INTRODUCTION TO THEORY OF COMPUTATIONS

- As word suggests 'TOC' is the study of 'mathematical' machines or systems called automata and the computational problems that can and cannot be solved using these machines, i.e. what is the extent to which a problem is solvable on a computer. Theory of computation can be considered as the study of all kinds of computational model in the field of computer science and it also considers how efficiently the problem can be solved (but not is depth).

PROBLEM

- Now a day's machines (digital, analog, mechanical) play a very important role in the development of human, we need some mechanism (language) to communicate with the machines.

SOLUTION

- We need a language for communication with machines.
- But we do not require natural languages to communicate with the machines, as natural languages are very complex and machine interaction require very fewer complex languages compare to natural languages.
- Languages can be of two types formal languages and informal languages, here in this subject we will only discuss formal languages.
- Dictionary defines the term informally as 'a system suitable for the expression of certain ideas, facts or concepts including a set of symbols for their manipulation'.

MATHEMATICL DEFINATION OF LANGUAGE

SYMBOL- Symbols are the basic building blocks, which can be any character/token. (cow, sheep, sun, white flag etc.) (in English we called them as letters).

ALPHABET- An alphabet is a finite non empty set of symbols, (every language has its own alphabet). here in toc, we use symbol Σ for depicting alphabet. e.g. $\Sigma = \{0,1\}$. for English $\Sigma = \{a, b, c, …., z\}$ (in English also alphabet is a set of letters, thought in general we called them as alphabet).

STRING - It is a sequence of symbols (which are the member of set alphabet). E.g. $\Sigma = \{a, b\}$ String- aabb, aa, b, so on. (in English we called them as words).

LANGUAGE - A language is defined as a set of strings. (in natural language (set of words(predefined) and grammar) we apply this model from words to sentence).

- In the next level we consider programs as a string and programming constructs/tokens like int, floats as letters/symbols.

METHODS TO DEFINE LANGUAGE

- In natural language we define the list of words in a dictionary because they are finite and predefined, but we cannot list all the sentence which can be formed using these words as they are infinite. So, we have a mechanism called grammar/rules using which we can decide which sentence is valid and which is invalid.
- Similarly, in our system we have finite number of symbols/letters but using those letters we can generate infinite strings/words. So, we may have languages that have infinite number of words, so it is not possible for us to list them, we have to use some framework, which can somehow represent the same language.
    - There are mainly two methods to represent a language
    - by a grammar that generates a language [CFG generate CFL]
    - by a machine that accepts a language [PDA accept CFL language]

SOME BASIC OPERATIONS ON STRINGS

So, before we proceed further let's do a little home work on string, which will help us throughout the subject.

Length of a string - It is defined as number symbol in the string. Denoted like |W|, e.g. length of string |00110| = 5.

Concatenation of string- Let x and y be two strings, then concatenation is defined as the string formed by making a copy of string x followed by a copy of string y. (NOTE- It's not commutative)

E.g. w = ab, x= ba

wx = abba

xw = baab

$w = w_1w_2w_3......w_m$ $x = x_1x_2x_3......x_n$

$wx = w_1w_2w_3......w_mx_1x_2x_3......x_n$

$|wx| = |w| + |x|$

Reverse of a string – if there is a string w then reverse of a string a denoted by w^r it is just the same string but written in reverse order.

e.g. $W = w_1w_2w_3......w_n$

$w^r = w_n......w_3w_2w_1$

$|w|=|w^r|$

Empty/Null String- The string with zero occurrence of symbols. It is denoted by ∈, |∈|=0. If there is a string w, then w^n stands for the strings obtained by repeating w, n times.

$w^1 = w$

$w^2 = ww$

$w^3 = www$

$w^0 = \varepsilon$

$w\in = \in w = w$

Substring- Any string of consecutive symbols in some string 'w' can be collectively said as a substring. E.g. w= abab its substrings can be ab, a, ba...etc.

Q Consider a sting 'GATE' find the total number of substring possible?

	Substring of length 0	Substring of length 1	Substring of length 2	Substring of length 3	Substring of length 4
Number of sting possible	1	4	3	2	1
SUBSTRINGS	∈	G, A, T, E	GA, AT, TE	GAT, ATE	GATE

- If w is any string than empty string ∈ and the string w itself is called a trivial substring and the remaining of the other are the non-trivial sub string[[n(n+1)/2]-1]

Q Consider a sting 'GGGE' find the total number of substring possible?

- If a string has 'n' distinct symbols then total number of different sub string will be [n(n+1)/2] +1
- If there is a string of length n then no of prefix or suffix will be n+1

Q Consider a sting 'GATE' find the total number of prefix and suffix possible?

- if symbols are in order but not necessarily consecutive, then it is called subsequence.

Q if ∑ = {a, b} then, find the following?

$\sum^0$ =

$\sum^1$ =

$\sum^2$ =

$\sum^3$ =

$\sum^K$ is the set of all the strings from the alphabet ∑ of length exactly K.

$\sum^k$ = {W | |W| = K} (using the symbols from the alphabet ∑)

Kleene closure- If ∑ is a set of symbols, then we use ∑* to denote the set of strings obtained by concatenating zero or more symbols from ∑ of any length, in general any string of any length which can have only symbols specified in ∑.

$\sum^* = \bigcup_{i=0}^{i=\infty}\{w \mid |w| = i\}$ (using the symbols from the alphabet ∑)

Positive closure – If ∑ is a set of symbols, then we use $\sum^+$ to denote the set of strings obtained by concatenating one or more symbols from ∑ of any length, in general any string of any length which can have only symbols specified in ∑ (except ∈).

$\sum^* = \bigcup_{i=0}^{i=\infty}\{w \mid |w| = i\}$ (using the symbols from the alphabet ∑)

it is denoted by $\sum^+ = \sum^* - \in$

LANGUAGES

Since languages are sets, the union, intersection and difference of two languages are immediately defined.

- The complement of a language is defined with respect of ∑*
  - $L^c = \Sigma^* - L$
- The reverse of a language is the set of all the strings after reversal
  - $L^R = \{w^r \mid w \in L\}$
- The concatenation of two languages L_1 and L_2 is the set of all the strings obtained by concatenating any elements of L_1 and L_2.
  - $L_1L_2 = \{xy \mid x \in L_1, y \in L_2\}$
- Let L^n is defined as L is concatenated with itself n times.
  - $L^n = \{LLL\ldots\ldots\ldots L\}$ (n times)
  - $L^* = L^0 \cup L^1 \cup L^2 \cup \ldots\ldots \cup L^\infty$

Q Given the language L = {ab, aa, baa}, which of the following strings are in L*? **(GATE-2012) (1 Marks)**

1) abaabaaabaa **2)** aaaabaaaa **3)** baaaaabaaaab **4)** baaaabaa

(A) 1, 2 and 3 **(B)** 2, 3 and 4 **(C)** 1, 2 and 4 **(D)** 1, 3 and 4

Answer: (C)

Q The number of substrings that can be formed from string given by 'adefbghnmp' is? **(NET-DEC-2018)**

a) 10 **b)** 45 **c)** 55 **d)** 56

Ans: d

Q In a string of length n, how many proper prefixes can be generated

a) 2^n **b)** n **c)** n (n + 1)/2 **d)** n – 1

Q The number of substrings (of all lengths inclusive) that can be formed from a character string of length n is

a) n! **b)** n^2 **c)** $\frac{n(n-1)}{2}$ -1 **d)** $\frac{n(n+1)}{2}$ +1

Ans: d

Q Suppose L_1 = {10, 1} and L_2= {011, 11}. How many distinct elements are there in L = L_1L_2

a) 4 **b)** 3 **c)** 2 **d)** None of these

Q Let x_n denote the number of binary strings of length n that contain no consecutive 0^s, which of the following recurrences does x_n satisfy?

a) $x_n = 2x_{n-1}$ **b)** $x_n = x_{\lceil \frac{n}{2} \rceil} + 1$

c) $x_n = x_{\lceil \frac{n}{2} \rceil}$ + n **d)** $x_n = x_{n-1} + x_{n-2}$

Ans: d

Q The value of x_5 is?

a) 5 **b)** 7 **c)** 13 **d)** 16

Ans: c

Q Match the following

1. Maximum number of prefixes for the n length string	A. 2 n – 1
2. Maximum number of proper suffixes for n length string	B. n
3. Maximum number of proper non-empty sub strings for n length string	C. $\frac{n(n+1)}{2} - 1$
4. The number of strings of length n over the alphabet {0, 1}	D. n + 1
	E. n – 1
	F. 2n

a) 1-A, 2-B, 3-D, 4-F

b) 1-B, 2-C, 3-E, 4-A

c) 1-D, 2-B, 3-C, 4-F

d) 1-E, 2-F, 3-B, 4-C

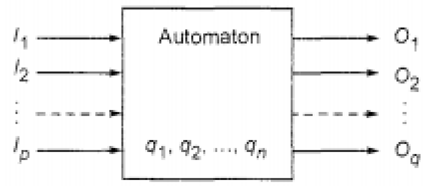
Ans: c

Automaton

An automaton is defined as a system where energy, materials and information are transformed, transmitted and used for performing some functions without direct participation of man.

Example: automatic machine tools, automatic packing machines, and automatic photo printing machines.

- Below is the figure of a discrete automaton:



The characteristics of automaton are now described as:

- **Input.** At each of the discrete instants of time t1, t2, ... tm the input values I1, I2.... Ip each of which can take a finite number of fixed values from the input alphabet ∑, are applied to the input side of the model.

- **Output.** O_0, O_1 … Oq are the outputs of the model, each of which can take a finite number of fixed values from an output O.

- **States**. At any instant of time the automaton can be in one of the states ql, q2, ... qn

- **State relation.** The next state of an automaton at any instant of time is determined by the present state and the present input.

- **Output relation.** The output is related to either state only or to both the input and the state. It should be noted that at any instant of time the automaton is in some state. On 'reading' an input symbol, the automaton moves to a next state which is given by the state relation.

Some definitions of various automatons

- An automaton in which the output depends only on the input is called an automaton without a memory.
- An automaton in which the output depends on the states as well is called automaton with a finite memory.
- An automaton in which the output depends only on the states of the machine is called a Moore machine.
- An automaton in. which the output depends on the state as well as on the input at any instant of time is called a Mealy machine.

FINITE AUTOMATA

A **Finite automaton** is a model that has a finite set of states (represented in the figure by circles) and its control moves from one state to another state in response to external inputs (represented by arrows).

Finite automata can be broadly classified into two types-

- **Finite automata without output**
  - Deterministic finite automata.
  - non deterministic finite automata.
  - Non deterministic finite automata with ∈
- **Finite automata with output**
  - Moore machine
  - Mealy machine

In general, this type of automata is characterized by machine having no temporary storage, as it is severely limited in its capacity to remember things during the computation.

Only finite amount of information can be in the control unit by placing the unit into a specific state but since the number of states is finite, a finite automaton can only deal with situation in which the information to be stored at any time is strictly bounded.

DETERMINISTIC FINITE AUTOMATA

A **deterministic finite automaton (DFA)** is defined by 5-tuple (Q,Σ,δ,S,F) where:

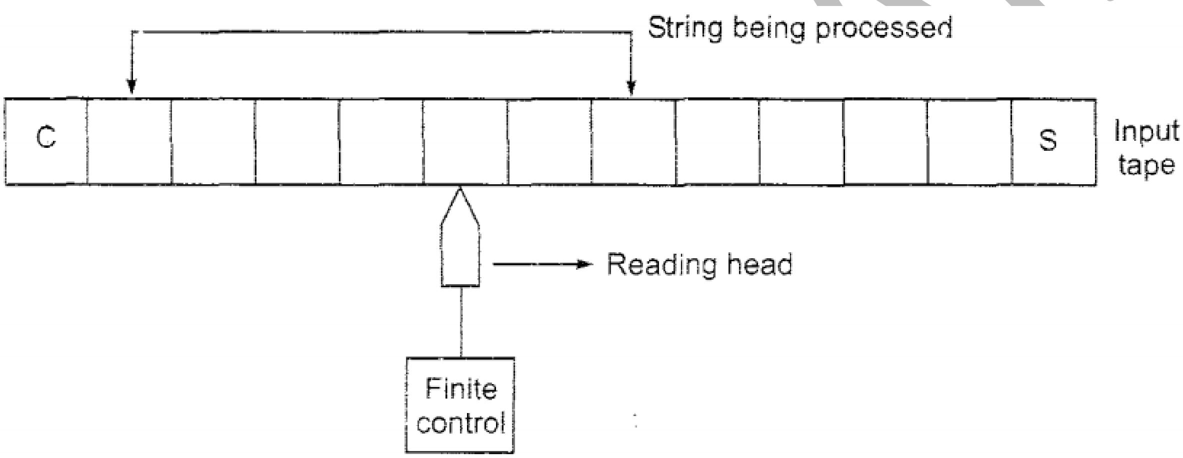
Q is a finite and non-empty set of states

Σ is a finite non-empty set of finite input alphabet

δ is a transition function, ($\delta: Q \times \Sigma \rightarrow Q$)

S is **initial state** (always one) ($S \in Q$)

F is a set of final states ($F \subseteq Q$) ($0<=|F|<=N$, where n is the number of states)



The various components of the block diagram are explained as follows:

- **Input tape**: The input tape is divided into squares, each square containing a single symbol from the input alphabet ∑. The end squares of the tape contain the end marker ¢ at the left end and the end marker $ at the right end. The absence of end markers indicates that the tape is of infinite length. The left-to-right sequence of symbols between the two end markers is the input string to be processed.
- **Reading head (R-head)**: The head examines only one square at a time and will move one square to the right.
- **Finite control:** is the inference engine take care of transition

NOTE-

- Produces a unique computation (or run) of the automaton for each input string.
- Deterministic refers to the uniqueness of the computation.
- DFAs are useful for doing lexical analysis (Spell check) in compiler design

Representation

TRANSITION STATE DIAGRAM- (Graphical, easy to understand, easy to design) A transition graph or a transition system is a finite directed labelled graph in which each circle represents a state and the directed edges indicate the transition of one state to another state. The initial state is represented by a circle with an arrow pointing towards it, the final state by two concentric circles.

TRANSITION TABLE- it is a two-dimensional table where number of columns is equal to number of input alphabets and number of rows is equal to number of states.

<table>
<tr><td colspan="2" rowspan="2">δ</td><td colspan="2">Σ</td></tr>
<tr><td>a</td><td>b</td></tr>
<tr><td rowspan="2">Q</td><td>q_0</td><td></td><td></td></tr>
<tr><td>q_1</td><td></td><td></td></tr>
</table>

TRANSITION ID- δ {q_i, a) = q_j, this means that on a state q_i take an input symbol a machine will make a transition q_j

ACCEPTANCE BY DFA

Let 'w' be any string designed from the alphabet Σ, corresponding to w, if there exist a transition for which it starts at the initial state and ends in any One of the final states, then the string 'w' is said to be accepted by the finite automata. $\delta(q_0, w) = q_f$ for some $q_f \in F$.

Mathematically, it can be represented as: - $L(M) = \{w \in \Sigma^* \mid \delta^*(S, w) \in F\}$

Q Consider the following table of an FA

δ	a	b
start	q_1	q_0
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_2
q_3	q_4	q_3
q_4	q_4	q_4

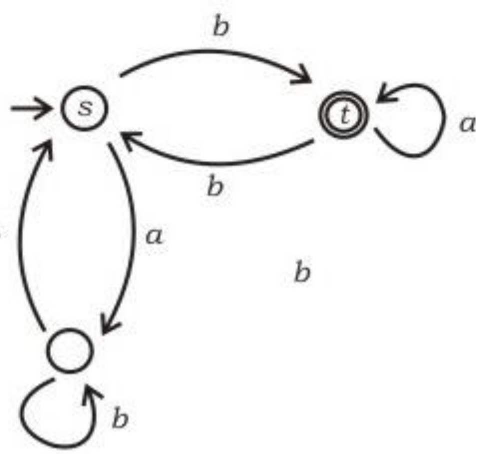
IF the final state is q_4, the which of the following strings will be accepted?

1. aaaaa **2.** aabbaabbbbb **3.** bbabababbb

a) 1 and 2 **b)** 2 and 3 **c)** 3 and 1 **d)** all of the above

Ans: d

Q In the automaton below, s is the start state and t are the only final state. **(GATE-2006) (2 Marks)**



Consider the strings u = ababab, v = bab, and w = aabb. Which of the following statements is true?

(A) The automaton accepts u and v but not w

(B) The automaton accepts each of u, v, and w

(C) The automaton rejects each of u, v, and w

(D) The automaton accepts u but rejects v and w

Answer: (D)

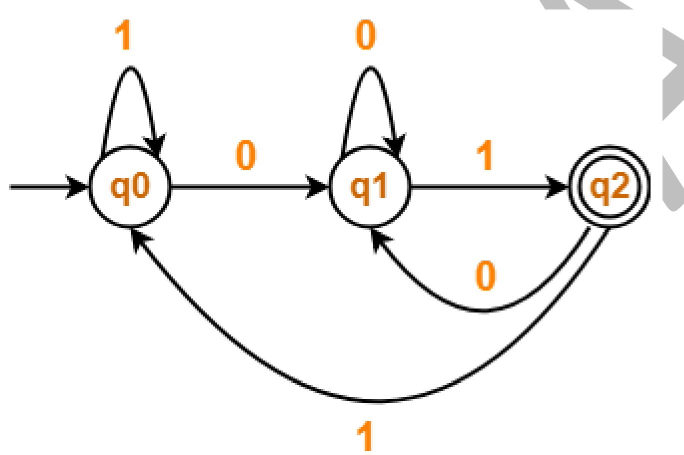
DFA Designing

A language a said to be regular language if it can be accepted by a DFA

The best knowledge about DFA can be only understood by designing a number of DFA, by doing so we will first understand the process of DFA designing and secondly, we will understand Regular Language.

Q Draw a DFA for the language accepting strings ending with '01' over input alphabets ∑ = {0, 1}?

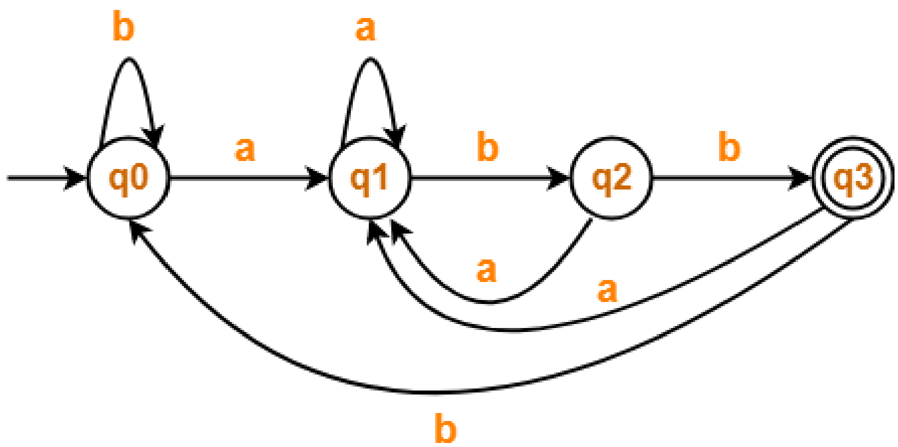
Ans. The required DFA for the above problem will be:



- Initially make a skeleton of the basic DFA ending with 01.
- Then fill up the required states accordingly.
- State q_0 can accept any number of 1's and if it sees 0 as a current input symbol it changes states to q_1, similarly for q_1 if it sees 1 as a current input it will immediately change state to q_2 else it will self-loop the 0's.
- q_3 acts as an accepting state here, if q_3 has an input 0 so it will change state to q_1 and wait for 1 to occur else if it has 1 as an input the whole pattern needs to be reset.

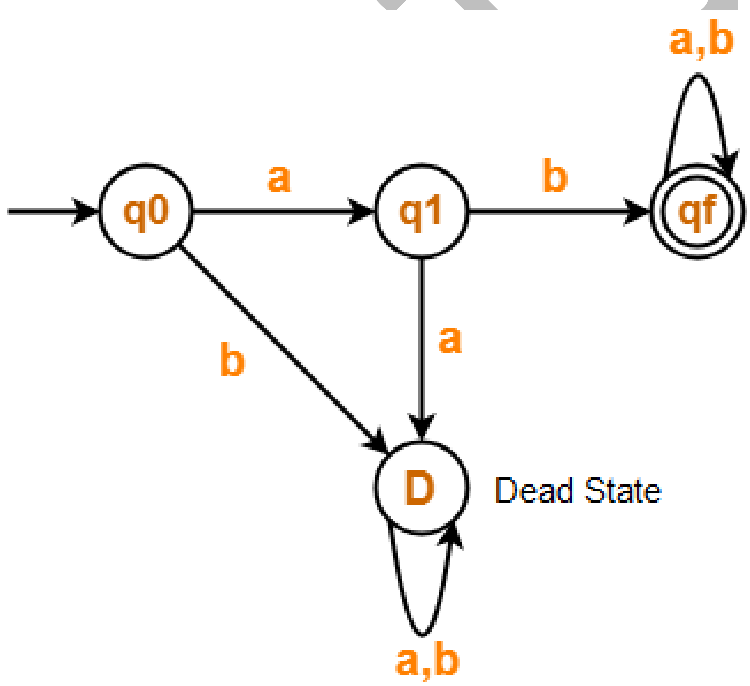
Q Draw a DFA for the language accepting strings ending with 'abb' over input alphabets ∑ = {a, b}?

Ans. The required DFA for the above problem will be:



Q Draw a DFA for the language accepting strings starting with 'ab' over input alphabets ∑ = {a, b}?

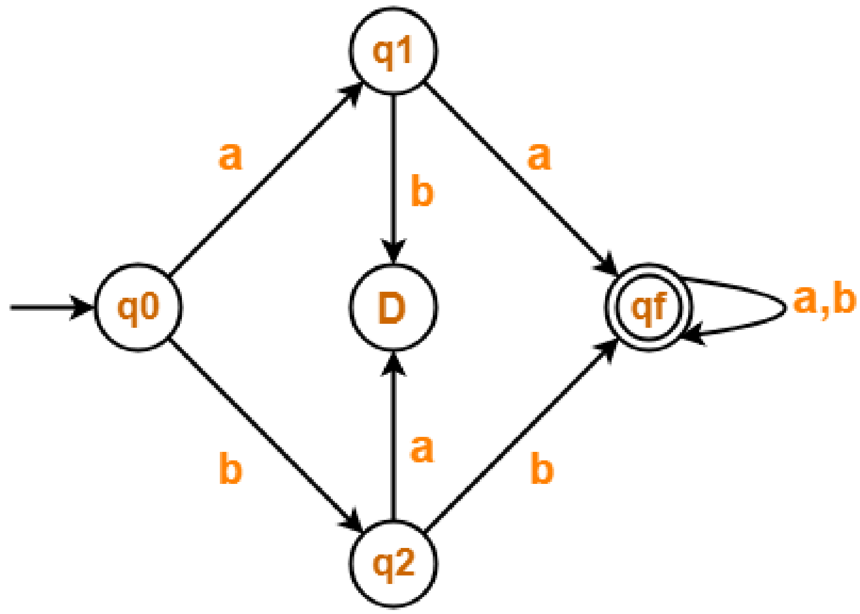
Ans.



- If the string starts with b we direct the transition to a dead state such that it never gets accepted.
- If we encounter two simultaneous a's in the beginning, we direct the transition to dead state.

Q Construct a DFA that accepts a language L over input alphabets ∑ = {a, b} such that L is the set of all strings starting with 'aa' or 'bb'?

Ans.



- In this case D acts as a dead state, if we encounter ab or ba as starting inputs we put the transitions into a dead state.

Q Design a minimal DFA that accepts a language ‘L’, where L={a} over the alphabet ∑={a}.

Q design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, where every accepted string ‘w’ starts with substring s

i) s = b **ii)** s = ab **iii)** s = abb

conclusion→ if w = sx, |s| = m, then no of states in the DFA is m+2

Q design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}. where every accepted string ‘w’ ends with substring ‘s’.

i) s = ab **ii)** s = aa **iii)** s = bab

conclusion→ if w = xs, |s| = m, then no of states in the DFA is m+1

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}. where every accepted string ‘w’ contains sub string s.

i) abb **ii)** aba

conclusion→ if w = xsx, |s| = m, then no of states in the DFA is m+1

Q No of states in a minimal in a that accepts all the strings of a, b such that every string must have ‘aabab’ as sub stray

a) 5 **b)** 6 **c)** 7 **d)** none

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string start and end with a.

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string start and end with same symbol.

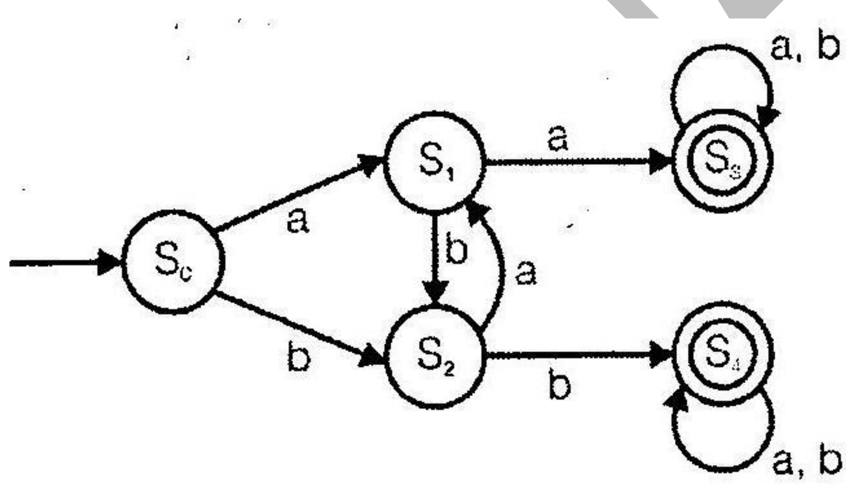
Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string start and end with different symbol.

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string w, is like w=SX.
i) s= aa/bb ii) s=aaa/bbb

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string w, is like w=XS.
i) s= aa/bb ii) s=aaa/bbb

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string w, is like w=XSX.
i) s= aa/bb ii) s=aaa/bbb

Q Consider the machine M shown below L(M)?



a) L(M) = {words starting with aa or bb} **b)** L(M) = {words ending with aa or bb}
c) L(M) = {words containing aa or bb as a sub word} **d)** None of these

Q Let L be the set of all binary strings who's last two symbols are the same. The number of states in the minimum state deterministic finite 0 state automation accepting L is **(GATE-1998) (1 Marks)**
a) 2 **b)** 5 **c)** 8 **d)** 3
Ans: b

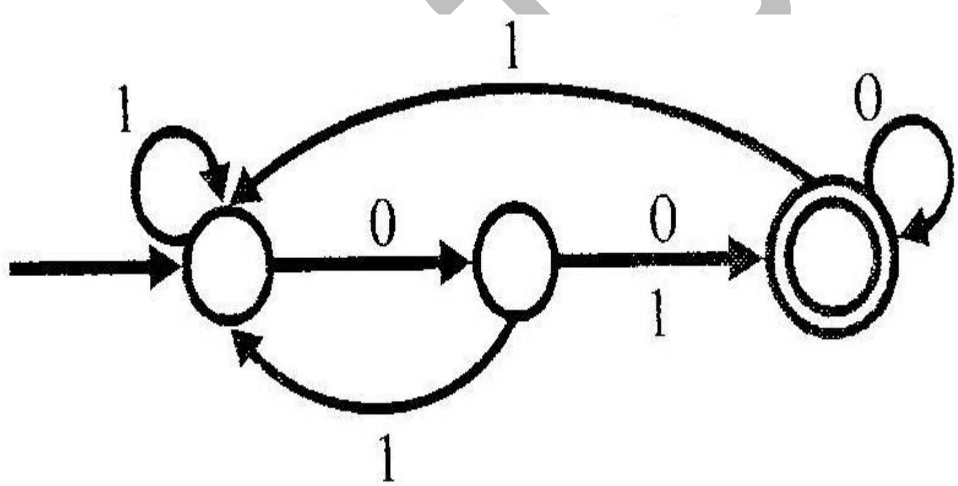
Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string 'w' accepted must be like
i) |w| = 3 **ii)** |w|<=3 **iii)** |w|>=3

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain exactly two a's.
Conclusion→ no of states will be n+2, if number of symbols is n

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain at least two a's.
Conclusion→ no of states will be n+1, if number of symbols is n

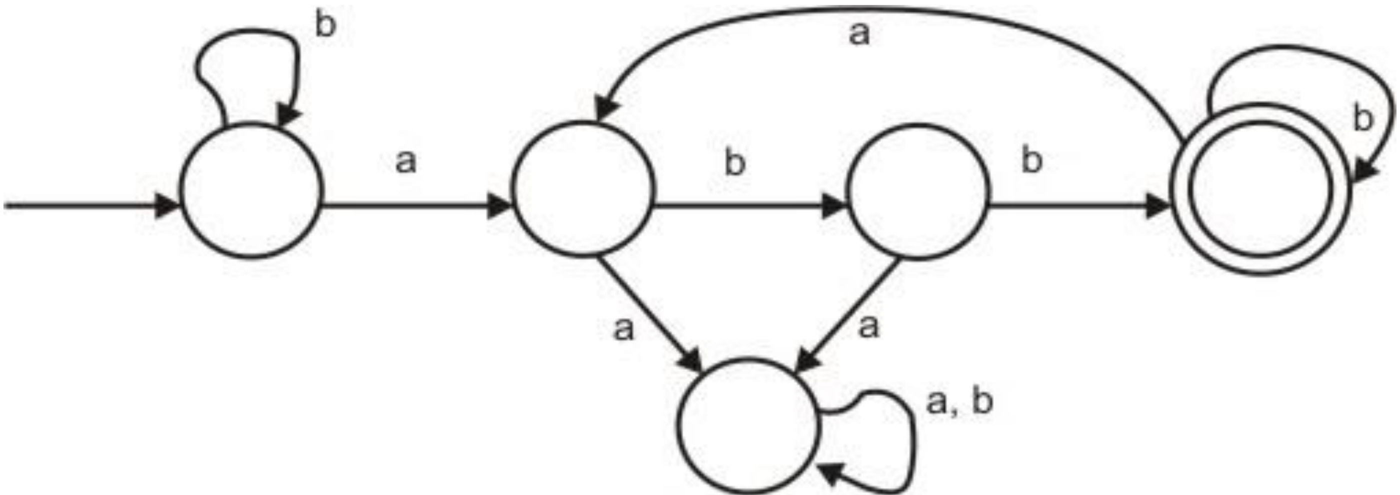
Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain at most two a's.
Conclusion→ no of states will be n+2, if number of symbols is n

Q The following DFA accepts the set of all strings over {0, 1} that **(GATE-2009) (1 Marks)**



a) Begin either with 0 or 1
b) End with 0
c) End with 00
d) Contain the substring 00

Q Consider the machine M **(GATE-2005) (2 Marks)**



The language recognized by M is :

(A) {w ∈ {a, b}* / every a in w is followed by exactly two b's}

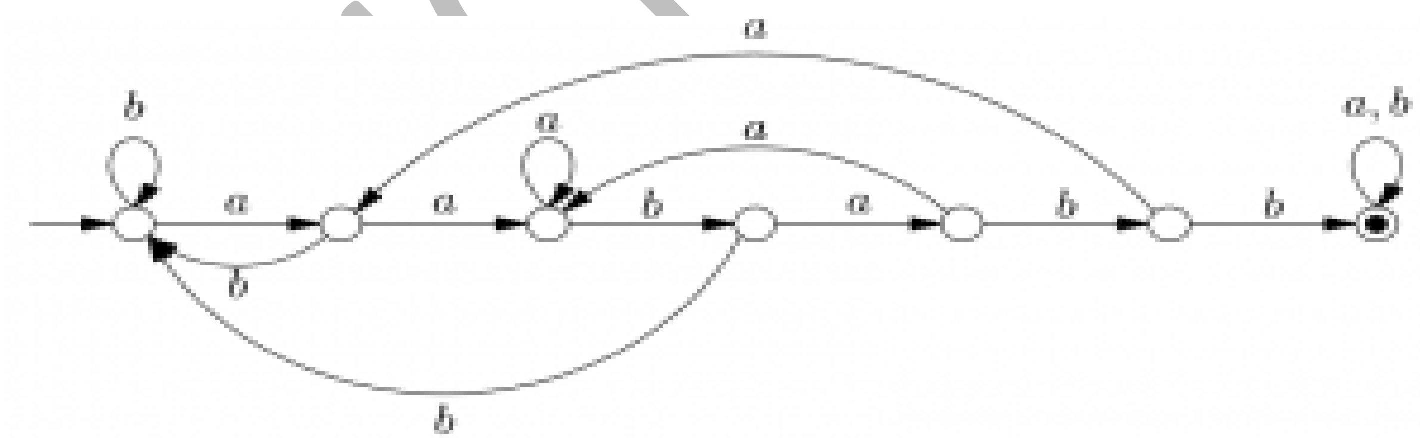
(B) {w ∈ {a, b}* every a in w is followed by at least two b'}

(C) {w ∈ {a, b}* w contains the substring 'abb'}

(D) {w ∈ {a, b}* w does not contain 'aa' as a substring}

Answer: (B)

Q Consider the following Deterministic Finite Automata **(GATE-2015) (2 Marks)**



Which of the following is true?

