CS & IT ENGINEERING

THEORY OF COMPUTATION

[redacted]

Regular Language

Lecture No.- 01

By- Venkat sir

Recap of Previous Lecture

- **Topic** Properties of Regular Expression
- **Topic** Construction of Regular Expression
- **Topic** DFA States

F.A → Regular Expressions → state elimination
→ Arden's method

{ Regular Expression → F.A } ε-NFA, NFA, DFA

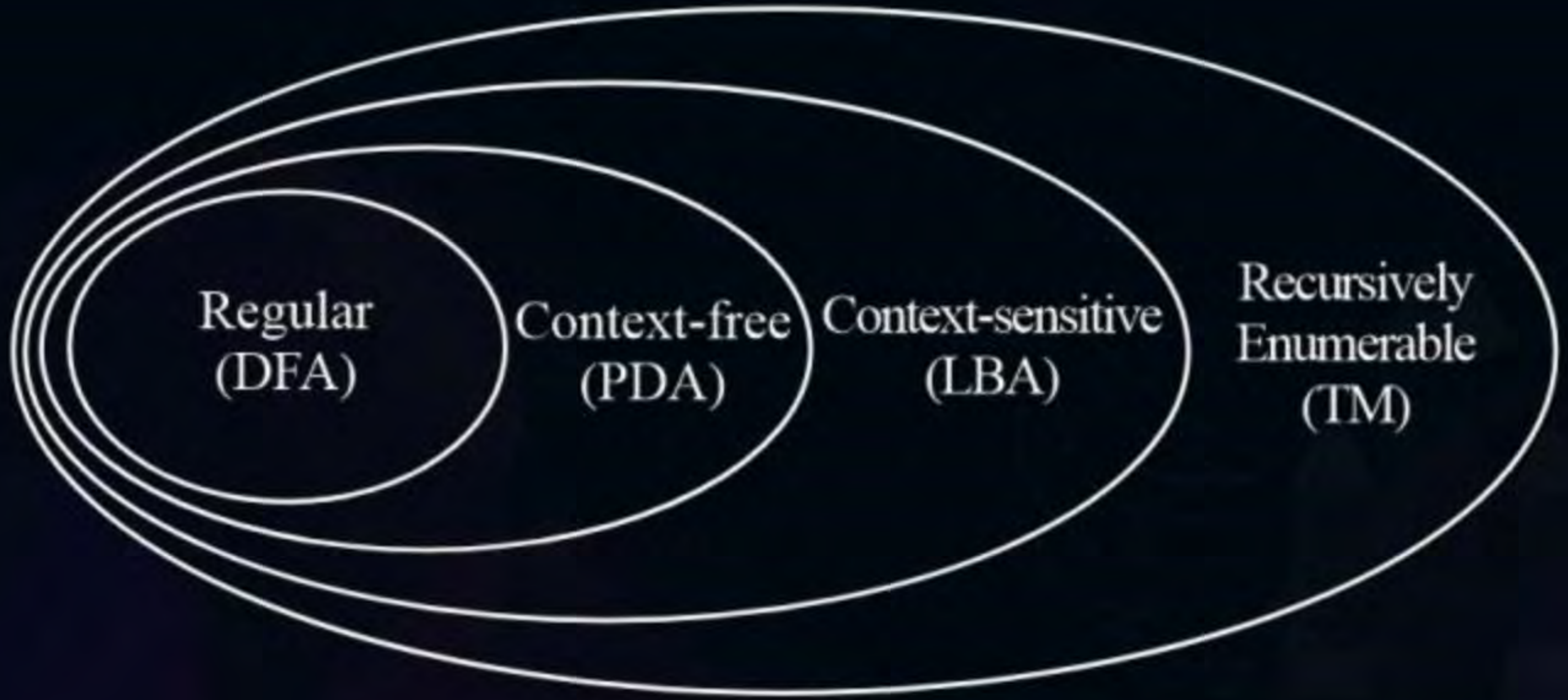
Topics to be Covered

Topic Conversion from $\in$ NFA to NFA

Topic ??

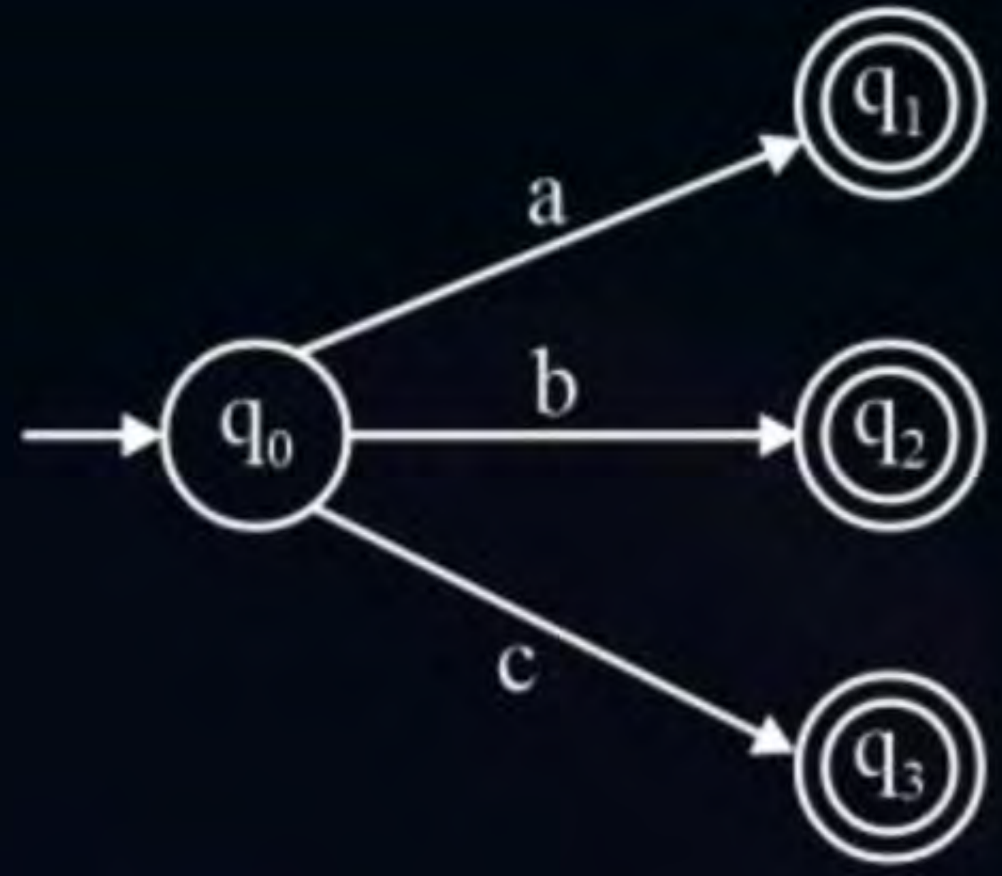
Topic ??

