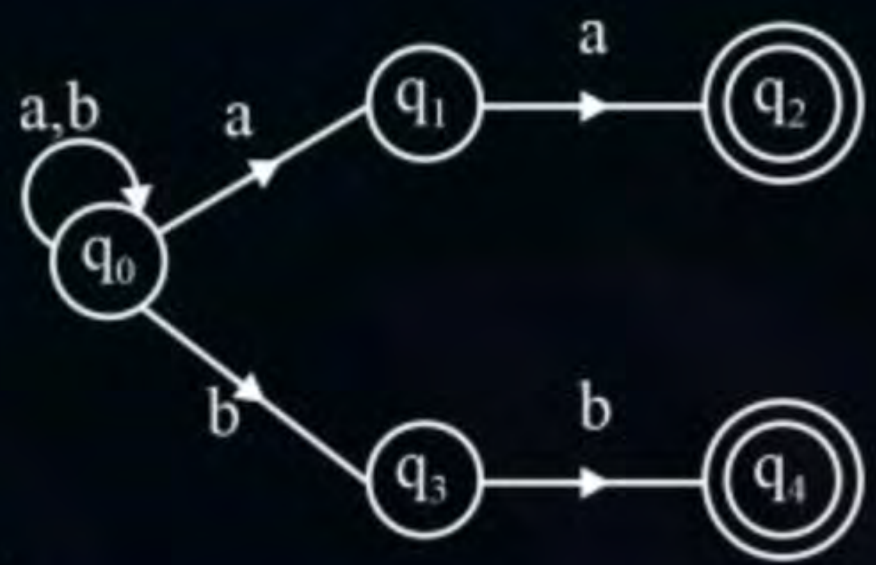


Each string ending with aba or bab

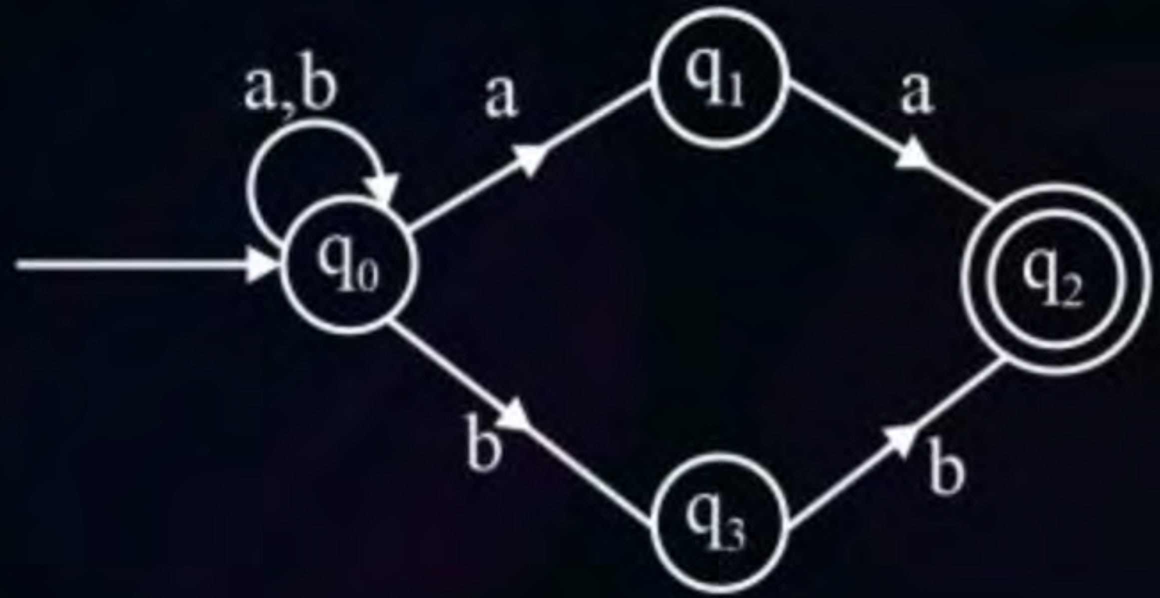


Topic : NFA