(A) It accepts all strings with prefix as "aababb"

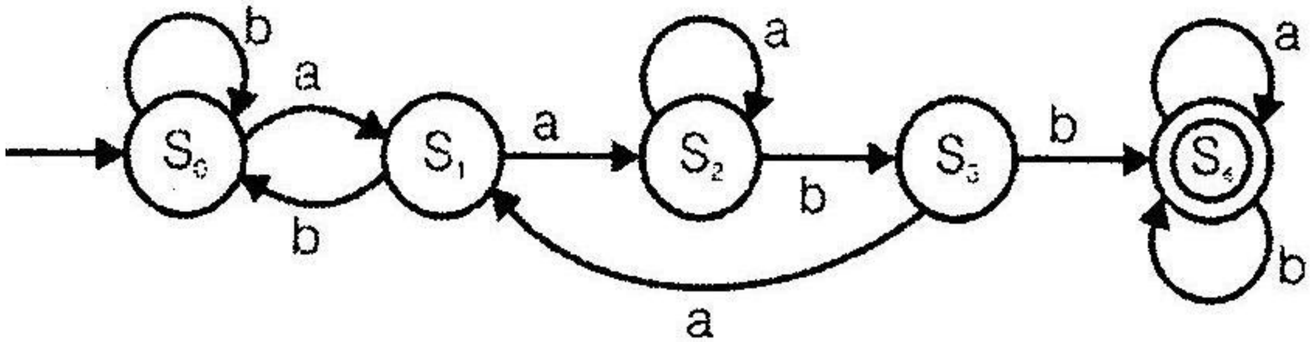
(B) It accepts all strings with substring as "aababb"

(C) It accepts all strings with suffix as "aababb"

(D) None of the above

Answer: (B)

Q Consider the following machine M



What is the language L(M) accepted by this machine?

a) L(M) = {Set of all words starting with aabb}

b) L(M) = {Set of all words having aabb as a sub word}

c) L(M) = {Set of all words ending with aabb}

d) L(M) = {Set of all words with exactly one occurrence of aabb}

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string 'w' accepted must be like

i) |w| = 0(mod 3) **ii)** |w|<=1(mod 4)

Conclusion→ no of states will be n, if the equation is on format m (mod n)

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain

i) number of a's = 0(mod 2) **ii)** number of b's = 1(mod 2)

iii) number of a's = 2(mod 3) **iv)** number of a's = 1(mod 4)

Conclusion→ no of states will be n, if the equation is on format m (mod n)

Q How many minimum states are required in a DFA to find whether a given binary string has odd number of 0's, there can be any number of 1's. (GATE-2015) (2 Marks)

(A) 1 **(B)** 2 **(C)** 3 **(D)** 4

Answer: (B)

Q The smallest finite automation which accepts the language {x | length of x is divisible by 3} has (GATE-2002) (2 Marks)

(A) 2 states **(B)** 3 states **(C)** 4 states **(D)** 5 states

Answer: (B)

Q Let ∑ = {0, 1}, then an automaton A accepting only those words from ∑ having an odd number of 1's requires __________ states including the start state

a) 2 b) 3 c) 4 d) 5

Ans: a

Q The smallest finite automaton which accepts the language {x | length of x is divisible by 500} has

a) 200 states b) 300 states c) 400 states d) 500 states

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain odd occurrence of the substring 'ab'.

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain even occurrence of the substring 'ab'.

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain even occurrence of the substring 'baa'.

Q how many DFA can be designed, with a fixed initial state, over an alphabet ∑ = {a, b}, and the number of states be two.

Q how many DFA can be designed, with a fixed initial state, over an alphabet ∑ = {a, b}, and the number of states be three.
Conclusion → if no of states is |Q| = n, and no of input alphabet |∑| = m
Then no of states will be $n^{n*m} * 2^n$

Q There are exactly ___________ different finite automata with three states x, y and z over the alphabet {a, b} where x is always the start state. **(NET-DEC-2015)**

a) 64 **b)** 56 **c)** 1024 **d)** 5832

Ans d

Q The possible number of finite automata with 3 states x, y and z, where x is always initial state, over the alphabet {a, b} are

a) 5830 **b)** 5832 **c)** 5834 **d)** 583

Q how many DFA can be constructed over the alphabet, such that no of states |Q| = 2 and size of input alphabet is |∑| = 2, that accepts empty language(Φ)?

Q how many DFA can be constructed over the alphabet, such that no of states |Q| = 2 and size of input alphabet is |∑| = 2, that accepts universal language (Σ^*)?

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that for every accepted string 2nd from left end is always b.

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that for every accepted string 4th from left end is always a.

Conclusion→ no of states will be n+2, if position is n

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that for every accepted string 2nd from right end is always b.

Conclusion→ no of states will be 2^n, if position is n

Q The no of final states in a minimal DFA, that accept all string over a, b such that 2nd symbol from right end is always b.

a) 1 **b)** 2 **c)** 4 **d)** none

Ans: 2

Q Design a minimal DFA that accepts all strings over the alphabet ∑= {a, b}, such that every string 'w' which is accepted starts with 'a' & length is divisible by 3, i.e. 0(mod 3)?

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string 'w' which is accepted starts with 'ab' & length is divisible by 2(mod 4)?
Conclusion→ no of states will be n, n = first condition + second condition -1

Q No of states in a minimal DFA that accept all string over a, b such that every string starts with 'abb' and length is concurrent to 4 (mod 7) is _______________

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string 'w' which is accepted starts with 'abbb' & length is divisible by 3(mod 15)?

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain be like, no of a = 0(mod 2) || no of b = 0(mod 2) ? cross product method(one more approach , where concentrate on even and odd logic)

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain be like, no of a = 0(mod 2) && no of b = 0(mod 2)?

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain be like, a = 0(mod 2) && no of b = 0(mod 3) && no of c = 0(mod 5)?
Conclusion → if, n (mod m) && p (mod r), then the no of states will be n*m, if n & m are relatively prime to each other

Q A minimum state deterministic finite automaton accepting the language L={w | w ε {0,1} *, number of 0s and 1s in w are divisible by 3 and 5, respectively} has **(GATE-2007) (2 Marks)**

(A) 15 states **(B)** 11 states **(C)** 10 states **(D)** 9 states

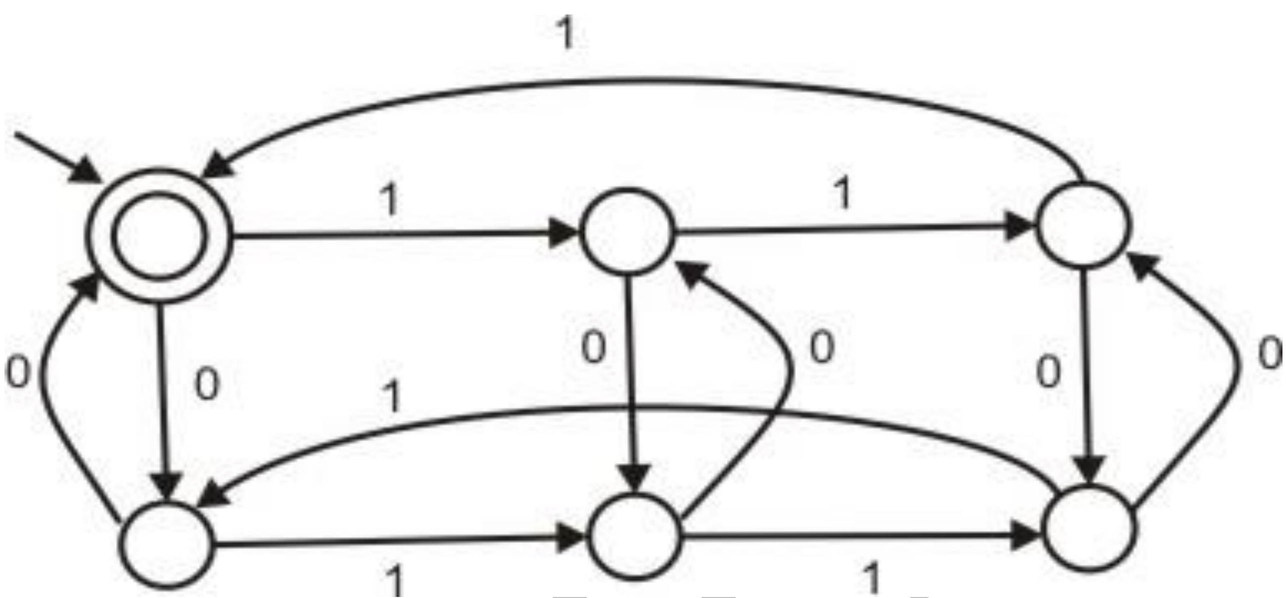
Answer: (A)

Q Consider a DFA over ∑ = {a, b} accepting all strings which have number of a's divisible by 6 and number of b's divisible by 8. What is the minimum number of states that the DFA will have? **(GATE-2001) (2 Marks)**

(A) 8 **(B)** 14 **(C)** 15 **(D)** 48

Answer: (D)

Q The following finite state machine accepts all those binary strings in which the number of l's and 0's is respectively. **(GATE-2004) (2 Marks)**



(A) divisible by 3 and 2

(B) odd and even

(C) even and odd

(D) divisible by 2 and 3

Answer: (A)

Q Given below are two finite state automata (→ indicates the start state and F indicates a final state) Which of the following represents the product automaton Z×Y? **(GATE-2008) (2 Marks)**

Y:

	a	b
→ 1	1	2
2 (F)	2	1

Z:

	a	b
→ 1	2	2
2 (F)	1	1

(A)

	a	b
→ P	S	R
Q	R	S
R (F)	Q	P
S	Q	P

(B)

	a	b
→ P	S	Q
Q	R	S
R (F)	Q	P
S	P	Q

(C)

	a	b
→ P	Q	S
Q	R	S
R (F)	Q	P
S	Q	P

(D)

	a	b
→ P	S	Q
Q	S	R
R (F)	Q	P
S	Q	P

Answer: (A)

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {0,1}, such that every string 'w' which is accepted has a decimal equivalent

i) 0(mod 3) **ii)** 2(mod 4) **iii)** 0(mod 5) **iv)** 2(mod 6) **v)** 3(mod 8)

Conclusion→ on a format of m (mod n)

1) if n is odd, number of states will be n

2) if n is even and $n=2^m$, then number of states will be m+1

3) if n is even and n! $=2^m$, then for number of states no direct formula

COMPLIMENT OF DFA

Q Design a minimal DFA that accepts all strings over the alphabet $\Sigma = \{a, b\}$, such that every string accepted must not contain a substring aaa?

Note: -

- The finite automata which is obtained by interchanging final & non-final states is known as complement of the finite automata.
- $L(FA) \cup L(FA^C) = \Sigma^*$ $L(FA) \cap L(FA^C) = \Phi$
- $L(FA)$ subset of Σ^* and $L(FA)^C$ subset of Σ^*
- No of states in FA = no of states in complement of FA
- Complement always exist for DFA (as it is a complete system)

Q Deterministic automata of a language over alphabets {0, 1}, which does not contain 3 consecutive 0's. Minimum how many states 'S' DFA will have and how many of them will final states, F?

a) ISI = 5 and IFI = 1

b) ISI = 5 and IFI = 4

c) ISI = 4 and IFI = 1

d) ISI = 4 and IFI = 3

Ans: d

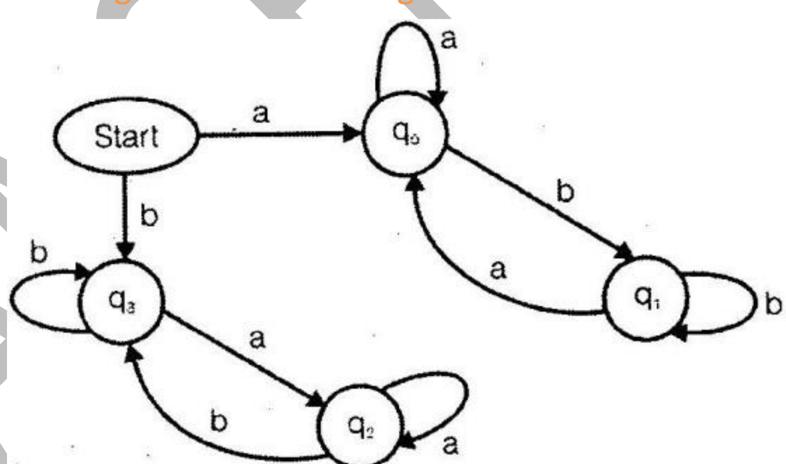
Q construct the DFA for the following languages:
i) L= {a^m b^n | m, n >= 0}
ii) L = {a^m b^n c^p | m, n, p >= 0}
conclusion →if type is L = {a^m b^n | m>= 0, n>=0}, then no of states is no of alphabet +1

Q construct the DFA for the following languages:
i) L= {a^m b^n | m>= 0, n>=1}
ii)L= {a^m b^n | m>= 0, n>=2}
conclusion→ if type is L = {a^m b^n | m>= 0, n>=j}, then no of states is j+2

Q construct the DFA for the following languages:
i) L = {a^m b^n | m>= 1, n>=0}
ii)L = {a^m b^n | m>= 2, n>=0}
conclusion →if type is L = {a^m b^n | m>= i, n>=0}, then no of states is i+3

Q construct the DFA for the following languages:
L = {a^m b^n | m >= 1, n>= 2}
conclusion →if type is L = {a^m b^n | m>= i, n>=j}, then no of states is i+j+2

Q Consider the transition diagram of an DFA as given below:



Q Which should be the final state(s) of the DFA if it should accept strings starting with 'a' and ending with 'b'?

a) q_0 **b)** q_1 **c)** q_0, q_1 **d)** q_3

Ans: b

Q Which of the following strings will be accepted if q_0 and q_1 are accepting states?

1. ababab **2.** Babaaa **3.** aaaba

a) 1,2 **b)** 2,3 **c)** 1,3 **d)** None of these

Ans: c

Q Which of the following represents the set of accepting states if the language to be accepted contains strings having the same starting and ending symbols?

a) $\{q_0\}$ **b)** $\{q_0, q_3\}$ **c)** $\{q_3\}$ **d)** $\{q_3, q_1\}$

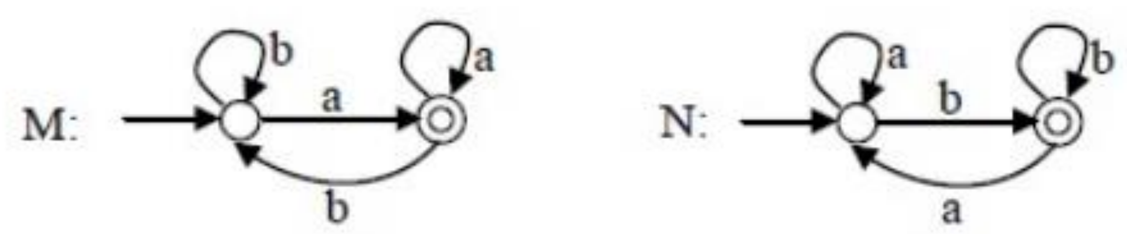
Ans: b

Q Definition of a language L with alphabet $\{a\}$ is given as following.
L = $\{a^{nk} \mid k>0$, and n is a positive integer constant$\}$
What is the minimum number of states needed in DFA to recognize L?(CS-2011)

(A) k+1 **(B)** n+1 **(C)** $2^{(n+1)}$ **(D)** $2^{(k+1)}$

Answer: (B)

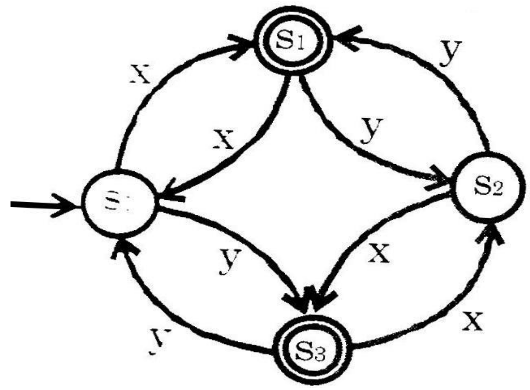
Q Consider the DFAs M and N given above. The number of states in a minimal DFA that accepts the language L(M) ∩ L(N) is ___________. **(GATE-2015) (2 Marks)**



(A) 0 **(B)** 1 **(C)** 2 **(D)** 3

Answer: (B)

Q Consider the following DFA in which s_0 is the start and s_1, s_3 are the final sates. What language does this DFA recognize? **(GATE-2007) (2 Marks)**



a) All strings of x and y

b) All strings of x and y and even either even number of x and even number of y of odd number or x and odd number of y

c) All strings of x and y which have equal number of x and y

d) All strings of x and y with either even number of x and odd number of y or odd number of x and even number of y.

Ans: D

MINIMIZATION OF FINITE AUTOMATA

It is sometimes difficult to design a minimal DFA directly so, a better approach is to first design the DFA and then minimize it.

Based on **productivity**, the states of DFA can be mainly classified in two types-

- **PRODUCTIVE STATES**- State is said to be productive, if it adds any accepting power to the machine that is its presence and absence effect the language accepting capability of the machine.
- **NON- PRODUCTIVE STATES**- These states don't any add anything to the language accepting power to the machine. They can further be divided into three types-
  - **Dead State**- It is basically created to make the system complete, can be defined as a state from which there is no transition possible to the final state. In a DFA there can be more than one dead state but logically always one dead state is sufficient to complete the functionality.
  - **Unreachable Sate**- It is that state which cannot be reached starting from initial state by parsing any input string.
  - **Equal State**-These are those states that behave in same manner on each and every input string. That is for any string w where $w \in \Sigma^*$ either both of the states will go to final state or both will go to non-final state. (remember the example of an equal state DFA). More formally, two states q_l and q_2 are equivalent (denoted by $q_1 \cong q_2$) if both $\delta(q_1, x)$ and $\delta(q_2, x)$ are final states or both of them are non-final states for all $x \in \Sigma^*$. If q_1 and q_2 are k-equivalent for all $k \geq 0$, then they are *k*-equivalent.

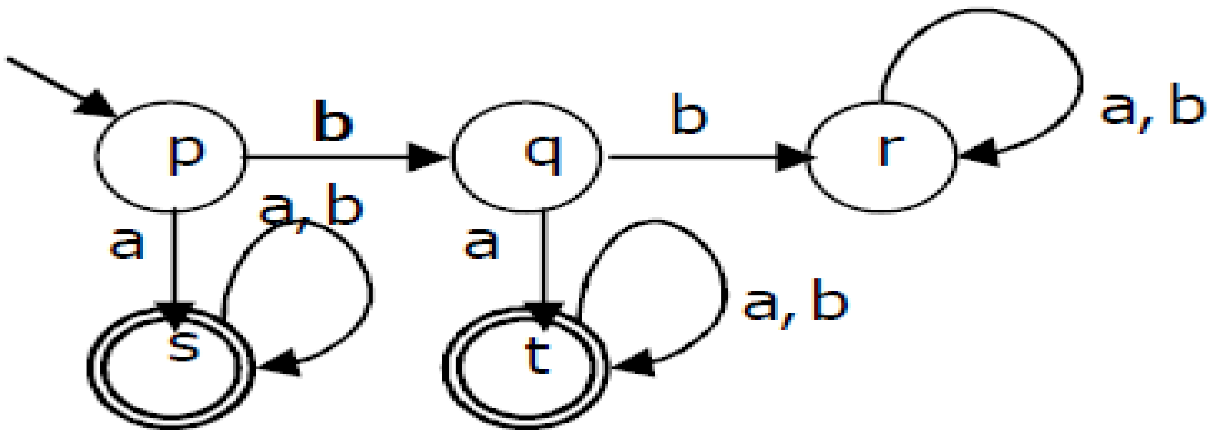
The process of elimination of states whose presence or absence doesn't affect the language accepting capability of deterministic Finite Automata is called minimization of automata and the result is minimal deterministic finite automata or commonly called as minimal finite automata as MFA.

NOTE- MFA is always unique for a language.

Procedure of Minimization

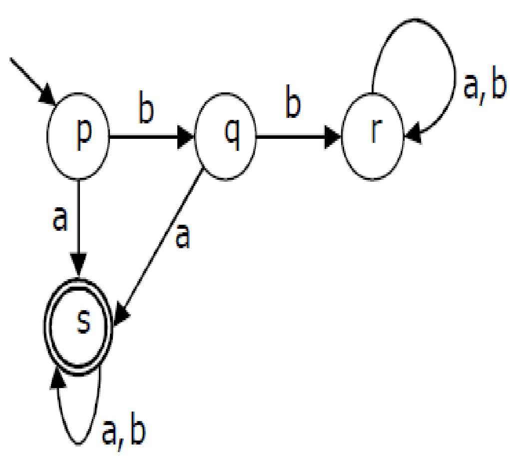
For this first of all, group all the non-final states in one set and all final states in another set. Now, on both the sets, individually check, whether any of the underlying elements (states) of that particular set are behaving in the same way, that is are they having same transition on each input alphabet, if the answer is yes, then these two states are equal, otherwise not.

Q A deterministic finite automation (DFA)D with alphabet {a, b} is given below **(GATE-2011) (2 Marks)**

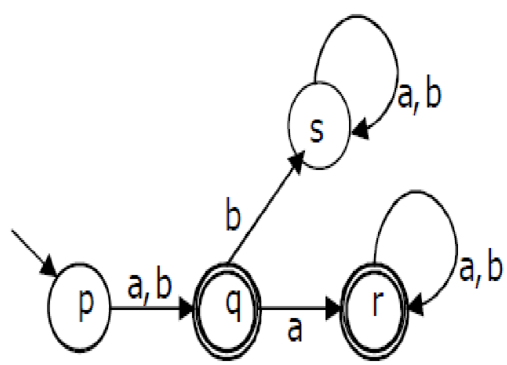


Which of the following finite state machines is a valid minimal DFA which accepts the same language as D?

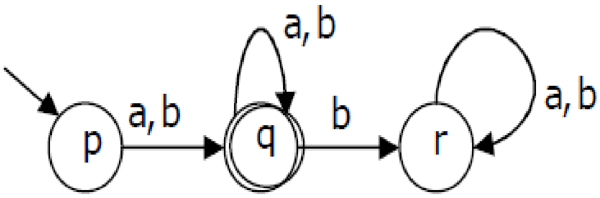
(A)



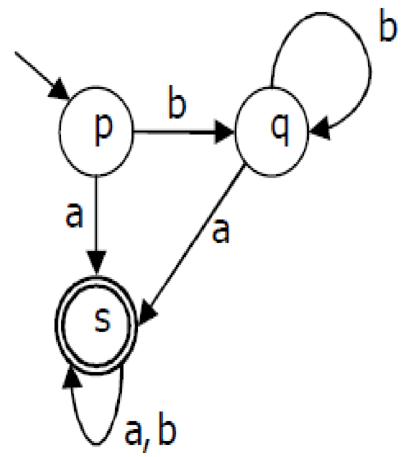
(B)



(C)



(D)



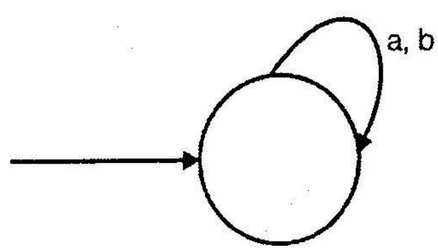
Answer: (A)

Q IF a DFA is represented by the following transition table, then how many states does the corresponding minimal DFA contains?

State	Input 0	Input 1
(start) → A	B	C
B	B	D
C	B	C
D	B	E
(final) → E	B	C

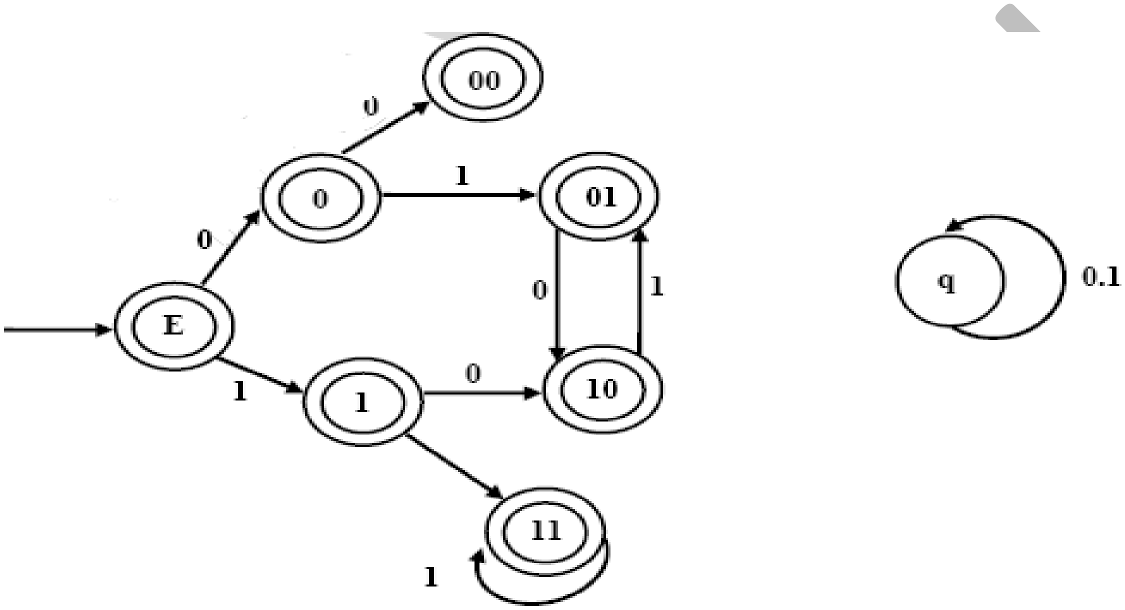
a) 2 **b)** 3 **c)** 4 **d)** 5

Q The FSM over an alphabet {a, b} shown in the figure accepts



a) all stings **b)** no stings **c)** ϵ - alone **d)** None of these

Q Consider the set of strings on {0,1} in which, every substring of 3 symbols has at most two zeros. For example, 001110 and 011001 are in the language, but 100010 is not. All strings of length less than 3 are also in the language. A partially completed DFA that accepts this language is shown below. **(GATE-2012) (2 Marks)**



The missing arcs in the DFA are

(A)

	00	01	10	11	q
00	1	0			
01				1	
10	0				
11			0		

(B)

	00	01	10	11	q
00		0			1
01		1			
10				0	
11		0			

(C)

	00	01	10	11	q
00		1			0
01		1			
10			0		
11		0			

(D)

	00	01	10	11	q
00		1			0
01				1	
10	0				
11			0		

Answer: (D)

Q Which of the following set can be recognized by a deterministic Finite state Automation? **(GATE–1998) (2 Marks)**

a) The numbers 1,2,4,8, 2^n, written in binary

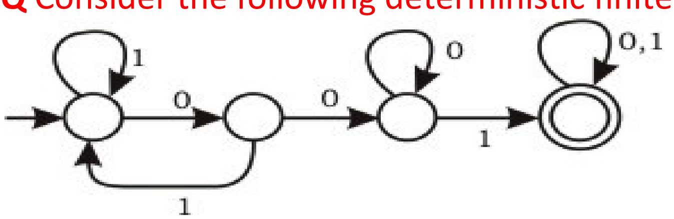
b) The numbers 1,2,4, 2^n, written in unary

c) The set of binary string in which the number of zeroes is the same as the number of ones.

d) The set { 1, 101, 11011, 1110111,}

Ans a

Q Consider the following deterministic finite state automaton M. **(GATE-2003) (2 Marks)**



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

Answer: (C)

Q Minimal deterministic finite automaton for the language L = $\{0^n \mid n \geq 0, n \neq 4\}$ will have: **(NET-JUNE-2015)**

a) 1 final state among 5 states

b) 4 final states among 5 states

c) 1 final state among 6 states

d) 5 final states among 6 states

Ans: d

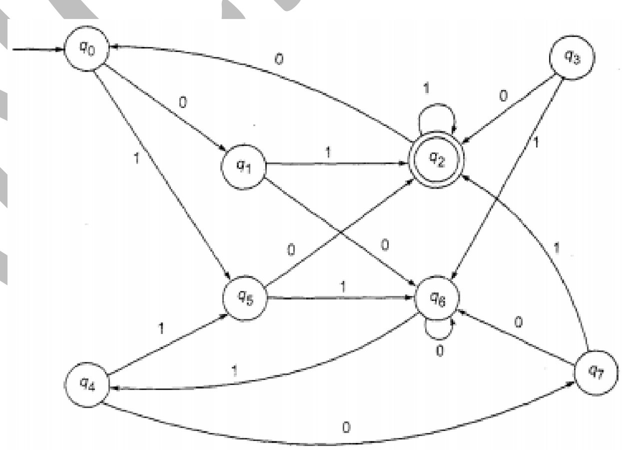
Q Find the no of states in MDFA where no of a" is divisible by 2 or no of b" is at most 1.

a) 4 **b)** 3 **c)** 6 **d)** 5

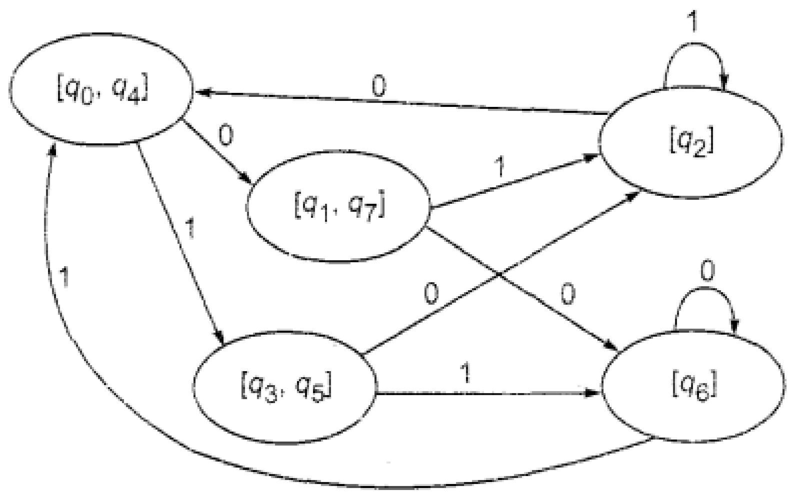
Q If w ϵ (a, b) * satisfies abw = wab, then | ω| is

a) even **b)** odd **c)** null **d)** none of these

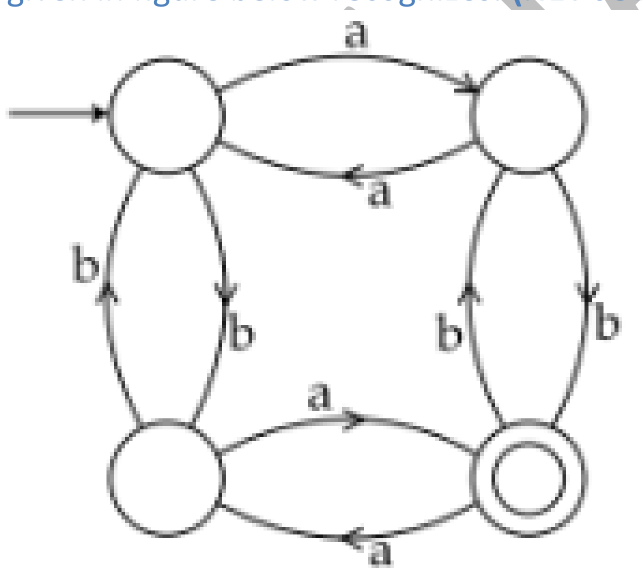
Q Convert the DFA below to MDFA?



