CS & IT ENGINEERING

Theory of Computation

DFA

Lecture No.- 01

By- Venkat Rao sir

THEORY OF COMPUTATION

(7-10) ✓ 70

50%

Topic 1 Finite Automaton & Regular Languages.

25%

Topic 2 Pushdown Automata & Context free Languages.

Topic 3 Turing Machine & Recursive Enumerable Languages.

25%

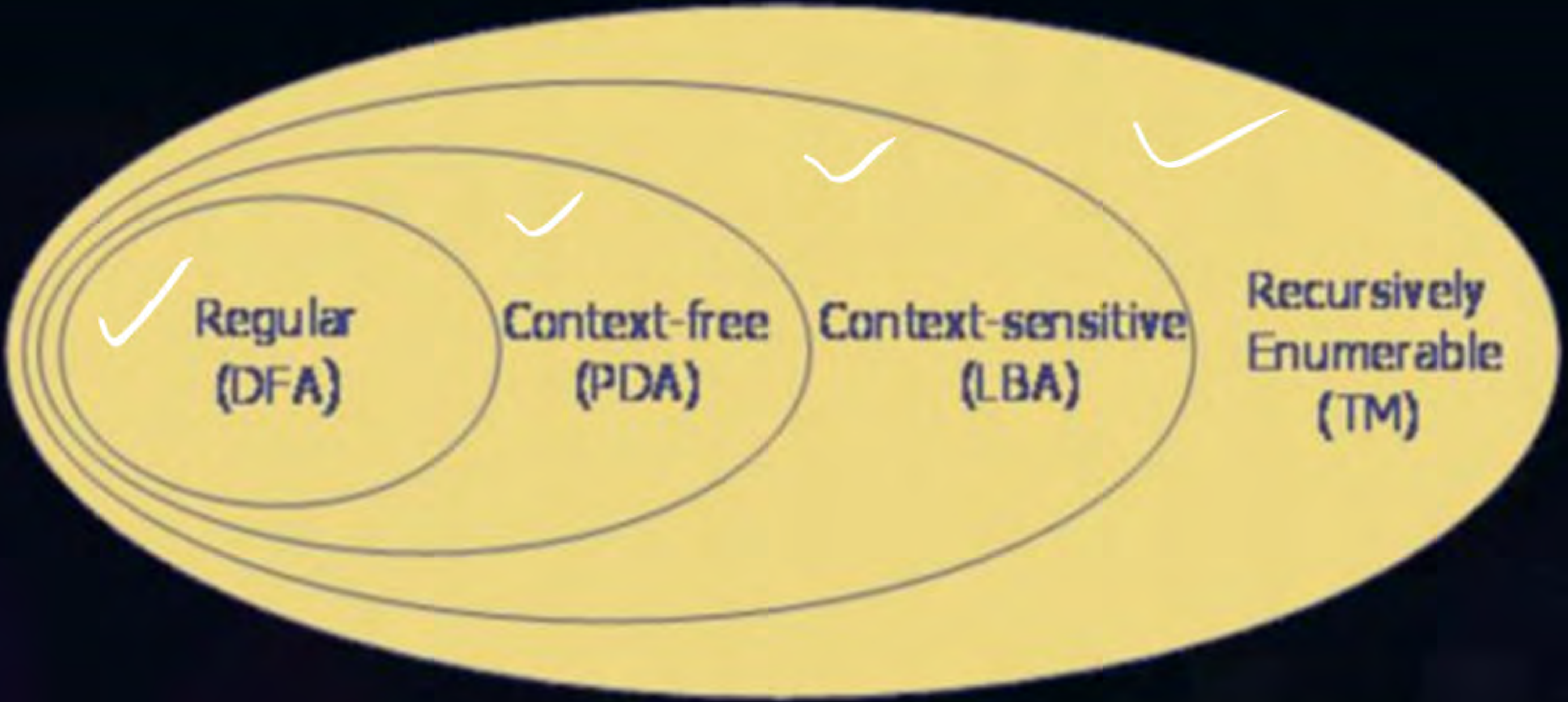
Topic 4 Undecidability.

BOOKS:

1. PETER LINZ
2. MICHAEL SIPSER
3. HOPCROFT & ULLMAN

$L = \{ \quad \}$

Topic : Theory of Computation



Topic : Introduction:

It is the mathematical study of computing machines and their capability

or

It is the study of automata theory and formal languages.

Topic : Introduction:

Applications of Theory of Computation:

- Algorithm design and analysis
- Compiler design
- Cryptography and network security
- Artificial intelligence and machine learning
- Database systems and query optimization
- Software verification and model checking

Topic : Introduction:

Decidable Problem : Algo exist

Undecidable Problem no algo

Topic : Terminologies:

Alphabet(Σ): Finite non-empty set of symbols

Ex:- {a, b} – ✓

{a, 1, 2} – ✓

{ } – ✗

{अ, आ} ✓

Topic : Terminologies:

Alphabet(Σ): Finite non-empty set of symbols

Ex:- {a, b} –

{a, 1, 2} –

{ } –

Topic : String:

String: Finite sequence of symbols over the given alphabet Σ.

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

Ex:-

$\underline{ab} \rightarrow 2$

$abaab \rightarrow 5$

$aaa = a^3 \longrightarrow 3$

Epsilon $= a^0 \longrightarrow$ zero

ϵ ✓

Topic : String:

Language :- Any set of strings over the given alphabet $\Sigma = \{a, b\}$.

$L_1 = \{ab, ba, abab\}$ - → finite Language

$L_2 = \{a, ab, aba, \ldots\ldots\}$ - → infinite Language

$L_3 = \{\ \}$ - Empty Language

$L_4 = \{\epsilon\}$ - 1 → finite Language

$L_5 = \{a\}$ - 1 → finite Language

$L_6 = \{\epsilon, a, b, aa, ab, ba, bb \text{------}\}$ → infinite Language } Σ^*

Complete Language

Topic : String:

Sub – String : Consecutive sequence of symbols over the given string.

Total no of substring for the given string $= n(n+1)/2 + 1$

FINITE AUTOMATA

Topic : Finite Automata

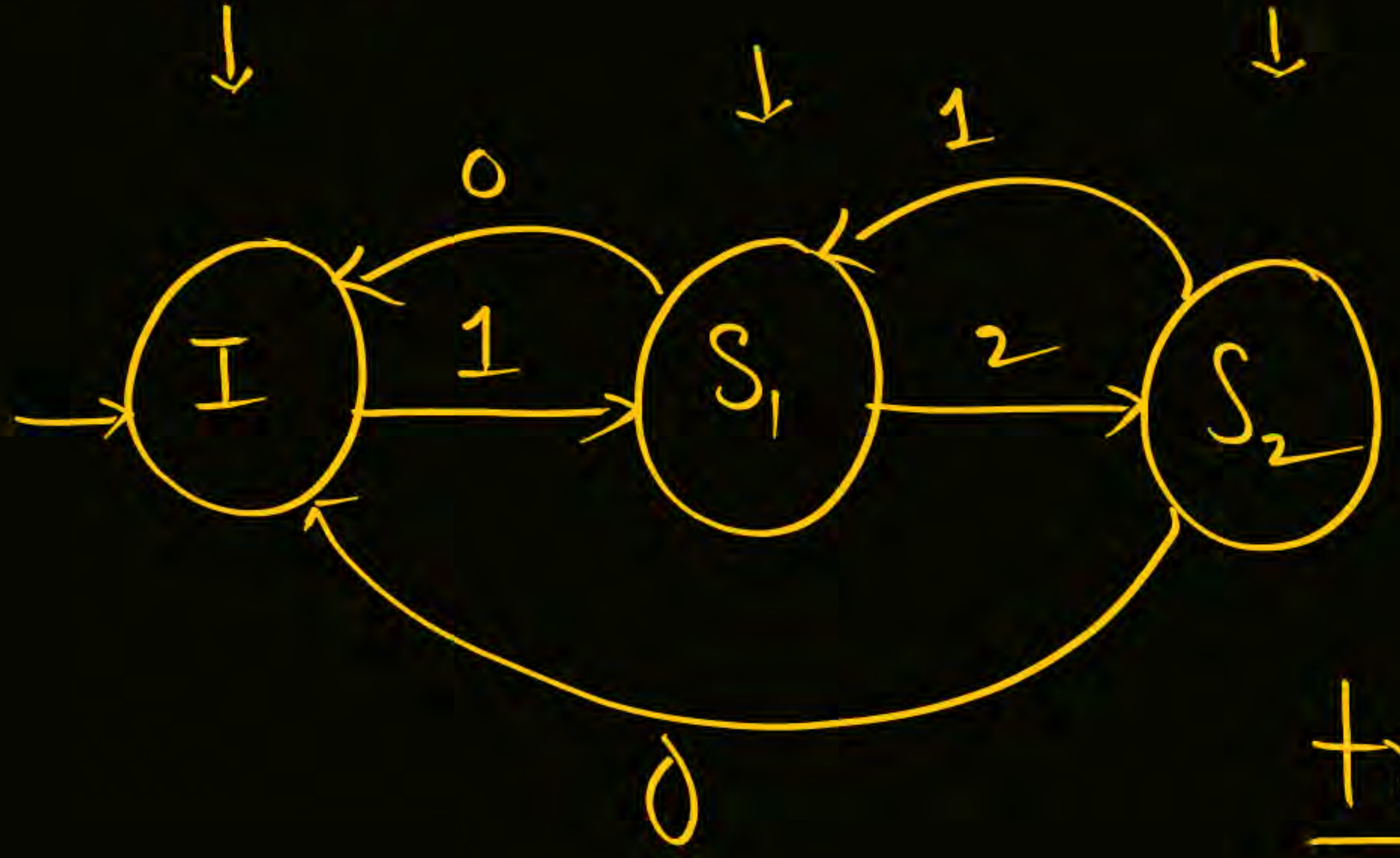
It is a mathematical model which contains finite number of states and transitions.