Last two symbol are same over alphabet {a, b}

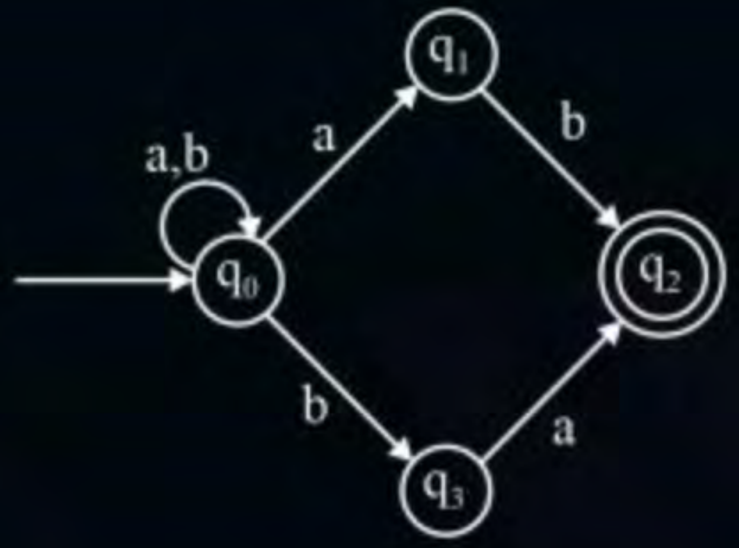


or



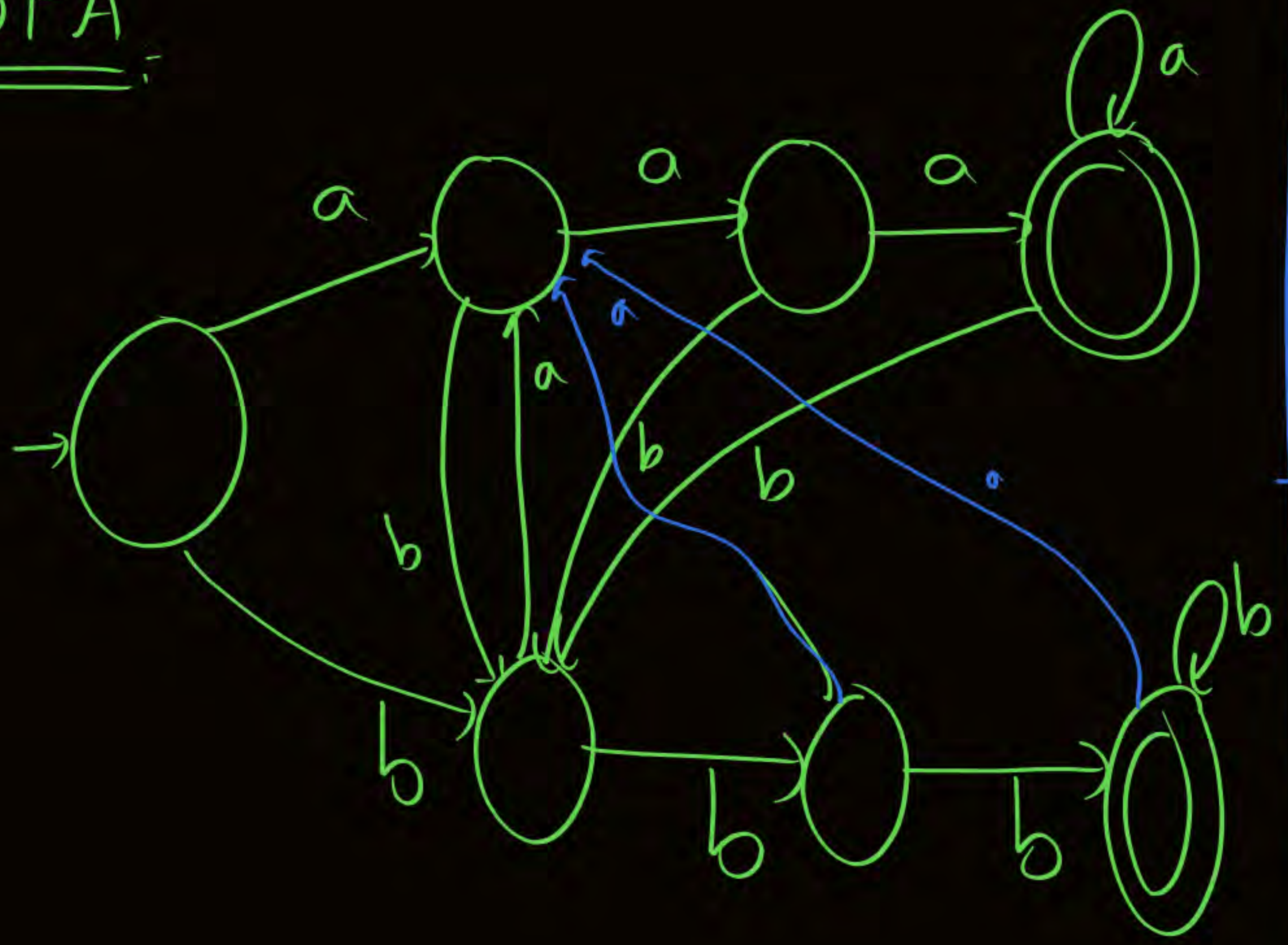
Topic : NFA

Last two symbols are