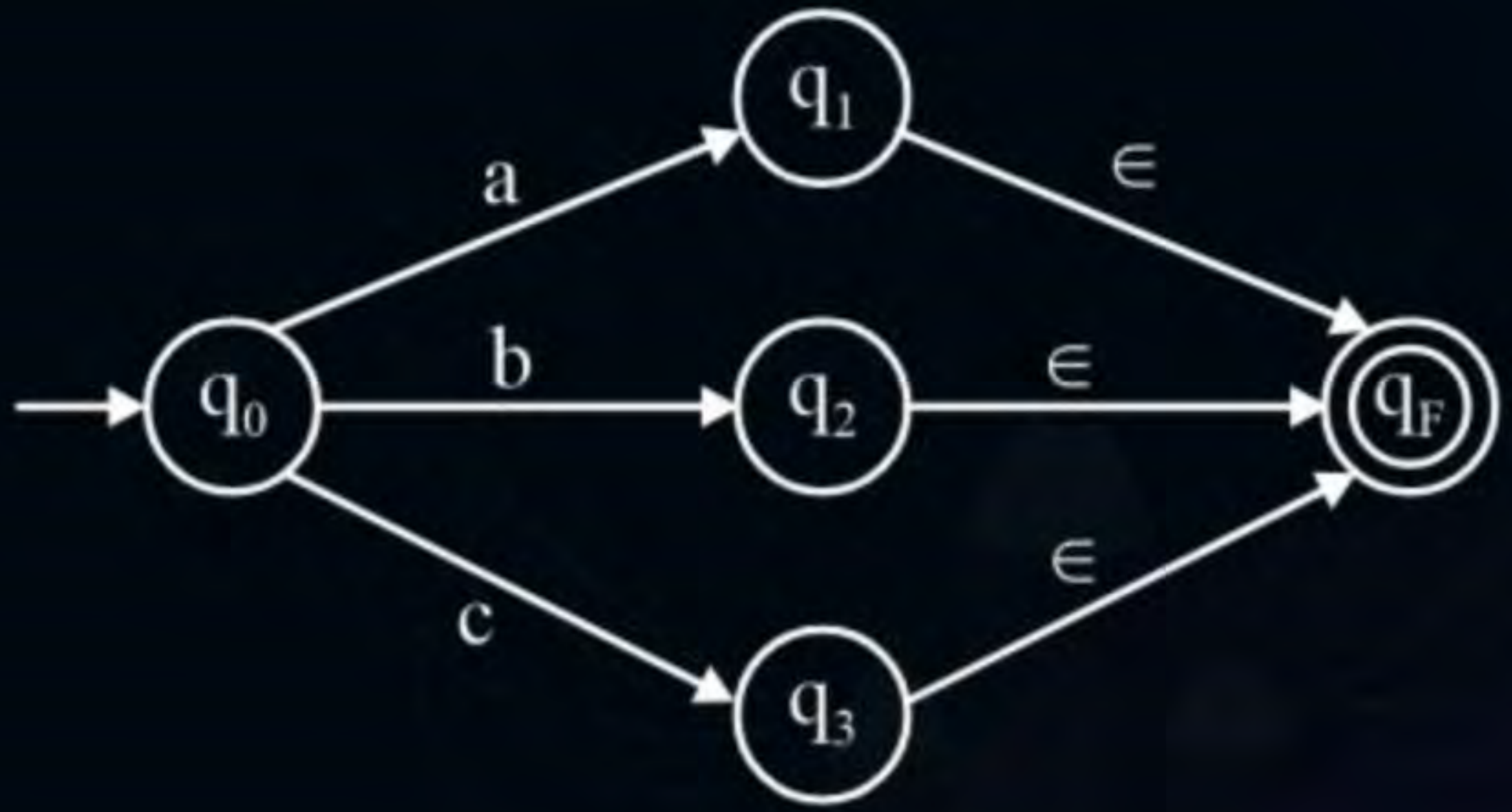
Topic : Theory of Computation



Topic : Finite Automata to Regular Expression

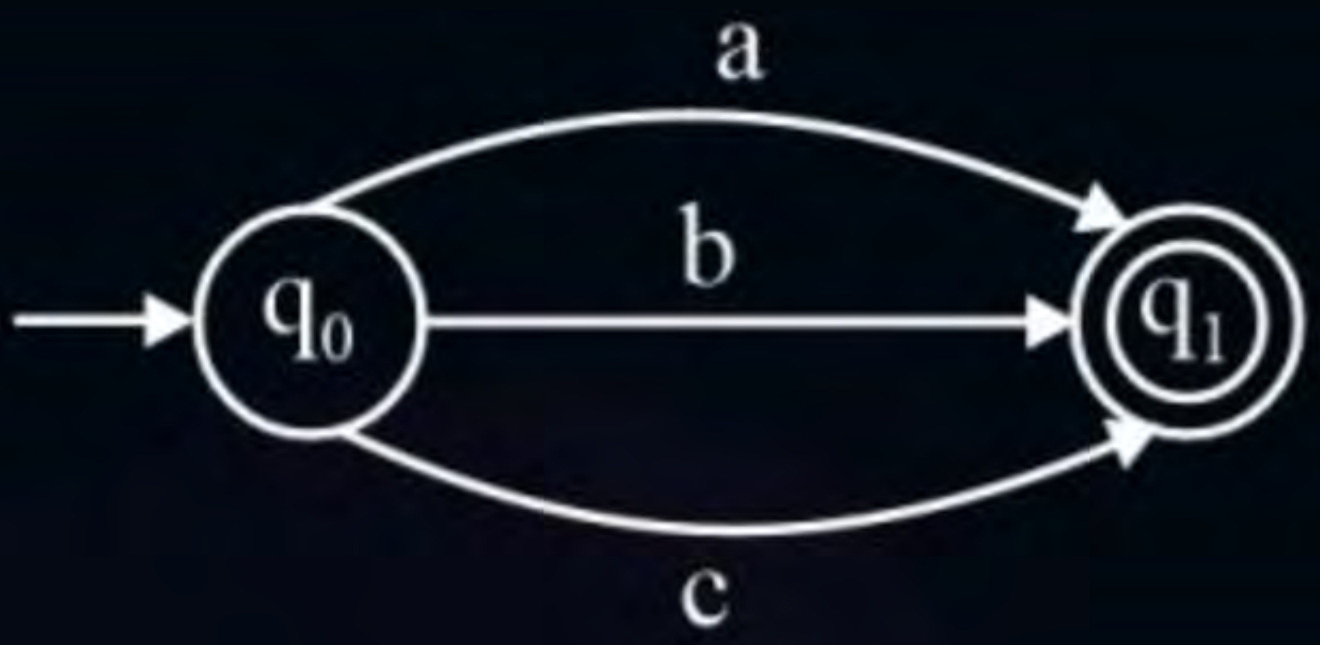
(1)

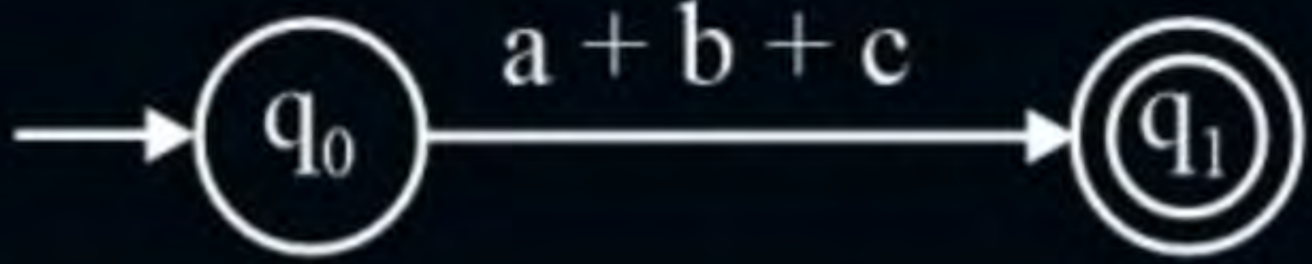




Topic : Finite Automata to Regular Expression

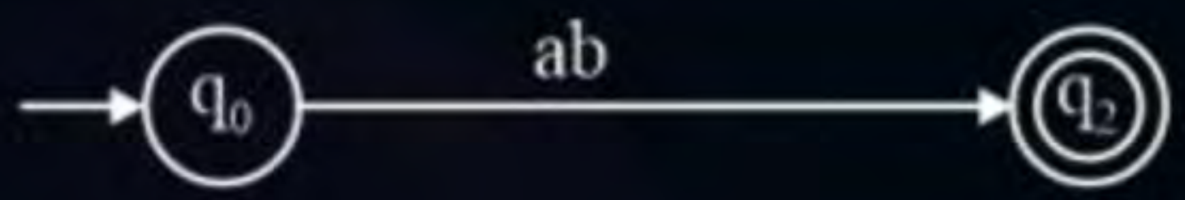
(2)





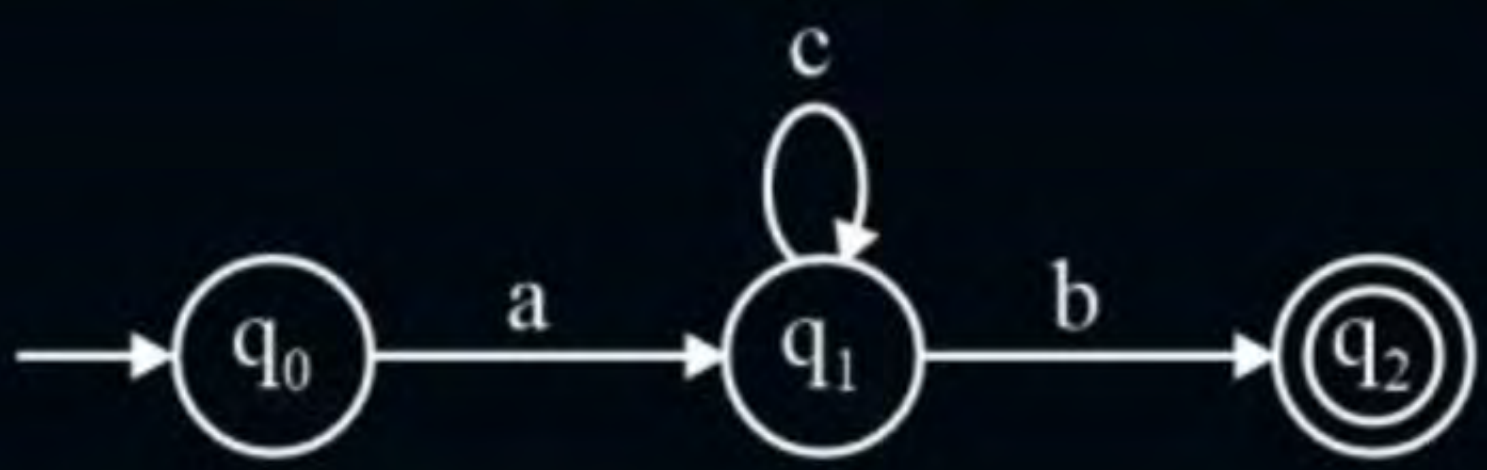
Topic : Finite Automata to Regular Expression

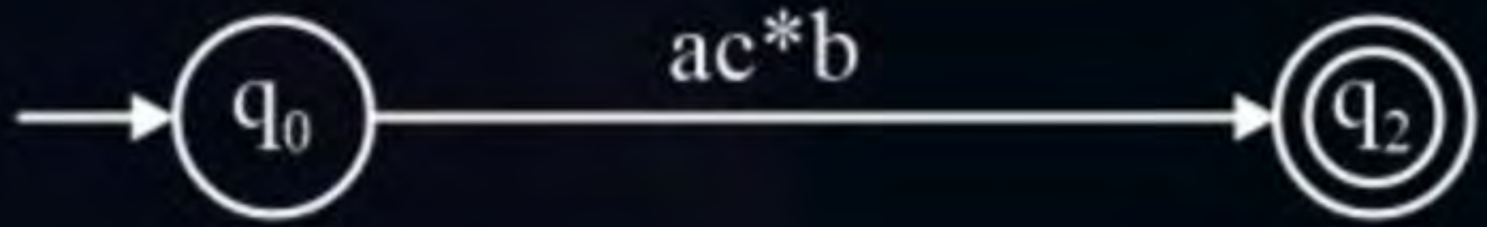
(3)