And the transition diagram of MDFA is:



Q The finite state machine given in figure below recognizes: **(NET-JULY-2018)**



(1) any string of odd number of a's
(2) any string of odd number of b's
(3) any string of even number of a's and odd number of b's
(4) any string of odd number of a's and odd number of b's

Consider the language L given by

$$L = \{2^{nk} \mid k > 0, \text{ and n is non-negative integer number}\}$$

The minimum number of states of finite automaton which accepts the language L is

(NET-DEC-2018)

a) N **b)** N+1 **c)** n(n+1)/2 **d)** 2^n

Ans: b

Q The transition function for the language L = {w | n_a (w) and n_b(w) are both odd} is given by:

δ (q_0, a) = q_1

δ (q_0, b) = q_2

δ (q_1, a) = q_0

δ (q_1, b) = q_3

δ (q_2, a) = q_3

δ (q_2, b) = q_0

δ (q_3, a) = q_2

δ (q_3, b) = q_1

The initial and final states of the automata are: **(NET-JUNE-2015)**

A) q_0 and q_0 respectively

b) q_0 and q_1 respectively

c) q_0 and q_2 respectively

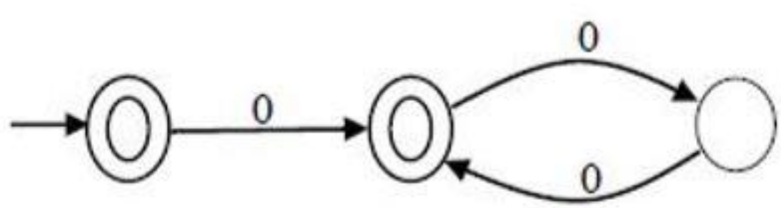
d) q_0 and q_3 respectively

Q Given a language L, define L^i as follows:

$L^0 = \{\varepsilon\}$

$L^i = L^{i-1} \cdot L$ for all i>0

The order of a language L is defined as the smallest k such that $L^k = L^{k+1}$. Consider the language L_1 (over alphabet 0) accepted by the following automaton.



The order of L_1 is ______. **(GATE-2018) (1 Marks)**

(ANSWER- 2)

Q The minimum possible number of states of a deterministic finite automaton that accepts a regular language L = {$w_1 a w_2$ | w_1, w_2 $\in$ {a,b}* , |w_1| = 2, w_2>=3} is_______ **(GATE-2017) (2 Marks)**

ANSWER 8

Q Let Σ be the set of all bijections from {1, ..., 5} to {1, ..., 5}, where id denotes the identity function, i.e. id(j) = j, $\forall j$. Let $\circ$ denote composition on functions. For a string $x = x_1x_2 \ldots x_n \in \Sigma^n$, $n \geq 0$, let $\pi(x) = x_1 \circ x_2 \circ \ldots \circ x_n$. Consider the language $L = \{x \in \Sigma^* \mid \pi(x) = id\}$. The minimum number of states in any DFA accepting L is **(GATE-2019) (2 Marks)**

(ANSWER- 120)

NON DETEMINISTIC FINITE AUTOMATA

Non determinism means choice of move for an automaton. So rather than prescribing a unique move in each situation, we allow a set of possible moves.

Non deterministic machine are only theoretical machine i.e. in the first place they are not implementable and neither we want to implement them, the only reason we study non-determinism is because they are easy to design and easily be converted into deterministic machine.

FORMAL DESCRIPTION OF NDFA

A Non-Deterministic finite automaton (NDFA) is a 5-tuple (Q,Σ,δ,S,F) where:

Q is a finite and non-empty set of states

Σ is a finite non-empty set of finite input alphabet

δ is a transition function $\delta: Q \times \Sigma \rightarrow 2^Q$

q_0 is **initial state** (always one) ($q_0 \in Q$)

F is a set of final states ($F \subseteq Q$) ($0<=|F|<=N$), where n is the number of states

Some points to remember

- Every DFA is also an NFA.
- Each NFA can be translated to an equivalent DFA, that they are equivalent in power.
- NFAs like DFA's only recognize ***regular languages***.
- It need not to be a complete system. There can be a state that doesn't have any transition on some input symbol.
- It is possible that a single state led to multiple transition on same input to different states.

NOTE- A null transition is also possible for NFA, such special NFA are called Null-NFA.

PROPERTIES OF NFA

i) Accepting power of NDFA= Accepting power of DFA.

ii) NDFA is a theoretical engine and is not implementable, but it is very easy to design compare to DFA.

iii) No concept of dead state, therefore complementation of DFA is also not possible.

iv) NDFA will respond for only valid strings and no need to respond for invalid strings. (it is a complete system.

ACCEPTANCE BY NDFA

Let 'w' be any string defined over the alphabet Σ, corresponding to w, there can be multiple transitions for NFA starting from initial state, if there exist at least one transition for which we start at the initial state and ends in any One of the final state, then the string 'w' is said to be accepted by the non-deterministic finite automata, otherwise not.

Mathematically, it can be represented as, $L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$

Q design a NDFA that accepts all strings over the alphabet ∑ = {a, b}, where every accepted string 'w' starts with substring s

i) s = ab **ii)** s = bb

conclusion→ if w = sx, |s| = m, then no of states in the NDFA is m+1

Q Design a NDFA that accepts all strings over the alphabet ∑ = {a, b}. where every accepted string 'w' ends with substring 's'.

i) s = ab **ii)** s = bab

conclusion→ if w = sx, |s| = m, then no of states in the NDFA is m+1

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b}. where every accepted string 'w' contains sub string s.

i) s = bb **ii)** s = aba

conclusion→ if w = xsx, |s| = m, then no of states in the DFA is m+1

Q Let w be any string of length n is {0,1}*. Let L be the set of all substrings of w. What is the minimum number of states in a non-deterministic finite automaton that accepts L? (GATE-2010) (1 Marks)

(A) n-1 **(B)** n **(C)** n+1 **(D)** 2n-1

Answer: (C)

Q Design a NDFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string start and end with same symbol.

Q Design a NDFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string start and end with different symbol.

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string w, is like w=SX.

i) s= aa/bb **ii)** s=aaa/bbb

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string w, is like w=XS.

i) s= aa/bb **ii)** s=aaa/bbb

Q Design a minimal DFA that accepts all strings over the alphabet ∑ = {a, b} such that every accepted string w, is like w=XSX.

i) s= aa/bb **ii)** s=aaa/bbb

Q Design a NFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string ‘w’ accepted must be like

i) $|w| = 3$ **ii)** $|w| <= 3$ **iii)** $|w| >= 3$

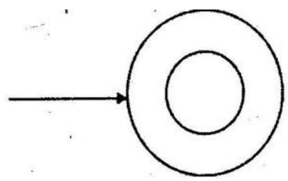
Q Design a NFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain exactly two a’s.

Q Design a NFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain at least two a’s.

Q Design a NFA that accepts all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain at most two a’s.

Q Design a NDFA that accepts all strings over the alphabet ∑ = {a, b} such that for every accepted string 2nd from right end is always A.

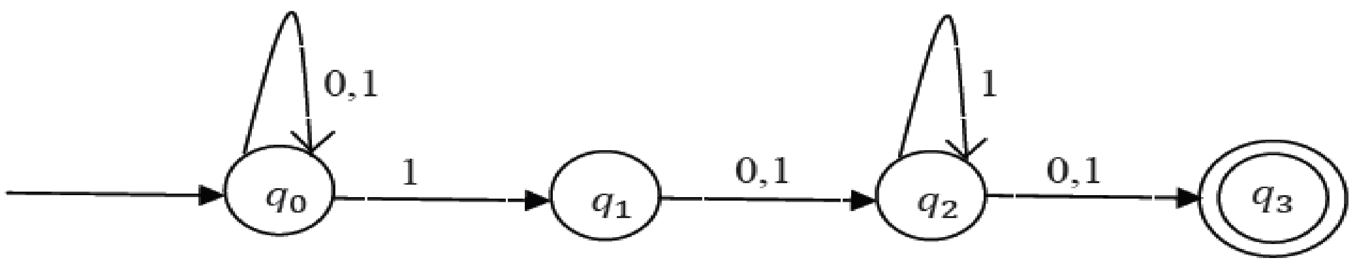
Q The FSM shown in the figure accepts



a) all strings **b)** no strings **c)** ϵ - alone **d)** None of these

Q (GATE-2014) (2 Marks)

Consider the finite automaton in the following figure.

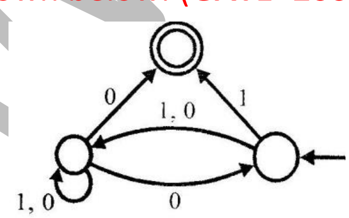


What is the set of reachable states for the input string 0011?

(A) {q0, q1, q2} **(B)** {q0, q1} **(C)** {q0, q1, q2, q3} **(D)** {q3}

Answer: (A)

Q Consider the NFAM shown below. (**GATE–2003) (2 Marks)**

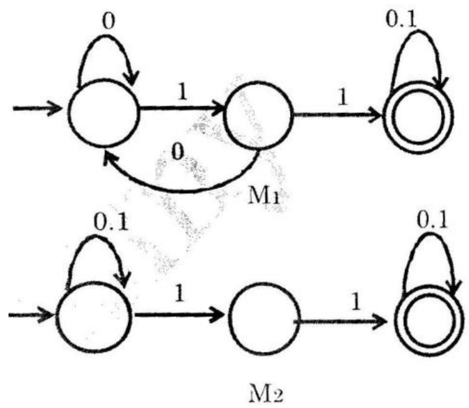


Let the language accepted by M be L. Let L_1 be the language accepted by the NFA M_1, obtained by changing the accepting state and by changing the non – accepting state of M to accepting states. Which of the following statements is True?

a) L_1 = {0, 1} * - L **b)** L_1 = {0, 1} * **c)** $L_1 \subset$ L **d)** L_1= L

Ans: b

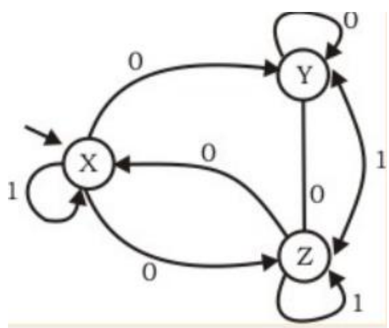
Q Consider the following two finite automata. M_1 accepts L_1 and M_2 accepts L_2 Which one of the following is TRUE? **(GATE-2008) (2 Marks)**



a) $L_1 = L_2$ **b)** $L_1 \subset L_2$ **c)** $L_1 \cap \overline{L_2} = \phi$ **d)** $L_1 \cup L_2 \neq L_1$

Ans: a

Q Consider the non-deterministic finite automaton (NFA) shown in the figure. **(GATE-2005) (2 Marks)**



State X is the starting state of the automaton. Let the language accepted by the NFA with Y as the only accepting state be L1. Similarly, let the language accepted by the NFA with Z as the only accepting state be L2. Which of the following statements about L1 and L2 is TRUE?

(A) L1 = L2 **(B)** L1 ⊂ L2 **(C)** L2 ⊂ L1 **(D)** None of the above

Answer: (D)

Q The minimum number of states of the non-deterministic finite automation which accepts the language {a b a b^n | n >=0} U {a b a^n | n >= 0} is **(NET-DEC-2012)**

(A) 3 **(B)** 4 **(C)** 5 **(D)** 6

Ans: c

Q Let w be any string of length n is {0, 1}*. Let L be the set of all substrings of w. What is the minimum number of states in a non-deterministic finite automaton that accepts L? **(GATE-2010) (2 Marks)**

a) n-1 **b)** n **c)** n+1 **d)** 2n-1

ANSWER C

NFA and DFA Equivalence

- In this topic we will be learning about the equivalence of NFA and DFA and how an NFA can be converted to equivalent DFA. Let us take an example and understand the conversion.
- Since every NFA and DFA has equal power that means, for every language if a NFA is possible, then DFA is also possible.
- So, every NFA can be converted to DFA and vice versa.
- The process of conversion of an NFA into a DFA is called Subset Construction.
- If NFA have 'n' states which is converted into DFA which 'm' states than the relationship between n and m will be $1 <= m <= 2^n$
- Consider the following NFA:

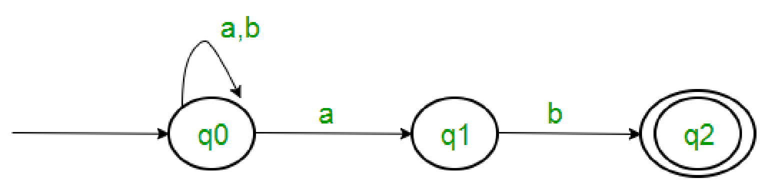


Figure 1

- We first draw the transition table of the NFA as:

State	a	b
q0	q0,q1	q0
q1		q2
q2		

- Now, the transitions such as: q_0, q_1 are treated as a new state.

State	a	b
q0	{q0,q1}	q0

- Now, we convert the NFA into equivalent DFA such that we will be taking transitions from the new states too. The transitions over the alphabets of the new states will be the union of all the transitions over a particular input symbol. Such that:

State	a	B
q0	{q0,q1}	q0
{q0,q1}	{q0,q1}	{q0,q2}

- For instance, state {q_0, q_1, b} => {q_0, q_2} as q_1 on b initially transitions to q_2 and q_0 on b transitions to q_0, so we take the union of the two and treat it as a new state. Similarly, after the final transitions we get the table as:

State	a	B
q0	{q0,q1}	q0
{q0,q1}	{q0,q1}	{q0,q2}
{q0,q2}	{q0,q1}	q0

- And we convert the table into the DFA as:

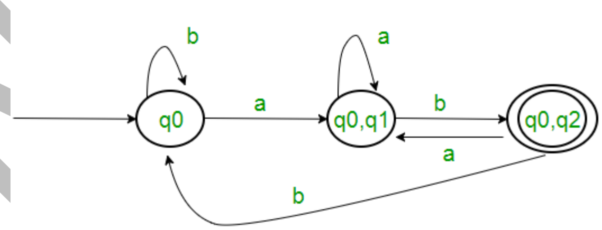


Figure 2

Q Can a DFA simulate NFA?

a) no **b)** yes **c)** sometime **d)** depends on NFA

Q Given an arbitrary non-deterministic finite automaton (NFA) with N states, the maximum number of states in an equivalent minimized DFA is at least (GATE-2001) (2 Marks)

(A) N^2 **(B)** 2^N **(C)** 2N **(D)** N!

Answer: (B)

Q Let N be an NFA with n states and let M be the minimized DFA with m states recognizing the same language. Which of the following in NECESSARILY true? **(GATE-2008) (2 Marks)**

(A) $m \le 2^n$ **(B)** $n \le m$ **(C)** M has one accept state **(D)** $m = 2^n$

Answer: (A)

Q Let N be an NFA with n states. Let k be the number of states of a minimal DFA which is equivalent to N. Which one of the following is necessarily true? **(GATE-2018) (1 Marks)**

a) $k \ge 2^n$ **b)** $k \ge n$ **c)** $k \le n^2$ **d)** $k \le 2^n$

(ANSWER- D)

Q IF an NDFA accepting L is denoted by (Q, S, d, q_0, F) the equivalence DFA is denoted M' = (Q', ∑, δ, q_0', F') Which of the following is true?

a) $Q' \subseteq 2^Q$

b) $q_0' = q_0$

c) F' is the set containing all elements of F

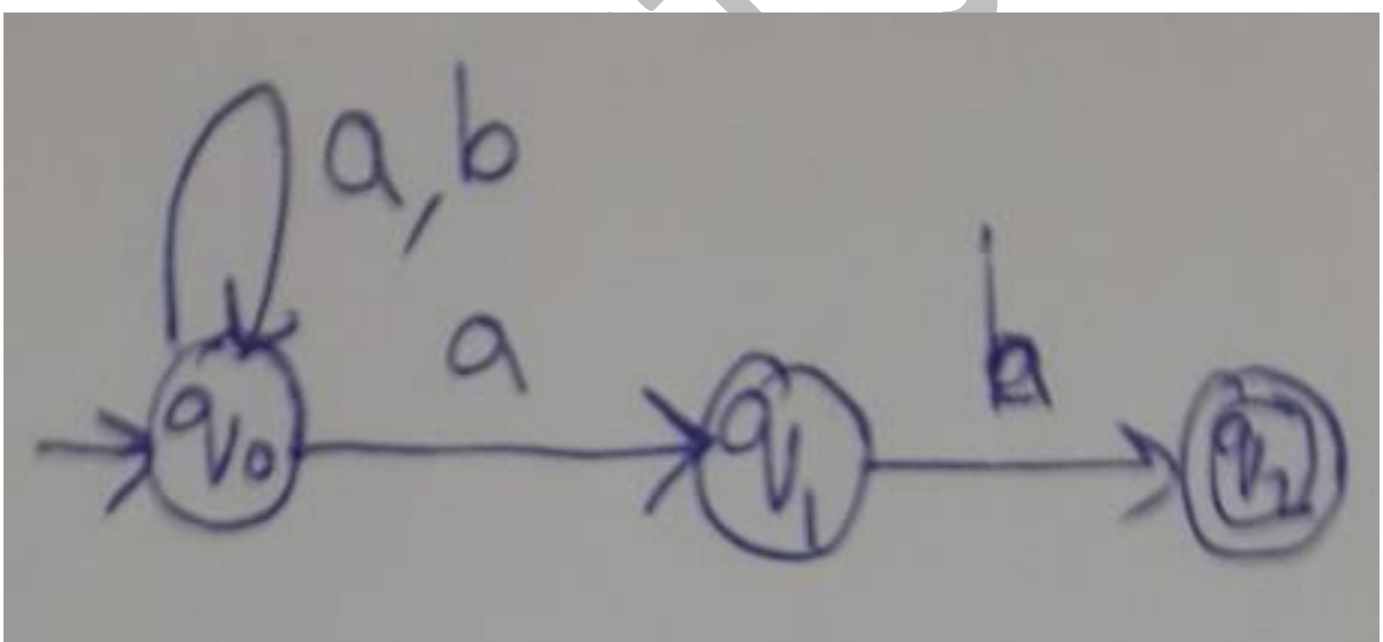
d) all of above

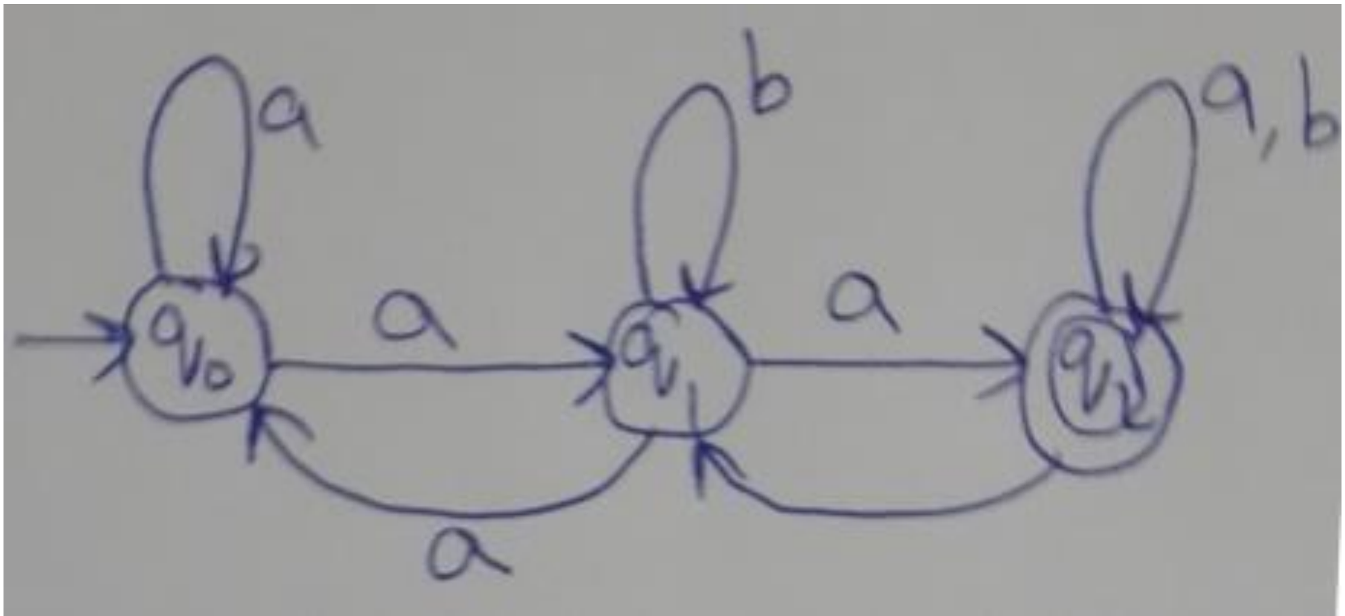
Procedure for Conversion

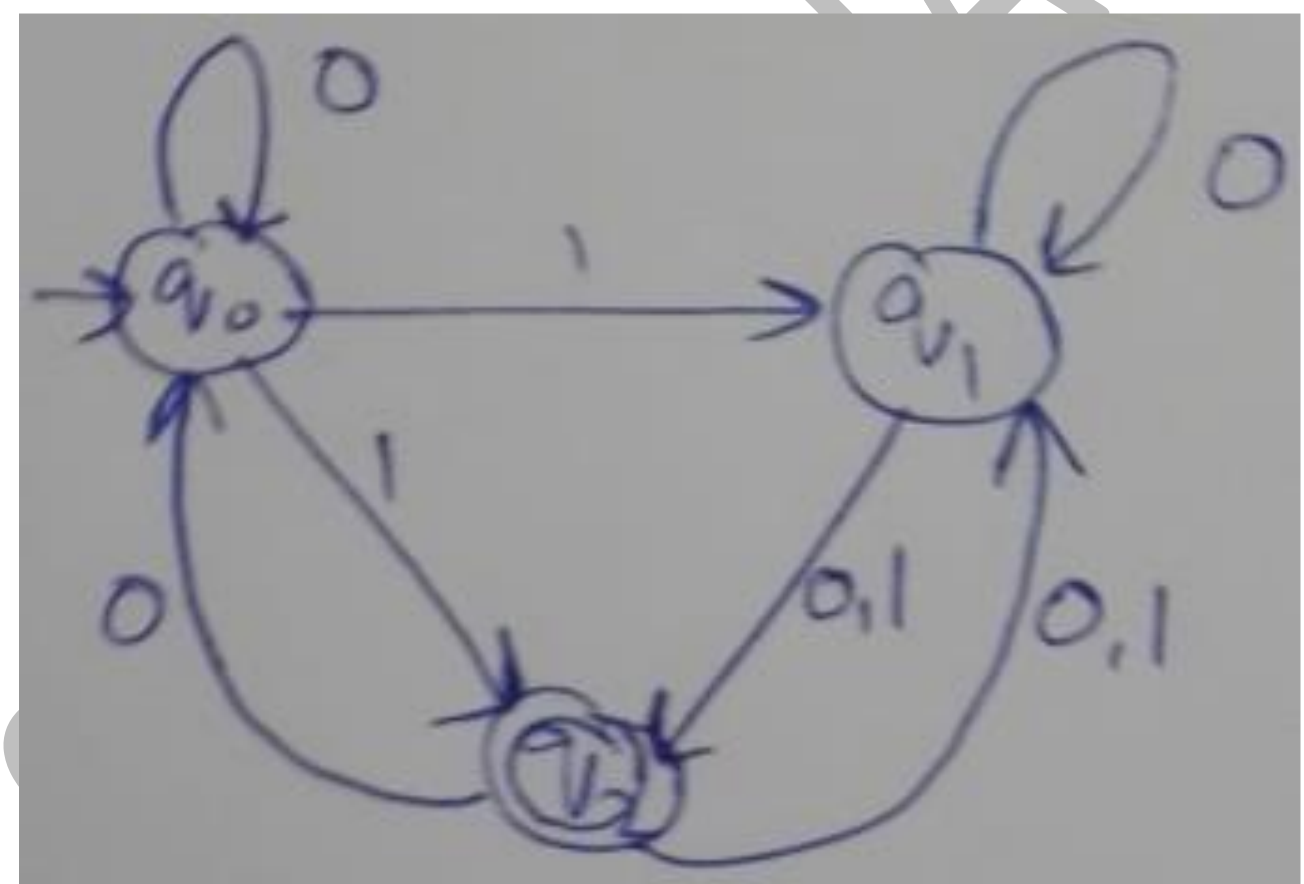
There lies a fixed algorithm for the NFA and DFA conversion. Following two things must be considered-

- Initial state will always remain same.
- Start the construction of δ' with the initial state & continue for every new state that comes under the input column and terminate the process whenever no new state appears under the input column.
- Every subset of states that contain the final state of the NFA is a final state in the resulting DFA.
- $\delta'(q_0, q_1, q_2, q_3\text{--------}, q_{n-1}, a) = \bigcup_{i=0}^{i=n-1} \delta(q_i, a)$

For e.g.




a
b
a,b
a
a
a


0
0
1
1
0
0,1
0,1

Q Given a Non-deterministic Finite Automation (NFA) with states p and r as initial and final states respectively and transition table as given below:

	a	b
P	–	q
Q	r	s
R	r	s
S	r	s

The minimum number of states required in Deterministic Finite Automation (DFA) equivalent to NFA is **(NET-JUNE-2013)**

(A) 5 **(B)** 4 **(C)** 3 **(D)** 2

Ans: b

NFA WITH EPSILON MOVES (€-NFA)

An automaton that consist of null transitions is called a Null- NFA i.e. we allow a transition on null means empty string.

€-NFA is a 5-tuple (Q,Σ,δ,S,F) where:

Q is a finite and non-empty set of states

Σ is a finite non-empty set of finite input alphabet

δ is a transition function δ: $(Q \times \{\Sigma \cup \varepsilon\}) \rightarrow 2^Q$

S is **initial state** (always one) ($S \in Q$)

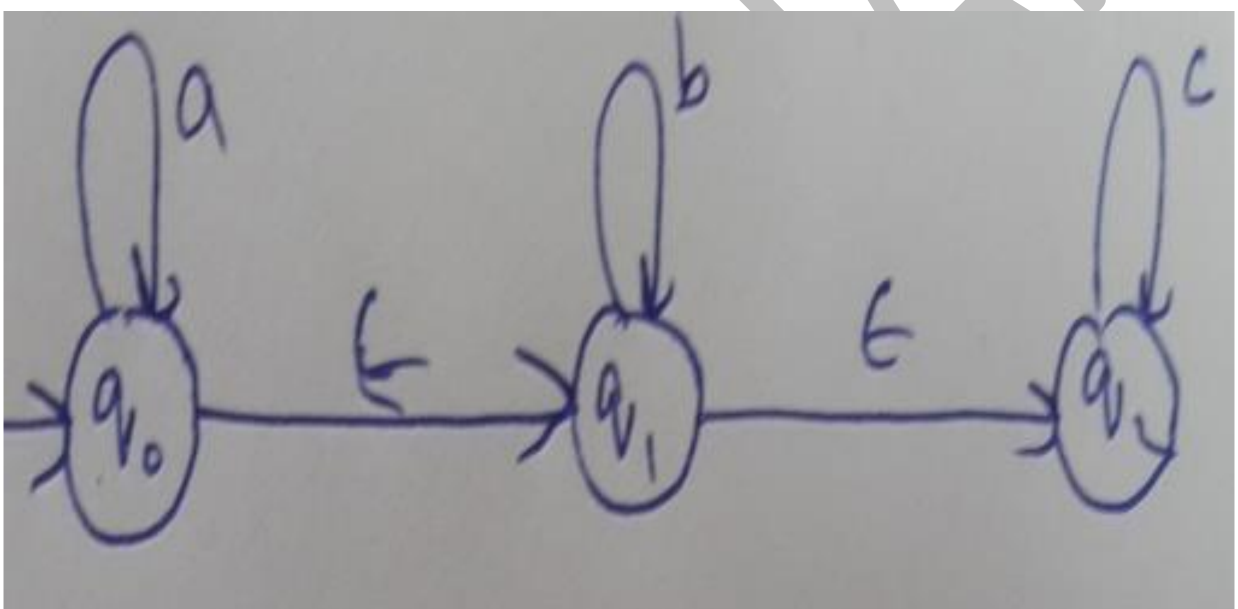
F is a set of final states ($F \subseteq Q$) ($0 <= |F| <= N$, where n is the number of states

NULL-CLOSURE

- Null closure of a set Q is defined as a set of all the states, which are at zero distance from the state Q. A set of all the states, that can be reached from the state and along a null-transition.
- **€-Closure(q_i),** The set of all the states which are at zero distance from the state q_i is called €-closure(q_i). Or the set of all the states that can be reached from the state q_i along € labelled transition path, is known as €-closure(q_i).