different over alphabet {a, b}

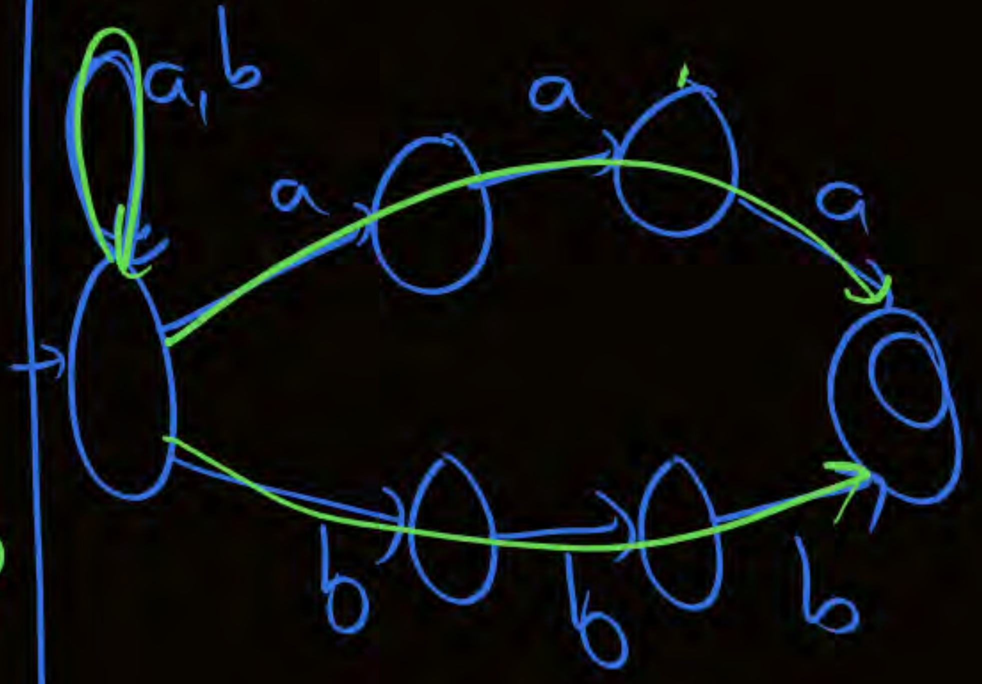


$\Sigma = \{a, b\}$ Last 3 Symbols are Same. $\{a, b\}^* (aaa)$ ✓ (bbb) ✓

DFA :