Compiler Design **Finite Automata** Hardware Design

- input → **Without Output** → Yes / → NO
  1. → DFA ✓
  2. → NFA
  3. → **ϵ - NFA**
- input → **With output** → output
  4. → Mealy machine
  5. → Moore machine

FAN

States



transitions

Lexical Analysis:

String Matching

Network Protocol Analysis:

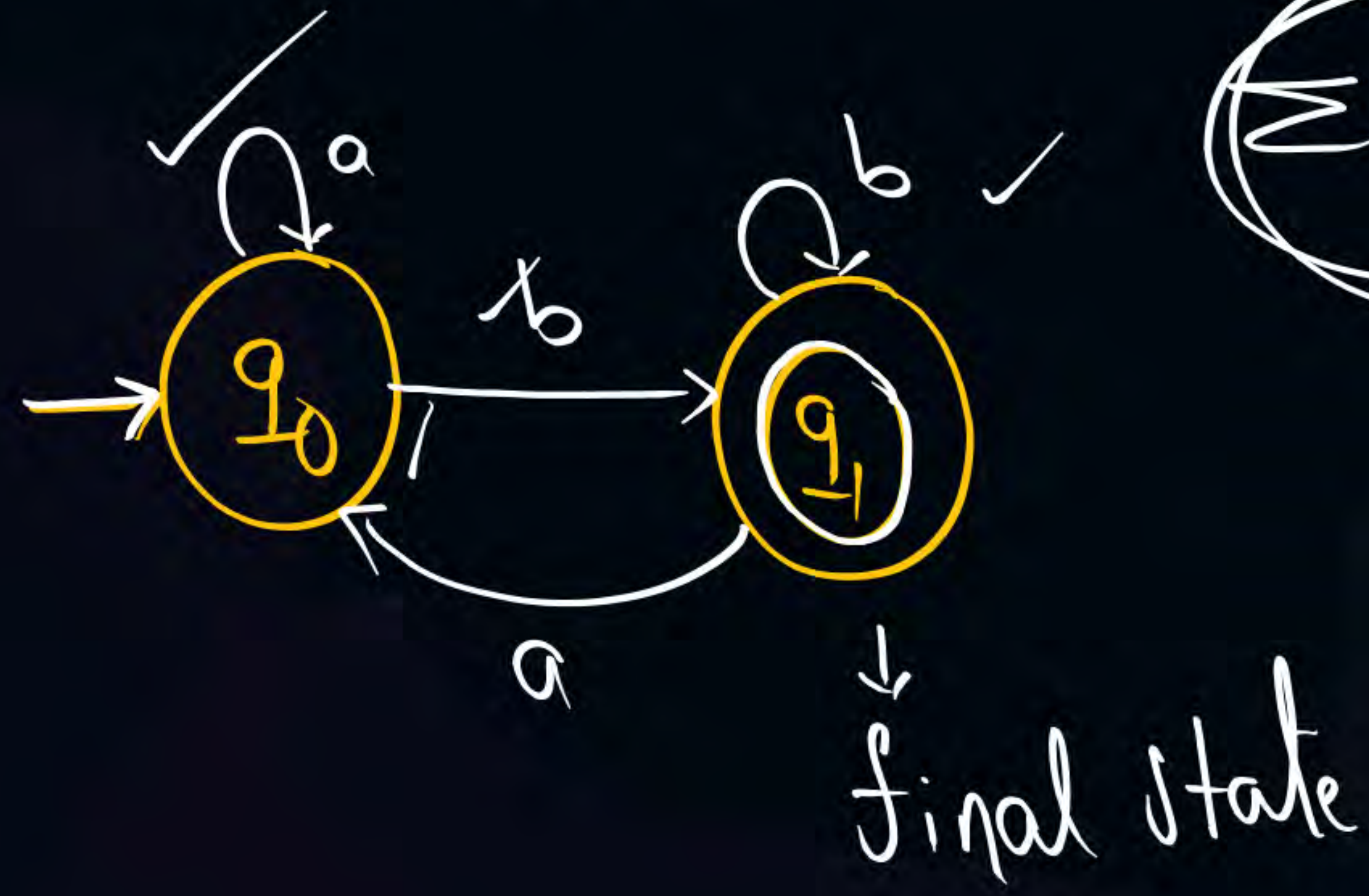
Digital Circuit Design:

Regular Expression Engines:

Natural Language Processing

Topic : Deterministic Finite Automata

DFA : It is a finite automata in which from every state on every input symbol exactly one transition should exits.



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

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 → one initial state

0, 1, 2, ---- all   any → F : Set of final states

δ : Transition function $\boxed{Q * \Sigma \rightarrow Q}$

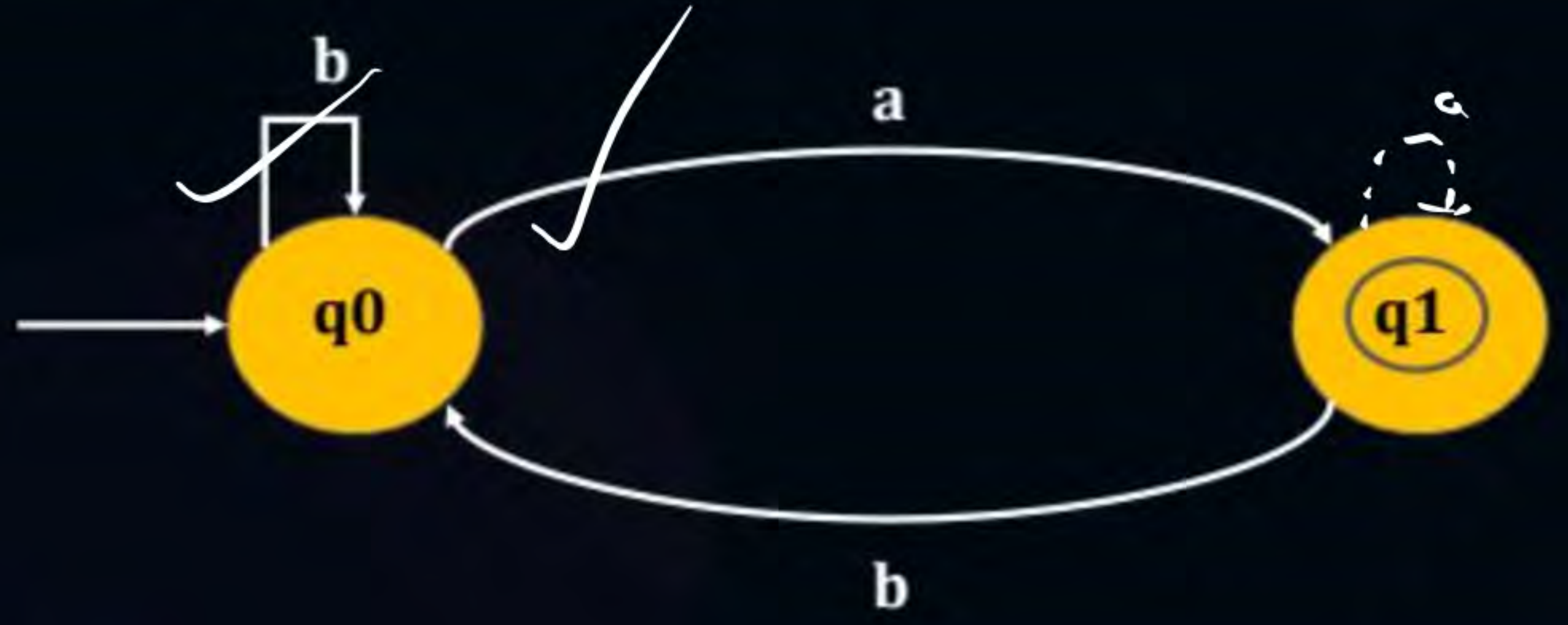
Topic : Deterministic Finite Automata

DFA?