Topic : Finite Automata to Regular Expression

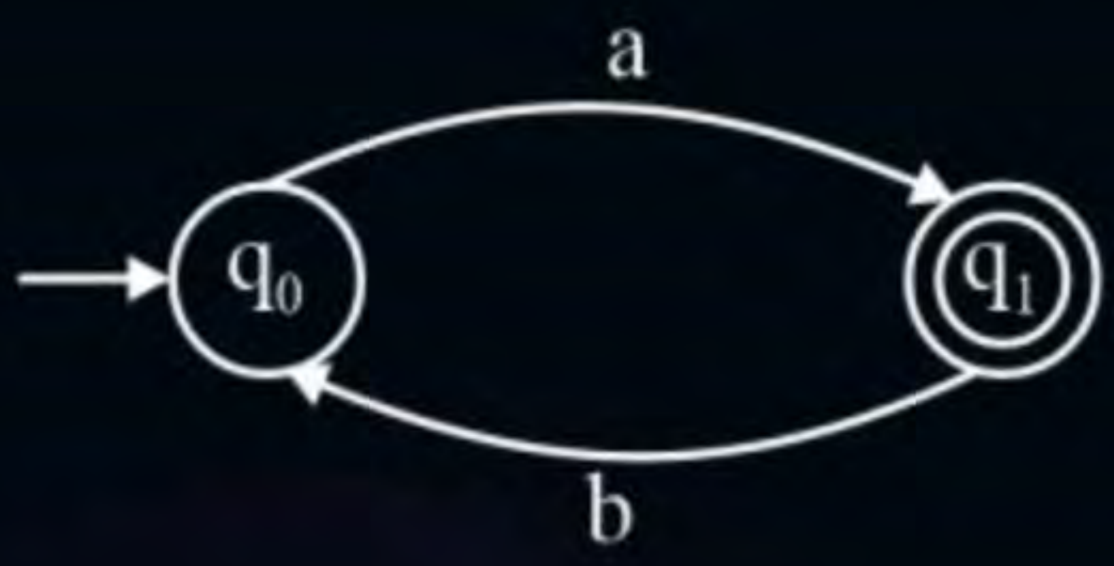
(4)

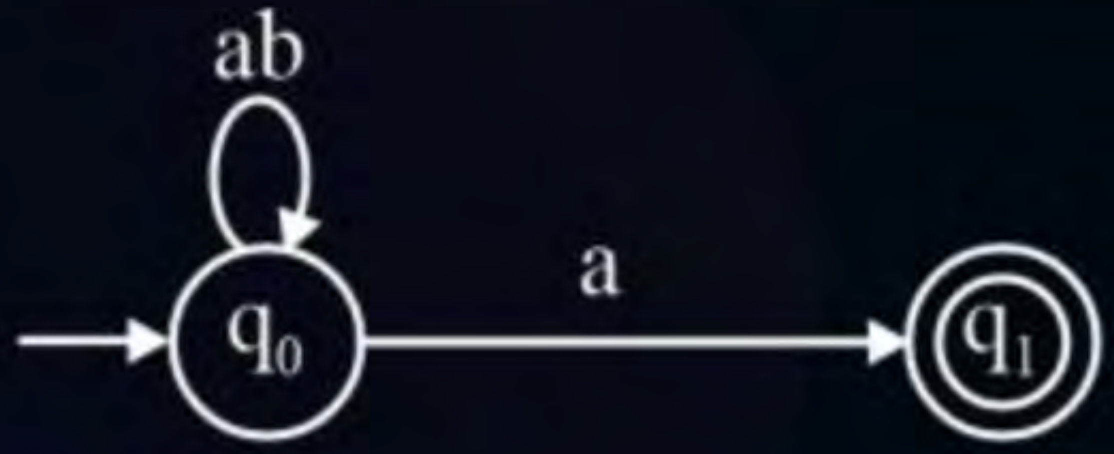




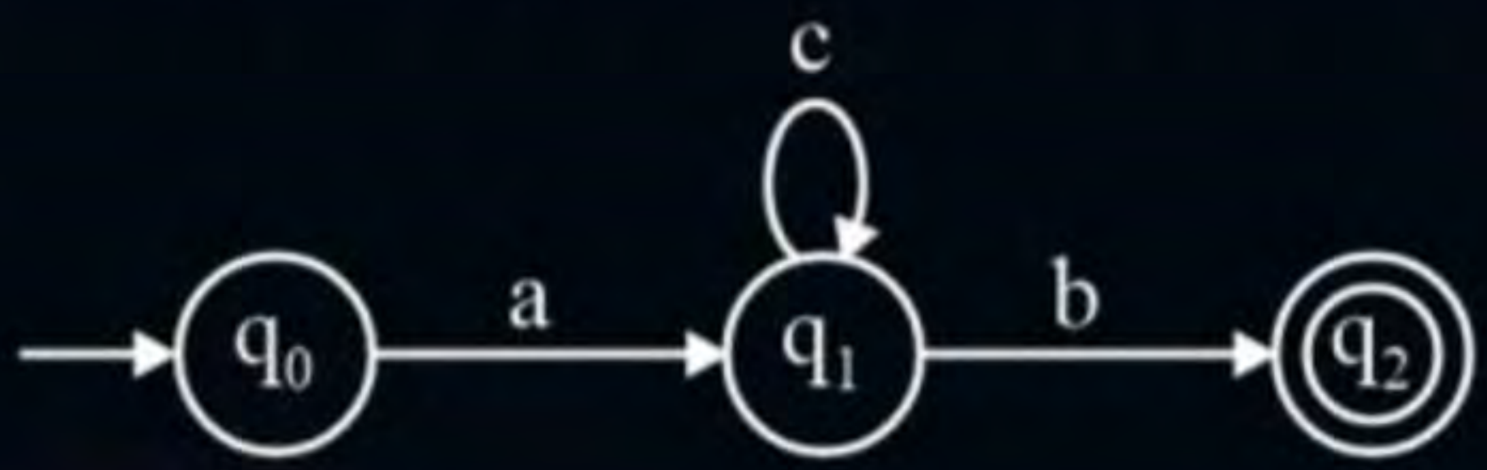
Topic : Finite Automata to Regular Expression

(5)

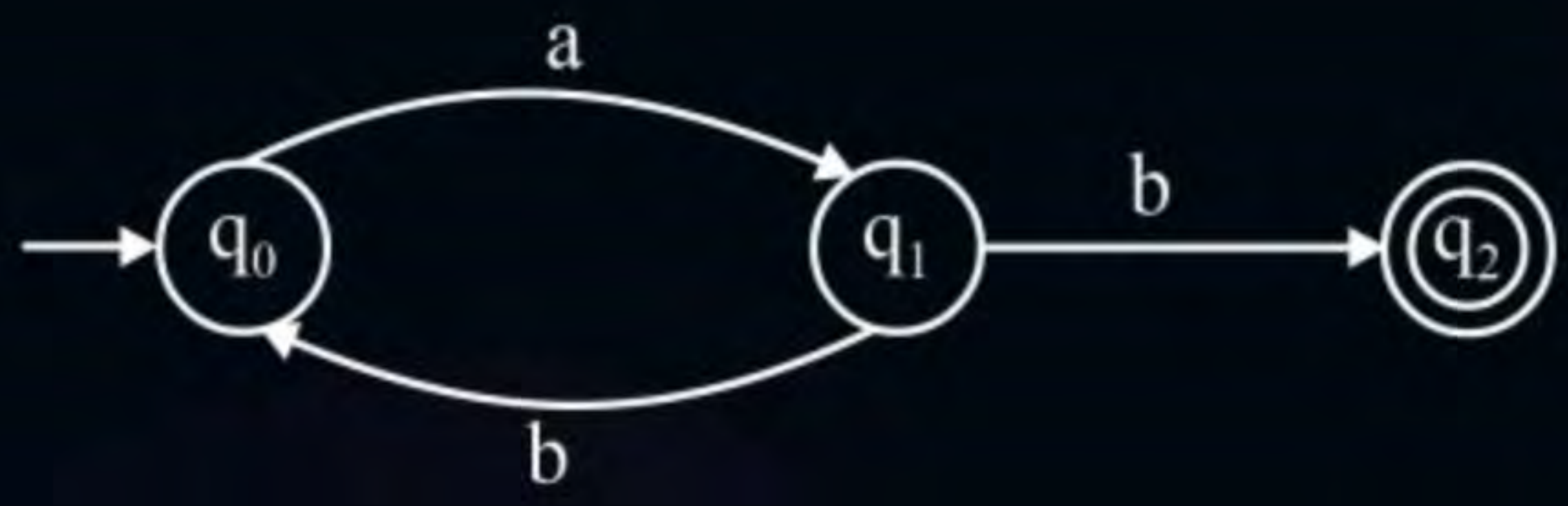




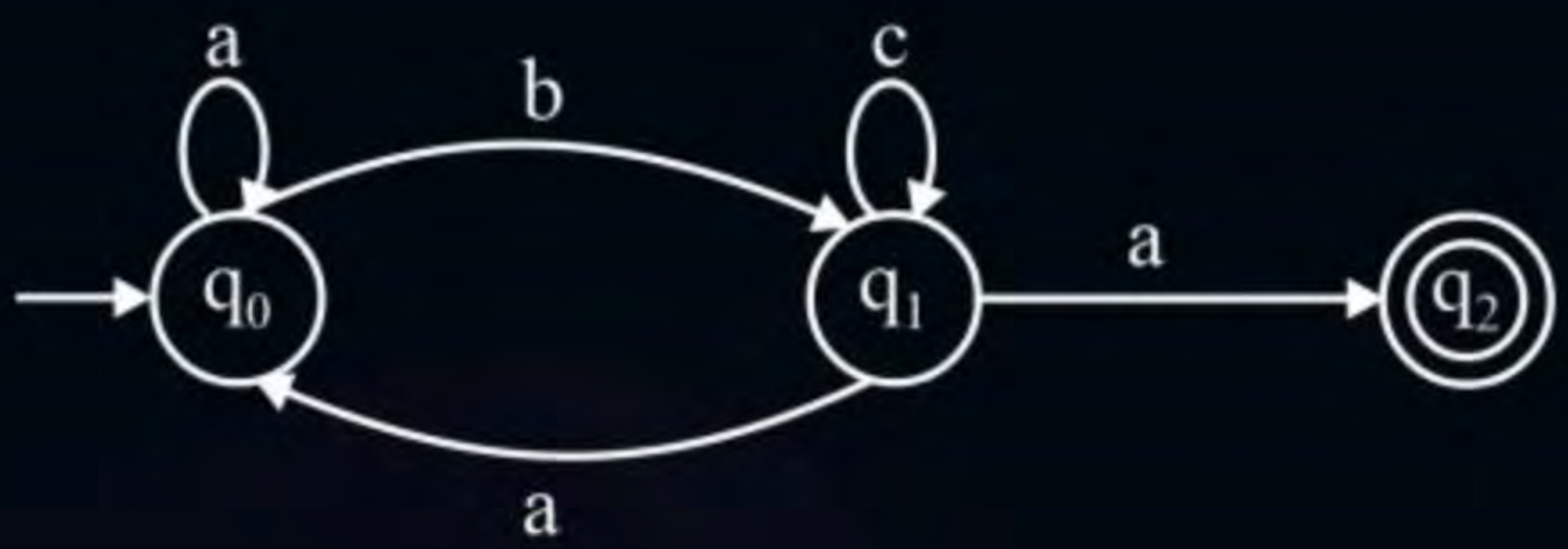
#Q. Construct Regular Expression for the following Finite Automata.