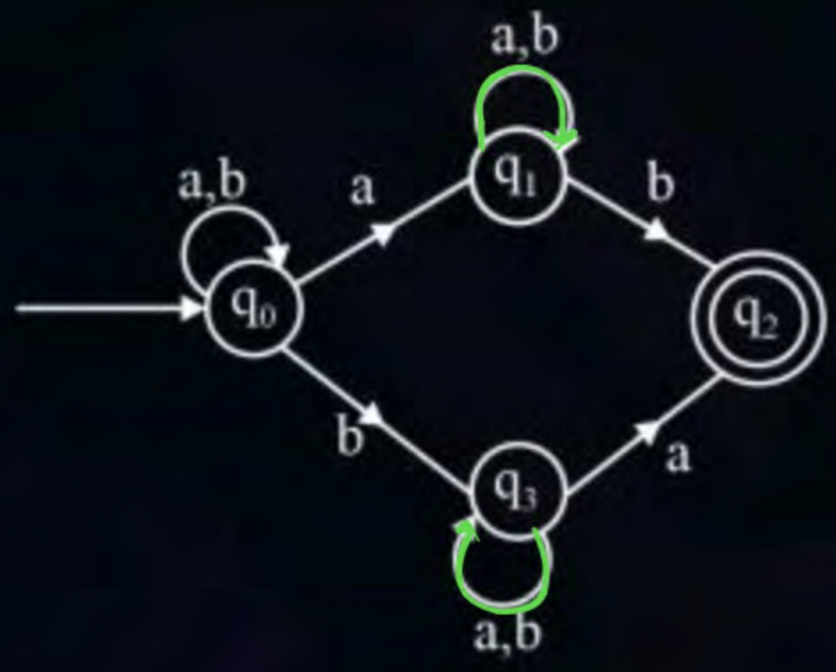


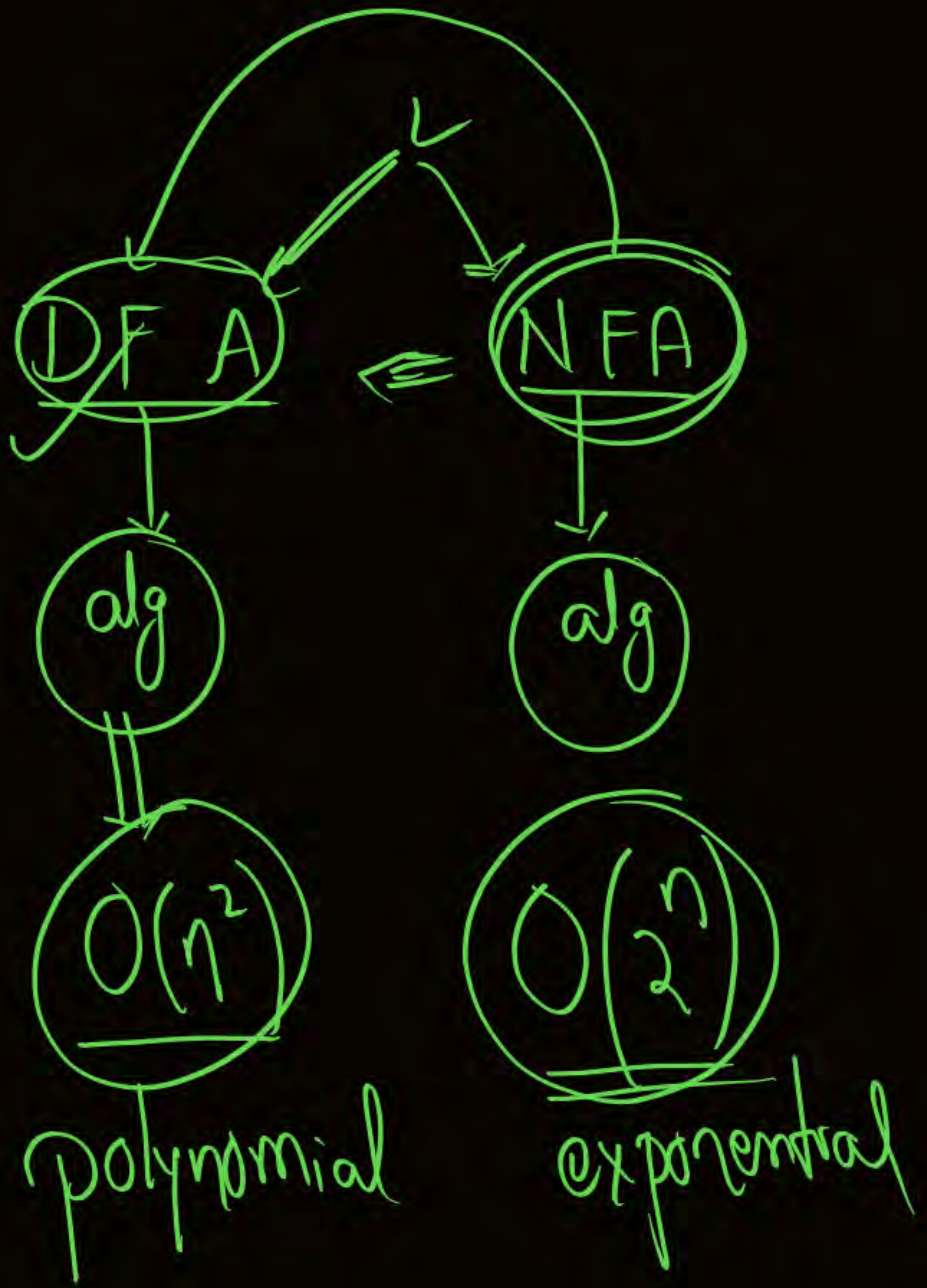
NFA



Topic : NFA

Starting & ending with different symbol.