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

Example of :

(1)



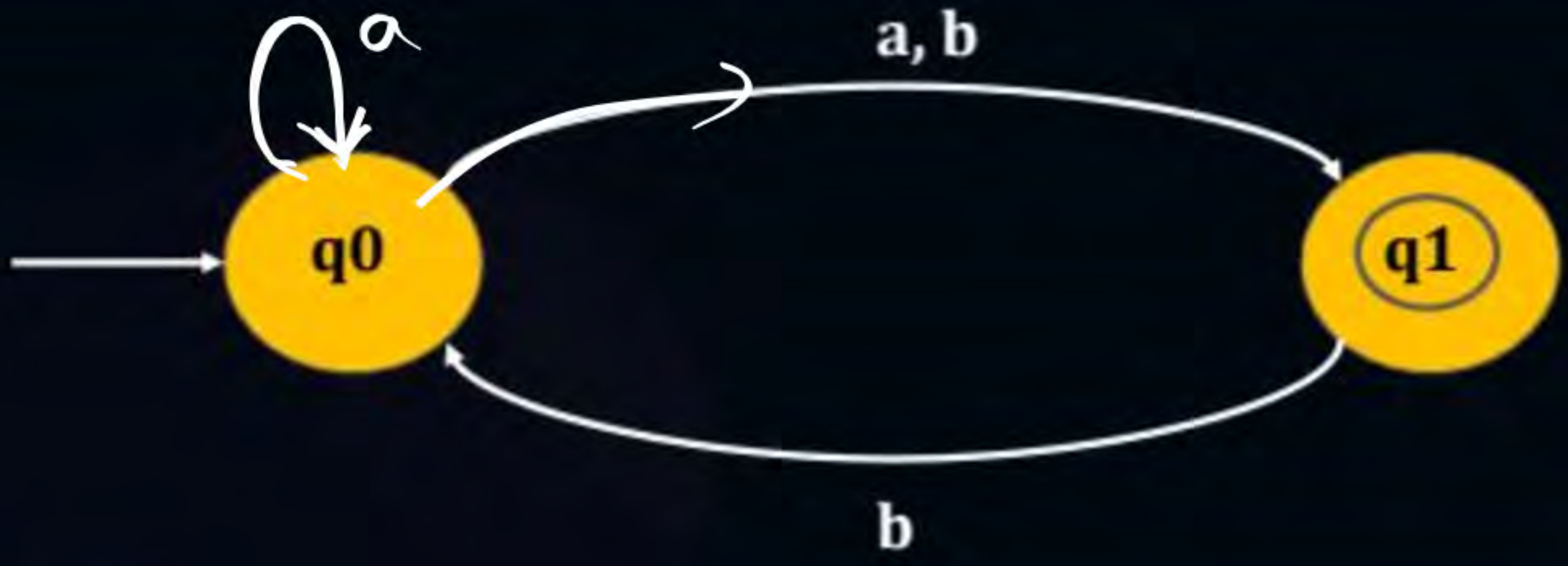
not a DFA.

Topic : Deterministic Finite Automata

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

Example of ~~[illegible]~~ : not DFA ?

(2)

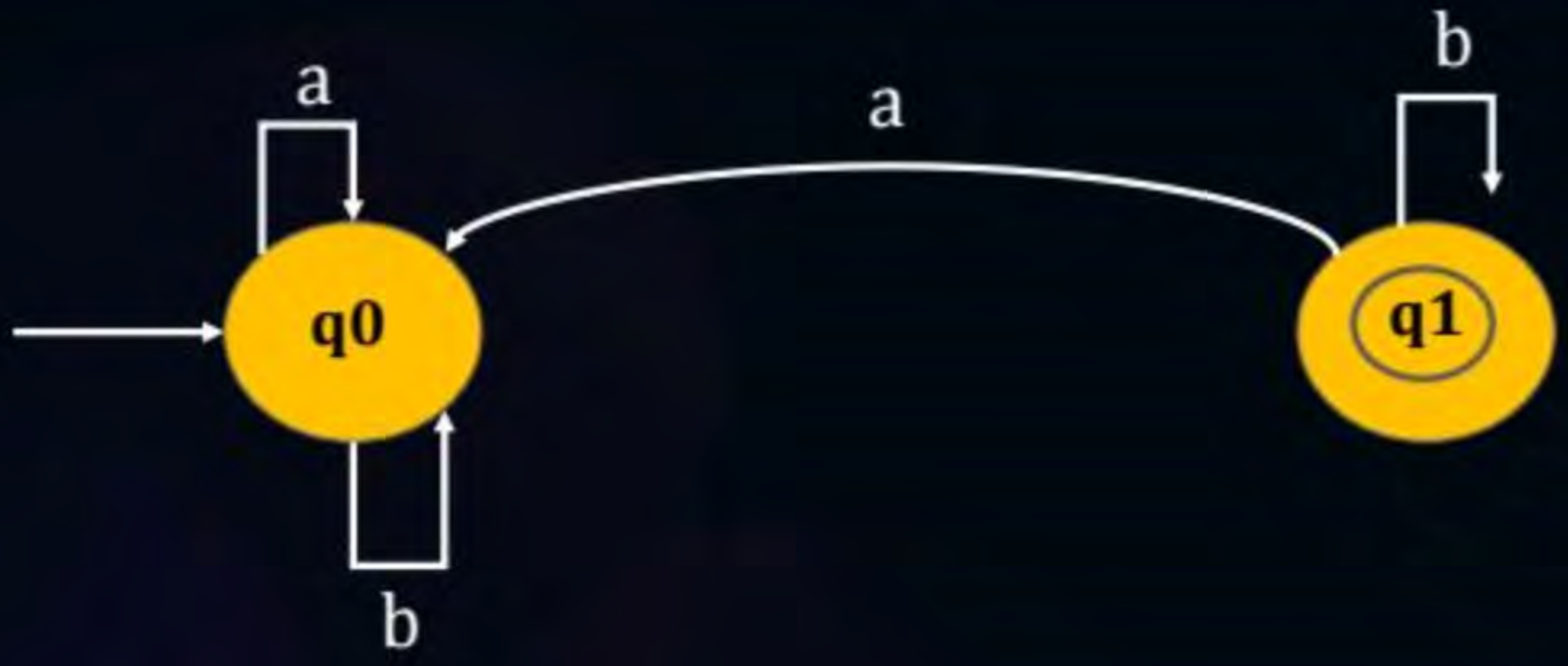


$\delta(q_0, a) = q_0 \text{ (or) } q_1$

Topic : Deterministic Finite Automata

Example of DFA :

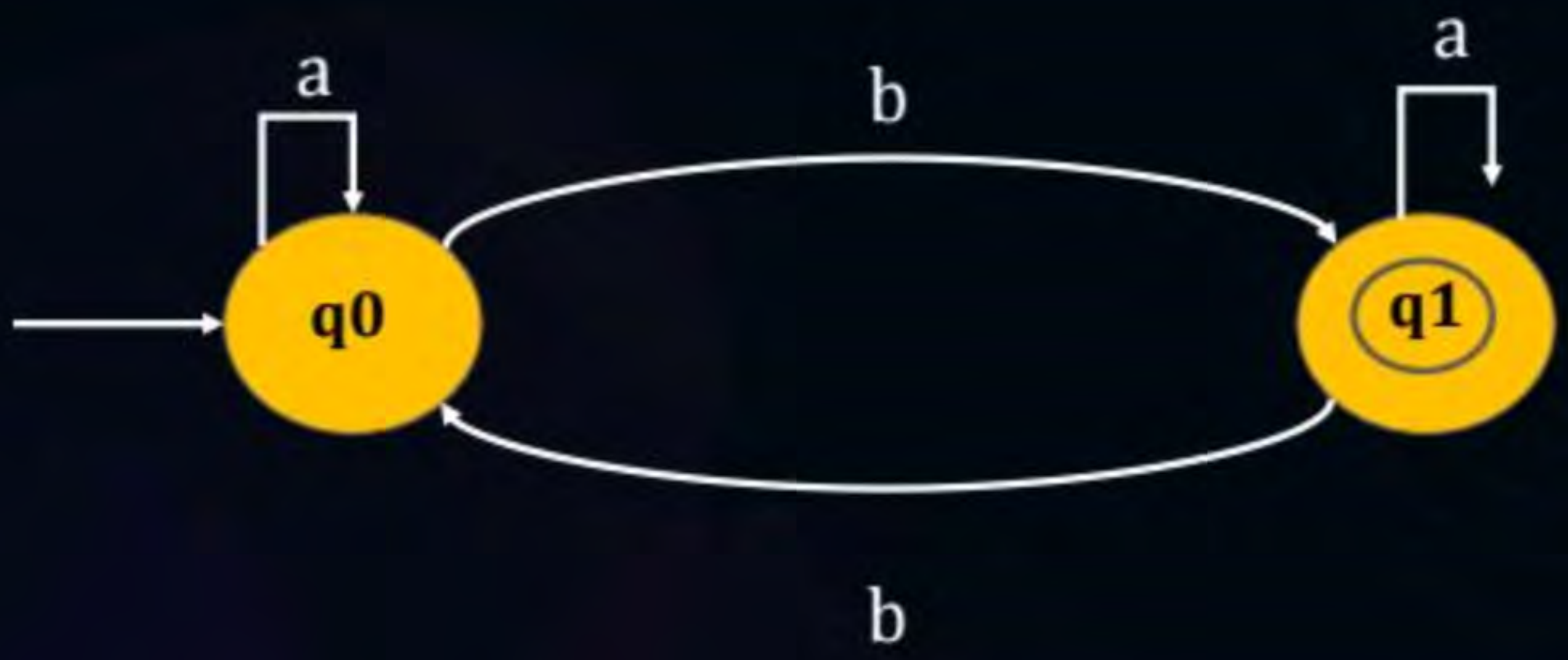
(3)