For e.g.

Q Consider the following null-NFA.



NULL CLOSURES-

Q_0 – {Q_0, Q_1, Q_2} Q_1 – {Q_1, Q_2} Q_2 –{Q_2}

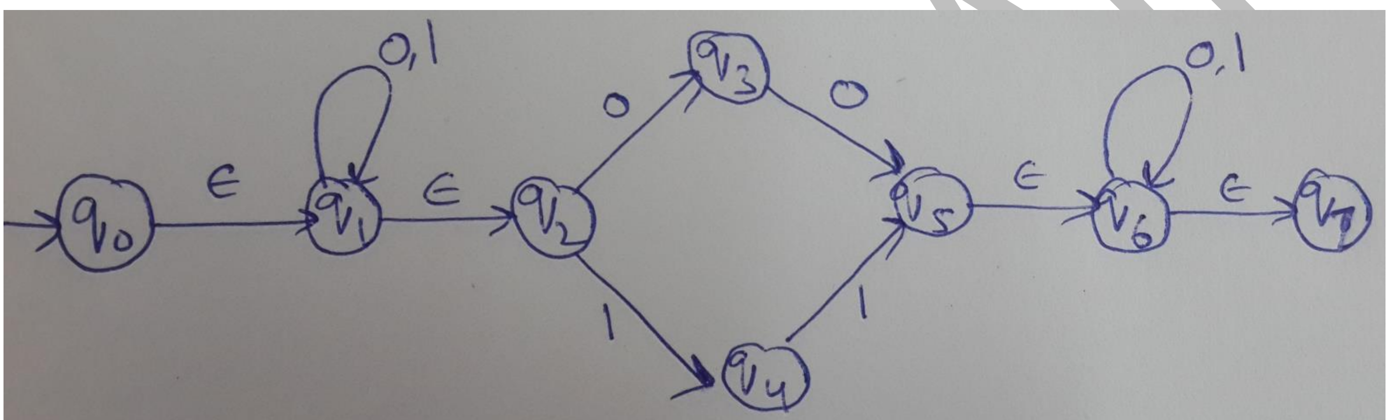
- Every state is at zero distance to itself.
- In NFA or DFA, distance between two states is always 1, because there could be no null transitions.
- The Null closure Q is always a non-empty and finite state, because every state's null closure is that state only.
- €-closure(Φ) = Φ

- €-closure (q_0, q_1, q_2, q_3--------, q_n) $= \bigcup_{i=0}^{i=n} \delta($ €-closure(q_i))

EQUIVALENCE BETWEEN NULL NFA TO NFA

- There will be no change in the initial state.
- No change in the total no. of states
- Me be change in the number of final states.
- All the states will get the status of the final state in the resulting NFA, whose €-closure contains at least one final state in the initial €-NFA.

Q Consider the following NFA,



Consider the following null NFA and convert it into corresponding DFA

Q Let us find the ε-Closure to above examples:

Q	ε-Closure
q_0	$\{q_0, q_1, q_2\}$
q_1	$\{q_1, q_2\}$
q_2	$\{q_2\}$

Now, we check out for the transitions that will be made using: $\delta'(q_i, x)$ = **ε-Closure [δ[ε-Closure(q_i), *x*]]**

Let us check for $\delta(q_0, a)$ = **ε-Closure [δ[ε-Closure(q_0), *a*]]**

- first, we find **ε-Closure** of q_0: $\{q_0, q_1, q_2\}$

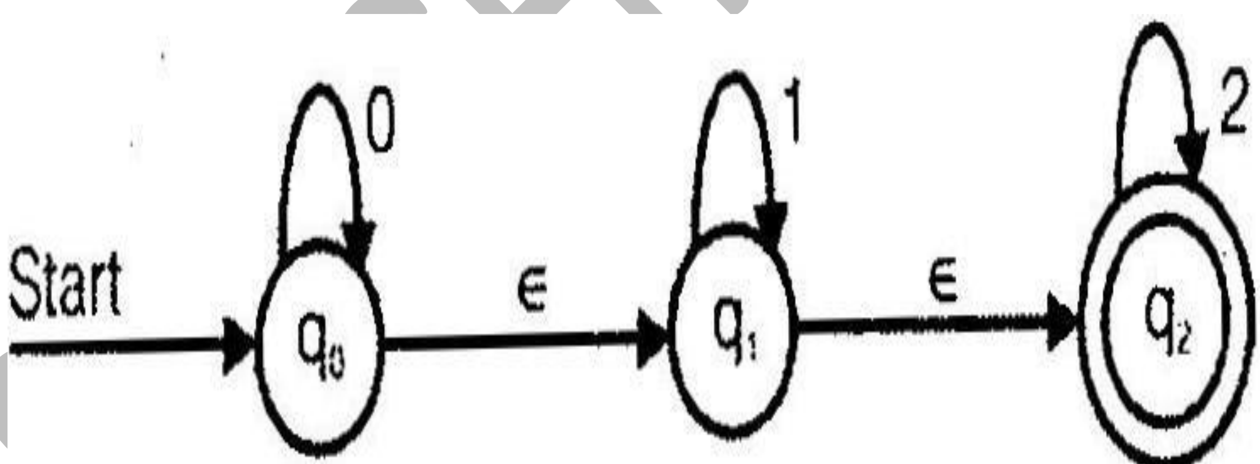
- Now check transition of all **three** states on input symbol *a* we get: q_0
- After that we again calculate the **ε-Closure** of the above result q_0, we get the result as: **$\{q_0, q_1, q_2\}$**
- Similarly, we will check it for every state.

The resulting table of NFA will be:

NFA	a	b	c
→ q0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q1	Φ	$\{q_1, q_2\}$	$\{q_2\}$
q2	Φ	Φ	$\{q_2\}$

- All those states will be considered as a final state in which we have q_2(final state in initial **ε-NFA**) in there ε-Closure, i.e. all states will be final in this case.

Q Consider the following NFA with ϵ moves.



Q if the given NFA is converted to NFA without ϵ - moves, which of the following denotes the set of final states?

a) $\{q_2\}$ **b)** $\{q_1, q_2\}$ **c)** $\{q_0, q_1, q_2\}$ **d)** can't be determined

Q Which of the following strings will not be accepted by the given NFA?

a) 00 11 22 **b)** 11 22 **c)** 21 **d)** 22

Q Let δ denote the transition function and δ' denoted the extended transition function of the ε-NFA whose transition table is given below: **(GATE-2017) (2 Marks)**
Then, δ' (q_2, aba) is

δ	ϵ	a	b
→ q_0	{q_2}	{q_1}	{q_0}
q_1	{q_2}	{q_2}	{q_3}
q_2	{q_0}	∅	∅
q_3	∅	∅	{q_2}

a) ∅ **b)** {q1, q2, q3} **c)** {q0, q1, q2} **d)** {q0, q2, q3}

Ans: c

Q Which of the following is false?
a) The languages accepted by FAs are regular languages
b) Every DFA is an NFA
c) There are some NFAs for which no DFA can be constructed
d) If L is accepted by an NFA with ϵ transition then L is accepted by an NFA without ϵ transition
Answer: (c)

Q An FSM can be used to add two given integers. This remark is
a) true **b)** false **c)** may be true **d)** none of the above

Consider the following languages and find which of them are regular?

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

$L = \{a^m b^n c^p \mid m, n, p >= 0\}$

$L = \{a^{x1} b^{x2} ---- z^{x26} \mid x_i >= 0, 0 <= I <= 26\}$

$L = \{a^m b^n \mid 1 <= m <= 100, 1 <= n <= 1200\}$

$L = \{a^m b^n \mid m*n = \text{finite}\}$

$L = \{a^n b^n \mid 1 <= n <= 2^{|GATE|}\}$

$L = \{a^n b^n \mid 1 <= n <= 2^{37\text{th prime}}\}$

$L = \{a^m b^n \mid m = n, 1 <= n <= 2^{2\wedge 10}\}$

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

$L = \{a^m b^n \mid m < n \mid m, n >= 0\}$

$L = \{a^m b^n \mid m != n \mid m, n > 0\}$

$L = \{a^m b^n \mid m \text{ is divisible by } n\}$

$L = \{a^m b^n \mid m = n^p, p>=1\}$

$L = \{a^m b^n \mid G (1) \mid m, n >= 1\}$

$L = \{a^m b^n \mid m + n = \text{even}\}$

$L = \{a^m b^n c^p \mid m = n = p\}$

$L = \{a^m b^n c^p \mid m + p = n\}$

$L = \{a^m b^n \mid m + n = \text{odd}\}$

$L = \{w c w^r \mid w \in \Sigma^*\}$

$L = \{w c w \mid w \in \Sigma^*\}$

$L = \{w c w^r \mid w \in \Sigma^+\}$

$L= \{w c w \mid w \in \Sigma^+\}$

$L = \{w w \mid w \in \Sigma^*\}$

$L = \{w\ w^r \mid w \in \Sigma^+\}$

$L = \{w\ c\ w^r \mid c, w \in \Sigma^*\}$

$L = \{c\ w\ w^r \mid c, w \in \Sigma^*\}$

$L = \{w\ w^r\ c \mid c, w \in \Sigma^*\}$

$L = \{w\ c\ w^r \mid c, w \in \Sigma^+$

$L = \{c\ w\ w^r \mid c, w \in \Sigma^+\}$

$L = \{w\ w^r\ c \mid c, w \in \Sigma^+\}$

$L = \{w\ c\ w \mid c, w \in \Sigma^*\}$

$L = \{c\ w\ w \mid c, w \in \Sigma^*\}$

$L = \{w\ w\ c \mid c, w \in \Sigma^*\}$

$L = \{w\ c\ w \mid c, w \in \Sigma^+\}$

$L = \{c\ w\ w \mid c, w \in \Sigma^+\}$

$L = \{w\ w\ c \mid c, w \in \Sigma^+\}$

Q Which of the following are regular sets? **(GATE-2008) (2 Marks)**

I. $\{a^n b^{2m} \mid n \geq 0, m \geq 0\}$

II. $\{a^n b^m \mid n = 2m\}$

III. $\{a^n b^m \mid n \neq m\}$

IV. $\{xcy \mid x, y \in \{a,b\}^*\}$

(A) I and IV only **(B)** I and III only **(C)** I only **(D)** IV only

Answer: (A)

If $L_1 = \{a^n \mid n \geq 0\}$ and $L_2 = \{b^n \mid n \geq 0\}$, consider

(I) $L_1 \cdot L_2$ is a regular language

(II) $L_1 \cdot L_2 = \{a^n b^n \mid n \geq 0\}$

Which one of the following is CORRECT? **(GATE-2014) (2 Marks)**

(A) Only (I) **(B)** Only (II) **(C)** Both (I) and (II) **(D)** Neither (I) nor (II)

Answer: (A)

Q Which one of the following is TRUE? **(GATE-2014) (1 Marks)**

(A) The language $L = \{a^n b^n \mid n \geq 0\}$ is regular.

(B) The language $L = \{a^n \mid n \text{ is prime}\}$ is regular.

(C) The language $L = \{w \mid w \text{ has } 3k+1 \text{ } b's \text{ for some } k \in \mathbb{N} \text{ with } \Sigma = \{a, b\}\}$ is regular.

(D) The language $L = \{ww \mid w \in \Sigma^* \text{ with } \Sigma = \{0,1\}\}$ is regular.

Answer: (C)

Q Consider the following two statements **(GATE-2001) (2 Marks)**

S1: $\{0^{2n} \mid n \geq 1\}$ is a regular language

S2: $\{0^m 1^n 0^{m+n} \mid m \geq 1 \text{ and } n \geq 1\}$ is a regular language

(A) Only S1 is correct **(B)** Only S2 is correct

(C) Both S1 and S2 are correct **(D)** None of S1 and S2 is correct

Answer: (A)

Q Consider the following languages **(GATE-2001) (2 Marks)**

$$L1 = \{ww \mid w \in \{a, b\}^*\}$$

$$L2 = \{ww^R \mid w \in \{a, b\}^*, w^R \text{ is the reverse of w}\}$$

$$L3 = \{0^{2i} \mid \text{i is an integer}\}$$

$$L4 = \{0^{i^2} \mid \text{i is an integer}\}$$

Which of the languages are regular?

(A) Only L1 and L2

(B) Only L2, L3 and L4

(C) Only L3 and L4

(D) Only L3

Answer: (D)

Q Which of the following languages is/are regular? **(GATE-2015) (2 Marks)**

L_1: $\{w\,x\,w^R \mid w, x \in \{a, b\}^* \text{ and } |w|, |x| > 0\}$ w^R is the reverse of string w

L_2: $\{a^n b^m \mid m \neq n \text{ and } m, n \geq 0$

L_3: $\{a^p b^q c^r \mid p, q, r \geq 0\}$

(A) L1 and L3 only **(B)** L2 only **(C)** L2 and L3 only **(D)** L3 only

Answer: (A)

Q Which of the following languages is (are) non-regular? **(GATE-2008) (2 Marks)**

$L_1 = \{0^m 1^n \mid 0 \leq m \leq n \leq 10000\}$

L_2 = {w | w reads the same forward and backward}

L_3 = {w ∈ {0, 1} * | w contains an even number of 0's and an even number of 1's}

(A) L_2 and L_3 only **(B)** L_1 and L_2 only **(C)** L_3 only **(D)** L_2 only

Answer: (D)

Q Which of the following languages is regular? **(GATE-2007) (2 Marks)**

a) $\{w\,w^R \mid w \in \{0, 1\}^+\}$

b) $\{w\,w^R\,x \mid x, w \in \{0,1\}^+\}$

c) $\{w\,x\,w^R \mid x, w \in \{0,1\}^+\}$

d) $\{x\,w\,w^R \mid x, w \in \{0,1\}^+\}$

ANSWER C

Q Let P be a regular language and Q be context-free language such that $Q \subseteq P$. (For example, let P be the language represented by the regular expression p^*q^* and Q be $\{p^nq^n \mid n \in N\}$). Then which of the following is ALWAYS regular? (GATE-2011) (1 Marks)

(A) $P \cap Q$ **(B)** $P - Q$ **(C)** $\Sigma^* - P$ **(D)** $\Sigma^* - Q$

ANSWER C

Q Which of the following are not regular? **(NET-JAN-2017)**

(A) Strings of even number of a's.

(B) Strings of a's, whose length is a prime number.

(C) Set of all palindromes made up of a's and b's. Strings of a's whose length is a perfect square.

a) (A) and (B) only **b)** (A), (B) and (C) only

c) (B), (C) and (D) only **d)** (B) and (D) only

Ans: c

Q Consider the following two languages: **(NET-SEP-2013)**

L1 = $\{a^n b^l a^k \mid n + l + k > 5\}$

L2 = $\{a^n b^l a^k \mid n > 5, l > 3, k \leq l\}$

Which of the following is true?

(A) L1 is regular language and L2 is not regular language.

(B) Both L1 and L2 are regular languages.

(C) Both L1 and L2 are not regular languages.

(D) L1 is not regular language and L2 is regular language.

Ans: a

Q Given the following statements:

S1: If L is a regular language then the language $\{u\, v \mid u \in L, v \in L^R\}$ is also regular.

S2: $L = \{w\, w^R\}$ is regular language.

Which of the following is true? **(NET-DEC-2013)**

(A) S1 is not correct and S2 is not correct.

(B) S1 is not correct and S2 is correct.

(C) S1 is correct and S2 is not correct.

(D) S1 is correct and S2 is correct.

Ans: c

Q Given two languages: **(NET-DEC-2014)**

L_1= $\{(ab)^n a^k \mid n > k, k \geq 0\}$

L_2= $\{a^n b^m \mid n \neq m\}$

Using pumping lemma for regular language, it can be shown that

(A) L1 is regular and L2 is not regular.

(B) L1is not regular and L2is regular.

(C) L1is regular and L2is regular.

(D) L1is not regular and L2is not regular.

Ans: d

Q Given the following two languages:

L1 = $\{u\ w\ w^R\ v \mid u, v, w \in \{a, b\}^+\}$

L2 = $\{u\ w\ w^R\ v \mid u, v, w \in \{a, b\}^+, |u| > |v|\}$

Which of the following is correct ? **(NET-AUG-2016)**

a) L1 is regular language and L2 is not regular language.

b) L1 is not regular language and L2 is regular language.

c) Both L1 and L2 are regular languages.

d) Both L1 and L2are not regular languages.

Ans: a

Q If L is a regular language over ∑ = {a, b}, which one of the following languages is NOT regular? **(GATE-2019) (2 Marks)**

a) $L \cdot L^R$ $\{xy \mid x \in L, y^R \in L\}$

b) Suffix (L) = $\{y \in \Sigma^* \mid \exists x \in \Sigma^*$ such that $xy \in L\}$

c) Prefix (L) = $\{x \in \Sigma^* \mid \exists y \in \Sigma^*$ such that $xy \in L\}$

d) $\{w\ w^R \mid w \in L\}$

Ans: d

Consider the following two languages :

L_1 = {x | for some y with $|y| = 2^{|x|}$, xy ∈ L and L is regular language}

L_2 = {x | for some y such that |x| = |y|, xy ∈ L and L is regular language}

Which one of the following is correct ? **(NET-DEC-2018)**

a) Only L1 is regular language

b) Only L2 is regular language

c) Both L1 and L2 are regular languages

d) Neither L1 nor L2 is regular

Ans: d

Q Let L1 = {w ∈ {0, 1}* | w has at least as many occurrences of (110)’s as (011)’s}.

Let L2 = {w ∈ {0, 1}* | w has at least as many occurrences of (000)’s as (111)’s}.

Which one of the following is TRUE? **(GATE-2014) (2 Marks)**

(A) L1 is regular but not L2

(B) L2 is regular but not L!

(C) Both L2 and L1 are regular

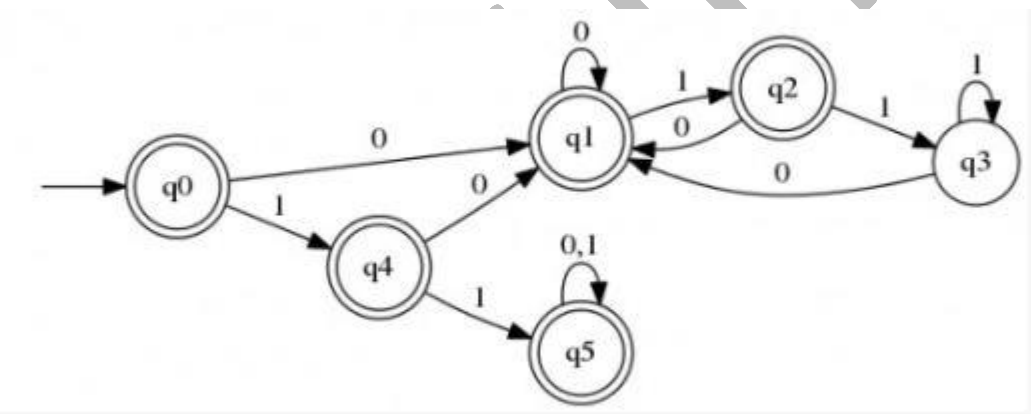
(D) Neither L1 nor L2 are regular

Answer: (A)

(A) is True. Though at first look both L1L1 and L2L2 looks non-regular, L1L1 is in fact regular. The reason is the relation between 110110 and 011011.

We cannot have two 110’110’s in a string without a 011011 or vice verse. And this would mean that we only need a finite number of states to check for acceptance of any word in this language.

That was just an intuitive explanation. Now I say that L contains all binary strings starting with 1111. Yes, if a binary string starts with 1111, it can never have more no. of 011011 than 110110.



Q If s is a string over (0 + 1)* then let $n_0(s)$ denote the number of 0’s in s and $n_1(s)$ the number of 1’s in s. Which one of the following languages is not regular? **(GATE-2006) (2 Marks)**

(A) $L = \{s \in (0+1)^* \mid n_0(s) \text{ is a 3-digit prime}\}$

(B) $L = \{s \in (0+1)^* \mid \text{ for every prefix } s' \text{ of } s, |n_0(s') - n_1(s')| \le 2\}$

(C) $L = \{s \in (0+1)^* \mid |n_0(s) - n_1(s)| \le 4\}$

(D) $L = \{s \in (0+1)^* \mid n_0(s) \text{ mod } 7 = n_1(s) \text{ mod } 5 = 0\}$

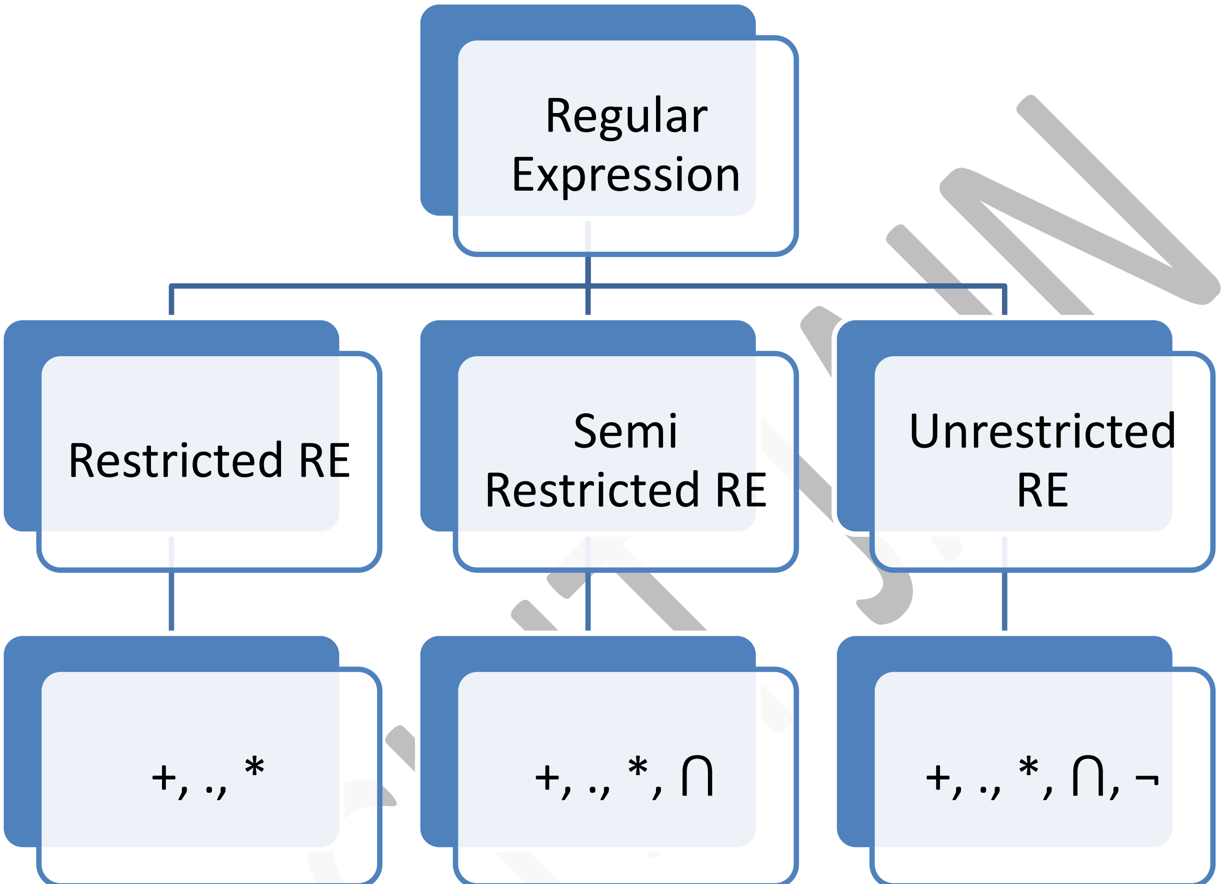
Answer: (C)

Q Let Σ be the set of all bijections from {1, ..., 5} to {1, ..., 5}, where id denotes the identity function, i.e. id(j) = j, $\forall_j$. Let ° denote composition on functions. For a string $x = x_1x_2 \dots x_n \in \Sigma^n$, $n \ge 0$, let $\pi(x) = x_1 \circ x_2 \circ \dots \circ x_n$. Consider the language $L = \{x \in \Sigma^* \mid \pi(x) = id\}$. The minimum number of states in any DFA accepting L is __________. **(GATE-2019) (2 Marks)**

Answer: a

REGULAR EXPRESSIONS

One way of describing regular language is via the notation of regular expression. An expression of strings which represents regular language is called regular expression.



The regular expressions are useful for representing certain sets of strings in an algebraic fashion. We give a formal recursive definition of regular expressions over ∑ as follows:

Any terminal symbol (i.e. an element of ∑), ∈ and Φ are regular expressions **<u>(Primitive Regular Expressions)</u>**. A regular expression is valid iff it can be derived from a primitive regular by a finite number of applications of the above **regular set.**

- When we view *a* in ∑ as a regular expression, we denote it by **a**.
- The union of two regular expressions $\mathbf{r_1}$ and $\mathbf{r_2}$, written as $\mathbf{r_1 + r_2}$, is also a regular expression.
- The concatenation of two regular expressions $\mathbf{r_1}$ and $\mathbf{r_2}$, written as $\mathbf{r_1 r_2}$, is also a regular expression.
- The iteration (or closure) of a regular expression **r** written as **r*,** is also a regular expression.
- If R is a regular expression, then **(r)** is also a regular expression.

Any set(language) represented by a regular expression is called a **regular set (regular language)**. If for example, a, b ∈ ∑, then

- a denotes the L = {a}
- a + b denotes L = {a, b}
- a.b denotes L = {ab} concatenation
- a* denotes the set {∈, a, aa, aaa, ...} known as Kleene closure.
- (a + b)* denotes {a, b}*
- a^+ Positive closure {a, aa, aaa…}
- R = {a.b + a}b

The precedence order to solve is: * (Kleene Closure) > $^+$ Positive Closure > Concatenation > Union

<u>**IDENTITIES FOR REGULAR EXPRESSIONS**</u>- Two regular expressions P and Q are equivalent (we write P = Q) if P and Q represent the same set of strings.

Q Let S and T be language over Σ = {a, b} represented by the regular expressions $(a + b^*)^*$ and $(a + b)^*$, respectively. Which of the following is true? **(GATE-2000) (1 Marks)**

(A) $S \subset T$ **(B)** $T \subset S$ **(C)** $S = T$ **(D)** $S \cap T = \phi$

Answer: (C)

- Every regular expression can generate only one regular language but, a regular language can be generated by more than one regular expression I.e. means two different regular expression can generate same language.
- Two regular expression are said to be equal if they generate same language.
- For e.g., $r_1 = a^*$, $r_2 = (aa)^*$

FINITE AUTOMATA AND REGULAR EXPRESSIONS

Q Design a regular express that represent a language 'L', where L={a} over the alphabet ∑={a}.

Q design a regular express that represent all strings over the alphabet ∑ = {a, b}, where every accepted string 'w' starts with substring s

i) s = b **ii)** s = ab **iii)** s = abb

Q design a regular express that represent all strings over the alphabet ∑ = {a, b}. where every accepted string 'w' ends with substring 's'.

i) s = ab **ii)** s = aa **iii)** s = bab

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b}. where every accepted string 'w' contains sub string s.

i) abb **ii)** aba

Q No of states in a minimal in a that accepts all the strings of a, b such that every string must have 'aabab' as sub stray

a) 5 **b)** 6 **c)** 7 **d)** none

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b} such that every accepted string start and end with a.

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b} such that every accepted string start and end with same symbol.

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b} such that every accepted string start and end with different symbol.

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b} such that every accepted string w, is like w=SX.

i) s= aa/bb   ii) s=aaa/bbb

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b} such that every accepted string w, is like w=XS.

i) s= aa/bb   ii) s=aaa/bbb

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b} such that every accepted string w, is like w=XSX.

i) s= aa/bb   ii) s=aaa/bbb

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b}, such that every string 'w' accepted must be like

i) |w| = 3   **ii)** |w|<=3   **iii)** |w|>=3

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain exactly two a's.

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain at least two a's.

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b}, such that every string accepted must contain at most two a's.

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b} such that for every accepted string 2nd from left end is always b.

Q Design a regular express that represent all strings over the alphabet ∑ = {a, b} such that for every accepted string 4th from left end is always a.

Q Design a regular express that represent all strings over the alphabet ∑= {a, b}, such that every string 'w' where |W| = 0(mod 3)?

Q Design a regular express that represent all strings over the alphabet $\sum= \{a, b\}$, such that every string 'w' where $|W| = 3(\text{mod } 4)$?

Q Design a regular express that represent all strings over the alphabet $\sum= \{a, b\}$, such that every string 'w' where $|W|_a = 0(\text{mod } 3)$?

Q Design a regular express that represent all strings over the alphabet $\sum= \{a, b\}$, such that every string 'w' where $|W|_a = 2(\text{mod } 3)$?

Q Design a regular express that represent all strings over the alphabet $\sum = \{a, b\}$, such that every string accepted must contain be like, no of a = 0(mod 2) || no of b = 0(mod 2)?