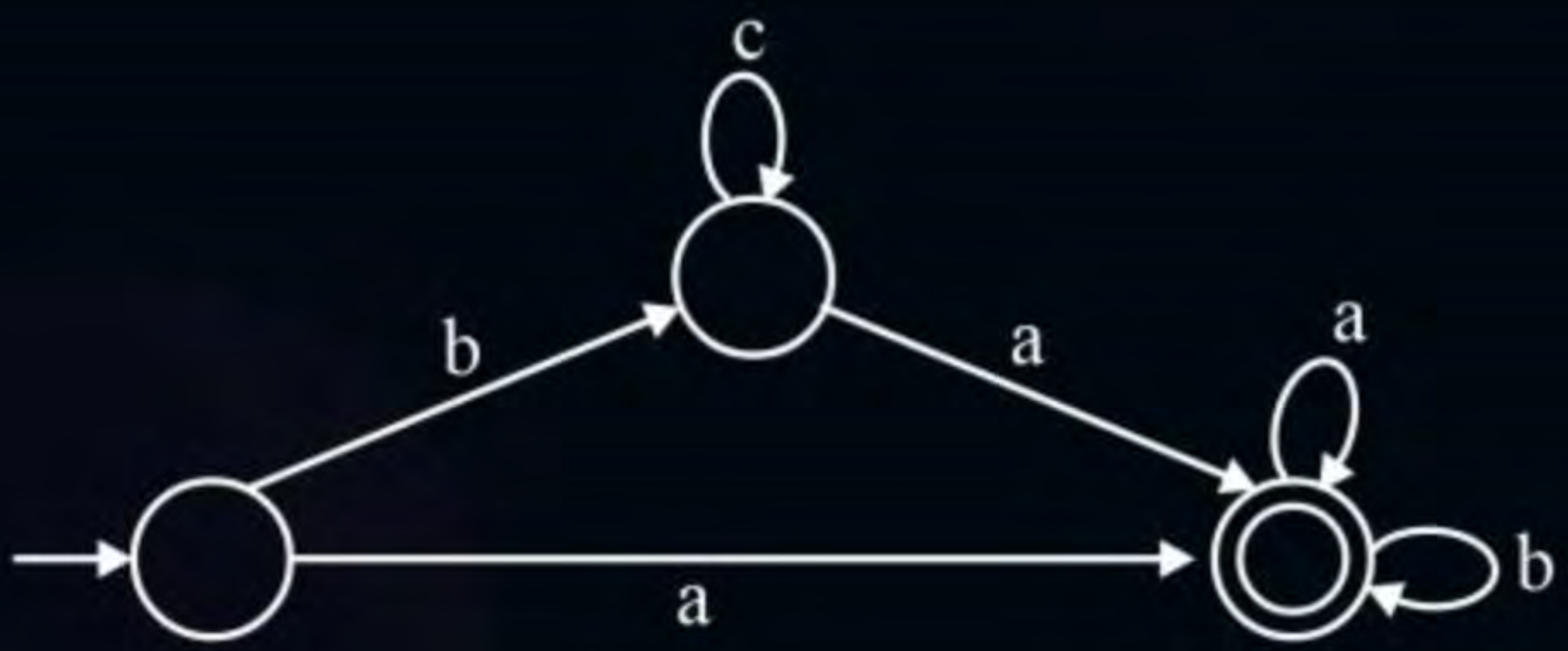
#Q. Construct Regular Expression for the following Finite Automata



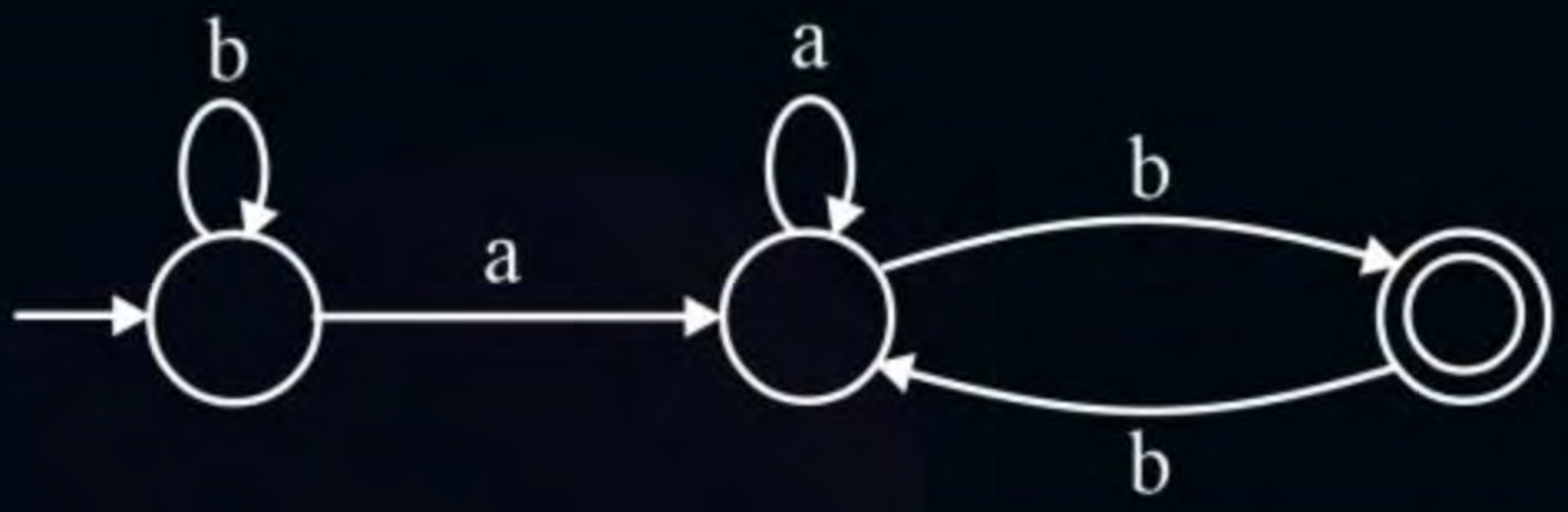
#Q. Construct Regular Expression for the following Finite Automata



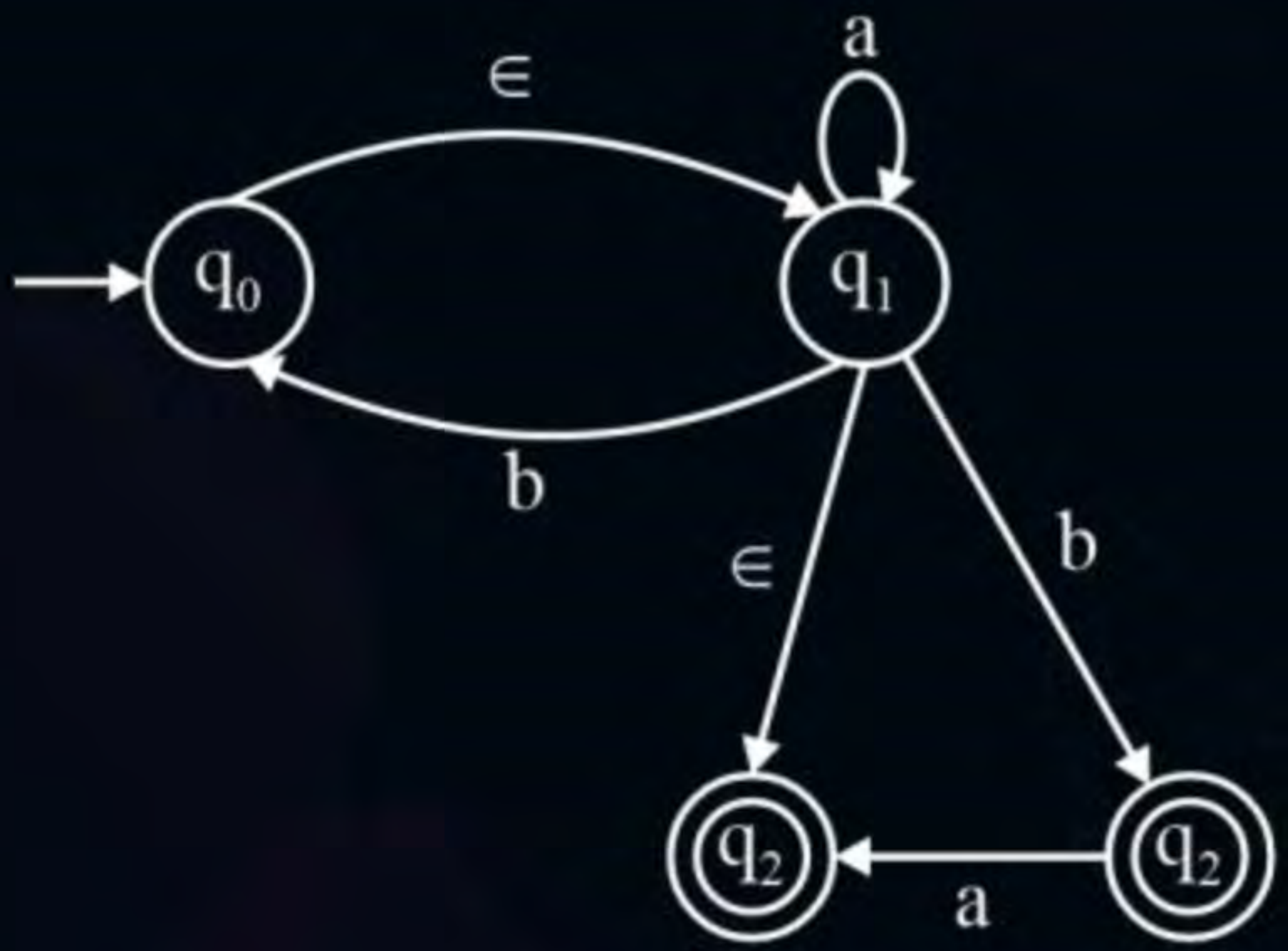
#Q. Construct Regular Expression for the following Finite Automata