Topic : Deterministic Finite Automata

Example of DFA :

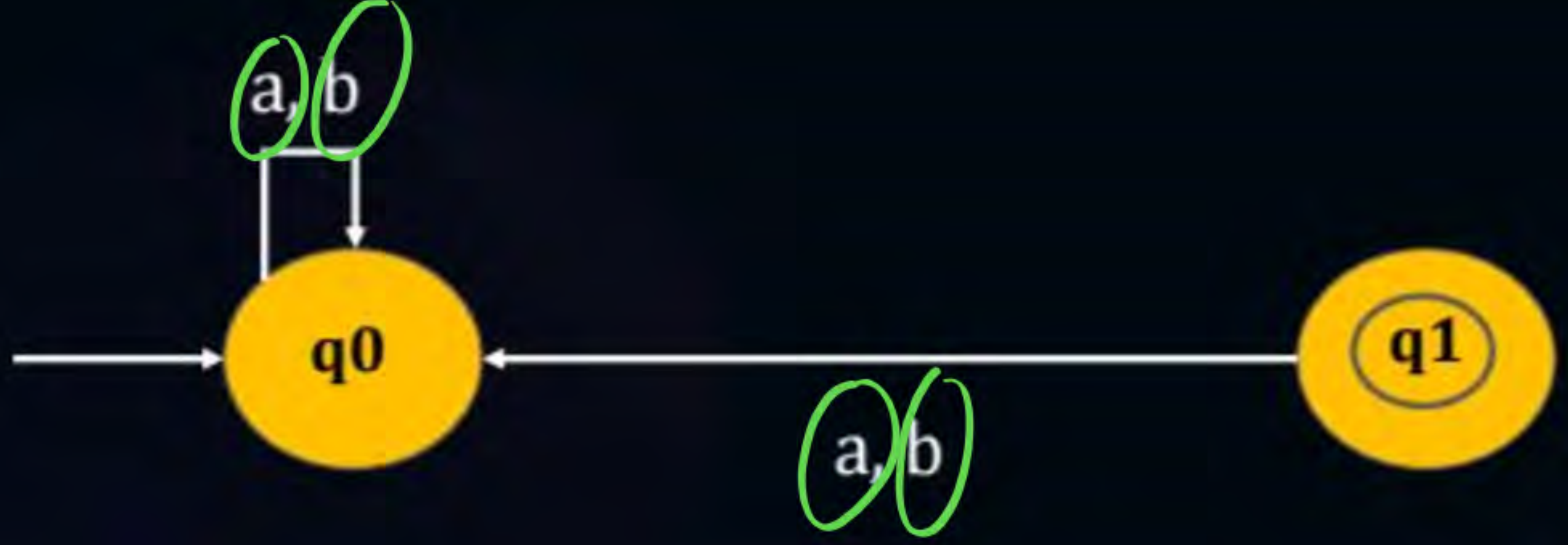
(4)



Topic : Deterministic Finite Automata

Example of DFA :

(5)



DFA ?

Q

Σ

q_0

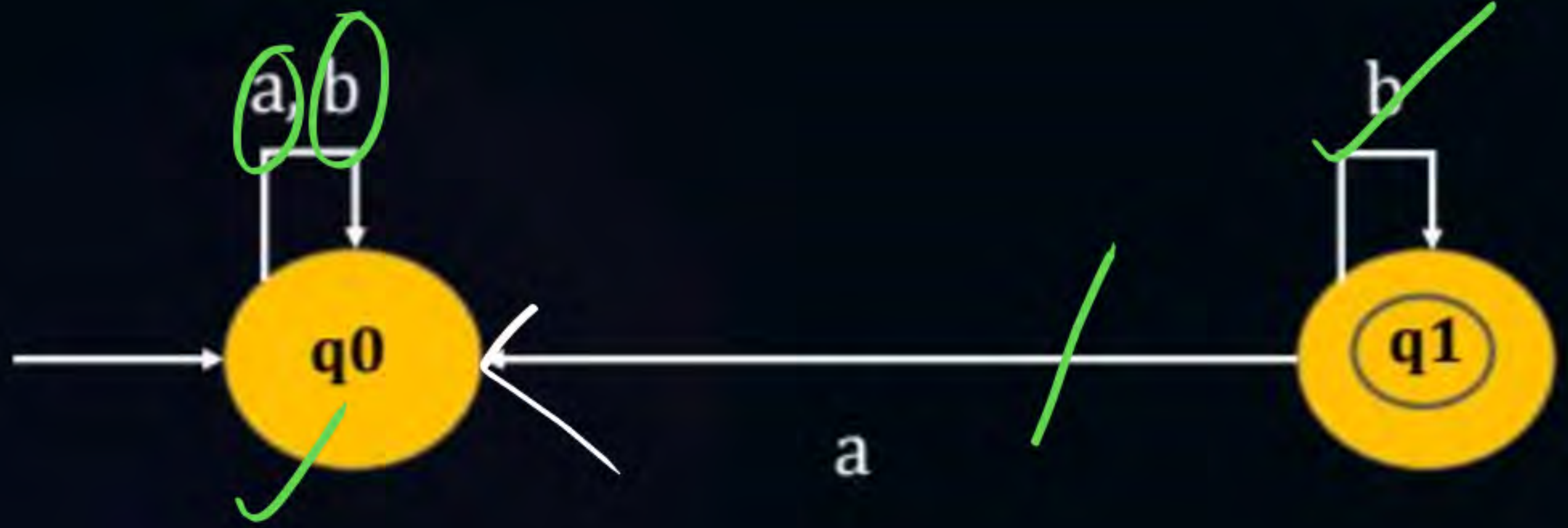
F

δ :

DFA ?

Example of DFA :

(6)



Q :

Σ :

q_0 :

F :

δ :

DFA ?

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

Example of :

(7)

Q

Σ

q_0

F : any number

δ

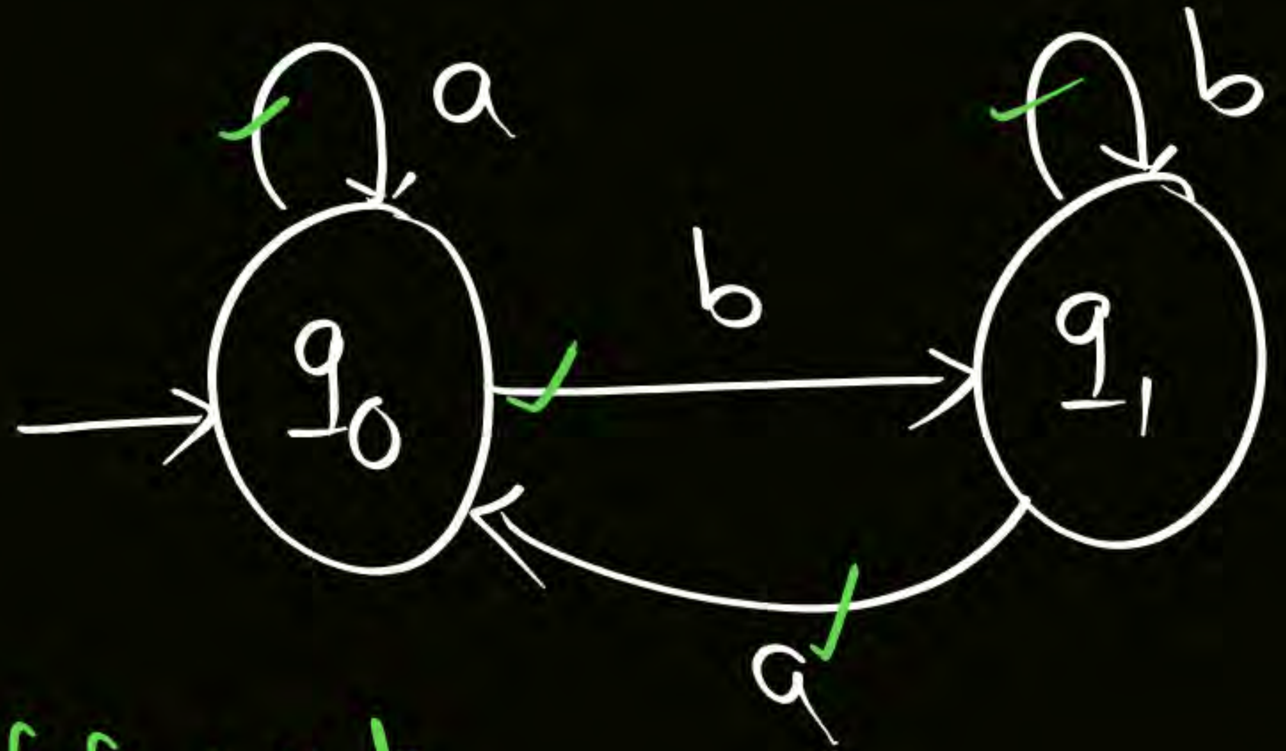
Q :