Q $L = \{a^m b^n \mid m, n >= 0\}$

Q $L = \{a^m b^n \mid m>=1, n >= 1\}$

Q $L = \{a^m b^n \mid m>=2, n >= 3\}$

Q $L = \{a^m b^n c^p \mid m, n, p >= 1\}$

Q $L = \{a^n b^n \mid n >= 1\}$

Q Write a regular expression for the language, $L = \{w\ c\ w^r \mid w, c\ (-\ \{a, b\}^+\}$

Q $L = \{a^m b^n \mid m + n = \text{even}\}$

Algebraic Properties of regular expressions-

- **Closure Property** - Regular expressions satisfy closure property with respect to Union, Concatenation and kleene closure. If R_1 and R_2 are regular expression then the following will also be regular expression.
  - $r_1 + r_2$
  - $r_1.r_2$
  - r_1*
  - r_1^+
- **Associative Property**- Regular expression satisfy associative property with respect to union and intersection.
  - $(r_1 + r_2) + r_3 = r_1 + (r_2 + r_3)$
  - $(r_1. r_2). r_3 = r_1. (r_2. r_3)$
- **Identity Property**- The identity property is satisfied as follows-
  - $r. \varepsilon = r$
  - $r + \varphi = r$
- **Inverse Property**- The inverse property is only satisfied with respect to concatenation, for union inverse property isn't satisfied.
  - $r. () = \varepsilon$
  - $r + () = \varphi$
- **Commutative Property**-Regular expressions are commutative with respect to union but not with respect to concatenation.
  - $r_1 + r_2 = r_2 + r_1$
  - $r_1. r_2 != r_2. r_1$
- **Distributive Property**-Regular expression satisfy this property as follows-
  - $r_1(r_2+r_3) = r_1r_2+r_1r_3$
  - $(r_1+r_2) r_3 = r_1r_3+r_2r_3$
  - $r_1+(r_2. r_3) != (r_1+r_2)(r_1+r_3)$
  - $(r_1.r_2) + r_3 = (r_1 + r_3). (r_2 + r_3)$
- **Idempotent Property**-Regular expressions satisfies idempotent property with respect to union but not with respect to concatenation.
  - $r_1+r_1=r_1$
  - $r_1. r_1 != r_1$

Identities for regular expressions

- $\phi + r = r$
- $\phi . r = r . \phi = \phi$
- $\in . r = r . \in = r$
- $\in^* = \in$ and $\phi^* = \in$

- $r^* r^* = r^*$
- $r . r^* = r^* . r$
- $r^+ . r^* = r^+$
- $r^+ r^+ = r^+ - r$

- $r^+ \cup r^* = r^*$
- $r^+ \cap r^* = r^+$

- $(r^*)^* = r^*$
- $(r^+)^* = r^*$
- $((r^+)^*)^* = r^*$
- $((r^*)^*) r^+ = r^+$

- $(r_1+r_2)^* = (r_1^* + r_2^*)^*$
- $(r_1+r_2)^* = (r_1 + r_2^*)^*$
- $(r_1+r_2)^* = (r_1^* + r_2)^*$
- $(r_1+r_2)^* = (r_1^* + r_2^*)$
- $(r_1+r_2)^* = (r_1^* . r_2^*)^*$
- $(r_1+r_2)^* = (r_1 . r_2^*)^*$
- $(r_1+r_2)^* = (r_1^* . r_2)^*$
- $(r_1+r_2)^* = (r_1 . r_2)^*$
- $(r_1+r_2)^* = (r_1^* . r_2^*)$
- $r_1(r_2.r_1)^* = (r_1.r_2)^* r_1$
-
- $L(r^*) = [L(r)]^*$

- $L(r^+) = [L(r)]^+$
- $L(r_1+R_2) = L(r_1) \cup L(r_2)$
- $L(r_1.r_2) = L(r_1).L(r_2)$
- $L((r_1)) = L(r_1)$

Q Which of the following are equal

a) a* **b)** (aa)* **c)** a(aa)* **d)** (a + ε) a*

Q Which one of the following languages over the alphabet {0,1} is described by the regular expression: (0+1)*0(0+1)*0(0+1)*? **(GATE-2009) (2 Marks)**

(A) The set of all strings containing the substring 00.

(B) The set of all strings containing at most two 0's.

(C) The set of all strings containing at least two 0's.

(D) The set of all strings that begin and end with either 0 or 1.

Answer: (C)

Q Which of the following regular expressions describes the language over {0, 1} consisting of strings that contain exactly two 1's? **(GATE-2008) (1 Marks)**

(A) (0 + 1) * 11(0 + 1) * **(B)** 0 * 110 *

(C) 0 * 10 * 10 * **(D)** (0 + 1) * 1(0 + 1) * 1 (0 + 1) *

Answer: (C)

Q Which of the following regular expression identities are true? **(NET-DEC-2012)**

(A) (r + s)* = r* s* **(B)** (r + s)* = r* + s*

(C) (r + s)* = (r*s*)* **(D)** r* s* = r* + s*

Ans: c

Q Consider the following identities for regular expressions: (a) (r + s)* = (s + r)* (b) (r*)* = r* (c) (r* s*)* = (r + s)* Which of the above identities are true? **(NET-AUG-2016)**

a) (a) and (b) only **b)** (b) and (c) only

c) (c) and (a) only

d) (a), (b) and (c)

Ans: d

Q The regular expression 0*(10*)* denotes the same set as **(GATE-2003) (1 Marks)**

(A) (1*0)*1* **(B)** 0 + (0 + 10)* **(C)** (0 + 1)* 10(0 + 1)* **(D)** none of these

Answer: (A)

Q Which of the following is correct

a) (xx)*y = x(xy)*

b) (xy)*x = x(yx)*

c) x(xy)* = (xx)*y

d) (xy)* = (yx)*

Ans: b

Q Which one of the following regular expressions is NOT equivalent to the regular expression (a + *b* + *c)* *? **(GATE-2004) (2 Marks)**

(A) (a* + b* + c*)* **(B)** (a*b*c*)* **(C)** ((ab)* + c*)* **(D)** (a*b* + c*)*

Answer: (C)

Q Consider the languages L1 = Φ and L2 = {a}. Which one of the following represents L1 $L2^*$U $L1^*$ **(GATE-2013) (1 Marks)?**

(A) $\{\epsilon\}$ (B) Φ (C) a^* (D) $\{\epsilon, a\}$

Answer:(A)

Q Consider the languages $L_1 = \phi$ and $L_2 = \{1\}$. Which one of the following represents $L_1{}^* \cup L_1{}^* L_2{}^*$? **(NET-JAN-2017)**

a) {∈} **b)** {∈, 1} **c)** Φ **d)** 1*

Ans: d

Let r = a(a + b)*, s = aa*b and t = a*b be three regular expressions.
Consider the following :

(i) $L(s) \subseteq L(r)$ and $L(s) \subseteq L(t)$

(ii) $L(r) \subseteq L(s)$ and $L(s) \subseteq L(t)$

Choose the correct answer from the code given below : **(NET-DEC-2018)**

a) Only I **b)** Only ii **c)** Both i and ii **d)** Neither i nor ii

Ans: a

Q Which one of the following regular expressions represents the language: the set of all binary strings having two consecutive 0s and two consecutive 1s? **(GATE-2016) (2 Marks)**

(A) $(0+1)^*0011(0+1)^* + (0+1)^*1100(0+1)^*$

(B) $(0+1)^*(00(0+1)^*11 + 11(0+1)^*00)(0+1)^*$

(C) $(0+1)^*00(0+1)^* + (0+1)^*11(0+1)^*$

(D) $00(0+1)^*11 + 11(0+1)^*00$

Answer: (B)

Q The regular expression for the complement of the language $L = \{a^n b^m \mid n \geq 4, m \leq 3\}$ is: **(NET-JULY-2016)**

a) (∈ + a + aa + aaa) b* + a* bbbb* + (a + b)* ba(a + b)*

b) (∈ + a + aa + aaa) b* + a* bbbbb* + (a + b)* ab(a + b)*

c) (∈ + a + aa + aaa) + a* bbbbb* + (a + b)* ab(a + b)*

d) (∈ + a + aa + aaa) b* + a* bbbbb* + (a + b)* ba(a + b)*

Ans: d

Q The regular expression corresponding to the language L where L = { x ϵ {0, 1}* |x ends with 1 and does not contain substring 00 } is: **(NET-JUNE-2015)**

a) (1 + 01) * (10 + 01) **b)** (1 + 01) * 01

c) (1 + 01) * (1 + 01)

d) (10 + 01) * 01

Ans: c

Q Regular expression for the language L = {w ∈ {0, 1}* | w has no pair of consecutive zeros} is **(NET-SEP-2013)**

(A) (1 + 010)*

(B) (01 + 10)*

(C) (1 + 010)* (0 + ∈)

(D) (1 + 01)* (0 + ∈)

Ans: d

Q Regular expression for the complement of language $L = \{a^n b^m \mid n \ge 4, m \le 3\}$ is **(NET-DEC-2014)**

(A) (a + b)* ba(a + b)*

(B) a* bbbbb*

(C) (∈ + a + aa + aaa)b* + (a + b)* ba(a + b)*

(D) None of the above

Ans: d

Q The number of strings of length 4 that are generated by the regular expression $(0|\in)1^{+}2(3|\in)$, where | is an alternation character and {+, *} are quantification characters, is: **(NET-AUG-2016)**

a) 08

b) 09

c) 10

d) 12

Ans: d

Q find the no of strings of length <= 3 generated by the $(a + ab)^{+}$

Q which of the following statements are correct

a) r^*, r^+ always represent infinite language.

b) r^*, r^+ always represent infinite language.

c) $r^* = r^+$ if and only if r = ε

d) $r^* = r^+ = r$ if r = Φ

EQUIVALENCE BETWEEN REGULAR EXPRESSION AND FINITE AUTOMATA-

ARDEN'S THEOREM is the mechanism for the construction of a regular expression from a finite automaton.

Steps used-

1.For, every individual state of the DFA, write an expression for every incoming and outgoing input alphabet.

2.Apply Arden's theorem as follows-

a) If P is free from NULL, then equation R=Q+RP has unique solution, R=QP*

b) If P contains NULL, then equation R=Q+RP has many infinitely many solutions.

Q Consider a DFA and convert it into regular expression using Arden's theorem?

$\delta(A, a) = A$

$\delta(A, b) = B$

$\delta(B, a) = B$

$\delta(B, b) = B$

A is the initial state and B Is the final state

Q Consider a DFA and convert it into regular expression using Arden's theorem?

$\delta(A, b) = A$

$\delta(A, b) = B$

$\delta(B, a) = C$

$\delta(C, b) = C$

A is the initial state and B,C Is the final state

Q Consider alphabet $\sum = \{0, 1\}$, the null/empty string $\in$ and the sets of strings X_0, X_1 and X_2 generated by the corresponding non-terminals of a regular grammar. X_0, X_1 and X_2 are related as follows **(GATE-2015) (2 Marks)**

$X_0 = 1\ X_1$ $\quad$ $X_1 = 0\ X_1 + 1\ X_2$ $\quad$ $X_2 = 0\ X_1 + \{\in\}$

Which one of the following choices precisely represents the strings in X0?

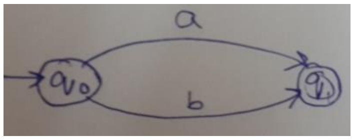
(A) 10 (0* + (10)*)1 **(B)** 10 (0* + (10)*)*1

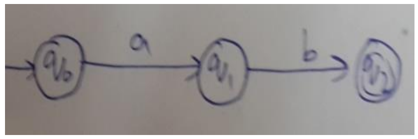
(C) 1(0* + 10)*1 **(D)** 10 (0 + 10)*1 + 110 (0 + 10)*1

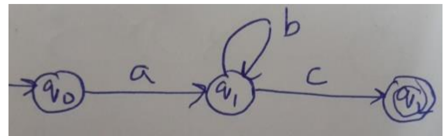
Answer: (C)

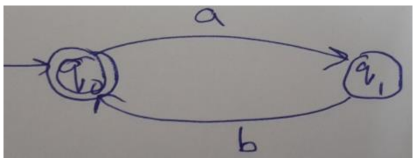
Conversion from Finite Automata to Regular expression

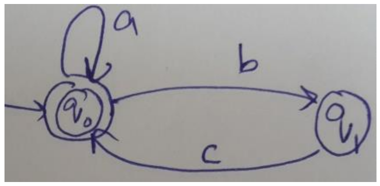
Q Write regular expressions for the following machines

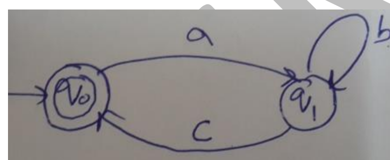


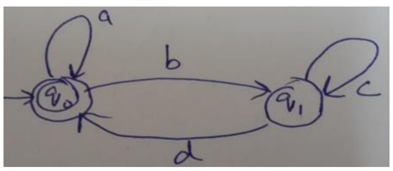


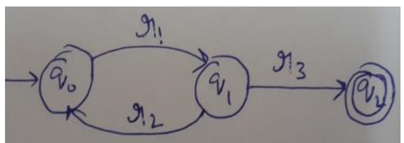
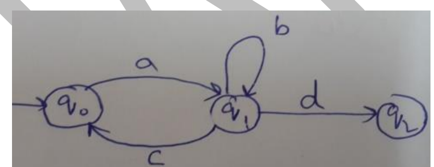


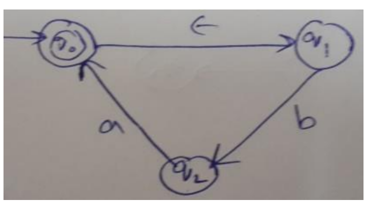


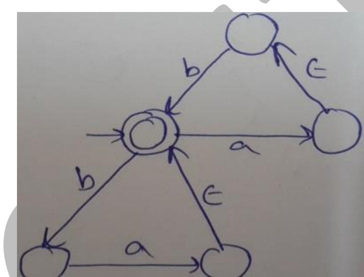


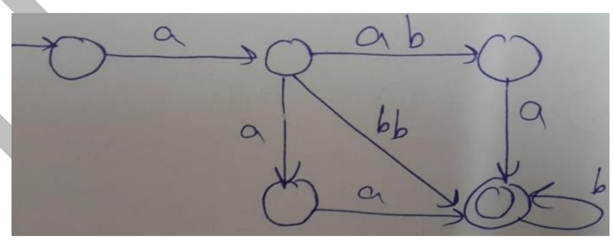


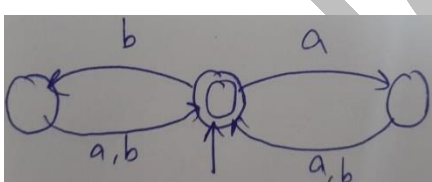


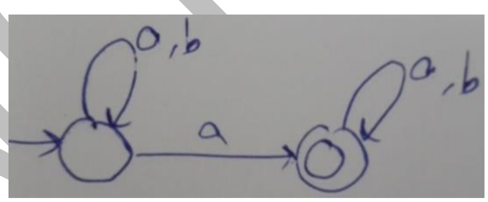



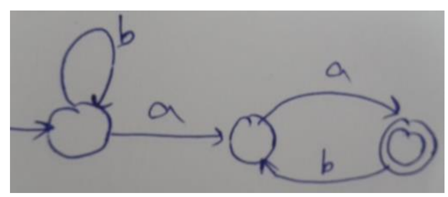


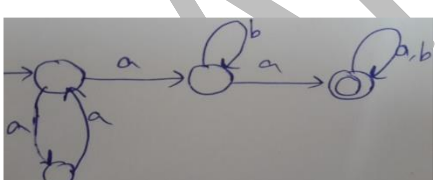


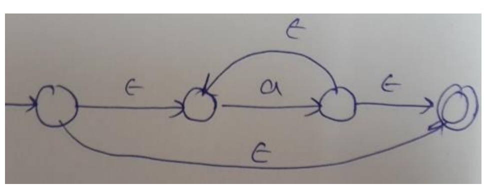


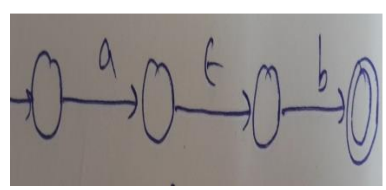


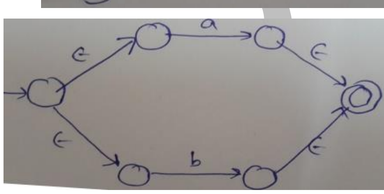




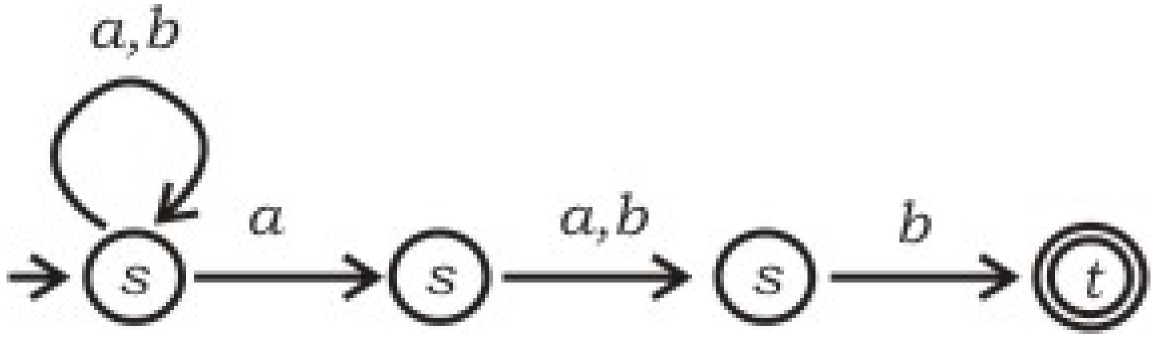








Q Which regular expression best describes the language accepted by the non-deterministic automaton below? **(GATE-2006) (1 Marks)**



(A) (a + b)* a(a + b)b **(B)** (abb)* **(C)** (a + b)* a(a + b)* b(a + b)* **(D)** (a + b)*

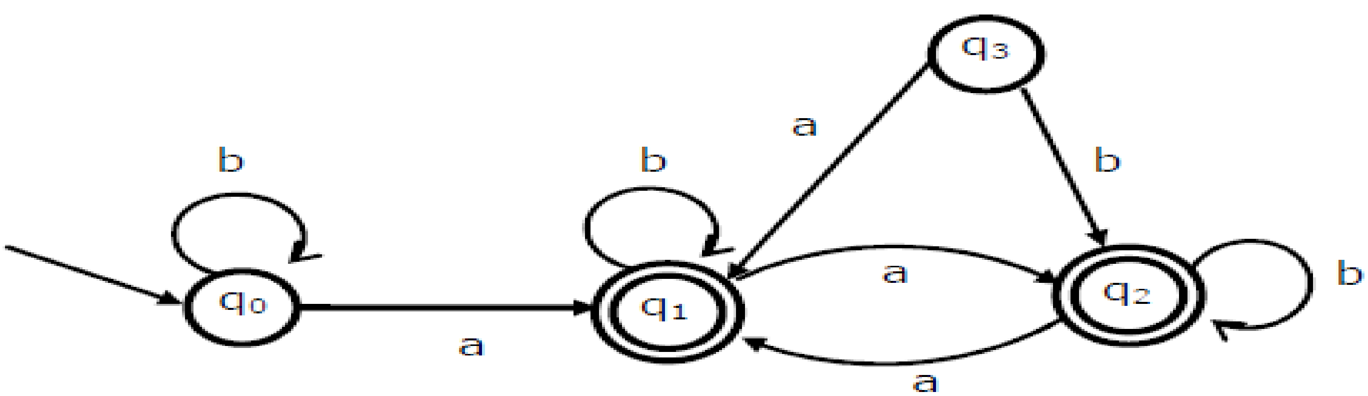
Answer: (A)

Q Let L = {w ∈ (0 + 1)* | w has even number of 1s}, i.e. L is the set of all bit strings with even number of 1s. Which one of the regular expressions below represents L? **(GATE-2010) (2 Marks)**

a) (0 * 10 * 1)*
b) 0 * (10 * 10*)*
c) 0*(10 * 1*)*0*
d) 0 * 1(10 * 1)*10*

ANSWER B

Q Consider the following Finite State Automaton. The language accepted by this automaton is given by the regular expression **(GATE-2007) (2 Marks)**



(A) $b^*ab^*ab^*ab^*$ (B) $(a+b)^*$ (C) $b^*a(a+b)^*$ (D) $b^*ab^*ab^*$

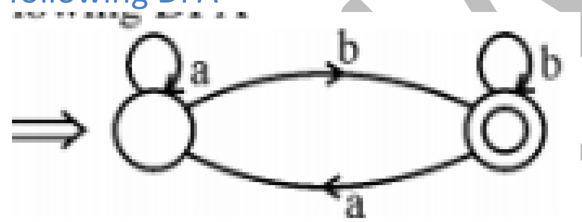
Answer: (C)

Q Consider the automata given in previous question. The minimum state automaton equivalent to the above FSA has the following number of states **(GATE-2007) (1 Marks)**

(A) 1 **(B)** 2 **(C)** 3 **(D)** 4

Answer: (B)

Q The regular expression for the following DFA

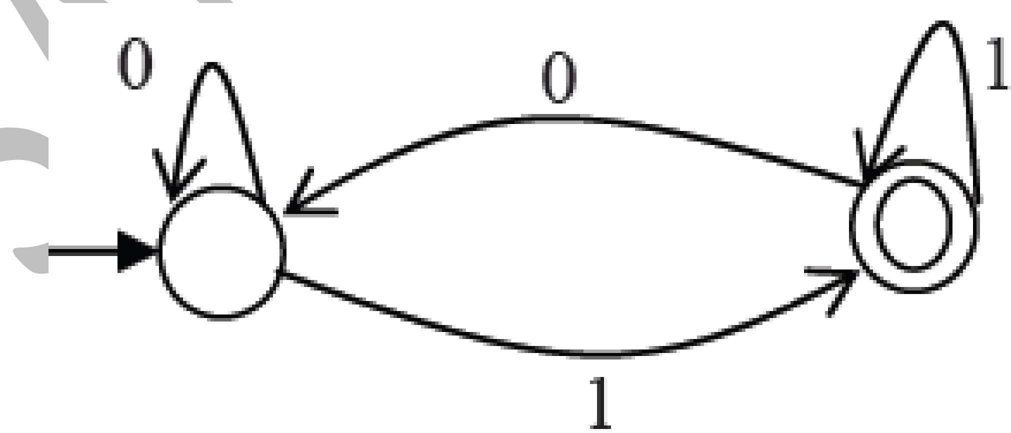


Is **(NET-JUNE-2012)**

(A) ab*(b + aa*b)*
(B) a*b(b + aa*b)*
(C) a*b(b* + aa*b)
(D) a*b(b * + aa*b)*

Ans: d

Q Which of the regular expressions given below represent the following DFA? **(GATE-2014) (2 Marks)**

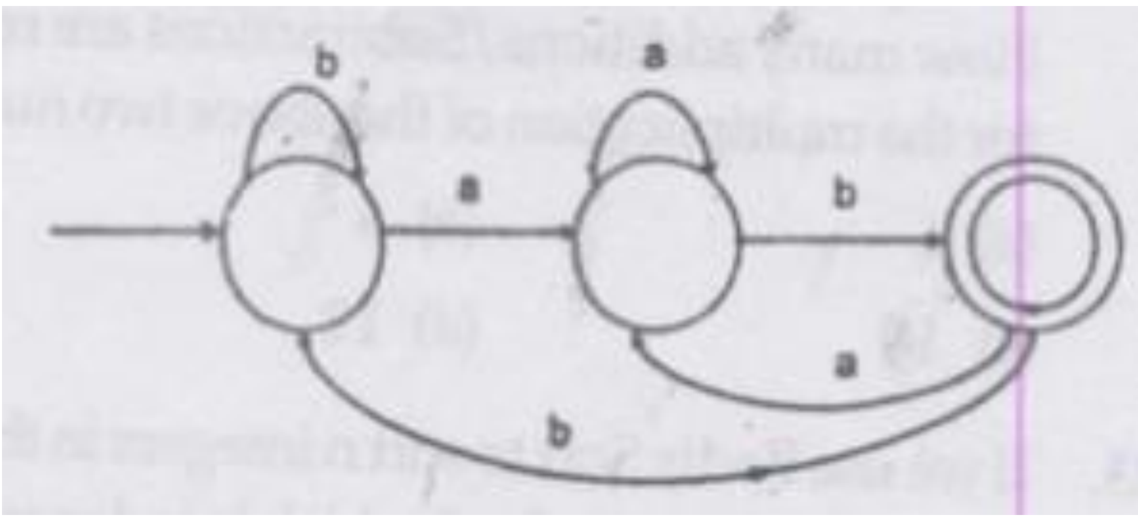


I) 0*1(1+00*1)* **II)** 0*1*1+11*0*1 **III)** (0+1)*1

(A) I and II only **(B)** I and III only **(C)** II and III only **(D)** I, II, and III

Answer: (B)

Q If the final states and non-final states in the DFA below are interchanged, then which of the following languages over the alphabet {a, b} will be accepted by the new DFA? **(GATE-2008) (1 Marks)**



(A) Set of all strings that do not end with ab
(B) Set of all strings that begin with either an a or a b
(C) Set of all strings that do not contain the substring ab,
(D) The set described by the regular expression b*aa*(ba)*b*
Answer: (A)

Conversion from Regular expression in Finite Automata

R^*

$(R_1.R_2)^*$

$(R_1+ R_2)^*$

$(R_1^*R_2.R_3^*)$

R = a*b(ab)*

R = (a+ ba)*ab*

R = (aa+ aaa)*

R = (a+ aaaaa)*

{a.b + a}b

0 * 10 * 10 *

((ab)* + c*)*

(∈ + a + aa + aaa)b* + (a + b)* ba(a + b)*

0*(10 * 1*)*0*

a*b(b * + aa*b)*

((11*0+0)(0 + 1)*0*1*)

(ab+ bc+ acc).(a+ bc)*.(∈+ bc*a)*

Q Consider the regular language L = (111 + 11111)*. The minimum number of states in any DFA accepting this language is: **(GATE-2006) (1 Marks)**

(A) 3 **(B)** 5 **(C)** 8 **(D)** 9

Answer: (D)

Q The number of states in the minimal deterministic finite automaton corresponding to the regular expression (0 + 1)*(10) is _____________ **(GATE-2015) (1 Marks)**

(A) 2 **(B)** 3 **(C)** 4 **(D)** 5

Answer: (B)

Q The number of states in the minimum sized DFA that accepts the language defined by the regular expression $(0+1)^*(0+1)(0+1)^*$ is ____________________ **(GATE-2016) (2 Marks)**

[Note that this question was originally asked as Fill-in-the-Blanks type]

(A) 2 **(B)** 3 **(C)** 4 **(D)** 5

Answer: (A)

Q Let T 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)? **(GATE-2015) (2 Marks)**

(A) 4 **(B)** 5 **(C)** 6 **(D)** 8

Answer: (B)

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 ______. **(GATE-2017) (1 Marks)**

ANSWER 4

Q Which of the following statements is TRUE about the regular expression 01*0? **(GATE-2005) (1 Marks)**

(A) It represents a finite set of finite strings.

(B) It represents an infinite set of finite strings.

(C) It represents a finite set of infinite strings.

(D) It represents an infinite set of infinite strings

Answer: (B)

Q Which of the following regular expressions, each describing a language of binary numbers (MSB to LSB) that represents non-negative decimal values, does not include even values? **(NET-NOV-2017)**

a) 0*1+0*1* **b)** 0*1*0+1* **c)** 0*1*0*1+ **d)** 0+1*0*1*

Ans: c

Q The length of the shortest string NOT in the language (over Σ = {a, b}) of the following regular expression is ______________. **(GATE-2014) (2 Marks)**

a*b*(ba)*a*

(A) 2 **(B)** 3 **(C)** 4 **(D)** 5

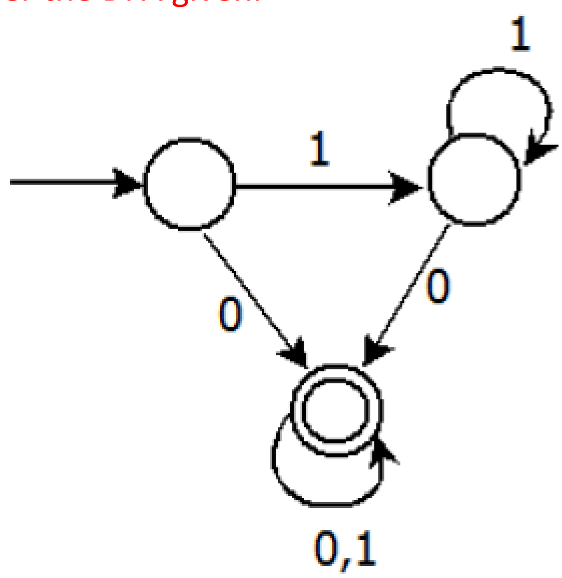
Answer: (B)

Q Consider the regular expression (a + b) (a + b) ... (a + b) (n-times). The minimum number of states in finite automaton that recognizes the language represented by this regular expression contains **(NET-JUNE-2012)**

(A) n states
(B) n + 1 states
(C) n + 2 states
(D) 2n states

Ans: b

Q Consider the DFA given.