DFA
NFA
alg
alg
O(n^2)
O(2^n)
polynomial
exponential

NFA

length of the string

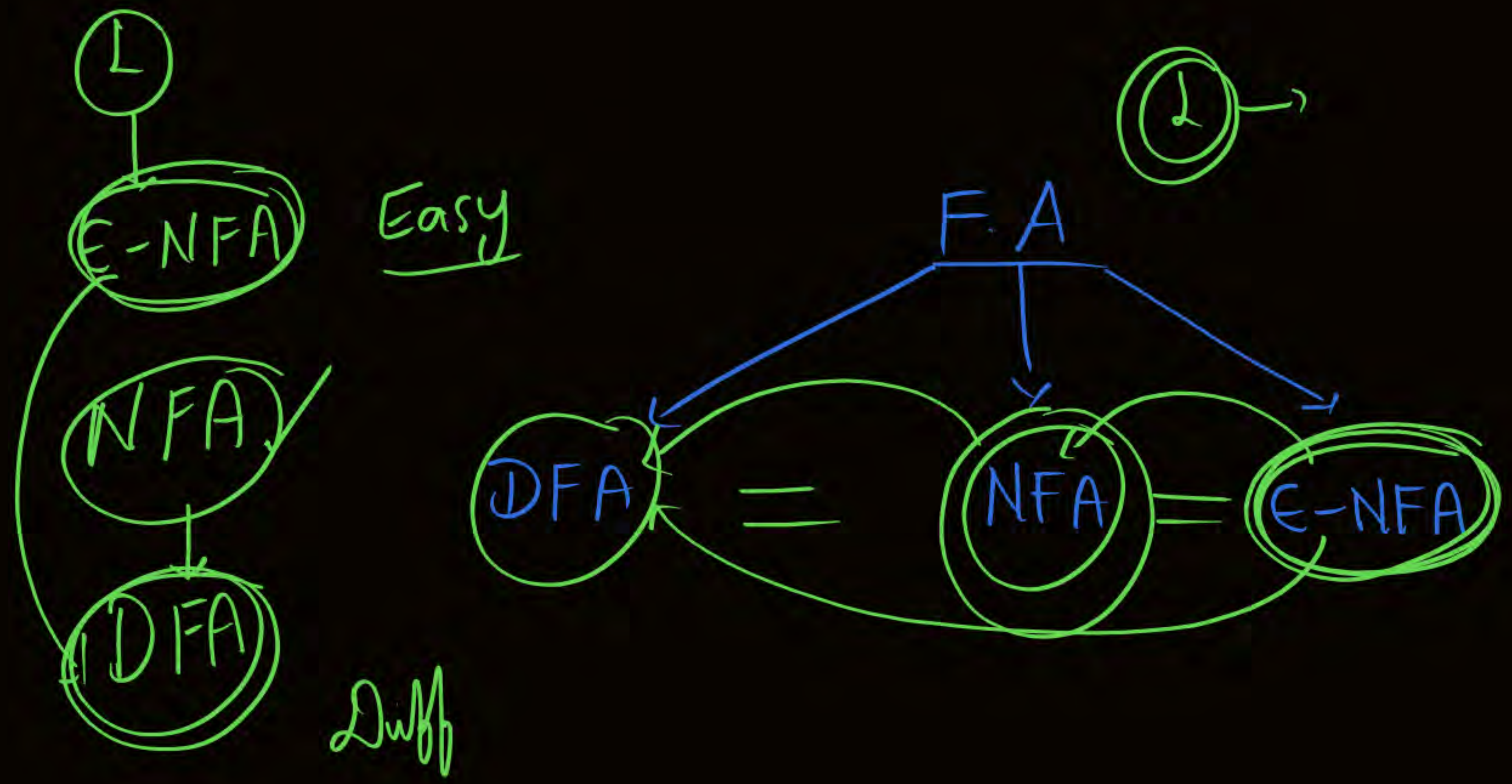
Exactly $n \rightarrow n+1$ states

atleast $n \rightarrow n+1$

at most $n \rightarrow n+1$

divisible by $n \rightarrow n$

Expressive power : No. of Languages accepted by Automata is known as expressive power.