Σ :

q_0 : one

F : any no. of final states

$\delta : \underline{Q \times \Sigma \rightarrow Q}$

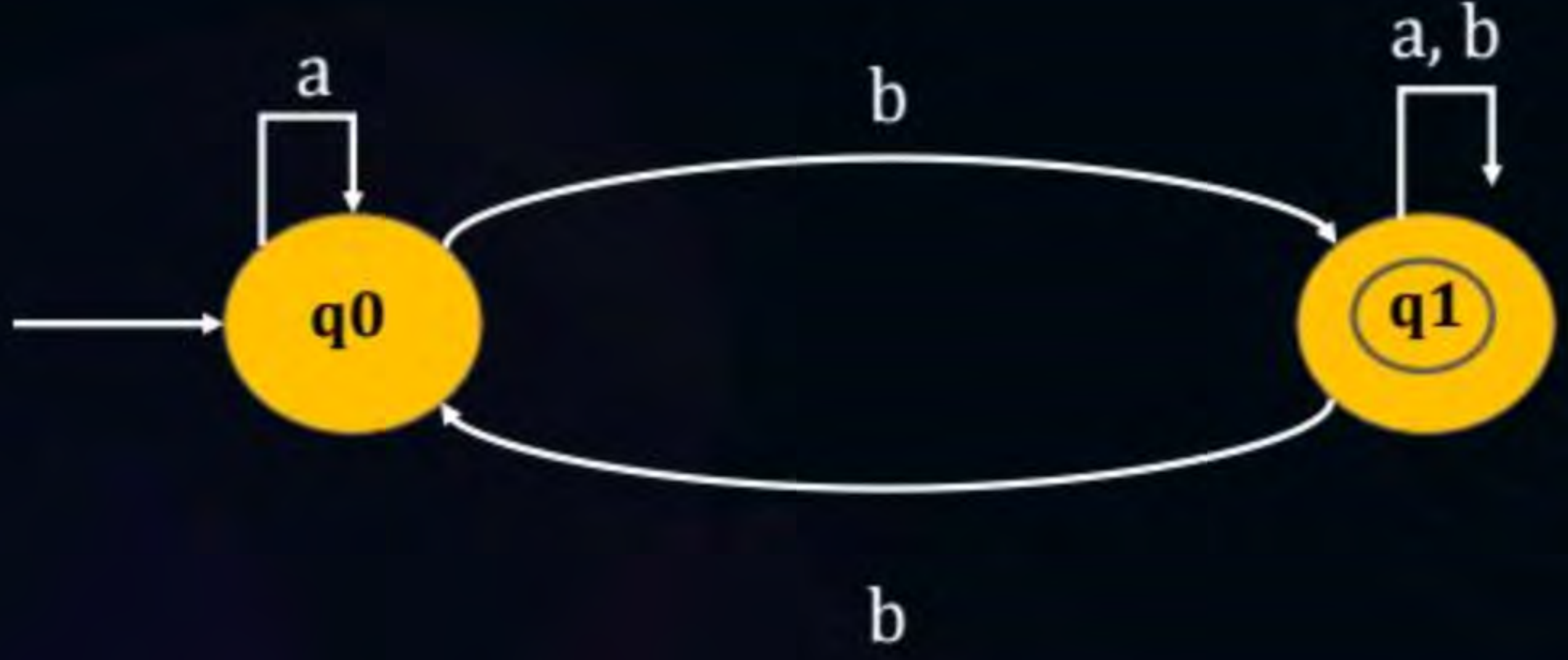


DFA

Topic : Deterministic Finite Automata

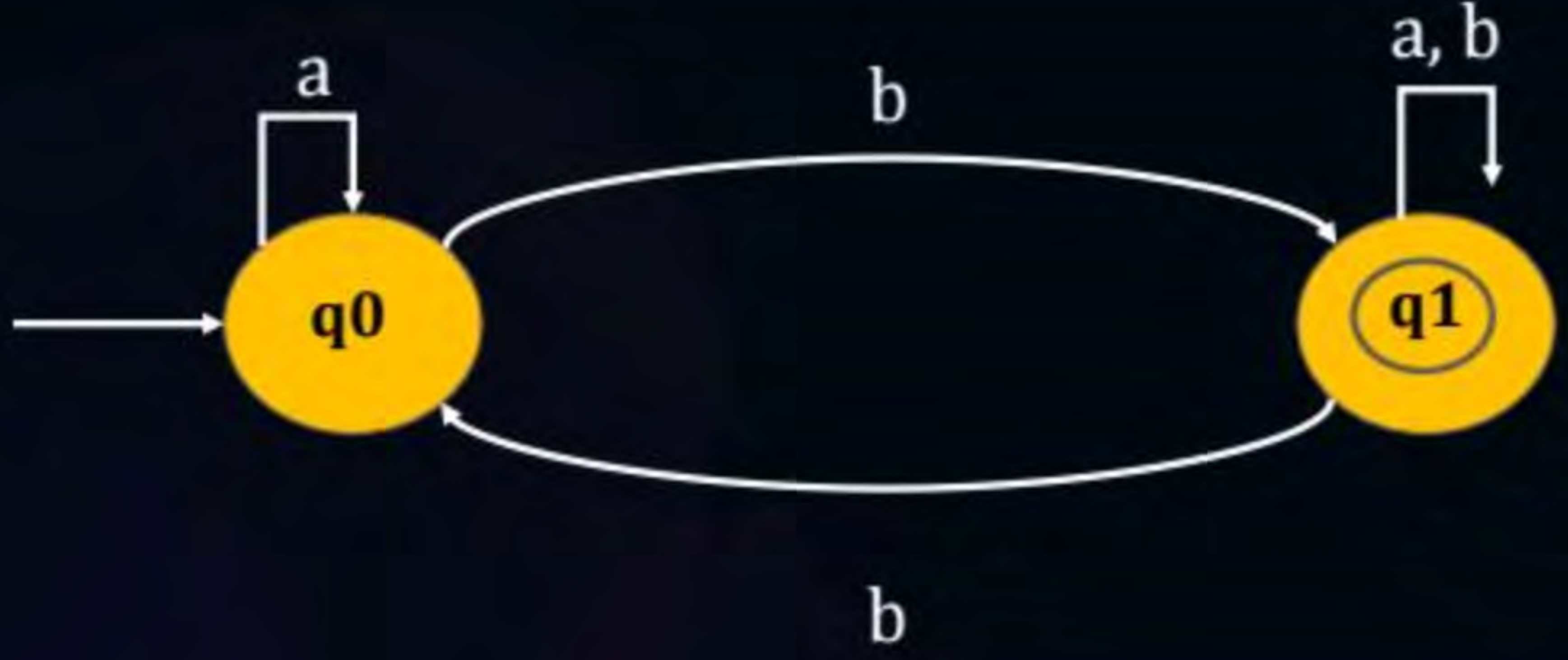
Example of DFA :

(8)



Example of DFA :

(8)



DFA acceptance method:

1. **Start at the Initial State**: Begin at the initial state of the DFA.

2. **Read Input Symbols**: For each symbol in the input string, read it one by one.

3. **Follow Transitions**: Based on the current state and the input symbol being read, follow the transition defined by the transition function of the DFA. This transition function specifies the next state of the automaton for each combination of current state and input symbol.

4. **Repeat Until End of Input**: Continue this process of reading input symbols and following transitions until you reach the end of the input string.

5. **Final State**: Once you have processed all input symbols, check the current state of the DFA. If it is one of the accepting states (states designated as final states), then the input string is accepted. Otherwise, it is rejected.

6. **Acceptance**: If the DFA halts in an accepting state after reading the entire input string, then the Input is accepted.