Which of the following are FALSE? **(GATE-2013) (2 Marks)**

1. Complement of L(A) is context-free.
2. L(A) = L((11*0+0)(0 + 1)*0*1*)
3. For the language accepted by A, A is the minimal DFA.
4. A accepts all strings over {0, 1} of length at least 2.

a) 1 and 3 only
b) 2 and 4 only
c) 2 and 3 only
d) 3 and 4 only

Ans: d

Q which of the following re, over the Σ = {0, 1} denotes the set of all strings that not contain 100 as sub-string

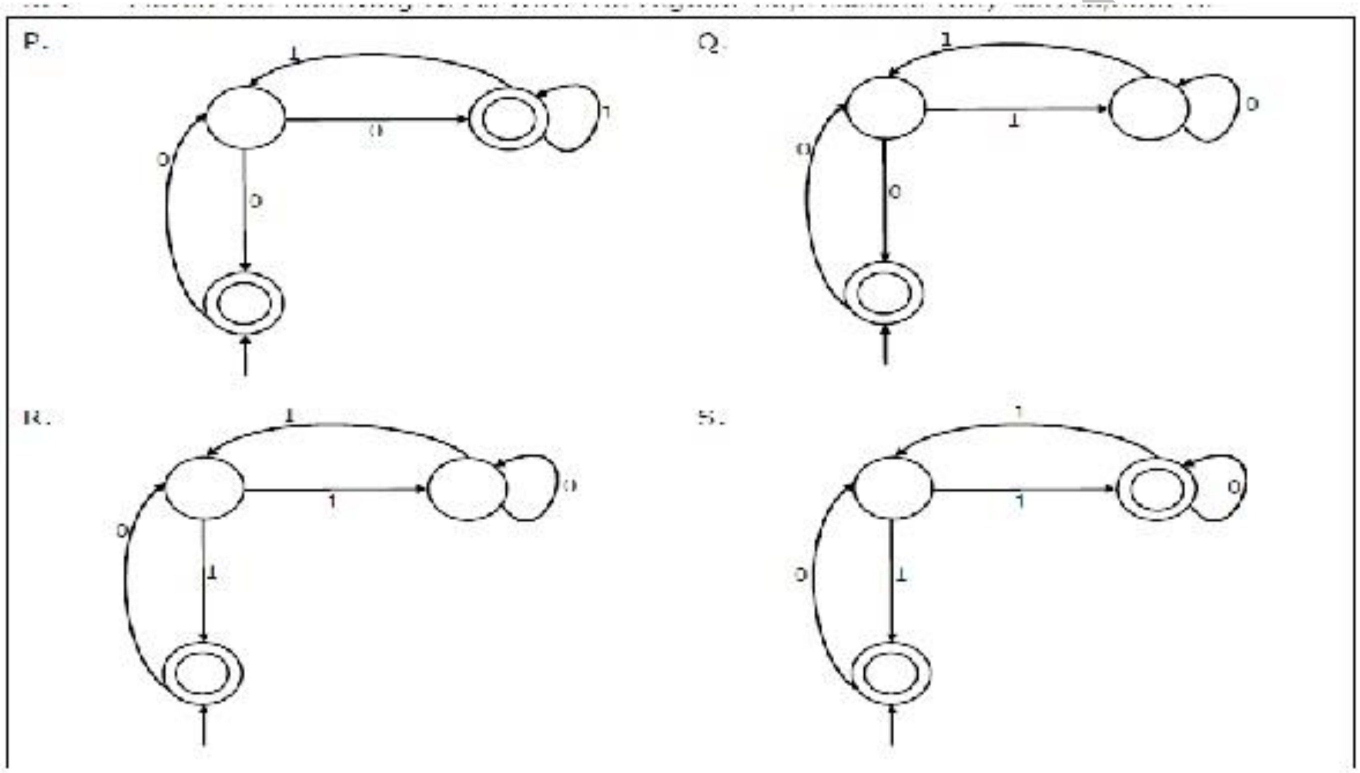
a) 0*(0+1)*
b) 0*1010*
c) 0*1*01
d) 0*(10+1)*

Q the string 1101 does not belong to the set

a) 110*(0+1)
b) 1(0+1)*101
c) (10)*(10)*(00+11)*
d) (00 + (11)*01)*

Q Match the following NFAs with the regular expressions they correspond to **(GATE-2008) (2 Marks)**

1. ϵ + 0(01*1 + 00) * 01*
2. ϵ + 0(10 *1 + 00) * 0
3. ϵ + 0(10 *1 + 10) *1
4. ϵ + 0(10 *1 + 10) *10 *



(A) P – 2, Q – 1, R – 3, S – 4
(B) P – 1, Q – 3, R – 2, S – 4
(C) P – 1, Q – 2, R – 3, S – 4
(D) P – 3, Q – 2, R – 1, S – 4

Answer: (C)

Q Pushdown automata can recognize language generated by_________. **(NET-JULY-2018)**
(1) Only context free grammar
(2) Only regular grammar
(3) Context free grammar or regular grammar
(4) Only context sensitive grammar
Ans: c

Q Which of the following languages is generated by the given grammar? (**GATE-2016) (2 Marks)**

S → aS | bS | ε

(A) $\{a^n b^m \mid n, m \geq 0\}$

(B) {w ∈ {a, b}* | w has equal number of a's and b's}

(C) $\{a^n \mid n \geq 0\} \cup \{b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$

(D) {a, b}*

ANSWER -D

Q Let L denotes the language generated by the grammar S → 0S0/00. (**GATE-2000) (1 Marks)**
Which of the following is true?
a) $L = 0^+$
b) L is regular by not 0^+
c) L is context free but not regular
d) L is not context free
Ans: b

Q Language L_1 is defined by the grammar: **S1 -> aS1b | ε**

Language L_2 is defined by the grammar: **S2 -> abS2 | ε**

Consider the following statements:

P: L_1 is regular

Q: L_2 is regular

Which one of the following is TRUE? (**GATE-2007) (2 Marks)**

a) Both P and Q are true

b) P is true and Q is false

c) P is false and Q is true

d) Both P and Q are false

ANSWER C

Q S -> aSa | bSb | a | b **(GATE-2009) (2 Marks)**
The language generated by the above grammar over the alphabet {a, b} is the set of
(A) All palindromes
(B) All odd length palindromes.
(C) Strings that begin and end with the same symbol

(D) All even length palindromes

Answer: (B)

Q The grammar with production rules S → aSb | SS | λ generates language L given by: **(NET-JUNE-2013)**

(A) L = {w ∈ {a, b}* | n_a (w) = n_b(w) and n_a(v) ≥ n_b(v) where v is any prefix of w}

(B) L = {w ∈ {a, b}* | n_a(w) = n_b(w) and n_a(v) ≤ n_b(v) where v is any prefix of w}

(C) L = {w ∈ {a, b}* | n_a (w) ≠ n_b(w) and n_a (v) ≥ n_b(v) where v is any prefix of w}

(D) L = {w ∈ {a, b}* | n_a (w) ≠ n_b(w) and n_a(v) ≤ n_b(v) where v is any prefix of w}

Ans: a

Q Which of the following definitions generates the same Language as L, where L = {WW^R | W ∈ {a, b}*} **(NET-DEC-2012)**

(A) S → asb | bsa | ∈

(B) S → asa | bsb | ∈

(C) S → asb | bsa | asa | bsb | ∈

(D) S → asb | bsa | asa | bsb | ∈

Ans: b

Q The following CFG **(NET-JUNE-2012)**

S → aB | bA

A → a | as | bAA

B → b | bs | aBB

generates strings of terminals that have

(A) odd number of a's and odd number of b's

(B) even number of a's and even number of b's

(C) equal number of a's and b's

(D) not equal number of a's and b's

Ans: c

Q A context free grammar for L = {w | n_0 (w) > n_1 (w)} is given by: **(NET-JUNE-2015)**

a) S → 0 | 0 S | 1 S S

b) S → 0 S | 1 S | 0 S S | 1 S S | 0 | 1

c) S → 0 | 0 S | 1 S S | S 1 S | S S 1

d) S → 0 S |1 S | 0 | 1

Ans: c

Q The following Context-Free Grammar (CFG): **(NET-DEC-2014)**

S → aB | bA

A → a | as | bAA

B → b | bs | aBB

will generate

(A) odd numbers of a's and odd numbers of b's

(B) even numbers of a's and even numbers of b's

(C) equal numbers of a’s and b’s
(D) different numbers of a’s and b’s
Ans: a

Q Identity the language generated by following grammar where S is the start variable. (**GATE-2017) (2 Marks)**

S-->XY

X--> aX | a

Y--> aYb | ∈

a) $\{a^m b^n | m >= n, n > 0\}$

b) $\{a^m b^n | m >= n, n >= 0\}$

c) $\{a^m b^n | m > n, n >= 0\}$

d) $\{a^m b^n | m > n, n > 0\}$

Ans: c

Q The equivalent production rules corresponding to the production rules
S → Sα1 |Sα2 | β1 | β2 is **(NET-JUNE-2013)**
(A) S → β1 | β2, A → α1A | α2A | λ
(B) S → β1| β2| β1A | β2A, A → α1A | α2A
(C) S → β1 | β2, A → α1A | α2A
(D) S → β1 | β2 | β1A | β2A, A → α1A | α2A | λ
Ans: d

Q Which one of the following grammars generates the language $L = \{a^i b^j \mid i \neq j\}$ **(GATE-2006) (1 Marks)**

(A)

$S \to AC|CB$

$C \to aCb|a|b$

$A \to aA|\in$

$B \to Bb|\in$

(B) $S \to aS|Sb|a|b$

(C)

$S \to AC|CB$

$C \to aCb|\in$

$A \to aA|\in$

$B \to Bb|\in$

(D)

$S \to AC|CB$

$C \to aCb|\in$

$A \to aA|a$

$B \to Bb|b$

Answer: (D)

Q Which of the following definitions below generates the same language as L, where **(GATE -1995) (2 Marks)**

L = { x^n y^n such that n > = 1} ?

I. E → xEy | xy **II.** xy | (x^+ syy^+) **III.** x^+ y^+

a) I only **b)** I and II **c)** II and III **d)** II only

Ans: a

Q Consider the following context-free grammars:

G_1: $S \to aS|B$, $B \to b|bB$

G_2: $S \to aA|bB$, $A \to aA|B|\varepsilon$, $B \to bB|\varepsilon$

Which one of the following pairs of languages is generated by G1 and G2, respectively **(GATE-2016) (2 Marks)**

(A) $\{a^m b^n | m > 0 \text{ or } n > 0\}$ and $\{a^m b^n | m > 0 \text{ and } n > 0\}$

(B) $\{a^m b^n | m > 0 \text{ and } n > 0\}$ and $\{a^m b^n | m > 0 \text{ or } n \geq 0\}$

(C) $\{a^m b^n | m \geq 0 \text{ or } n > 0\}$ and $\{a^m b^n | m > 0 \text{ and } n > 0\}$

(D) $\{a^m b^n | m \geq 0 \text{ and } n > 0\}$ and $\{a^m b^n | m > 0 \text{ or } n > 0\}$

Ans: d

Q In the context-free grammar below, S is the start symbol, a and b are terminals, and ϵ denotes the empty string **(GATE-2006) (1 Marks)**

S → aSa | bSb | a | b | ϵ

Which of the following strings is NOT generated by the grammar?

(A) aaaa **(B)** baba **(C)** abba **(D)** babaaabab

Answer: (B)

Q Consider the context-free grammars over the alphabet {a, b, c} given below. S and T are non-terminals. (**GATE-2017) (2 Marks)**

G1: S → aSb | T,T → cT | ϵ

G2: S → bSa | T,T → cT | ϵ

The language L(G1) ∩ L(G2) is

(A) Finite

(B) Not finite but regular

(c) Context-Free but not regular

(D) Recursive but not context-free

ANSWER B

Q Identity the language generated by following grammar where S is the start variable. **(GATE-2017) (2 Marks)**

S --> XY

X --> aX | a

Y --> aYb | ∈

a) $\{a^m b^n | m>=n, n>0\}$

b) $\{a^m b^n | m>=n, n>=0\}$

c) $\{a^m b^n | m>n, n>=0\}$

d) $\{a^m b^n | m>n, n>0\}$

ANSWER C

Q The two-grammar given below generate a language over alphabet {x, y, z) **(GATE – 2007) (2 Marks)**

G1: S → x | z | xS | zS | yB

B → y | z | yB | zB | zB

G2: S → y | z | yS | zS | xB

B → y | y | yS

Which one of the following choices describes the properties satisfied by the strings in these languages?

a) G1: No y appears before any x, G2: Every x is followed by at least one y

b) G1: No y appears before any x, G2: No x appears before any y

c) G1: No y appears after any x, G2: Every x is followed by at least one y

d) G1: No y appears after any x, G2 Every y is followed by at least one x

Ans: a

Q Given the following statements: **(NET-DEC-2013)**

S1: The grammars S → asb | bsa | ss | a and S → asb | bsa | a are not equivalent.

S2: The grammars S → ss | sss | asb | bsa | λ and S → ss | asb |bsa | λ are equivalent.

Which of the following is true?

(A) S1 is correct and S2 is not correct. **(B)** Both S1 and S2 are correct.

(C) S1 is not correct and S2 is correct. **(D)** Both S1 and S2 are not correct.

Ans: b

Q Consider the grammar given below **(GATE – 2007) (2 Marks)**

S → xB | yA

A → x | x S| y AA

B → y | y S| y BB

Consider the following strings.

i) xxyyx **ii)** xxyyxy **iii)** xyxy **iv)** yxxy **v)** yxx **vi)** xyx

Which of the above strings are generated by the grammar?

a) i), ii) and iii) **b)** ii), v) and vi) **c)** ii), iii) and iv) **d)** i), iii) and iv)

Ans: c

Q The context free grammar given by **S → XYX, X → aX | bX | λ, Y → bbb** generates the language which is defined by regular expression: **(NET-DEC-2015)**

a) (a + b)*bbb **b)** abbb(a + b)*

c) (a + b)*(bbb)(a + b)* **d)** (a + b)(bbb)(a + b)*

Ans C

Q The context free grammar for the language (NET-DEC-2013)

L = {$a^n b^m$ | n ≤ m + 3, n ≥ 0, m ≥ 0} is

(A) S → aaa A; A → aAb | B, B → Bb | λ

(B) S → aaaA | λ, A → aAb | B, B → Bb | λ

(C) S → aaaA | aa A | λ, A → aAb | B, B → Bb| λ

(D) S → aaaA | aa A | aA | λ, A → aAb | B, B → Bb | λ

Ans: d

Q Consider a CFG with the following productions. (GATE-2008) (1 Marks)

S → AA | B

A → 0A | A0 | 1

B → 0B00 | 1

S is the start symbol, A and B are non-terminals and 0 and 1 are the terminals. The language generated by this grammar is

(A) $\{0^n 10^{2n} \mid n \geq 1\}$

(B) $\{0^i 10^j 10^k \mid i, j, k \geq 0\} \cup \{0^n 10^{2n} \mid n \geq l\}$

(C) $\{0^i 10^j \mid i, j \geq 0\} \cup \{0^n 10^{2n} \mid n \geq l\}$

(D) The set of all strings over {0, 1} containing at least two 0's

(E) None of the above

Answer: (E)

Q Consider the following context-free grammar over the alphabet ∑ = {a, b, c} with S as the start symbol: **(GATE-2017) (2 Marks)**

S → abScT | abcT

T → bT | b

Which of the following represents the language generated by the above grammar?

a) $\{(ab)^n(cb)^n \mid n >= 1\}$

b) $\{((ab)^n c\, b^{m1} c\, b^{m2} \ldots c\, b^{mn} \mid n, m_1, m_2, \ldots, m_n >= 1\}$

c) $\{(ab)^n (cb^m)^n \mid n >= 1\}$

d) $\{(ab)^n (cb^n)^m \mid m, n >= 1\}$

ANSWER B

T→ bT | b, this production will generate any number of b's > 1

S→ abScT | abcT, this production will generate equal number of "ab" and "c" and for every "abc" any number of b's (> 1) after "abc".

Q The context free grammar for the language **$L = \{a^n b^m c^k \mid k = |n - m|, n > 0, m > 0, k > 0\}$** is (NET-JUNE-2014)

(A) S → S1 S3, S1 → aS1c | S2 | λ, S2 → aS2 b |λ, S3 → aS3 b| S4 | λ, S4 → bS4 c |λ

(B) S → S1 S3, S1→ aS1 S2 c | λ, S2 → aS2 b| λ, S3 → aS3 b| S4 |λ, S4 → bS4 c |λ

(C) S → S1|S2, S1→ aS1 S2 c | λ, S2 → aS2 b | λ, S3 → aS3 b | S4 |λ, S4 → bS4 c |λ

(D) S → S1 | S3 , S1→ aS1 c|S2 | λ, S2 → aS2 b | λ, S3 → a S3 b| S4 | λ, S4 → bS4 c | λ

Q The equivalent grammar corresponding to the grammar **(NET-JUNE-2012)**
G: S → aA
A → BB
B → aBb | ∈ is

(A) S → aA
A → BB
B → aBb

(B) S → a | aA
A → BB
B → aBb | ab

(C) S → a | aA
A → BB | B
B → aBb

(D) S → a |aA
A→ BB | B
B→ aBb | ab
Ans: d

Q Non-deterministic pushdown automaton that accepts the language generated by the grammar: S → aSS | ab is **(NET-SEP-2013)**

(A) $\delta(q_0, \lambda, z) = \{ (q_1, z)\};$

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

$\delta(q_0, b, B) = \{ (q_1, \lambda)\},$

$\delta(q_1, \lambda, z) = \{ (q_f, \lambda)\}$

(B) $\delta(q_0, \lambda, z) = \{ (q_1, Sz)\};$

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

$\delta(q_0, b, B) = \{ (q_1, \lambda)\},$

$\delta(q_1, \lambda, z) = \{ (q_f, \lambda)\}$

(C) $\delta(q_0, \lambda, z) = \{ (q_1, Sz)\};$

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

$\delta(q_0, b, \lambda) = \{ (q_1, B)\},$

$\delta(q_1, \lambda, z) = \{ (q_f, \lambda)\}$

(D) $\delta(q_0, \lambda, z) = \{ (q_1, z)\};$

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

$\delta(q_0, b, \lambda) = \{ (q_1, B)\},$

$\delta(q_1, \lambda, z) = \{ (q_f, \lambda)\}$

Ans B

Q A CFG G is given with the following productions where S is the start symbol, A is a non-terminal and a and b are terminals.

S → aS | A

A → aAb | bAa | ϵ

Which of the following strings is generated by the grammar above?

(A) aabbaba **(B)** aabaaba **(C)** abababb **(D)** aabbaab

Answer: (D)

Q What is the equivalent CFL for the following CFG?

S → aS / aSbS / ϵ

a) {x|x is a palindrome}

b) {x|x = $a^n b^n \ for\ n \geq 0$}

c) {x| each prefix of x has atleast as many a'S as b'S

d) {x|x has equal number of a'S and b'S}

Q The grammar **S → aaSbb | ab** can generate the set

a) $\{a^n b^n | n = 1,2,3, \dots\}$ **b)** $\{a^{2n+1} b^{2n+1}$|= n = 0, 1, 2,..}

c) $\{a^{2n+1}b^{2n+1}$ |= n = 1, 2, 3,..}   **d)** none of these

Q Which of the following grammar generates the language **L = {a^i b^j | i≠ j}**

a) S → AC / CB   C → aCb / a / b   A → aA / ε   B → Bb / ε

b) S → as / sb / a / b

c) S → Ac / CB   C → aCb / ε   A → aA / ε   B → Bb / ε

d) S → AC / CB   C → aCb / ε   A → aAa   B → bB / b

Q The following CFG

S → aS | bS | a | b is equivalent to the regular expression

1. (a* + b)*   **2.** $(a + b)^+$ **3.** (a + b)(a + b)*   **4.** (a + b)* (a + b)

a) 2 and 3 only   **b)** 2,3 and 4   **c)** All of the above   **d)** 3 and 4 only

Q The set $\{a^n b^n | n = 1,2,3 \ldots\}$ can be generated by the CFG

a) S → ab |aSb|ϵ   **b)** S → aaSbb | ab   **c)** S → ab | aSb   **d)** None of these

Q In the context – free grammar below, S is the start symbol, a and b are terminals, and ε denotes the empty string.

S → aSAb | ε

A → bA | ε

The grammar generates the language

a) ((a + b)*b)   **b)** $\{a^m b^n | m \leq n\}$

c) $\{a^m b^n | m = n\}$ **d)** a*b*

Q The following CFG

S → aB | bA

A → ba | aS | bAA

B → b | bS | aBB

generates strings of terminals that have

a) equal number of a's and b's

b) odd number of a's and even number of b's

c) even number of a's and even number of b's

d) odd number of a's and even number of a's

Q L_1 has the following grammar

S → aB / BA

A → bAA / aS / a

B → b / bS / aBB

L_2 has the following grammar
S → Sba / a
Which of the following statement is true about?
$L_3 = L_1 \cap L_2$ and $L_4 = L_{.1}.L_1 *$?
a) Both L_3 and L_4 are not $context$ $free$
b) L_3 is context free but L_4 is not
c) Both L_3 and L_4 are context free
d) L_4 is context free, but not L_3

Q What is the equivalent CFL for the following CFG **S → OS 1/ϵ**?
a) {X/X is a palindrome}
b) {X/X = 0^n1^n for $n \geq 0$}
c) {X/X = 0^n1^n for $n > 0$ }
d) {X/X = 0^n1^n for $n > 1$}

Q If G, = ({S}, {a}, {S → SS}, S), find language generated by G.
a) L(G) = φ
b) L(G) = a^n
c) L(G)= a*
d) L(G)= a^nba^n

Q Consider a grammar G as follows
S → aA
A → bbA
A → c
L(G) =?
a) L(G) = {abbc}
b) L(G) = {a b^n c; n ≥ 0}
c) L(G) = {a $b^{2n}c$; n > 0}
d) L(G) = {a b^{2n} c; n ≥ 0}

Q LL grammar for the language **L = {a^n b^m c^{n+m} | m≥0, n≥0}** is
(A) S → aSc | S1 ; S1 → bS1c | λ
(B) S → aSc | S1| λ ; S1 → bS1c
(C) S → aSc | S1| λ ; S1 → bS1c| λ
(D) S → aSc | λ ; S1 → bS1c| λ

Decision properties

- Following properties are decidable in case a CFL. Here we will use Grammar model to proof decision properties.

i) Emptiness

ii) Non-emptiness

iii) Finiteness

iv) Infiniteness

v) Membership

- Following properties are Undecidable in case a CFL.

i) Equality

ii) Ambiguity

Q Which of the following are decidable? **(GATE-2008) (2 Marks)**

I. Whether the intersection of two regular languages is infinite

II. Whether a given context-free language is regular

III. Whether two push-down automata accept the same language

IV. Whether a given grammar is context-free

1. I and II **2.** I and IV **3.** II and III **4.** II and IV

ANSWER B

Q Consider the following decision problems? **(GATE-2000) (2 Marks)**

(P_1) Does a given finite state machine accept a given string

(P_2) Does a given context free grammar generate an infinite number of stings

Which of the following statements is true?

(A) Both (P_1) and (P_2) are decidable **(B)** Neither (P_1) nor (P_2) are decidable

(C) Only (P_1) is decidable **(D)** Only (P_2) is decidable

Ans: a

Q Which one of the following statements is FALSE? **(GATE-2004) (1 Marks)**
a) There exist context free languages such that all the context free grammars generating them age ambiguous.
b) An unambiguous context free grammar always has a unique parse tree for each string of the language generated by it.
c) Both deterministic and non – deterministic pushdown automata always accept the same set of languages
d) A finite set of string from one alphabet is always a regular language.
Ans: b

Q Which of the following problems is undecidable? **(GATE-2014) (1 Marks)**
(A) Deciding if a given context-free grammar is ambiguous.
(B) Deciding if a given string is generated by a given context-free grammar.
(C) Deciding if the language generated by a given context-free grammar is empty.
(D) Deciding if the language generated by a given context-free grammar is finite.
Ans: a

Q Which of the following problems is undecidable? **(GATE-2007) (1 Marks)**
(A) Membership problem for CFGs
(B) Ambiguity problem for CFGs.
(C) Finiteness problem for FSAs.
(D) Equivalence problem for FSAs.
Answer: (B)

Q Which of the following problems is undecidable? **(NET-NOV-2017)**
a) To determine if two finite automata are equivalent
b) Membership problem for context free grammar
c) Finiteness problem for finite automata
d) Ambiguity problem for context free grammar
Ans: d

Q Which of the following are decidable? **(GATE-2008) (1 Marks)**
i) Whether the intersection of two regular languages is infinite
ii) Whether a given context-free language is regular
iii) Whether two push-down automata accept the same language
iv) Whether a given grammar is context-free

a) I and II **b)** I and IV **c)** II and III **d)** II and IV

ANSWER B

Q Which of the following problems is undecidable? **(GATE-2014) (1 Marks)**

(A) Deciding if a given context-free grammar is ambiguous.

(B) Deciding if a given string is generated by a given context-free grammar.

(C) Deciding if the language generated by a given context-free grammar is empty.

(D) Deciding if the language generated by a given context-free grammar is finite.

Answer: (A)

Closure Properties of Deterministic Context Free Languages

- Regular languages are closed under following operations
  - Complement
  - Intersection with regular set
  - Inverse Homeomorphism

- Regular languages are not closed under following operations
  - Union
  - Concatenation
  - Kleen closure
  - homomorphism
  - Substitution
  - Reverse operator
  - Intersection

Closure Properties of Context Free Languages

- Regular languages are closed under following operations
  - Union
  - Concatenation
  - Kleen Closure
  - Substitution
  - Homomorphism
  - Inverse Homomorphism
  - Reverse Operator
  - Intersection with regular set

- Regular languages are not closed under following operations
  - Intersection
  - Complement
  - Symmetric Difference

Q Context – free languages are (**GATE – 1992) (1 Marks)**

a) Closed under union
b) Closed under complementation
c) Closed under intersection
d) Closed under Kleene closure

Ans: a, d

Q Let L_1 and L_2 be any context-free language and R be any regular language. Then, which of the following is correct? **(GATE-2017) (2 Marks)**

I. $L_1 \cup L_2$ is context-free.
II. L_1' is context-free.
III. L_1 - R is context-free.
IV. $L_1 \cap L_2$

a) I, II and IV only
b) I and III only
c) II and IV only
d) I only

Ans: b

Q Consider the following languages over the alphabet ∑= {a, b, c}.
Let $L_1 = \{a^n b^n c^m \mid m, n >= 0\}$ and $L_2 = \{a^m b^n c^n \mid m, n >= 0\}$.
Which of the following are context-free languages? **(GATE-2017) (2 Marks)**

I. $L_1 \cup L_2$
II. $L_1 \cap L_2$

a) I only
b) II only
c) I and II
d) Neither I nor II

ANSWER A

Q Which one of the following statements is FALSE? **(GATE-2013) (1 Marks)**

$L_1 = \{0^p 1^q 0^r \mid p, q, r \geq 0\}$
$L_2 = \{0^p 1^q 0^r \mid p, q, r \geq 0, p \neq r\}$

(A) L_2 is context-free.
(B) L_1 intersection L_2 is context-free.
(C) Complement of L_2 is recursive.
(D) Complement of L_1 is context-free but not regular.

Answer: (D)

Q Let L = $L_1 \cap L_2$, where L_1 and L_2 are languages as defined below: (**GATE – 2009) (2 Marks)**

$L_1 = \{a^m b^m c\ a^n b^n \mid m, n \geq 0\}$
$L_2 = \{a^i b^j c^k \mid I, j, k \geq 0\}$

a) Not recursive
b) Regular
c) Context – free but not regular
d) Recursively enumerable but not context – free

Ans: c

Q Consider the following languages. **(GATE-2008) (2 Marks)**

$L_1 = \{a^i b^j c^k \mid i = j, k \geq 1\}$ $L_2 = \{a^i b^j \mid j = 2i, i \geq 0\}$

Which of the following is true?

(A) L_1 is not a CFL but L_2 is

(B) $L_1 \cap L_2 = \emptyset$ and L_1 is non-regular

(C) $L_1 \cap L_2$ is not a CFL but L_2 is

(D) There is a 4-state PDA that accepts L_1, but there is no DPDA that accepts L_2

Answer: (B)

Q Consider the languages: **(GATE-2005) (2 Marks)**

$L_1 = \{a^n b^n c^m \mid n, m > 0\}$ $L_2 = \{a^n b^m c^m \mid n, m > 0\}$

Which one of the following statements is FALSE?

(A) $L_1 \cap L_2$ is a context-free language **(B)** $L_1 \cup L_2$ is a context-free language

(C) L_1 and L_2 are context-free language **(D)** $L_1 \cap L_2$ is a context sensitive language

Answer: (A)

Q Let L be a regular language and M be a context-free language, both over the alphabet Σ. Let L^c and M^c denote the complements of L and M respectively. Which of the following statements about the language $L^c \cup M^c$ is TRUE? **(GATE-2005) (2 Marks)**

(A) It is necessarily regular but not necessarily context-free

(B) It is necessarily context-free.

(C) It is necessarily non-regular.

(D) None of the above

Answer: (D)

Q Which of the following statements is true? **(GATE-2001) (1 Marks)**

(A) If a language is context free it can always be accepted by a deterministic push-down automaton

(B) The union of two context free languages is context free

(C) The intersection of two context free languages is context free

(D) The complement of a context free language is context free

Answer: (B)

Q If L_1 and L_2 are context free language and R a regular set, one of the languages below is not necessarily a context free language. Which one? **(GATE – 1996) (1 Marks)**

a) L_1, L_2 **b)** $L_1 \cap L_2$ **c)** $L_1 \cap$ R **d)** $L_1 \cup L_2$

Ans: b

Q Let L_1 is context free language and L_2 is a regular language which of the following is/are false **(GATE – 1999) (1 Marks)**

a) $L_1 - L_2$ is not context free **b)** $L_1 \cap L_2$ is context free

c) ~ L_1 is context free **d)** ~ L_2 is regular

Ans: c

Q Context – free language is closed under: (**GATE – 1999) (1 Marks)**

a) Union, intersection **b)** Union, Kleene closure

c) Intersection, complement **d)** Complement, Kleene closure

Ans: b

Q Let L be a context - free language and M a regular language. Then the language L ∩ M is **(GATE-2006) (1 Marks)**

a) Always regular **b)** Never regular

c) Always a deterministic context free language **d)** Always a context – free language

Ans: d

Q Context free grammar is not closed under: **(NET-NOV-2017)**

a) Concatenation **b)** Complementation

c) Kleene Star **d)** Union

Ans: b

Q Given the following statements:

(A) A class of languages that is closed under union and complementation has to be closed under intersection.

(B) A class of languages that is closed under union and intersection has to be closed under

complementation.
Which of the following options is correct? **(NET-JAN-2017)**

A) Both (A) and (B) are false.
b) Both (A) and (B) are true.
c) (A) is true, (B) is false.
d) (A) is false, (B) is true.

Ans: c

Q Let L = $\{0^n1^n | n \geq 0\}$ be a context free language. Which of the following is correct? **(NET-JULY-2016)**

a) L` is context free and L^k is not context free for any $k \geq 1$.
b) L` is not context free and L^k is not context free for any $k \geq 1$.
c) Both L` and L^k is for any $k \geq 1$ are context free.
d) Both L` and L^k is for any $k \geq 1$ are not context free.

Ans: c

Q Assume the statements S1 and S2 given as : **(NET-SEP-2013)**
S_1: Given a context free grammar G, there exists an algorithm for determining whether L(G) is infinite.
S_2: There exists an algorithm to determine whether two context free grammars generate the same language.
Which of the following is true?
(A) S_1 is correct and S_2 is not correct.
(B) Both S_1 and S_2 are correct.
(C) Both S_1 and S_2 are not correct.
(D) S_1 is not correct and S_2 is correct.

Ans: a

Q The statements s_1 and s_2 are given as : **(NET-JUNE-2013)**
s_1: Context sensitive languages are closed under intersection, concatenation, substitution and inverse homomorphism.
s_2: Context free languages are closed under complementation, substitution and homomorphism.
Which of the following is correct statement?
(A) Both s_1 and s_2 are correct.
(B) s_1 is correct and s_2 is not correct.
(C) s_1 is not correct and s_2 is correct.
(D) Both s_1 and s_2 are not correct.

Ans: b

Consider R to be any regular language and L_1, L_2 be any two context-free languages.

Which one of the following is correct ? **(NET-DEC-2018)**

a) $\overline{L_1}$ is context free

b) $\overline{(L_1 \cup L_2)} - R$ is context free

c) $L_1 \cap L_2$ is context free

d) $L_1 - R$ is context free

Q Suppose L_1 and L_2 are two language over Σ^*

L = Σ^*- ((Σ^* - L_1) ∪ (Σ^* - L_2))

L_1 and L_2 are CFL. Which of the following statement is true?

1. L is necessarily CFL **2.** L may or may not be CFL **3.** L ⊆ L_1

a) only 2 **b)** 1 and 3 **c)** 2 and 3 **d)** all of correct

Q CFG is not closed number

a) Intersection **b)** complement **c)** difference **d)** all of the above

Q Which of the following is not true?

a) CFLs are closed under union and concatenation.

b) Regular languages are closed under union and intersection

c) CFLs are not closed under intersection and complementation.

d) IF L is a CFL and R is a regular set then L∩R is not a CFL.