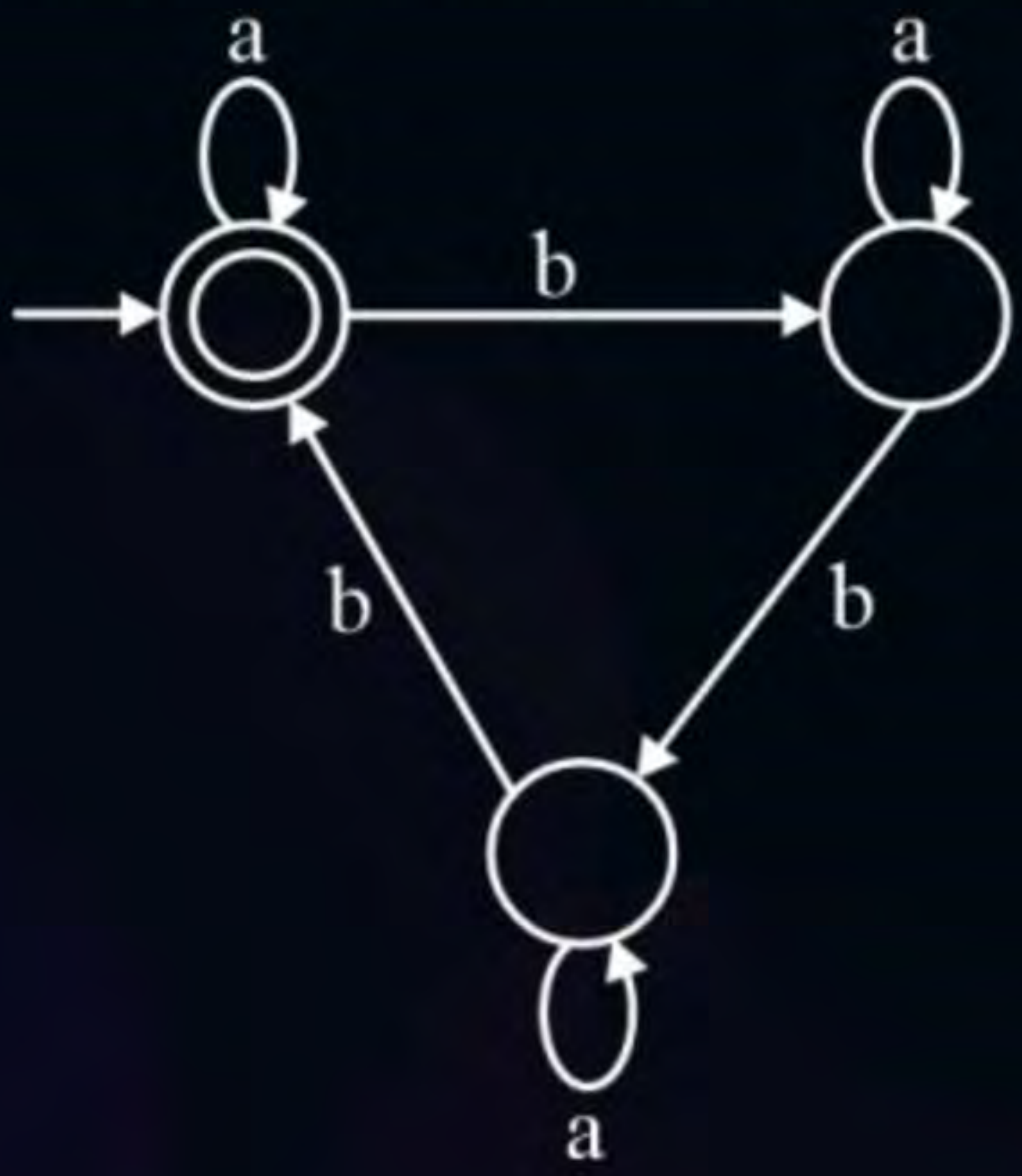
#Q. Construct Regular Expression for the following Finite Automata



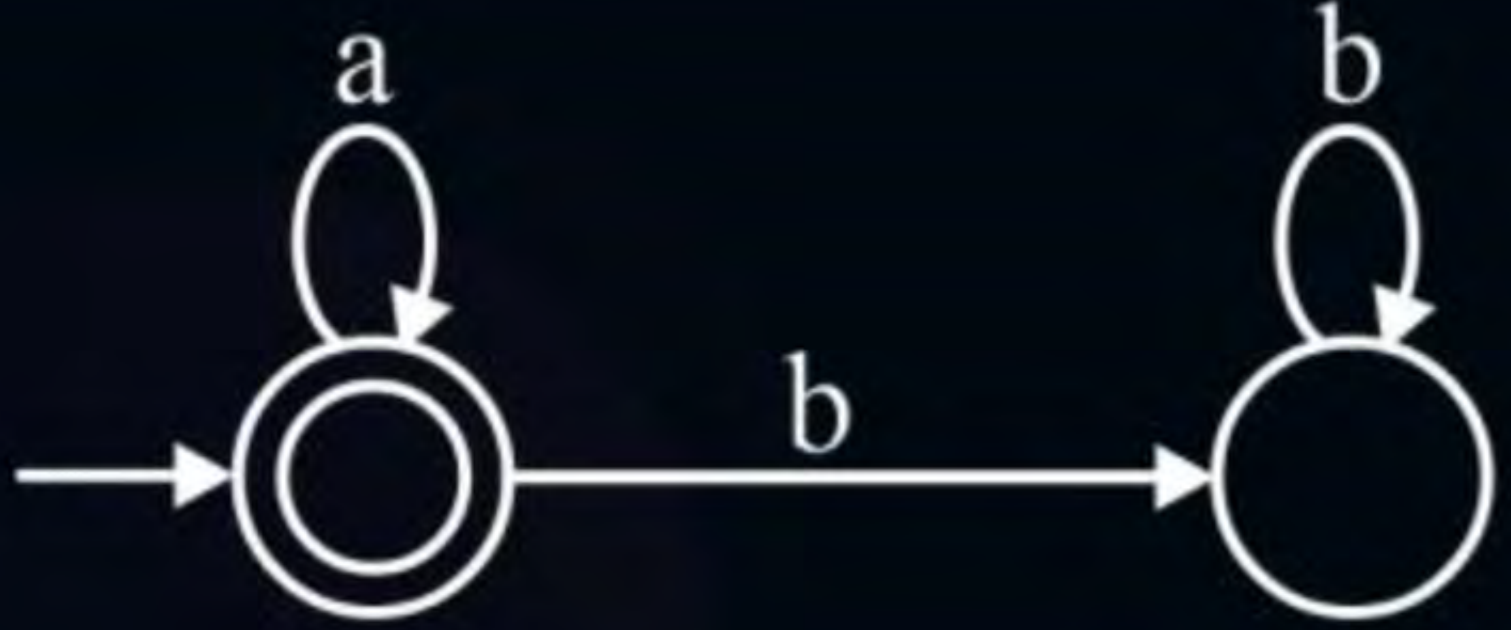
Q. Construct Regular Expression for the following Finite Automata



#Q. Construct Regular Expression for the following Finite Automata



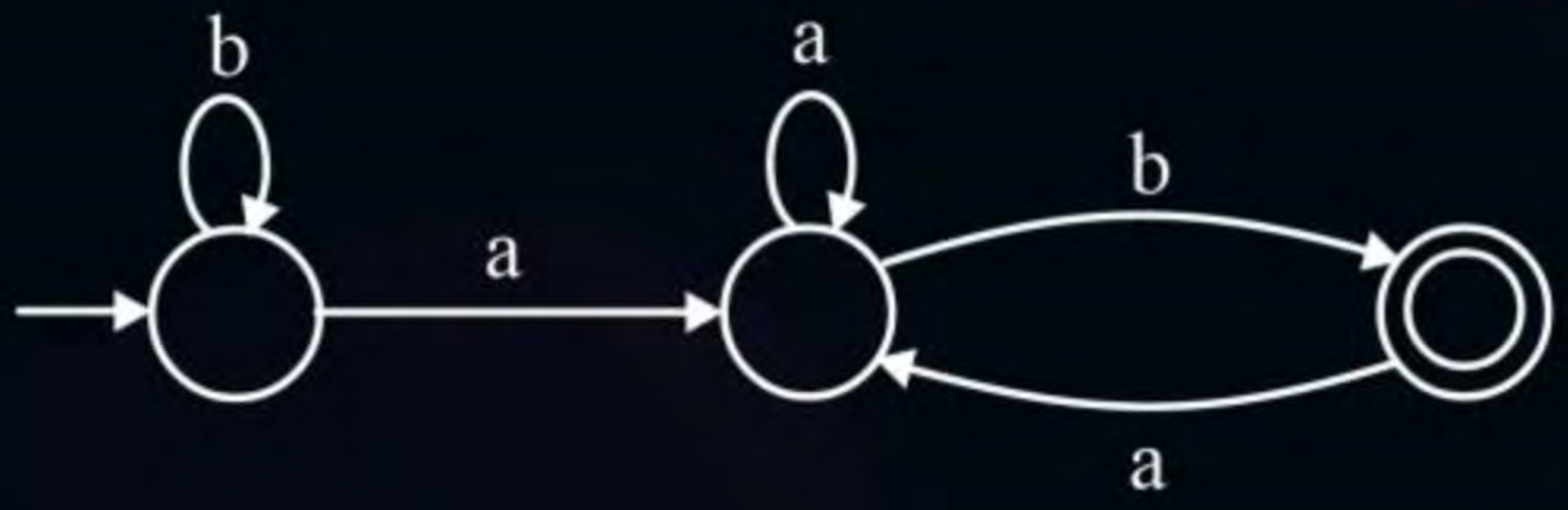
#Q. Construct Regular Expression for the following Finite Automata