#Q. Identify language accepted by given DFA

Set of all strings

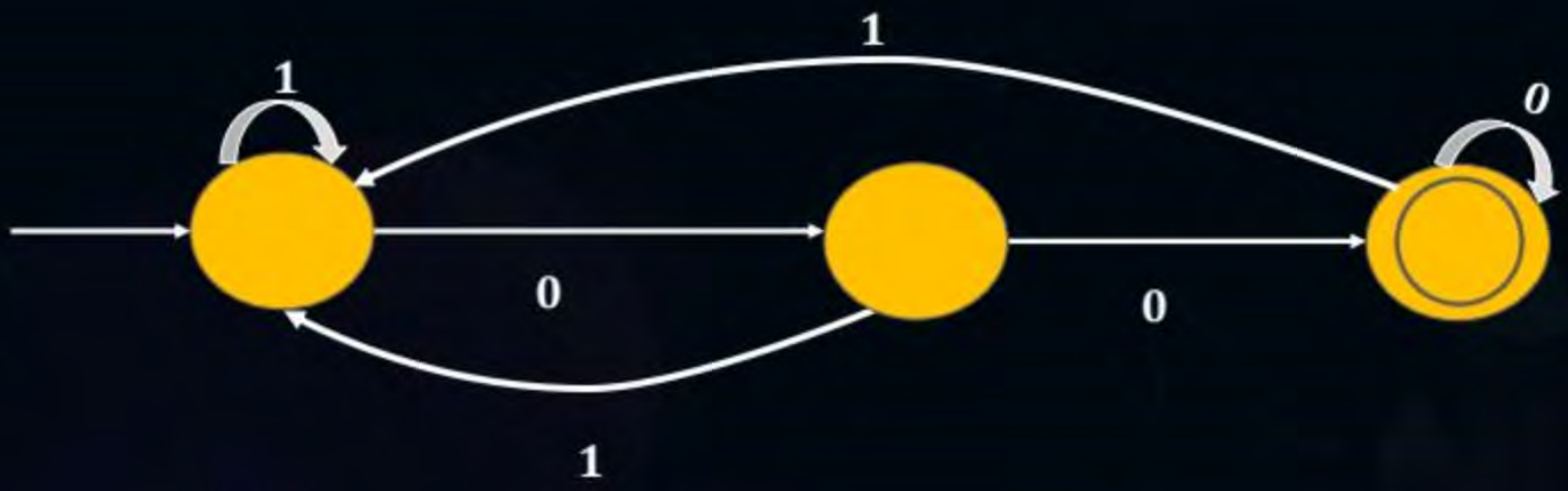
A Starting with bb

B Ending with bb

C Contains at least 2 b's

D None

#Q. Identify the language accepted by following DFA



A Starting with 00

B Substring 00

C Ending with 0

D None

#Q. Identify language accepted by following DFA

A $L = \{a^n b^m \mid n, m \geq 1\}$

B $L = \{a^n b^m \mid n \geq 1, m \geq 0\}$

C $L = \{a^n b^m \mid n, m \geq 0\}$

D None

#Q. Identify language accepted by following DFA

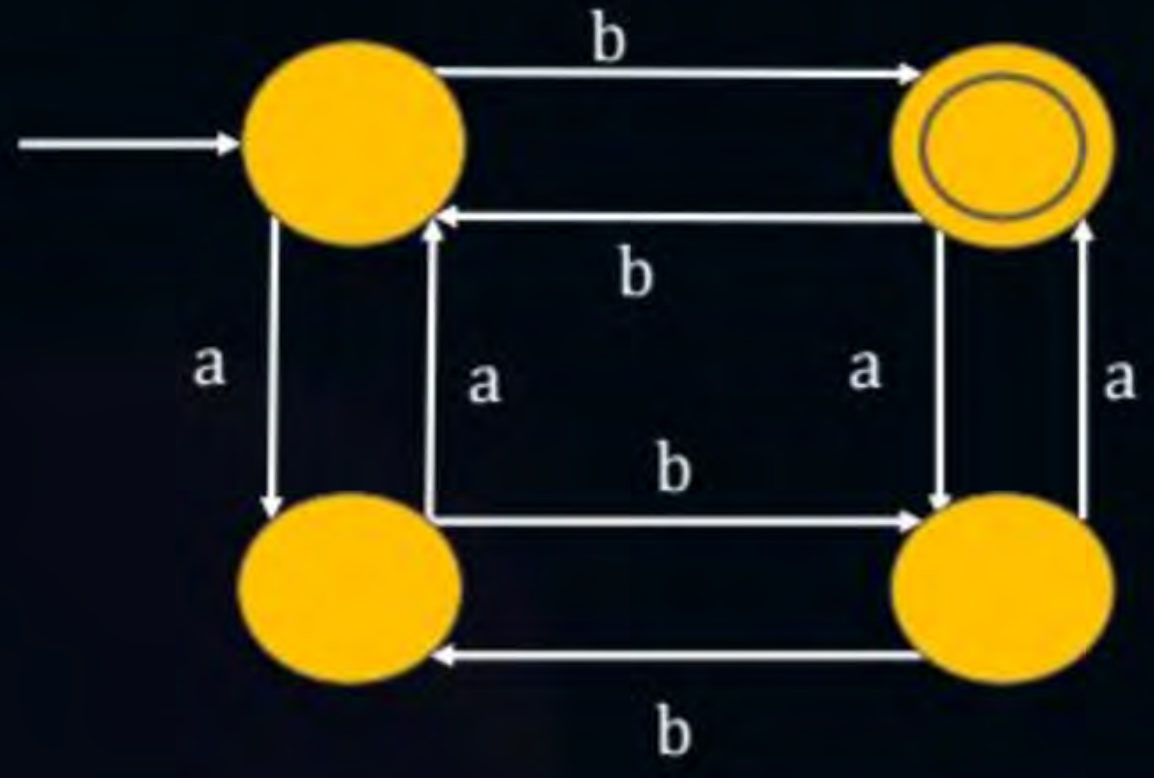
A $L = \{a^n b^m \mid n, m \geq 1\}$

B $L = \{a^n b^m \mid n \geq 1, m \geq 0\}$

C $L = \{a^n b^m \mid n, m \geq 0\}$

D None

#Q. Identify language accepted by following DFA



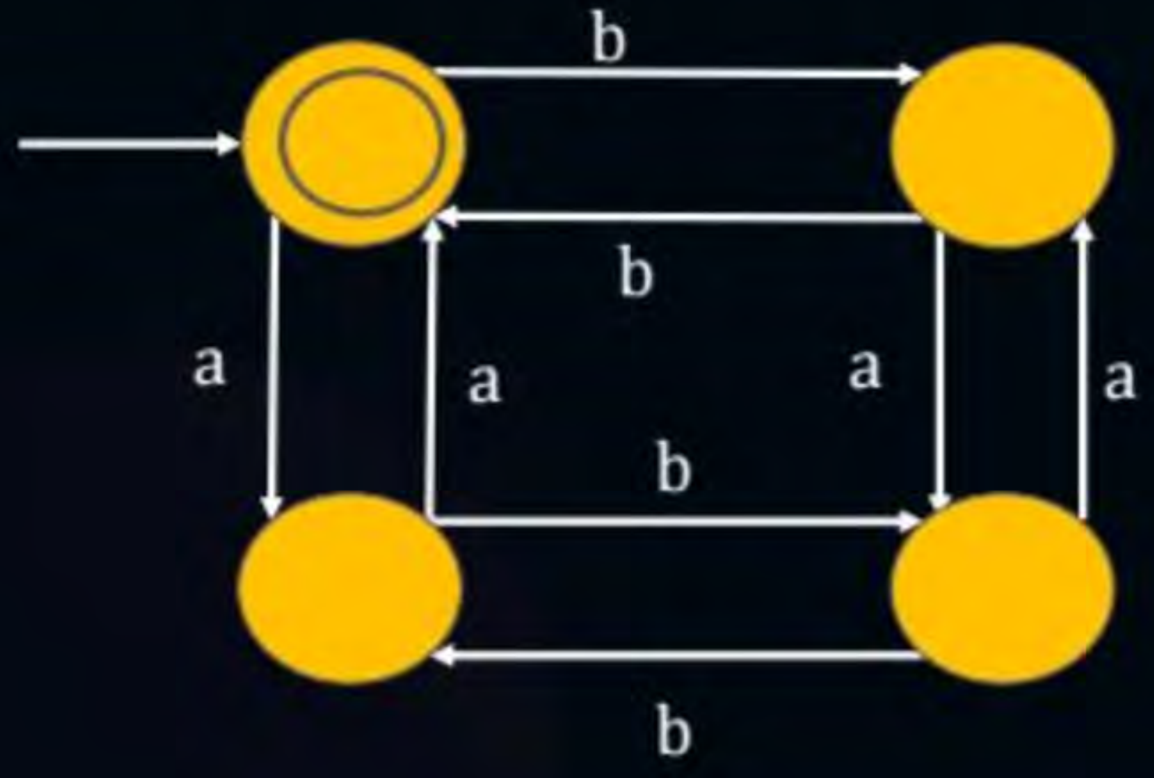
A # a's even and # b's even

B # a's odd and # b's even

C # a's odd and # b's odd

D # a's even and # b's odd

#Q. Identify language accepted by following DFA



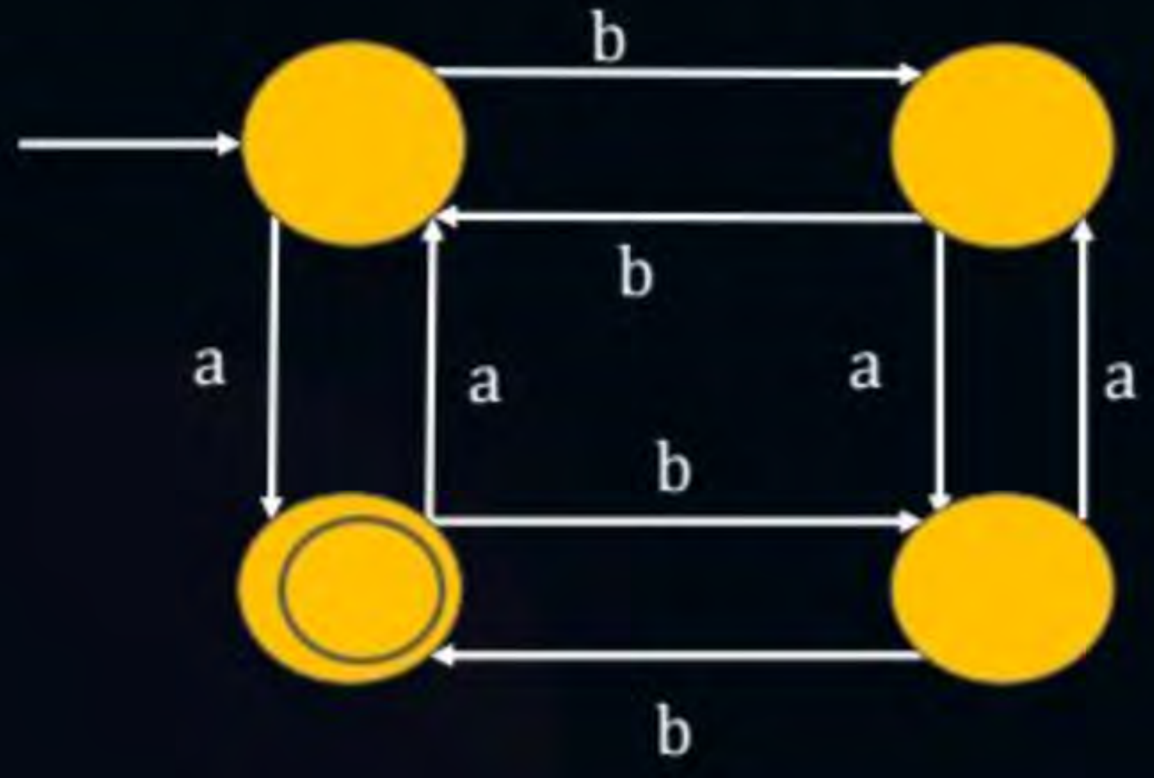
A # a’s even and # b’s even

B # a’s odd and # b’s even