Q If $L_1$is regular and L_2 is CFL over Σ^* which of the following statement is incorrect?

a) $L_1 \cup L_2$ is CFL **b)** $L_1 \cap L_2$ is regular **c)** $L_1{}^*$ is regular **d)** None of the above

Q CFLs are not closed under

a) union **b)** Concatenation **c)** Closure **d)** Intersection

Q CFLs are not closed under

a) union **b)** Kleene star **c)** Complementation **d)** product

Q Context free languages are closed under

a) Union **b)** Intersection **c)** Complementation **d)** Set difference

Q Context – free languages and regular languages are both closed under the operations (s) of

i) Union **ii)** Intersection **iii)** Concatenation

a) i) and ii) only **b)** ii) and iii) only **c)** i) and iii) only **d)** All of the above

Q Consider the statements.

S1: All Context Free Languages over one symbol alphabet are regular.

S2: All non-self-embedded context-free languages are regular.

Which of the above statements are true?

a) Only S1 **b)** Only S2 **c) B**oth S1 & S2 **d)** None of S1 & S2

Q Let L1 and L2 are context free languages, L3 and L4 are regular languages then which of the following languages are always context-free

a) $(L1 \cup L2) – (L1 \cap L4)$ **b)** $(L1 \cap L3) – (L2 \cap L4)$

c) $(L1 \cup L2) / (L3 \cap L4)$ **d)** $(L3 \cup L4) / (L1 \cup L2)$

Q Let G1 and G2 be Context Free Grammars and R be a regular expression. Then which of the following are decidable.

a) Is $L(G1) \cap L(G2) = \phi$? **b)** Is $L(G1) = L(G2)$?

c) Is $L(G1) = L(R)$? **d)** None of the above

Q Which of the following is undecidable

a) Equivalence of regular languages **b)** Equivalence of context free languages

c) Finiteness check on context free languages **d)** Emptiness of regular languages

Q CFG is not closed number

a) Intersection **b)** complement **c)** difference **d)** all of the above

Q Consider the following decision problems **(GATE-2000) (2 Marks)**

(P1) Does a given finite state machine accept a given string

(P2) Does a given context free grammar generate an infinite number of stings

Which of the following statements is true?

(A) Both (P1) and (P2) are decidable **(B)** Neither (P1) nor (P2) are decidable

(C) Only (P1) is decidable **(D)** Only (P2) is decidable

Answer: (A)

Q Consider the following languages. **(GATE-2008) (2 Marks)**

L_1 = $\{a^i b^j c^k \mid i = j, k \geq 1\}$ **L_2** = $\{a^i b^j \mid j = 2i, i \geq 0\}$

Which of the following is true?

(A) L_1 is not a CFL but L_2 is

(B) $L_1 \cap L_2 = \emptyset$ and L_1 is non-regular

(C) $L_1 \cap L_2$ is not a CFL but L_2 is

(D) There is a 4-state PDA that accepts L_1, but there is no DPDA that accepts L_2

Answer: (B)

Q Consider the language L_1, L_2, L_3 as given below. **(GATE-2011) (2 Marks)**

L_1 = $\{a^p b^q \mid p, q \in N\}$

L_2 = $\{a^p b^q \mid p, q \in N \text{ and } p=q\}$

L_3 = $\{a^p b^q c^r \mid p, q, r \in N \text{ and } p = q = r\}$

Which of the following statements is **NOT TRUE**?

(A) Push Down Automata (PDA) can be used to recognize L_1 and L_2

(B) L_1 is a regular language

(C) All the three languages are context free

(D) Turing machine can be used to recognize all the three languages

Answer: (C)

Q Consider the languages **(GATE-2010) (2 Marks)**

L_1 = $\{0^i 1^j \mid i \mathrel{!}= j\}$ **L_2** = $\{0^i 1^j \mid i = j\}$

L_3 = $\{0^i 1^j \mid i = 2j+1\}$ **L_4** = $\{0^i 1^j \mid i \mathrel{!}= 2j\}$

(A) Only L2 is context free **(B)** Only L2 and L3 are context free

(C) Only L1 and L2 are context free **(D)** All are context free

Answer: (D)

Q (GATE-2006) (1 Marks)

Let $L_1 = \{0^{n+m}1^n 0^m \mid n, m \geq 0\}$, $L_2 = \{0^{n+m}1^{n+m} 0^m \mid n, m \geq 0\}$, and

$L_3 = \{0^{n+m} 1^{n+m}0^{n+m} \mid n, m \geq 0\}$. Which of these languages are NOT context free?

(A) L1 only **(B)** L3 Only **(C)** L1 and L2 **(D)** L2 and L3

Answer: (D)

Consider the following languages over the alphabet $\Sigma = \{0,1,c\}$:

$$L_1 = \{0^n 1^n \mid n \geq 0\}$$
$$L_2 = \{wcw^r \mid w \in \{0,1\}^*\}$$
$$L_3 = \{ww^r \mid w \in \{0,1\}^*\}$$

Here, w^r is the reverse of the string w. Which of these languages are deterministic Context-free languages? **(GATE-2014) (2 Marks)**

(A) None of the languages

(B) Only L1

(C) Only L1 and L2

(D) All the three languages

Answer: (C)

Q Consider the following languages **(GATE-2016) (1 Marks)**
L_1 = $\{a^n b^m c^n: m, n >= 1\}$
L_2 = $\{a^n b^n c^{2n}: n >= 1\}$
Which one of the following is TRUE?
(A) Both L_1 and L_2 are context-free
(B) L_1 is context-free while L_2 is not context-free.
(C) L_2 is context-free while L_1 is not context-free
(D) Neither L_1 nor L_2 is context-free.
Answer: (B)

Q Consider the languages: **(GATE-2005) (2 Marks)**

L_1 = $\{w w^R \mid w \in \{0, 1\}^*\}$

L_2 = $\{w \# w^R \mid w \in \{0, 1\}^*\}$, where # is a special symbol

L_3 = $\{w w \mid w \in (0, 1\}^*)$

Which one of the following is TRUE?

(A) L1 is a deterministic CFL

(B) L2 is a deterministic CFL

(C) L3 is a CFL, but not a deterministic CFL

(D) L3 is a deterministic CFL

Answer: (B)

Q Which of the following languages are context-free? **(GATE-2015) (2 Marks)**

L_1 = $\{a^m b^n a^n b^m \mid m, n \geq 1\}$

L_2 = $\{a^m b^n a^m b^n \mid m, n \geq 1\}$

L_3 = {$a^m b^n$ | m = 2n + 1}

(A) L_1 and L_2 only
(B) L_1 and L_3 only
(C) L_2 and L_3 only
(D) L_3 only

Answer: (B)

Q **i)** {w w^R | w in (a + b)*} and **ii)** { w $ w^R | w in (a + b)*} are

a) Both accepted by deterministic push down automata

b) Only ii) is accepted by a DPDA and i requires a NDPDA

c) Only i is accepted by a DPDA and ii requires a NDPDA

d) None of the above

Q Which of the following languages can't be accepted by a deterministic PDA?

a) The set of palindromes over alphabet {a, b}

b) The set of all strings of balanced parenthesis

c) L = {W c $W^R | W\ in\ (0+1)*$}

d) L = {$0^n 1^n$ | n ≥ 0}

Q Which of the following language is true about the language L = (a^P |P is a prime)

a) It is not accepted by a T.M
b) It is regular but not C.F.L
c) It is CFL but not regular
d) It is neither regular non CFL

Q Which of the following language over {a, b, c} is accepted by a deterministic pushdown automaton? **(GATE – 1997) (1 Marks)**

a) {w ⊂ w^R | w ϵ {a, b) *}
b) {ww^R | w ϵ {a, b, c) *}
c) { $a^n b^n c^n$ | n ≥ 0}
d) {w| w is a palindrome over {a, b, c}}

Ans: a

Q Which of the following languages is accepted by a non – deterministic pushdown automaton (PDA) but NOT by a deterministic PDA?

a) {$a^n b^n c^n$|n ≥ 0}
b) {$a^1 b^m c^n | l \neq m \text{ or } m \neq n$}
c) {$a^n b^n$|n ≥ 0}
d) {$a^n b^n | m, n \geq 0$}

Q Which of the following languages is/ are context free?

1. $\{a^n b^n c^m d^m\mid$ n ≥ 1, m ≥ 1}

2. $\{a^n b^m c^m d^n\mid$ n ≥ 1, m ≥ 1}

3. $\{a^n b^m c^n d^m\mid$ n ≥ 1, m ≥ 1}

4. $\{a^m b^n c^m d^n\mid$ n ≥ 1, m ≥ 1}

a) 1 and 2 **b)** 3 and 4 **c)** 2 and 4 **d)** 1,2,3 and 4

Q Consider the language

$\boldsymbol{L_1}$= $\{a^n b^m c^n d^m$ |n ≥1, m ≥1} and $\boldsymbol{L_2}$= $\{a^n b^m c^m d^n$ |n ≥1, m ≥1}

a) Both L_1 and L_2 are context free

b) L_1 is not context free but L_2 is context free

c) both are not context free

d) L_1 is context free but L_2 is not context free

Q Which of the following languages over {a, b, c} is accepted by deterministic push down automata?

a) {ω| ω is palindrome over {a, b, c}}

b) {ω ω^R| ω ϵ $\{a, b, c\}$*}

c) {a^n b^n c^n | n ≥ 0}

d) {ω c ω^R| ω ϵ $\{a, b,) *\}$

Q Which of the following is a CFL?

a) L = $\{a^m b^m c^m\mid m \geq 1\}$

b) L = $\{a^m\ b^n\ c^m\ d^n\mid m \geq 1$ and n≥ 1}

c) L = $\{a^m b^n c^p\mid m < n \geq p\}$

d) {W W^R| W in (a + b)*}

Q Consider the following set of languages

$\boldsymbol{L_1}$ = $a^m b^n$ | n = m^2 $\boldsymbol{L_2}$ = $a^m b^m c^m d^n$ | m, n > 0 $\boldsymbol{L_3}$ = $a^m b^m c^n d^n$ | m, n > 0

Which of the above language is not context free?

a) L_1 and L_3 **b)** L_2 and L_3 **c)** L_1 and L_2 **d)** All L_1, $L_2$and L_3

Q Which of the following statement must always be true for A and B? Suppose A and B are two sets of strings from ∑*, such that B ⊆ A

i) If A is finite then, B is finite

ii) If A is regular then, B is regular

iii) If A is context free then, B is context free

a) i) only **b)** ii) only **c)** iii) only **d)** All three

Q Which of the following languages are context free?

$\boldsymbol{L_1}$ = $\{a^m b^m c^n\mid$ m ≥ 1 and n ≥ 1} $\boldsymbol{L_2}$ = $\{a^m b^m c^n\mid$ n ≥ 1} $\boldsymbol{L_3}$ = $\{a^m b^m c^m\mid$ m ≥ 1}

a) only L_1 **b)** L_2 and L_3 **c)** only L_2 **d)** L_3

Q L = $\{a^m b^n c^{m+n}$ | m, n ≥ 1}
a) Regular
b) C.F.L but not regular
c) C.S.L but not CFL
d) Type – 0 but not context sensitive

Consider the following languages :

$L_1 = \{a^{n+m} b^n a^m \mid n, m \geq 0\}$

$L_2 = \{a^{n+m} b^{n+m} a^{n+m} \mid n, m \geq 0\}$

Which one of the following is correct ? **(NET-JULY-2018)**
a) Only L1 is Context Free Language
b) Only L2 is Context Free Language
c) Both L1 and L2 are Context Free Languages
d) Neither L1 Nor L2 are context free languages
Ans: a

Q Given the following two languages:
L_1 = $\{a^n b^n \mid n \geq 0, n \neq 100\}$
L_2 = $\{w \in \{a, b, c\}^* \mid n_a(w) = n_b(w) = n_c(w)\}$
Which of the following options is correct? **(NET-JAN-2016)**
a) Both L1 and L2 are not context free language
b) Both L1 and L2 are context free language.
c) L1 is context free language, L2 is not context free language.
d) L1 is not context free language, L2 is context free language
Ans: c

Q Given the following two statements:
A. L = $\{w \mid n_a(w) = n_b(w)\}$ is deterministic context free language, but not linear.
B. L = $\{a^n b^n\} \cup \{a^n b^{2n}\}$ is linear, but not deterministic context free language.
Which of the following options is correct? **(NET-JAN-2017)**
a) Both (A) and (B) are false.
b) Both (A) and (B) are true
c) (A) is true, (B) is false.
d) (A) is false, (B) is true.
Ans: b

Q Given the following two languages:
L_1 = $\{a^n b a^n \mid n > 0\}$
L_2 = $\{a^n b a^n b^{n+1} \mid n > 0\}$
Which of the following is correct? **(NET-DEC-2015)**

a) L1 is context free language and L2 is not context free language
b) L1 is not context free language and L2 is context free language
c) Both L1 and L2 are context free languages
d) Both L1 and L2 are not context free languages
Ans: a

Q Given the following two languages: **(NET-JUNE-2014)**
L_1= $\{a^n b^n \mid n > 1\} \cup \{a\}$
L_2= $\{w C w^R \mid w \in \{a, b\}^*\}$
Which statement is correct?
(A) Both L1 and L2 are not deterministic.
(B) L1 is not deterministic and L2 is deterministic.
(C) L1 is deterministic and L2 is not deterministic.
(D) Both L1 and L2 are deterministic.
Ans: d

Q Consider the languages:
L_1 = $\{w w^R \mid w \in \{0, 1\}^*\}$
L_2 = $\{w \# w^R \mid w \in \{0, 1\}^*\}$, where # is a special symbol
L_3 = $\{ww \mid w \in (0, 1\}^*)$
Which one of the following is TRUE?
a) L_1 is a deterministic CFL
b) L_2 is a deterministic CFL
c) L_3 is a CFL, but not a deterministic CFL
d) L_3 is a deterministic CFL
Ans. B

Q The language $\{a^m b^n C^{m+n} \mid m, n \geq 1\}$ is
a) regular
b) Context-free but not regular
c) context sensitive but not context free
d) Type-0 but not context sensitive

Ans. B

Q Which of the following languages are context-free?
L_1 = $\{a^m b^n a^n b^m \mid m, n \geq 1\}$
L_2 = $\{a^m b^n a^m b^n \mid m, n \geq 1\}$
L_3 = $\{a^m b^n \mid m = 2n + 1\}$
a) L_1 and L_2 only
b) L_1 and L_3 only
c) L_2 and L_3 only
d) L_3 only
Ans. B

Q Which one of the following languages over $\sum = \{a, b\}$ is NOT context-free? **(GATE-2019) (2 Marks)**

a) $\{a^n b^i \mid i \in \{n, 3n, 5n\}, n \geq 0\}$
b) $\{w\, a^n\, w^R\, b^n \mid w \in \{a, b\}^*, n \geq 0\}$
c) $\{w\, w^R \mid w \in \{a, b\}^*\}$
d) $\{w a^n\, b^n\, w^R \mid w \in \{a, b\}^*, n \geq 0\}$

Ans: b

Q Consider the following languages:

I. $\{a^m b^n c^p d^q \mid m + p = n + q, \text{ where } m, n, p, q \geq 0\}$
II. $\{a^m b^n c^p d^q \mid m = n \text{ and } p = q, \text{ where } m, n, p, q \geq 0\}$
III. $\{a^m b^n c^p d^q \mid m = n = p \text{ and } p \neq q, \text{ where } m, n, p, q \geq 0\}$
IV. $\{a^m b^n c^p d^q \mid mn = p + q, \text{ where } m, n, p, q \geq 0\}$ Which of the above languages are context-free? **(GATE-2018) (2 Marks)**

a) I and IV only
b) I and II only
c) II and III only
d) II and IV only

(ANSWER-B)

Q The language $L = \{a^n\, b^n\, a^m\, b^m \mid n \geq 0, m \geq 0\}$ is **(NET-SEP-2013)**

(A) Context free but not linear
(B) Context free and linear
(C) Not Context free and not linear
(D) Not Context free but linear

Ans: a

Q Consider the following languages:

L1 = $\{a^m\, b^n \mid m \neq n\}$
L2 = $\{a^m\, b^n \mid m = 2n+1\}$
L3 = $\{am\ bm \mid m \neq 2n\}$

Which one of the following statements is correct? **(NET-NOV-2017)**

a) Only L1 and L2 are context free languages
b) Only L1 and L3 are context free languages
c) Only L2 and L3 are context free languages
d) L1, L2 and L3 are context free languages

Ans: d

Q The language **L** = $\{0^i 2 1^i \mid i \geq 0\}$ over the alphabet {0,1, 2} is: (**GATE-2007) (2 Marks)**

a) not recursive.
b) is recursive and is a deterministic CFL.
c) is a regular language.
d) is not a deterministic CFL but a CFL

Q Consider the following languages:

L_1 = {$a^n b^m c^n$: m, n >= 1} **L_2** = {$a^n b^n c^{2n}$: n >= 1}

Which one of the following is TRUE? (GATE-2016) (2 Marks)

a) Both L1 and L2 are context-free

b) L1 is context-free while L2 is not context-free.

c) L2 is context-free while L1 is not context-free.

d) Neither L1 nor L2 is context-free.

ANSWER B

Q Consider the following languages:

I. {$a^m b^n c^p d^q$ | m + p = n + q, where m, n, p, q ≥ 0}

II. {$a^m b^n c^p d^q$ | m = n and p = q, where m, n, p, q ≥ 0}

III. {$a^m b^n c^p d^q$ | m = n = p and p ≠ q, where m, n, p, q ≥ 0}

IV. {$a^m b^n c^p d^q$ | mn = p + q, where m, n, p, q ≥ 0}

Which of the above languages are context-free? **(GATE-2018) (1 Marks)**

a) I and IV only **b)** I and II only

c) II and III only **d)** II and IV only

(ANSWER-B)

Q The language {$a^m b^n C^{m+n}$ | m, n ≥ 1} is **(GATE-2004) (1 Marks)**

(A) regular **(B)** context-free but not regular

(C) context sensitive but not context free **(D)** type-0 but not context sensitive

Answer: (B)

Q The language **L**= {$0^i 2 1^i$ | i ≥ 0} over the alphabet {0,1, 2} is: **(GATE-2007) (1 Marks)**

(A) not recursive **(B)** is recursive and is a deterministic CFL.

(C) is a regular language. **(D)** is not a deterministic CFL but a CFL.

Answer: (B)

Q Which one of the following is FALSE? **(GATE-2009) (2 Marks)**

(A) There is unique minimal DFA for every regular language

(B) Every NFA can be converted to an equivalent PDA.

(C) Complement of every context-free language is recursive.

(D) Every nondeterministic PDA can be converted to an equivalent deterministic PDA.

Answer: (D)

Q The language **L** = {$a^i b c^i$ | i >= 0} over the alphabet {a, b, c} is:

a) a regular language.
b) not a deterministic context free language but a context free language.
c) recursive and is a deterministic context free language.
d) not recursive.

Q The language L is defined as

L = $\{a^n b^n c^m d^m \mid m, n \geq 1\}$ **L** = $\{a^n b^m c^m d^n \mid m, n \geq 1\}$ Otherwise

Then the language L is best described as

a) Regular **b)** Context-Free but not regular
c) Context-Sensitive but not Context-Free **d)** No one decide until the will be declared

Moore and Mealy Machine

- Both Moore and mealy machine are special case of DFA
- Both acts like o/p producers rather than language acceptors
- In Moore and mealy machine no need to define the final states
- No concepts of dead states and no concepts of final states
- Mealy and Moore Machines are equivalent in power.

Moore Machine

General Definitions

- **A Moore machine is a six-tuple (Q, ∑, Δ, δ, λ, q_0), where**
    1. Q is a finite set of states:
    2. ∑ is the input alphabet:
    3. Δ is the output alphabet.
    4. δ is the transition function ∑ x Q into Q
    5. λ is the output function mapping Q into Δ and
    6. q_0 is the initial state.
- In Moore machine for every state output is associated.
- If the length of i/p string is n, then length of o/p string will be n+1
- Moore machine response for empty string ∈

Examples: The below table shows the transition table of a Moore Machine.

Present state	*Next state* δ		*Output*
	a = 0	*a* = 1	λ
$\rightarrow q_0$	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

Here, For the input string 0111, the transition of states is given by $q_0 \rightarrow q_3 \rightarrow q_0 \rightarrow q_1 \rightarrow q_2$. The output string is 00010.

Q construct a Moore machine take all the string of a's and b's as i/p and counts the no of a's in the i/p string in terms of 1, ∑ = {a, b}, Δ = {0, 1}?

Q construct a Moore machine take all the string of a's and b's as i/p and counts the no of occurrence of sub-string 'ab' in terms of 1, ∑ = {a, b}, Δ = {0, 1}?

Q construct a Moore machine take all the string of a's and b's as i/p and counts the no of occurrence of sub-string 'aa' in terms of 1, ∑ = {a, b}, Δ = {0, 1}?

Q construct a Moore machine where ∑ = {0, 1}, Δ = {a, b, c}, machine should give o/p a, if the i/p string ends with 10, b if i/p string ends with 11, c otherwise?

<u>Mealy Machine</u>

- Mealy machine is a six-tuple **($Q, \sum, \Delta, \delta, \lambda, q_0$),** where all the symbols except λ have the same meaning as in the Moore machine. λ is the output function mapping ∑ x Q into Δ.
- in case of mealy machine, the output symbol depends on the transition.
- If the length of i/p string is n, then length of o/p string will be n
- Mealy machine do not response for empty string ∈

Example: The below table shows the transition table of a Mealy Machine.

Present state	Next state			
	a = 0		a = 1	
	state	output	state	output
$\to q_1$	q_3	0	q_2	0
q_2	q_4	1	q_4	0
q_3	q_2	1	q_1	1
q_4	q_4	1	q_3	0

For the input string 0011, the transition of states is given by $q_1 \to q_3 \to q_2 \to q_4 \to q_3$ and the output string is 0100.

Q construct a Mealy machine take all the string of a's and b's as i/p and counts the no of a's in the i/p string in terms of 1, ∑ = {a, b}, Δ = {0, 1}?

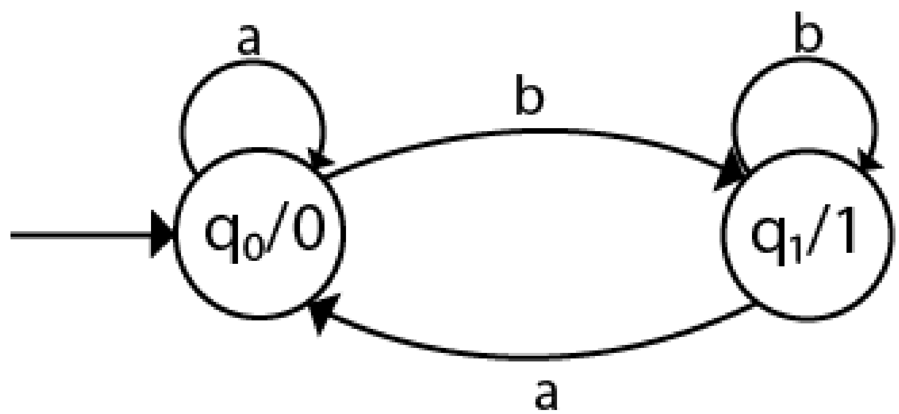
Q construct a Mealy machine take all the string of a's and b's as i/p and counts the no of occurrence of sub-string 'ab' in terms of 1, ∑ = {a, b}, Δ = {0, 1}?

Q construct a Mealy machine take all the string of a's and b's as i/p and counts the no of occurrence of sub-string 'aa' in terms of 1, ∑ = {a, b}, Δ = {0, 1}?

Q construct a Mealy machine where ∑ = {0, 1}, Δ = {a, b, c}, machine should give o/p a, if the i/p string ends with 10, b if i/p string ends with 11, c otherwise?

CONVERSION OF MOORE TO MEALY MACHINE

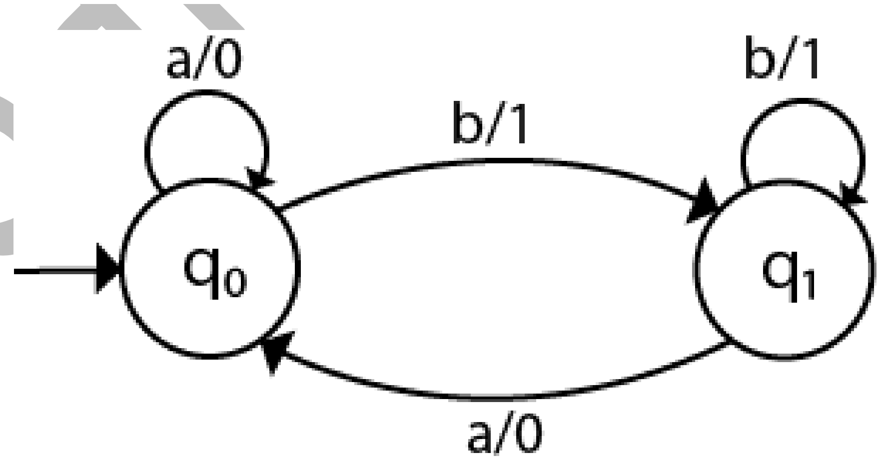
- Let us take an example to understand the conversion:
- Convert the following Moore machine into its equivalent Mealy machine.



- The transition table of given Moore machine is as follows:

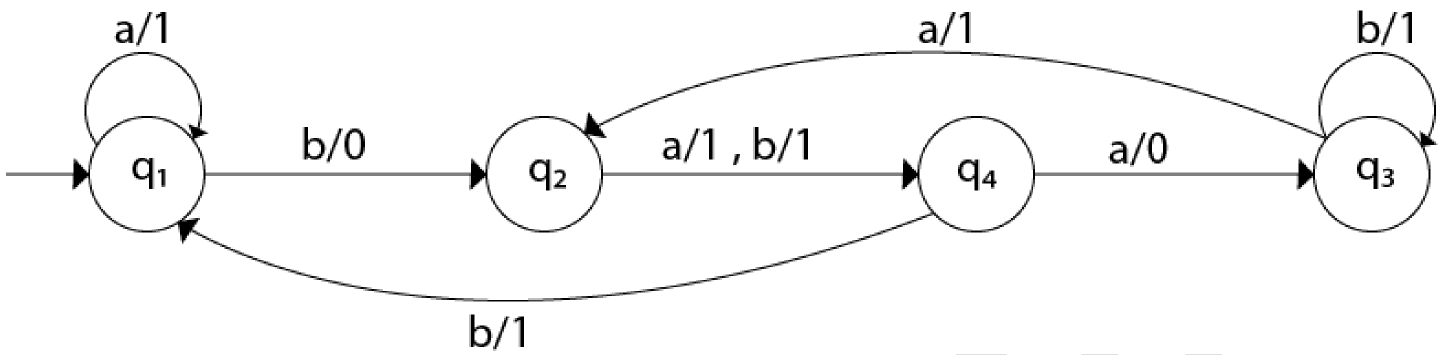
Q	a	b	Output(λ)
q0	q0	q1	0
q1	q0	q1	1

- To convert a mealy machine to moore machine all you need to do is just push out the outputs of states onto to the incoming transitions.
- While conversion from moore to mealy machine, the number of states will we same and there will be no extra states.
- The equivalent Moore Machine will be:



PROCEDURE FOR TRANSFORMING A MEALY MACHINE INTO A MOORE MACHINE

Consider the Mealy Machine:



Transition table for above Mealy machine is as follows:

Present State	Next State			
	a		b	
	State	O/P	State	O/P
q_1	q_1	1	q_2	0
q_2	q_4	1	q_4	1
q_3	q_2	1	q_3	1
q_4	q_3	0	q_1	1

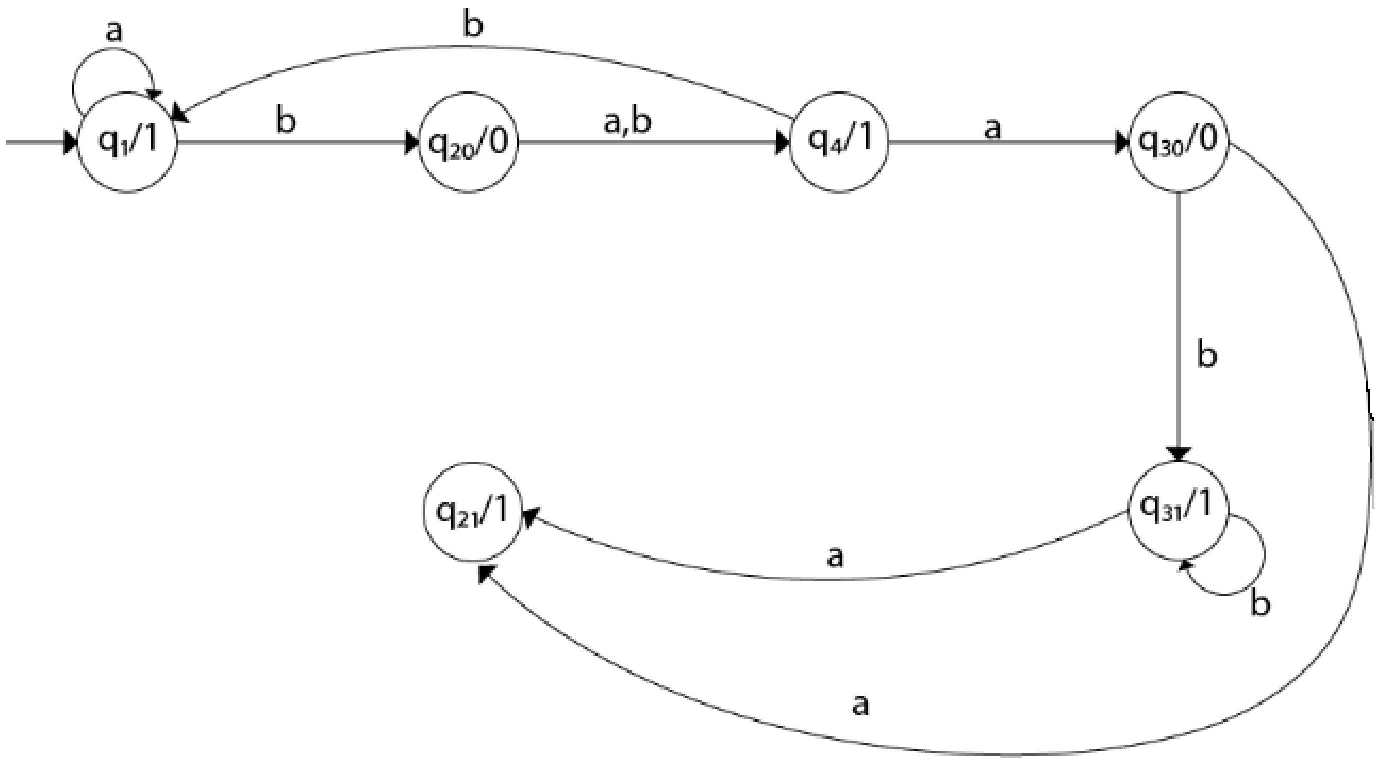
- For state q1, there is only one incident edge with output 0. So, we don't need to split this state in Moore machine.
- For state q2, there is 2 incident edge with output 0 and 1. So, we will split this state into two states q20 (state with output 0) and q21(with output 1).
- For state q3, there is 2 incident edge with output 0 and 1. So, we will split this state into two states q30 (state with output 0) and q31 (state with output 1).
- For state q4, there is only one incident edge with output 0. So, we don't need to split this state in Moore machine.