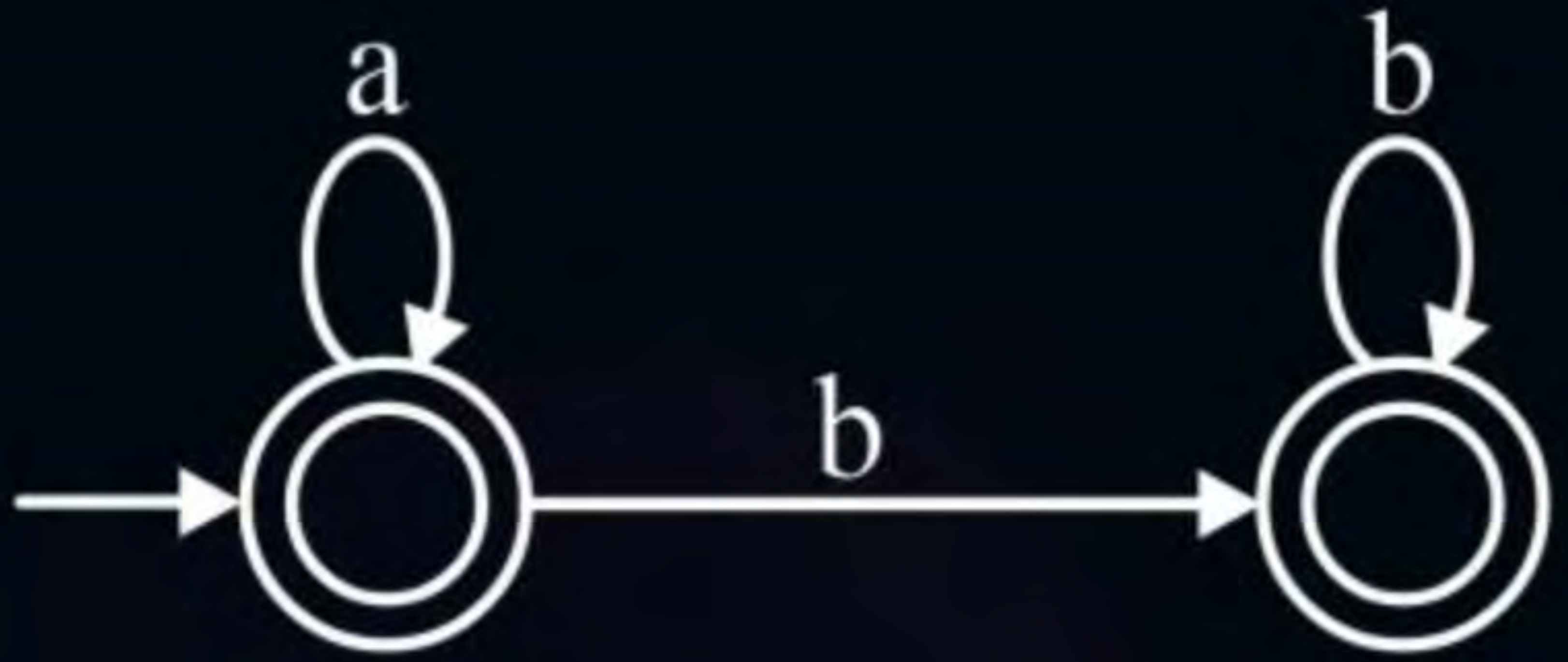
#Q. Construct Regular Expression for the following Finite Automata

#Q. Construct Regular Expression for the following Finite Automata

#Q. Construct Regular Expression for the following Finite Automata

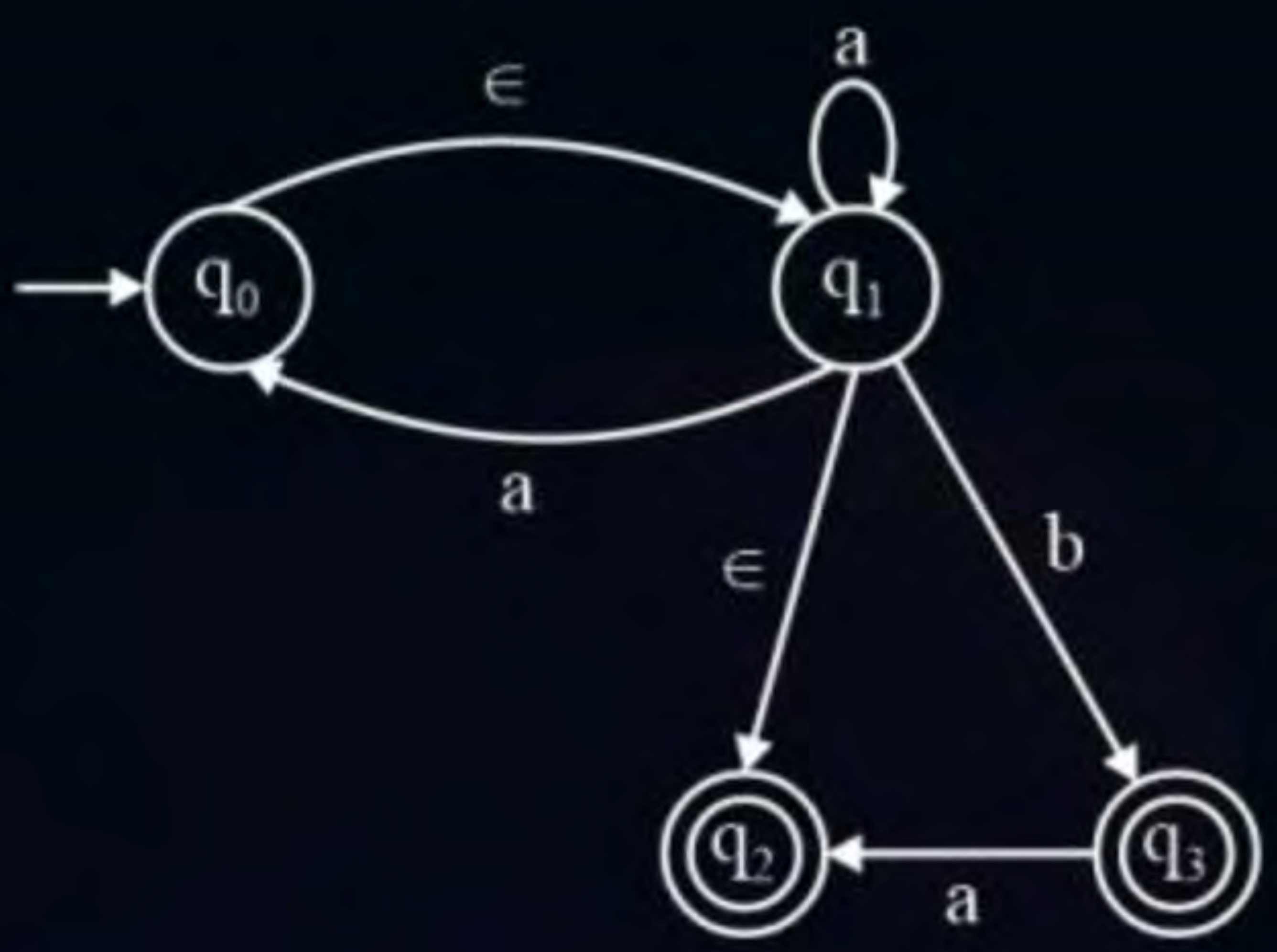


#Q. Construct Regular Expression for the following Finite Automata



#Q. Construct Regular Expression for the following Finite Automata

#Q. Construct Regular Expression for the following Finite Automata



#Q. Construct Regular Expression for the following Finite Automata

[MCQ]

#Q. Which one of the following regular expressions is equivalent to the language accepted by the DFA given below? [GATE-CS-shift-II-24: 1M]

A. $0^*1(0+10^*1)^*$

B. $0^*(10^*11)^*0^*$

C. $0(1+0^*10^*1)^*0^*$

D. $0^*1(010^*1)^*0^*$

[MCQ]

#Q. Let M be the 5-state NFA with ∈ - transitions shown in the diagram below. Which one of the following regular expressions represents the language accepted by M?

[GATE-CS-shift-II-24: 2M]

(A) $0^* + (1 + 0\,(00)^*)\,(11)^*$

(B) $(00)^* + 1\,(11)^*$

(C) $(00)^* + (1 + (00)^*)\,(11)^*$

(D) $0^+ + 1(11)^* + 0(11)^*$

Consider the languages $\boxed{L_1 = \phi}$ and $\boxed{L_2 = \{a\}}$. Which one of the following represents $\boxed{L_1 L_2^* \cup L_1^*?}$

[2013: 1 Mark]

A $\{\varepsilon\}$

B ϕ

C a^*

D (ε, a)

$$\left(\phi \cdot (a^*) + \phi^*\right)$$

$$\boxed{\phi + \epsilon}$$

$$\downarrow$$

$$\epsilon$$

hight

low

$*$

$\cdot$

high

$+$

low

Q

Let L be the language represented by the regular expression $\Sigma^*0011\Sigma^*$ where $\Sigma = \{0, 1\}$.

What is the minimum number of states in a DFA that recognizes ~~L'(complement of L)~~? above Language?

[2015-Set3: 1 Mark]

(A) 4

(B) 5

(C) 6

(D) 8

Σ^*

$(0+1)^*\,0011\,(0+1)^*$

$(n+1)$

5

Q

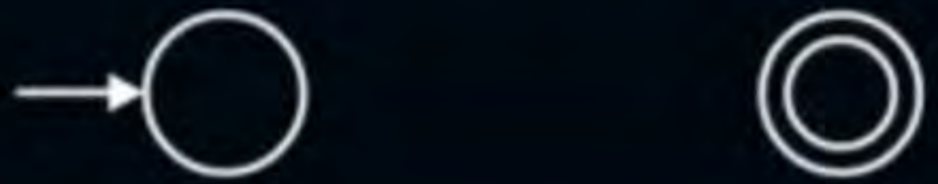
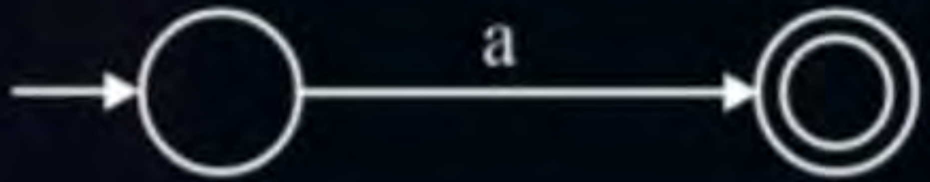
Consider the language L given by the regular expression $(a + b)^*b(a + b)$ over the alphabet $\{a, b\}$. The smallest number of states needed in a deterministic finite-state automaton (DFA) accepting L is ____.