C # a’s odd and # b’s odd

D # a’s even and # b’s odd

#Q. Identify language accepted by following DFA



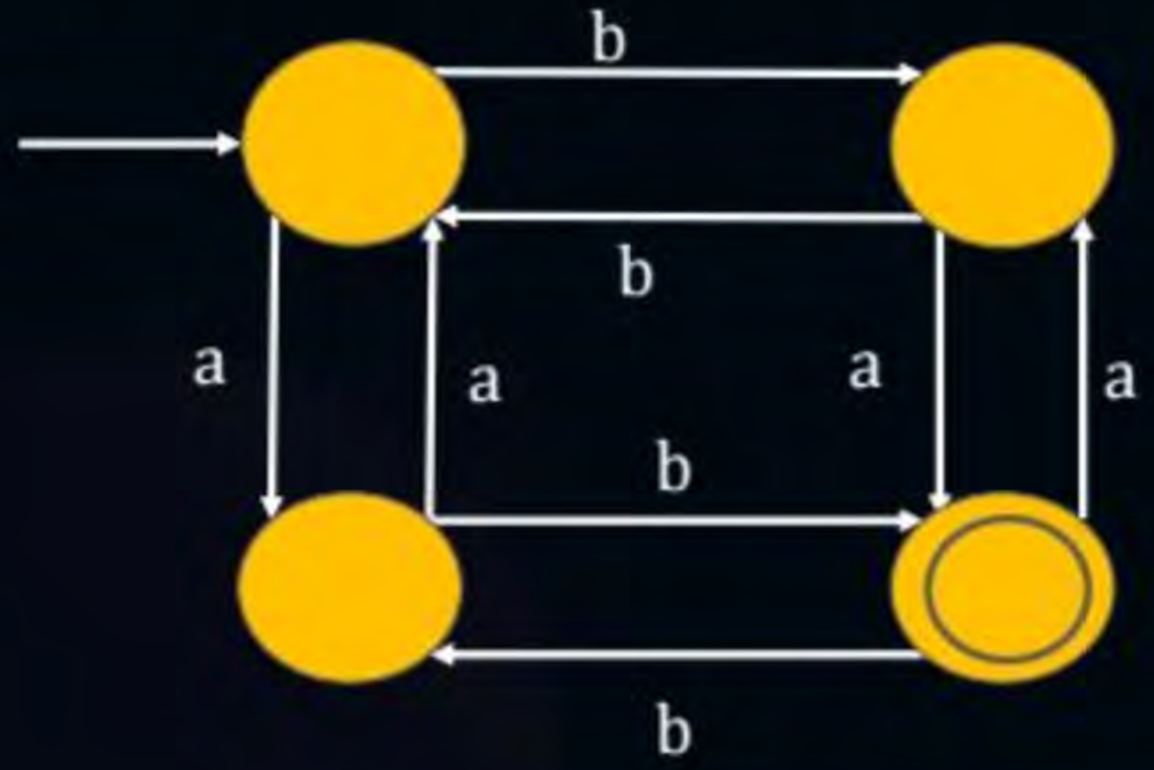
A # a's even and # b's even

B # a's odd and # b's even

C # a's odd and # b's odd

D # a's even and # b's odd

#Q. Identify language accepted by following DFA



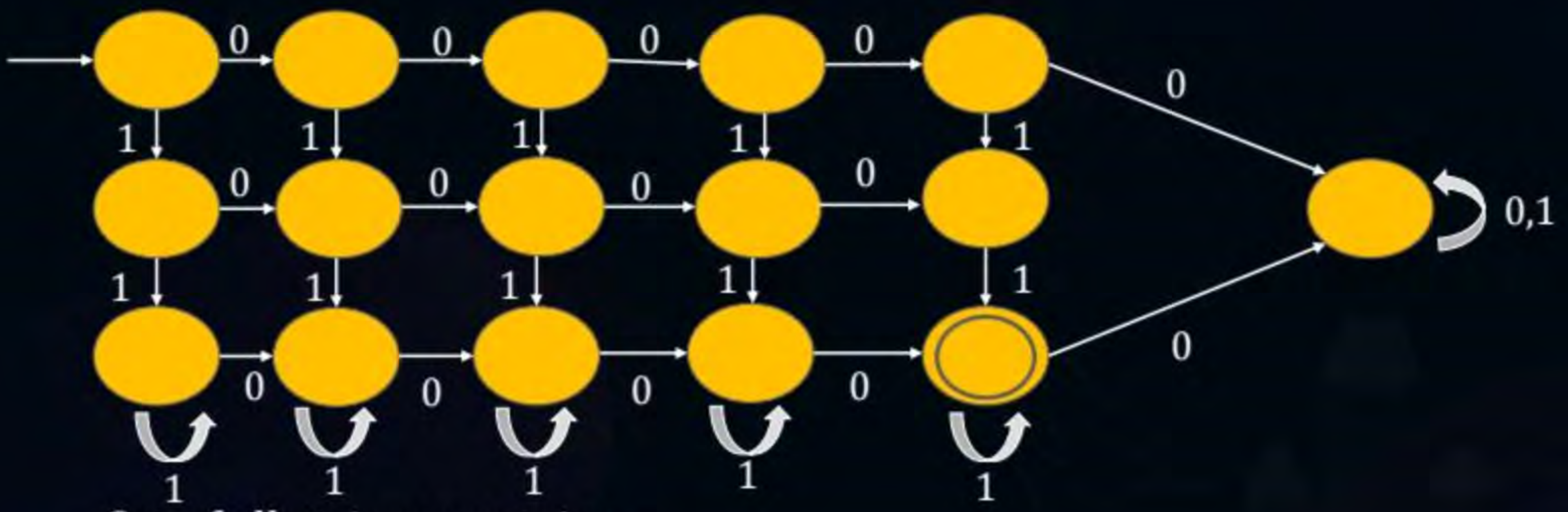
A # a's even and # b's even

B # a's odd and # b's even

C # a's odd and # b's odd

D # a's even and # b's odd

#Q. Identify language accepted by following DFA



Set of all strings contains

A Length of the string alteast 6

B # 0's exactly4 and 1's atleast 3

C # 0's atleast 4 and # 1's exactly 2

D None

MCQ

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

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

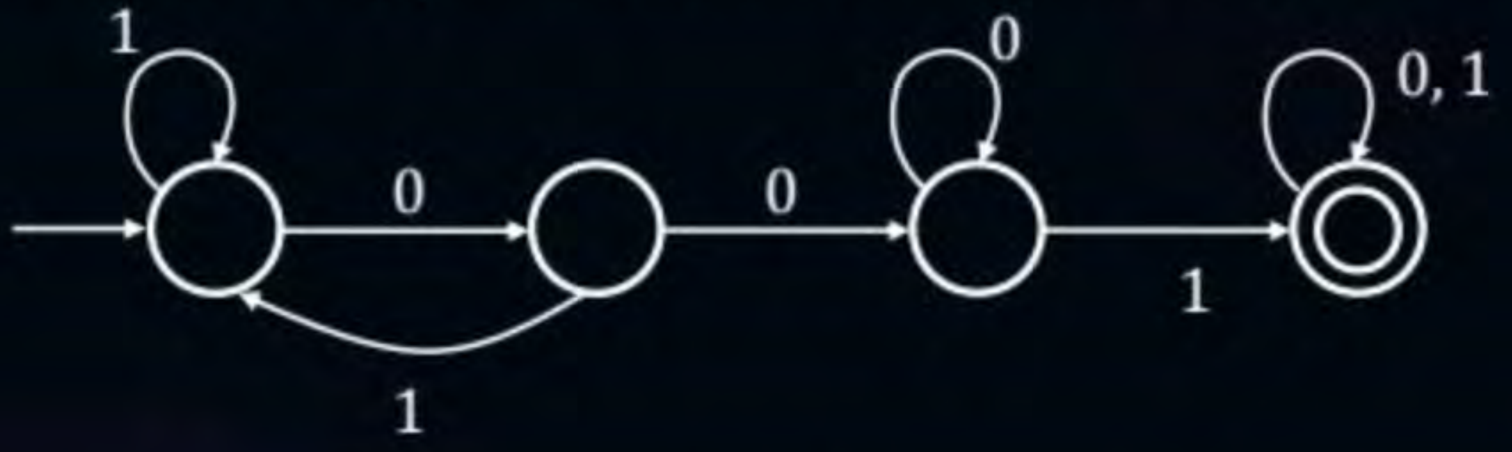
B $0^+ + 1(11)^* + 0(11)^*$

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

D $(00)^* + 1(11)^*$

MCQ

#Q. Consider the following deterministic finite state automaton M.