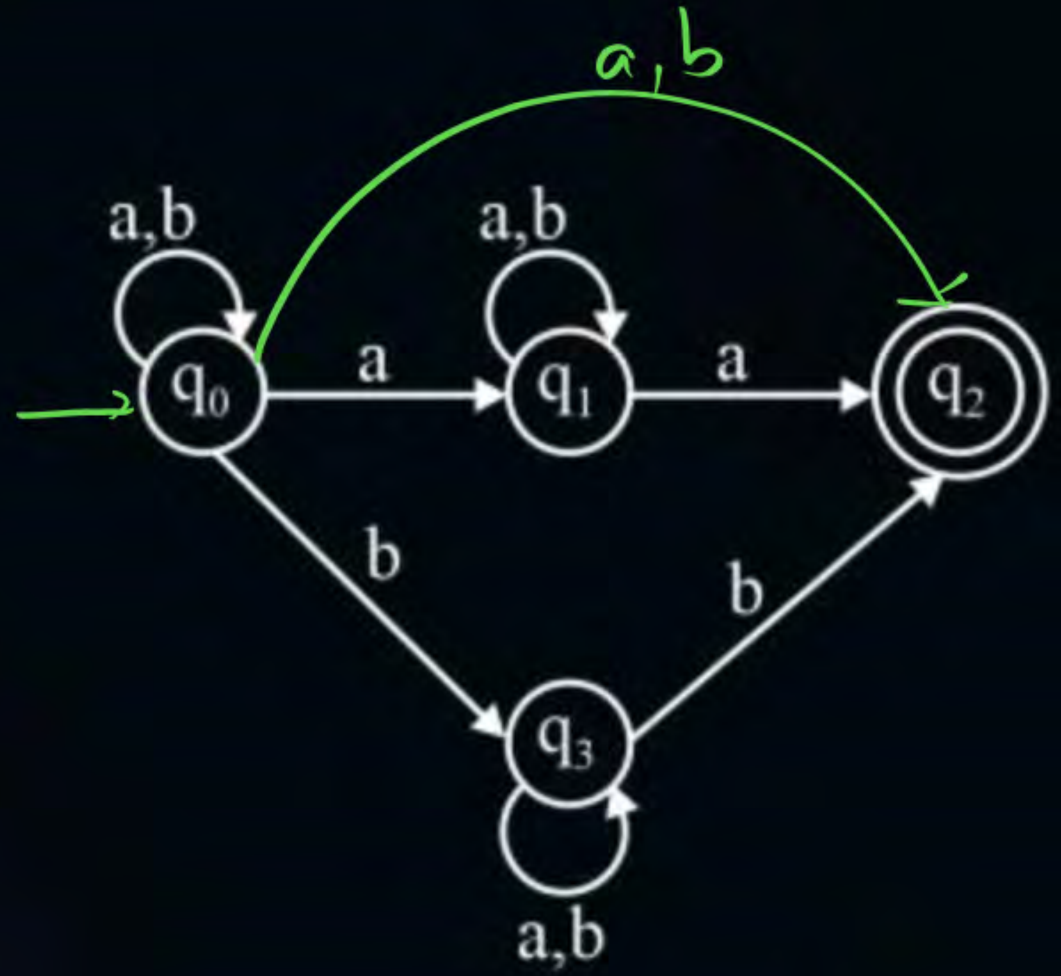


L
ε-NFA
Easy
NFA
DFA
Diff
L
F.A
DFA
=
NFA
=
ε-NFA

$\begin{pmatrix} a \\ b \end{pmatrix}$ $\underline{a}$ $\underline{a/b}$ $\underline{a/b}$