2^n

[2017-Set1: 2 Marks]

$$\left.(a+b)^* \, b \, (a+b)\right\} \; 2^2 = \underline{4 \text{ states}}$$

Topic : Regular Expression to Finite Automata Construction

Regular expression	$\in$-NFA
1. ϕ	
2. $\in$	
3. a	
4. $r_1 + r_2$	

Topic : Regular Expression to Finite Automata Construction

Regular expression	$\in$-NFA
5. $r_1 \cdot r_2$	
6. r^*	

(Q) How many states in min DFA for the Regular Expression

$\Sigma = \{1\}$ $\boxed{(11+111)^*} \rightarrow$ Regular language $\Rightarrow$ F.A

ϕ

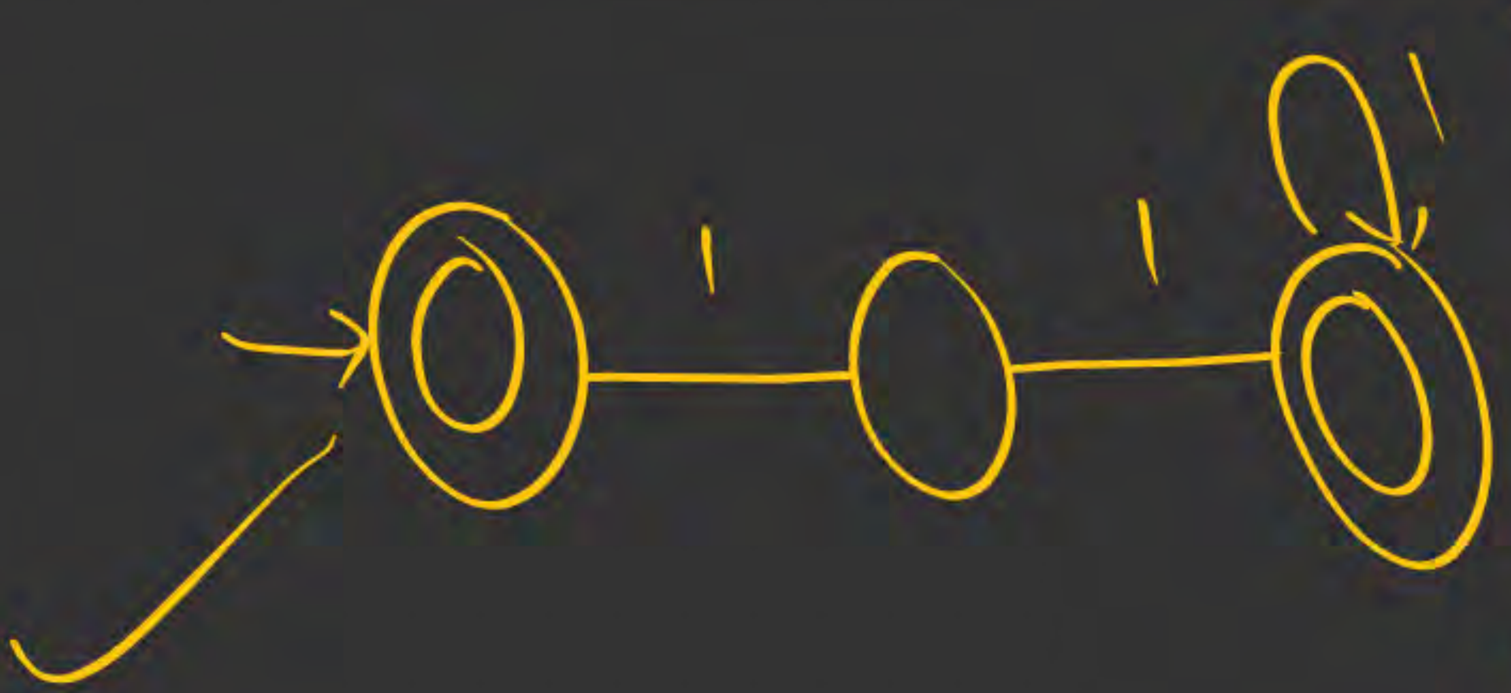
(a) 5

(b) 2

(c) ~~3~~

(d) 8

$\{\epsilon, 1^2, 1^3, 1^4, 1^5, 1^6, 1^7 \ldots\}$

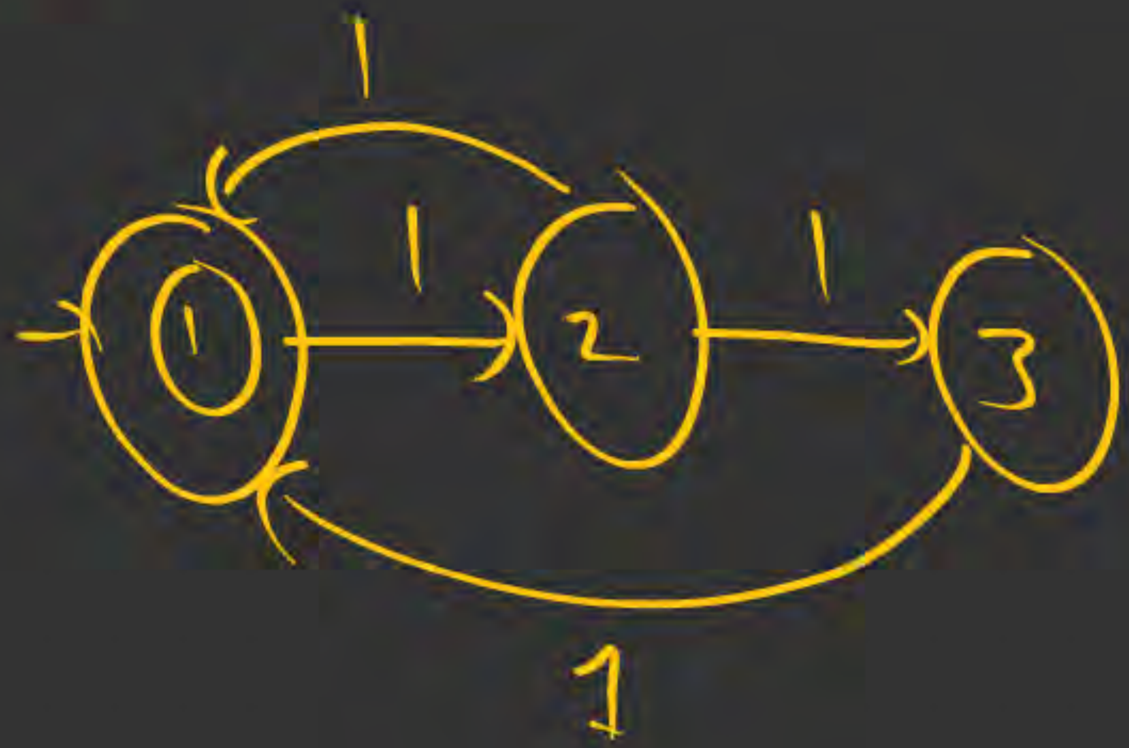


3 states in min DFA.

$$(11+111)^*$$

NFA

NFA



(Q) How many states in min DFA for the following Regular Expression?

$\boxed{\Sigma = \{1\}}$

$$(111 + 11111)^* \qquad (a+b)^*$$

GATE 2006

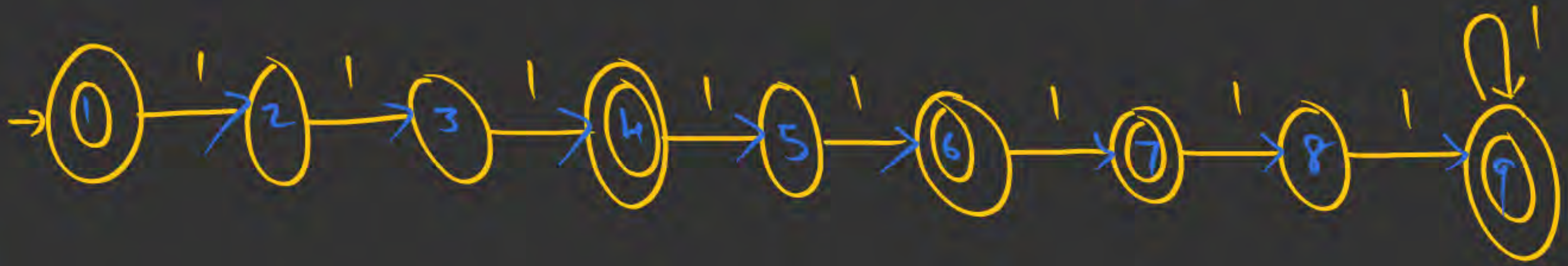
$$\{\epsilon, 1^3, 1^5, 1^6, 1^8, 1^9, 1^{10}, 1^{11}, 1^{12} \cdots\}$$

(a) 5

(b) 8

(c) 15

(d) 9



DFA

$L = \{a^{2n} / n \geq 1\} = \{a^2, a^4, a^6 - - - \}$

$L = \{a^{2^n}\} =$ no DFA $\}$ non Regular

#Q. Construct Finite Automata for the given regular expression

$\left.\begin{matrix}\text{min}\\ \text{DFA}\end{matrix}\right\}$ $(n+1)$ states.

$$\underline{a}^*\,\underline{b}^*\,\underline{c}^* \longrightarrow 4$$

$$a^*b^*c^*d^* \ldots z^* \longrightarrow 27$$

$$0^*1^*2^*3^*4^* \longrightarrow 6$$

(Q) How many states in min DFA for the following Regular Expression

① $(a+b)(a+b)(a+b) \xrightarrow{(n+2)} 5$

② $\underbrace{(a+b+\epsilon)(a+b+\epsilon)(a+b+\epsilon)}_{\text{atmost } 3} \xrightarrow{(n+2)} 5$