Let S denote the set of seven-bit binary strings in which the first, the fourth, and the last bits are 1. The number of strings in S that are accepted by M is

A. 1

B. 5

C. 7

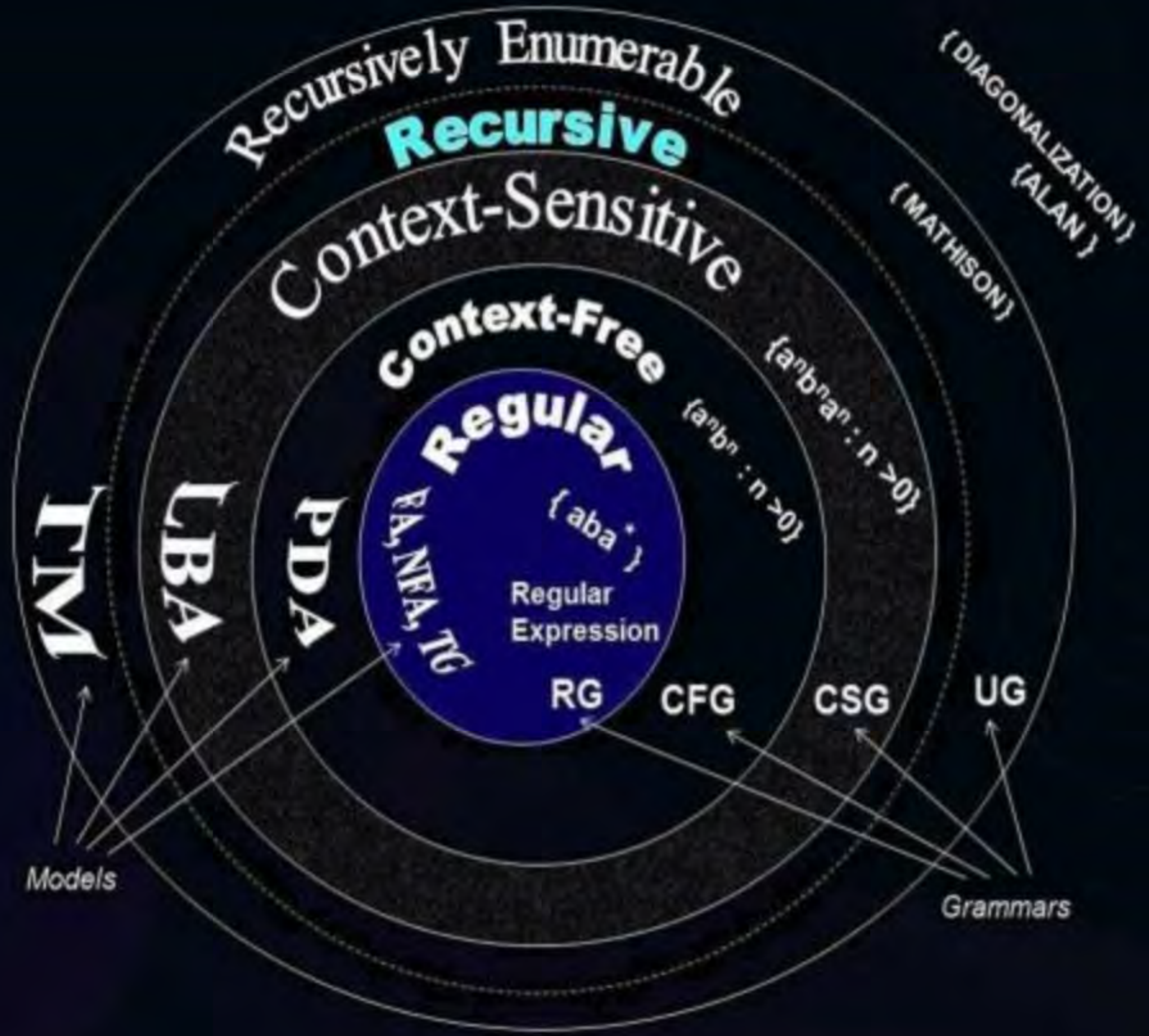
D. 8

Topic : DFA Construction

Construct DFA for the following Language.

1. $L = \{a^n b^m \mid n,\ m \geq 1\}$
2. $L = \{a^n b^n \mid n \geq 1\}$
3. $L = \{a^n b^m \mid n < m\}$
4. $L = \{a^n b^m \mid n \neq m\}$
5. $L = \{a^n b^m c^{n+m} \mid n,\ m \geq 1\}$
6. $L = \{a^n b^{2m} \mid n,\ m \geq 1\}$

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

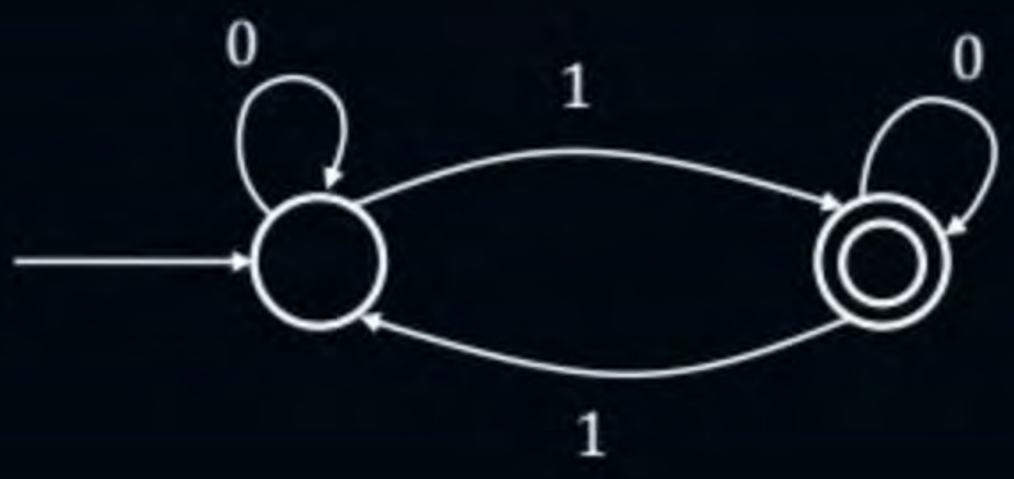
B M_1 is non deterministic turing machine

C M_1 Is non deterministic PDA

D None

MCQ

#Q. Which one of the following regular expressions is equivalent to the language accepted by the DFA given below?



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

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

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

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

THANK - YOU