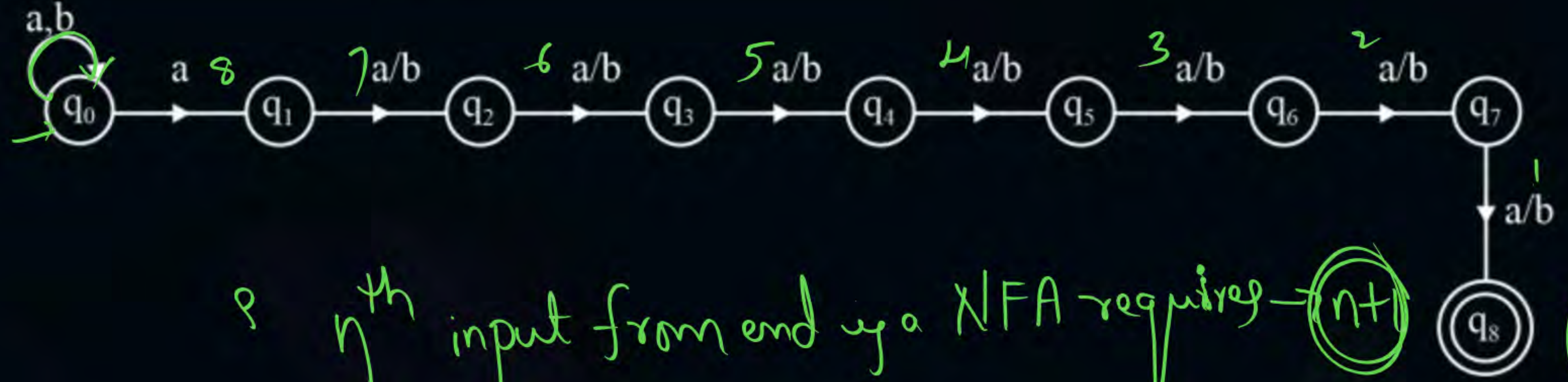
Topic : NFA

Starting and ending with same symbol



Topic : NFA

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



n^{th} input from end of a NFA requires → n+1

9

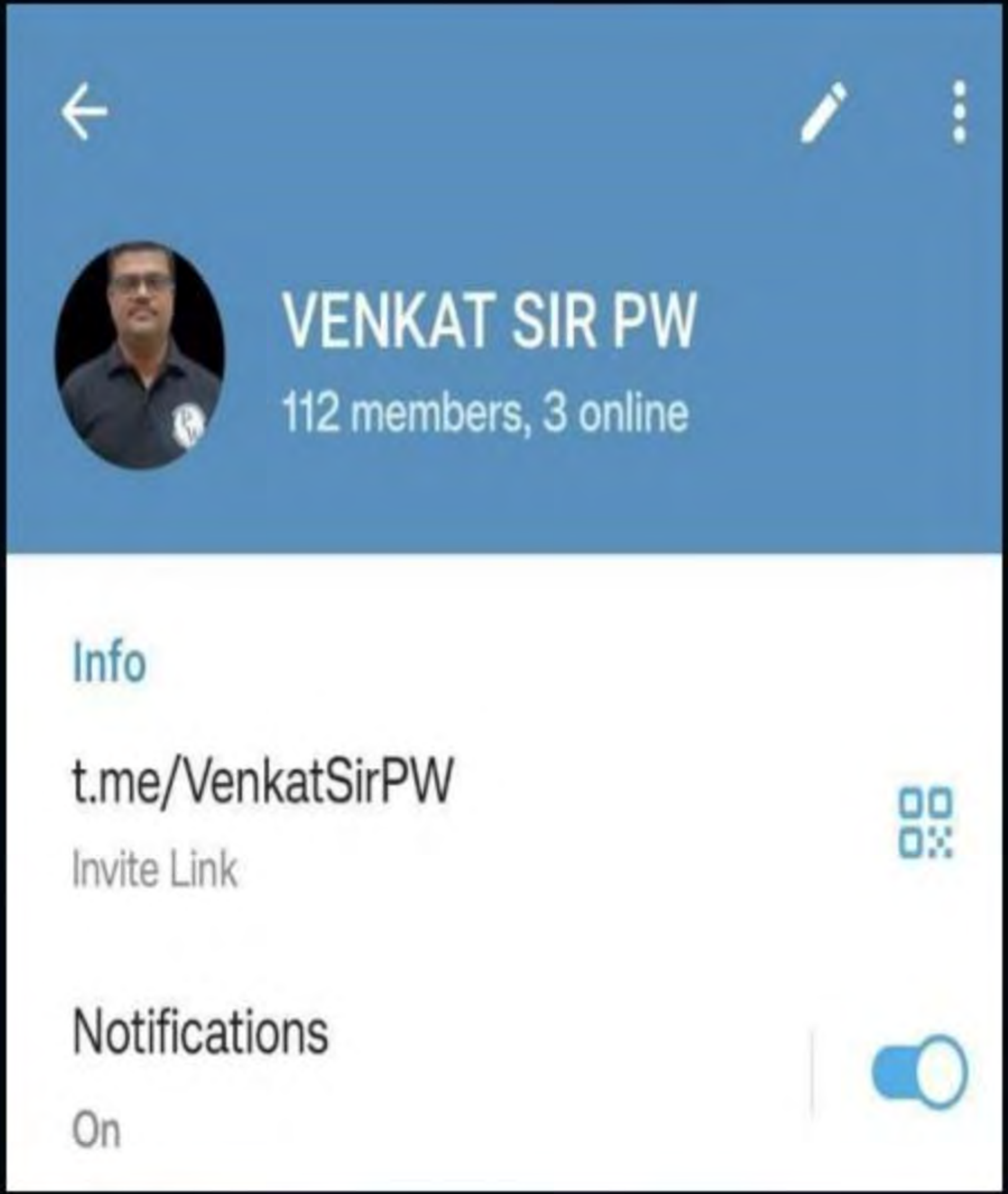
DFA → $2^8 = 256$

8th input symbol is a while reading from right side

5^{th} input from end is a $\longrightarrow$ DFA $\to 2^5 \to 32 = P$

NFA $\to 6 \quad = Q$

$P + Q = \underline{\underline{38}}$


VENKAT SIR PW
112 members, 3 online
Info
t.me/VenkatSirPW
Invite Link
Notifications
On


@VENKATSIRPW

THANK - YOU