③ $b^* a\, b^* a\, b^* a\, b^* a\, b^* \xrightarrow{\text{exactly } 4\ a\text{'s}} 6$

④ $b^*(a+\epsilon)b^*(a+\epsilon)b^* \xrightarrow{a\text{'s atmost } 2} 4$

⑤ $(a+b)(a+b)(a+b)^+ \xrightarrow{\text{atleast } 3\ (n+1)} 4$

min DFA

(6) $[(a+b)(a+b)(a+b)]^*$ $\xrightarrow{\text{Div by 3}}$ 3 States

(7) $[(a+b)(a+b)]^*(a+b)$ $\xrightarrow{\text{odd}}$ 2 states ✓

(8) $(b^*ab^*ab^*ab^*ab^*)^* + b^*$ $\xrightarrow{\text{a's div by 4}}$ 4

(9) $(a+b)^*\,a\,(a+b)(a+b)$ $\xrightarrow{2^n}$ $2^3 = 8$ states

end

(10) $(a+b)(a+b)(a+b)\,a\,(a+b)^*$ $\xrightarrow{(n+2)}$ 6 states

(11) $(a+b)^* \, abab \, (a+b)^* \xrightarrow{\text{Substring } (n+1)} 5$ states

(12) $(a+b)^* \, \underbrace{ababab} \xrightarrow{(n+1)} (n+1)$ states

$(a+b)^*$. Ending with n length $\longrightarrow (n+1)$ states

Regular Language

Detection

$\boxed{***}$

1. Regular Language Detection

2. Pumping Lemma

$\boxed{***}$

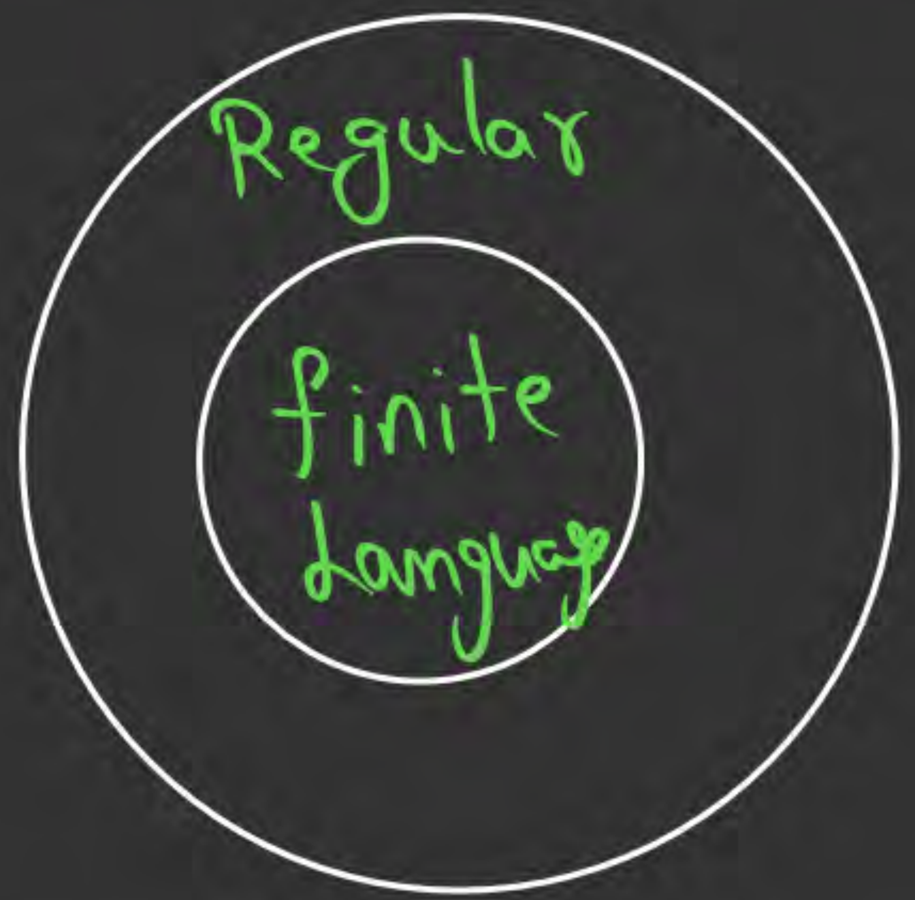
3. Closure properties

4. Mealy & Moore Machine

① Every finite language is Regular.

But every infinite language need not be Regular.

(may (or) may not be Regular)



(Q) Which of the following is non Regular ?

(a) $L = \{a^n b^n c^n d^n \mid n \leq 100\}$ ⟶ finite Lang ⟶ $\underline{\text{Regular}}$ ✓

(b) $L = \{a^n b^m \mid (n > m) \text{ and } (n < m)\} = \{\ \}$ → Empty Lang ⟶ Regular

(c) $L = \{a^{10^{10^{10}}}\} \Rightarrow \{1\}$ ⟶ finite Lang ⟹ Regular

(d) $L = \{a^n b^m \mid n > m,\ n \geq 1\}$ $\xrightarrow{\text{Dependency}}$ $\underline{\text{non Regular}}$

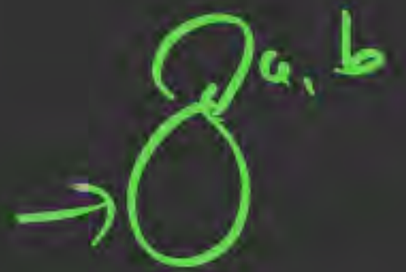
Infinite language

Dependency exist

non Regular

Dependency not exist

Regular



1. Empty Lang → Regular

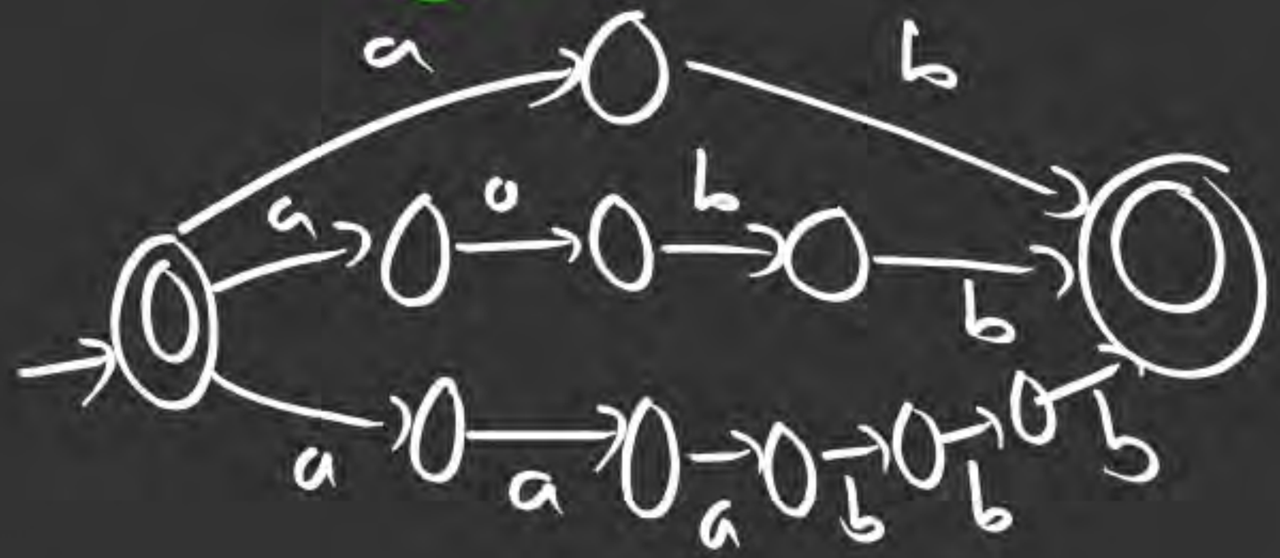
2. finite " → Regular

3. Infinite → Regular
   → Non Regular

$\{\epsilon, ab, a^2b^2, a^3b^3 \}$

$L_1 = \{a^n b^n \mid n \leq 3\}$ = Regular

infinite



$L_2 = \{a^n b^n \mid n \geq 1\}$ = Non Regular

infinite

Topic : Regular Language Detection

Which of these Languages are Regular

1. $L = \{a^n b^n c^n \mid 1 \leq n \leq 100\}$ $\xrightarrow{\text{finite}}$ Regular.
2. $L = \{a^n b^m \mid n + m = 10\}$ $\xrightarrow{\text{finite}}$ Regular
3. $L = \{a^n b^m \mid n - m = 5\}$ $\xrightarrow{\text{infinite}}$ non Regular
4. $L = \{a^n b^m \mid n * m = 100\}$ $\xrightarrow{\text{finite}}$ Regular
5. $L = \{a^n b^m \mid n = 2m + 1\}$ $\xrightarrow[\text{Dependency}]{\text{infinite}}$ Non Regular
6. $L = \{a^n b^m \mid n > m\}$ $\xrightarrow[\text{Dependency}]{\text{Infinite}}$ Non Regular
7. $L = \{a^n b^m \mid n >$ [redacted] $\xrightarrow[\text{no Dependency}]{\text{Infinite}}$ Regular

$$L = \{ a^n b^m \mid n = 2m+1 \}$$

$$\{ a^{2m+1} b^m \} \quad m \geq 0$$

0

1

2

3

⋮

Inf

$$L = \{ a^n b^m \mid n - m = 5 \}$$

$$n = m + 5 \}$$

$$\{ a^{m+5} b^m \} \; m \geq 0 \}$$

1 2 3 4 5

Infinite → Dependency } **Non Regular**

$$\{a^n b^m \mid n+m=10\}$$

$$\{a^0 b^{10}, a^1 b^9, a^2 b^8 \dots a^{10} b^0\}$$

TOC 10

Automata

1. DFA, NFA, ε-NFA 6 8
5
2. DPDA, NPDA
3. Turing machine

Languages 4

Very Easy

1. Regular languages 3
2. Contextfree language
3. Recursive Enumerable language

→ Detection

→ closure property

THANK - YOU