- if mealy machine have m states and the resultant moore machine have n states then $m <= n <= m * \mathbf{\Delta}$

Transition table for Moore machine will be:

Present State	Next State		Output
	a=0	a=1	
q_1	q_1	q_2	1
q_{20}	q_4	q_4	0
q_{21}	∅	∅	1
q_{30}	q_{21}	q_{31}	0
q_{31}	q_{21}	q_{31}	1
q_4	q_3	q_4	1

- Transition diagram for Moore machine will be:



a
b
q1/1
b
q20/0
a,b
q4/1
a
q30/0
b
q21/1
q31/1
a
b
a

Q Given the following state table of an FSM with two states A and B, one input and one output:

Present State A	Present State B	Input	Next State A	Next State B	Output
0	0	0	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	1	0	0
0	0	1	0	1	0
0	1	1	0	0	1
1	0	1	0	1	1
1	1	1	0	0	1

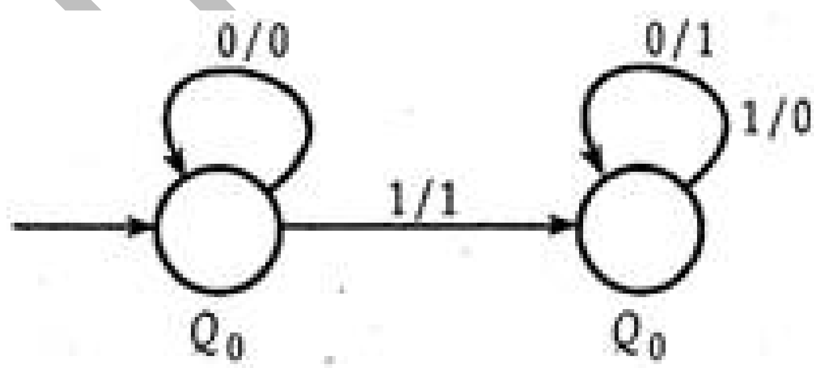
If the initial state is A=0, B=0, what is the minimum length of an input string which will take the machine to the state A=0, B=1 with Output = 1? **(GATE-2009) (2 Marks)**

(A) 3 **(B)** 4 **(C)** 5 **(D)** 6

Answer: (A)

Explanation: (0, 0) –1–> (0, 1) –0–> (1, 0) –1–> (0, 1) and output 1

Q The following diagram represents a finite state machine which takes as input a binary number from the least significant bit. **(GATE-2005) (1 Marks)**



Which one of the following is TRUE?

(A) It computes 1’s complement of the input number

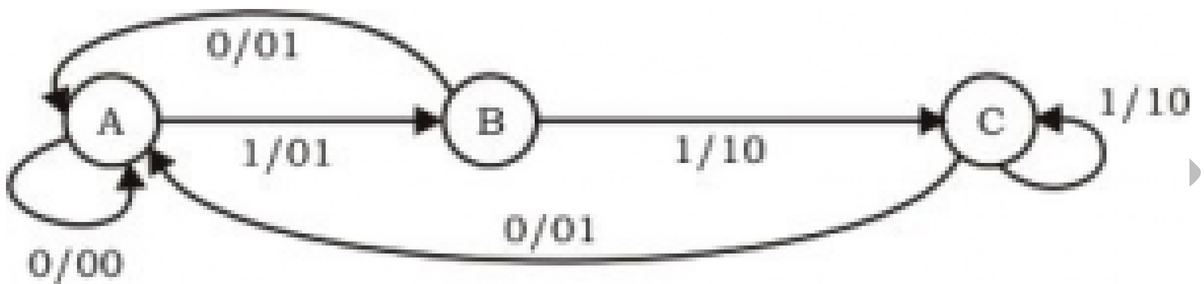
(B) It computes 2’s complement of the input number

(C) It increments the input number

(D) It decrements the input number

Answer: (B)

Q The Finite state machine described by the following state diagram with A as starting state, where an arc label is x / y and x stand for 1-bit input and y stands for 2- bit output **(GATE-2002) (2 Marks)**



(A) Outputs the sum of the present and the previous bits of the input.
(B) Outputs 01 whenever the input sequence contains 11.
(C) Outputs 00 whenever the input sequence contains 10.
(D) None of these
Answer: (A)

Q The Moore machine has six tuples (Q, ∑, Δ, δ, λ, d_0). Which of the following is true?

a) δ is the output function

b) δ is the transition function ∑ into Q

c) λ is the transition function ∑ x Q into Q

d) λ is the output function mapping Q into Δ

Q For the previous question, δ is the transition function from

a) Q to R **b)** δ x ∑ to d **c)** ∑ x Q to Q

CONTEXT-FREE LANGUAGES AND PUSH DOWN AUTOMATA

- Context-free languages are applied in parser design.
- They are also useful for describing block structures in programming languages.

Push Down Automata

BASIC DEFINITIONS

- We Already understand the limitation of finite automata, that it cannot do the infinite comparison between the symbols.
- Let us consider L = {a^n b^n | n >= 1}. This is not regular, as it has to remember the number of a's in a string and so it will require an infinite number of states, which is logically not possible.
- This difficulty can be avoided by adding an auxiliary memory in the form of a 'stack'. The reason we choose stack because it is the simplest memory possible.
- This type of arrangement where ***a finite automaton has a stack leads to the generation of a pushdown automaton.***

Problem
after this much study now we understand that there are number of languages which cannot be accepted by finite automata, actually finite automata is able to perform to generic operations it can count and it can remember order. But a task like comparison between two symbols cannot be performed by finite automata.

Solution
We require memory so that we can compare between two or more different symbols, though there are a number of options available for memory like array, stack, queue, link list, trees etc. out of them all stack is the most appropriate here as indexing is not required in stack as it is a zero-address data structure. It is also easy to check extreme conditions like over flow under flow.

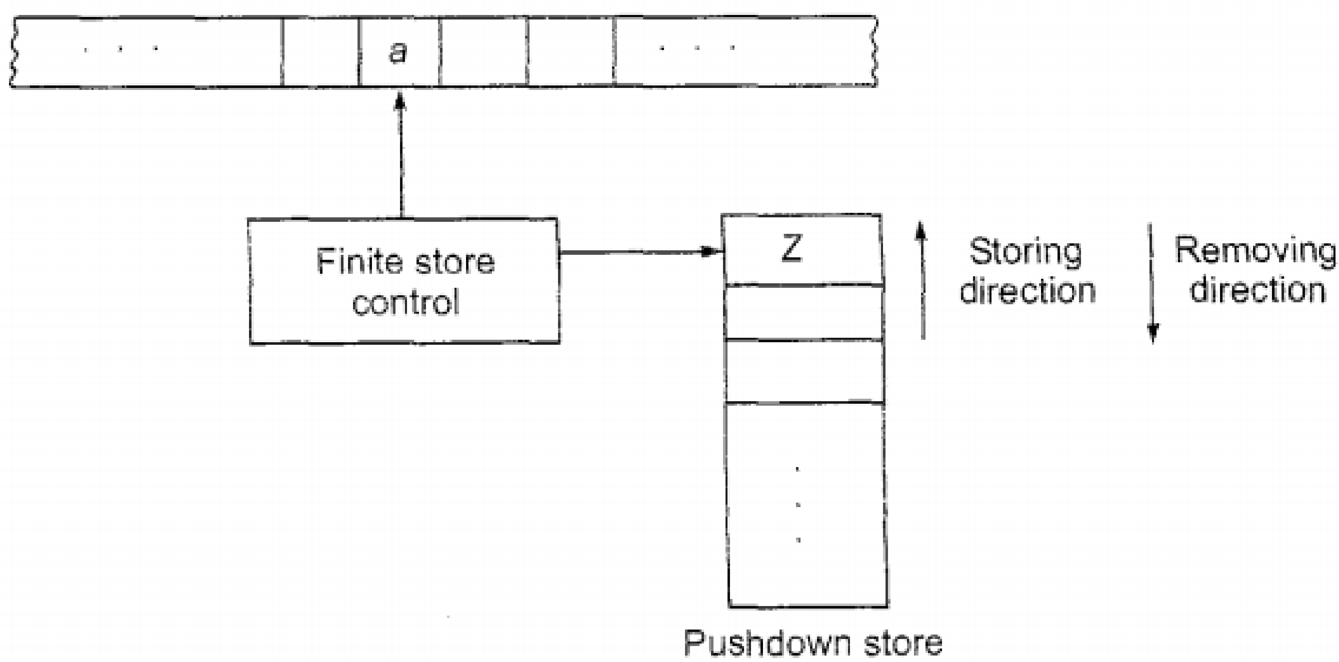
Finite automata + memory (1 stack) = push down automata

Application: - used in describing much of the syntax of high-level programming languages, as well as related languages like legal algebraic expressions and balanced strings of parentheses.

Context free Grammar generates context free language which is accepted by push down automata

BLOCK DIAGRAM OF PDA

- Finite control unit is also called as memory unit it is static and limited. So to process the i/p string if the static memory is not sufficient then we can use the stack.
- i/p tape is divided into cells where is cell is capable of holding one symbol at a time. At stack of infinite size, which support three operations push, pop and skip.
- The accepting power of a pda is more than that of finite automata and less than that of linear bounder automata.
- The power of non-deterministic pda is more than the power of deterministic pda.



Formal Definition of PDA

A PDA is a 7-tuple, namely $(Q, \sum, \Gamma, \delta, q_0, Z_0, F)$, where

(i) Q – is a finite nonempty set of states,

(ii) ∑ – is a finite nonempty set of input symbols,

(iii) Γ – is a finite nonempty set of pushdown symbols,

(iv) q_0 – is a special state called the initial state,

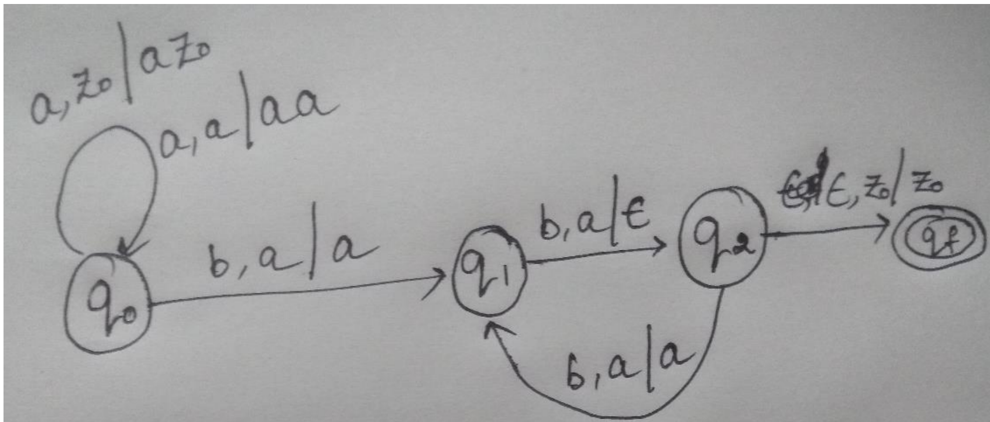
(v) Z_0 – is a special pushdown symbol called the initial symbol on the pushdown store.

(vi) F – is a set of final states, a subset of Q and

(vii) δ – is a transition function from $Q \times (\sum \cup \{\in\}) \times \Gamma$ to the set of finite subsets of $Q \times \Gamma^*$.

Representation of States

Presentation through states diagram



1.4 BASIC OPERATIONS ON STACK

(1) PUSH – one symbol can be inserted into the stack at one time.

$\delta(q_i, a, z_0) = (q_j, az_0)$

(2) POP – one symbol can be deleted from the stack at one time.

$\delta(q_i, a, z_0) = (q_j, \varepsilon)$

(3) SKIP – IT means no stack operation, status of the stack will remain same, before a after the operation

$\delta(q_i, a, z_0) = (q_j, z_0)$

note- if pda perform a push or a pop operation at least one's during processing of string than we say that pda is using the stack.

Instantaneous Description (ID)

- An **instantaneous description (ID)** is (q, *x*, α), where q ∈ Q, *x* ∈ ∑* and α ∈ Γ*.
- An initial ID is **(q_0, *x*, Z_0)**, this means that initially the pda is in the initial state q_0, the input string to be processed is *x* and the PDS has only one symbol, namely Z_0.

ACCEPTANCE BY PDA

- The set accepted by pda by final state is defined by

$$T(A) = \{w \in \Sigma^* | (q_0, w, Z_0) \overset{*}{\vdash} (q_f, \Lambda, \alpha) \text{ for some } q_f \in F \text{ and } \alpha \in \Gamma^*\}$$

Or acceptance by null stack/ empty stack,

$$N(A) = \{w \in \Sigma^* | (q_0, w, Z_0) \overset{*}{\vdash} (q, \Lambda, \Lambda) \text{ for some } q \in Q\}$$

In this case the special pushdown symbol is popped off.

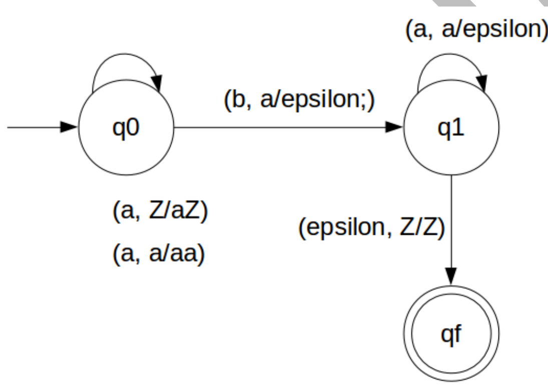
There is no change in the language acceptance capability of the pda either we accept by final state of empty state.

Example: Design a Deterministic Push Down Automata for {a^n b^n | n >= 1}

- The ***a's*** in the given string are added to the stack. When the symbol ***b*** is encountered in the input string, an ***a*** is removed from the stack. Thus, the matching of number of ***a's*** and the number of ***b's*** is accomplished.

Steps:

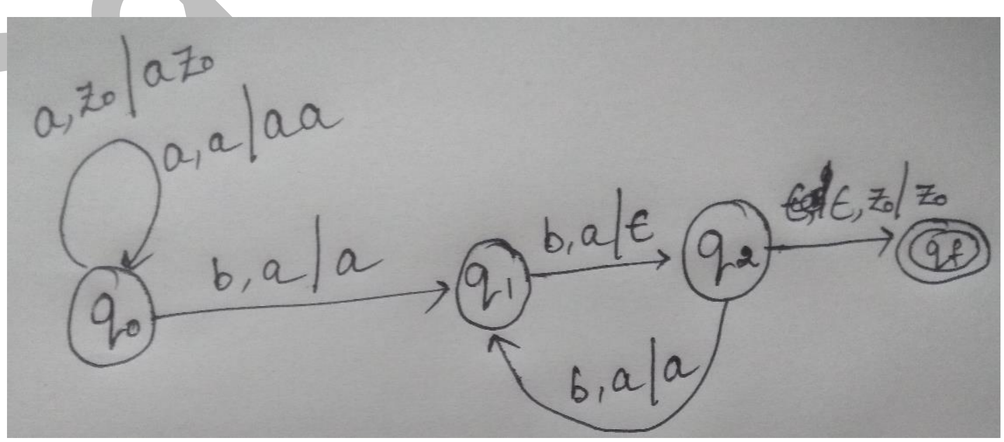
- First, we have to count number of a's and number of b should be equal.
- We will achieve this by pushing a's in STACK and then we will pop a's whenever "b" comes.
- So, in the end of the strings if nothing is left in the STACK then we can say that language is accepted in the PDA.



Example: Design a PDA for {a^nb^{2n} | n ≥ 1}

Steps:

- We will be pushing a's as they come and when we encounter a b we will change state without popping a.
- When we encounter 2nd 'b' we will pop out a 'a'.
- Loop this process until for every two b's, one 'a' is popped out and the stack empties.

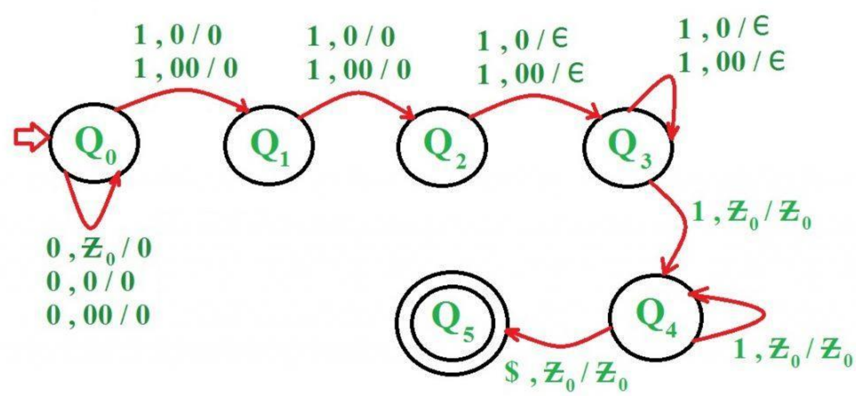


Example: Construct a PDA for language L = {0^n1^m | n >= 1, m >= 1, m > n+2}

<u>Design will consist:</u>

First 0's are pushed into stack. When 0's are finished, two 1's are ignored. Thereafter for every 1 as input a 0 is popped out of stack. When stack is empty and still some 1's are left then all of them are ignored.

- On receiving 0 push it onto stack. On receiving 1, ignore it and goto next state
- On receiving 1, ignore it and goto next state
- On receiving 1, pop a 0 from top of stack and go to next state
- On receiving 1, pop a 0 from top of stack. If stack is empty, on receiving 1 ignore it and goto next state
- On receiving 1 ignore it. If input is finished then goto last state

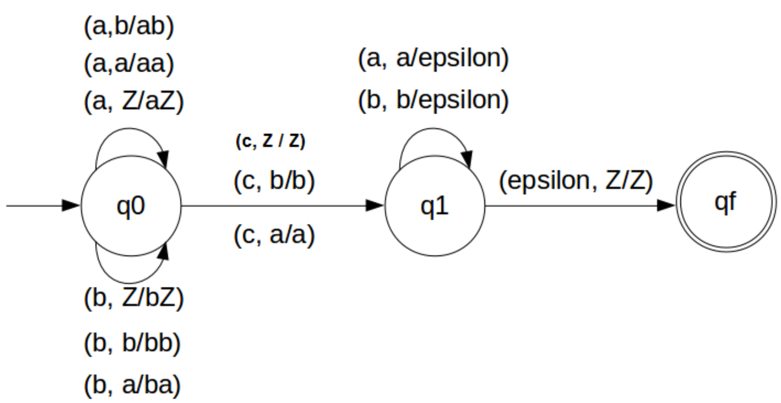


Example: Design a PDA for $w\,c\,w^r \mid w \in (a, b)^*$

W is a string and w^r is a reverse of a string. Ex: w = abc then w^r = cba

Steps

- Starting state will accept everything and we will be pushing everything until we get a ‘c’.
- ‘c’ is a string separator and we will not pop anything and skip to change state.
- Now in next state the topmost symbol and current symbol should match as stack by default acts as an string reverser, now for every matching symbols we will pop.
- We continue to do this until the stack is empty.



Q construct the PDA for the following languages?

a) L = {a, ab}

b) $L = \{a\, b^n \mid n >= 0\}$

c) $L = \{a^n b^n \mid n >= 1\}$

d) $L = \{a^n b^{2n} \mid n >= 1\}$

e) $L = \{a^n c^m b^n \mid n, m >= 1\}$

f) $L = \{w\, c\, w^r \mid w \in (a, b)^*\}$

Q consider the following mapping and find the correct language?

$\delta (q_0, 1, z_0) = (q_0, xz_0)$

$\delta (q_0, 1, x) = (q_0, xx)$

$\delta (q_0, 0, x) = (q_1, €)$

$\delta (q_1, 0, x) = (q_1, €)$

$\delta (q_1, 0, z_0) = (q_1, z_0)$

$\delta (q_1, €, z_0) = (q_F, z_0)$

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

b) $L = \{a^m b^n \mid m != n\}$

c) $L = \{a^m b^n \mid m >= n\}$

d) $L = \{a^m b^n \mid m <= n\}$

Ans: d

Q consider the following mapping and find the correct language?

$\delta (q_0, a, z_0) = (q_1, z_0)$

$\delta (q_0, b, z_0) = (q_2, z_0)$

$\delta (q_1, a, z_0) = (q_1, z_0)$

$\delta (q_1, b, z_0) = (q_2, z_0)$

$\delta (q_2, a, z_0) = (q_2, z_0)$

$\delta (q_2, b, z_0) = (q_1, z_0)$

$\delta (q_1, €, z_0) = (q_f, z_0)$

a) ending with a

b) ending with a, contain even number of a

c) ending with a, contain odd number of a

d) contain even number of b

Ans: d

Q construct PDA that accepts L = {|w|$_{a = b}$ | w € $(a, b)^+$} ?

Q construct pda that accepts a language L = {w w^r | w € $(a, b)^*$}?

Non- Deterministic PDA

Non- Deterministic PDA can also be defined using 7 tuples s.t. the transition function δ is defined as: Q x (∑ U {∈}) x Γ to the set of finite subsets of Q x Γ*.

δ: Q x **{∑ U** ∈**}** x **Γ → $2^{(Q \times \Gamma^*)}$**

i.e. on a given input symbol and stack symbol a NPDA can move to more than one state.

Rest all other tuples are same as DPDA.

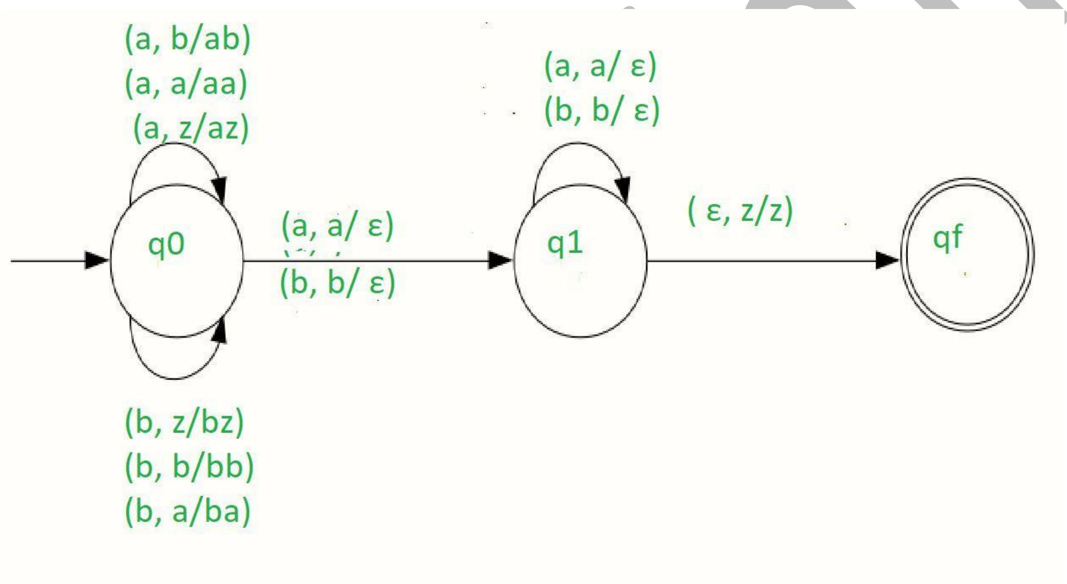
Let us take an example to understand about NPDA.

Example: Design a NPDA for ww^r | w∈ $(a, b)^+$

An example string for such case is: abba

As seen in the previous example we were aware about the center, but in this case we are not aware about the center. So we will be making a NPDA where the NPDA will be assuming two cases:

1. Center has been reached when we have two same symbols in input and top of stack as in case of a**bb**a, as this is the only way to determine center.
2. Center has not been reached and the NPDA will have to continue pushing symbol as in an example case of '**aaaa',** in this case we cannot assume that center is reached when we have two same symbols in input and top of stack.



Required NPDA

Here we are moving to two different states on inputs:

- (a, a / ∈) we are going to q1 and popping **a,** assuming that center has been reached.
- (a, a / aa) we are assuming that center has not reached and we keep on pushing a.
- (b, b /∈) we are assuming that center has not been reached.
- (b, b / bb) we are assuming that center has not reached and we keep on pushing b.

<u>**Some points to remember about PDA and NPDA**</u>

- Clearly for the above example we can design a NPDA but cannot design a DPDA, so NPDA and DPDA are not equivalent in power.
- NPDA is more powerful than DPDA.

<u>Prefix property</u>: - a language L is said to have prefix property if there is no two different string x and y in L such that one is the prefix of other. If a language is CFL and not RL and have prefix property then we can design a deterministic pda to accept it, hence the language is DCFL.

Q Let N_f and N_p denote the classes of languages accepted by non-deterministic finite automata and non-deterministic push-down automata, respectively. Let D_f and D_p denote the classes of languages accepted by deterministic finite automata and deterministic push-down automata, respectively. Which one of the following is TRUE? **(GATE-2005) (1 Marks)**

(A) $D_f \subset N_f$ and $D_p \subset N_p$ **(B)** $D_f \subset N_f$ and $D_p = N_p$

(C) $D_f = N_f$ and $D_p = N_p$ **(D)** $D_f = N_f$ and $D_p \subset N_p$

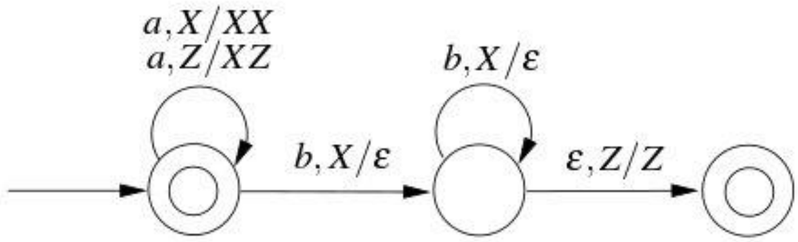
Answer: (D)

Q The language accepted by a Pushdown Automation in which the stack is limited to 10 items is best described as **(GATE-2003) (1 Marks)**

(A) Context Free **(B)** Regular **(C)** Deterministic Context Free **(D)** Recursive

Answer: (B)

Q Consider the transition diagram of a PDA given below with input alphabet ∑ = {a, b}and stack alphabet Γ = {X, Z}. Z is the initial stack symbol. Let L denote the language accepted by the PDA. **(GATE-2016) (2 Marks)**



Which one of the following is **TRUE**?

(A) $L = \{a^n b^n | n \geq 0\}$ and is not accepted by any finite automata

(B) $L = \{a^n | n \geq 0\} \cup \{a^n b^n | n \geq 0\}$ and is not accepted by any deterministic PDA

(C) L is not accepted by any Turing machine that halts on every input

(D) $L = \{a^n | n \geq 0\} \cup \{a^n b^n | n \geq 0\}$ and is deterministic context-free

Answer: (D)

Q The languages

1. $\{0^n 1^n \mid n>=1\}$ and **2.** $\{ 0^n 1^{2n} \mid n >= 1\} \cup \{0^n 1^n \mid n >= 1\}$ are

a) both accepted by deterministic push down automata

b) 1can be accepted by a DPDA but 2 requires a NDPDA

c) 2 can be accepted by a DPDA but 1 requires a NDPDA

d) None of the above

Q A PDM behaves like a TM when the number of auxiliary memories it has, is

a) 0 **b)** 1 or more **c)** 2 or more **d)** none of these

Q Consider the following set of languages over one symbol alphabet

L_1: set of left linear languages L_2: set of right linear languages

L_3: set of DCFLs L_4: set of CFLs

Choose the correct statement from the following

a) $L_1 \subset L_2 \subset L_3 \subset L_4$ **b)** $L_1 \subset L_2 \subset L_3 = L_4$

c) $L_1 = L_2 \subset L_3 = L_4$ **d)** $L_1 = L_2 = L_3 = L_4$

Q Which of the following is not true?

a) The set of languages accepted by deterministic and non-deterministic PDAs are not equal

b) L = {w c w^R|w in (0 + 1)* & c $\notin$ {0,1}} can be accepted by deterministic PDA

c) L = {w c w^R|w, c in (0 + 1)*} can not be accepted by a deterministic PDA

d) L= { $0^n\ 0^1$|$n\ \geq 0$} can be accepted by a deterministic PDA

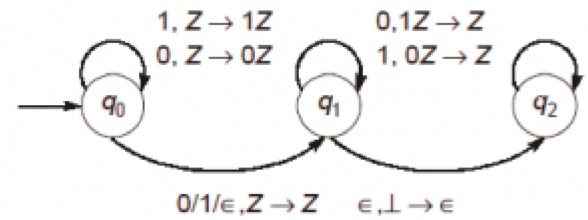
answer: (c)

Q The pushdown automation M = ({q_0, q_1, q_2}, {a, b}, {0, 1}, δ, q_0, 0, {q_0}) with (NET-DEC-2014)
δ(q_0, a, 0) = {(q1, 10)}
δ(q_1, a, 1) = {(q1, 11)}
δ(q_1, b, 1) = {(q2, ∈)}
δ(q_2, b, 1) = {(q2, ∈)}
δ(q_2, ∈, 0) = {(q0, ∈)}
Accepts the language
(A) L = {$a^n\ b^m$ | n, m ≥ 0}
(B) L = {$a^n\ b^n$| n ≥ 0}
(C) L = {$a^n\ b^m$ | n, m > 0}
(D) L = {$a^n\ b^n$| n > 0}
Ans: d

Q The language accepted by the non- deterministic pushdown automaton (NET-SEP-2013)
M= ({q_0, q_1, q_2}, {a, b}, {a, b, z}, δ, q_0, z, {q_2}) with transitions
δ (q_0 a, z) = {(q_1 a), (q_2, ∈)};
δ (q_1, b, a) = {(q_1, b)}
δ (q_1, b, b) = {(q_1 b)}, δ (q_1, a, b) = {(q_2, ∈)}
is
(A) L(abb*a)
(B) {a} U L(abb*a)
(C) L(ab*a)
(D) {a} U L(ab*a)
Ans: b

Q A pushdown automation M = (Q, Σ, Γ, δ, q_0, z, F) is set to be deterministic subject to which of the following condition(s), for every q ∈ Q, a ∈ Σ ∪ {∈} and b ∈ Γ
(s1) δ(q, a, b) contains at most one element
(s2) if δ(q, ∈, b) is not empty then δ(q, c, b) must be empty for every c ∈ Σ (NET-JUNE-2013)
(A) only s1
(B) only s2
(C) both s1 and s2
(D) neither s1 nor s2
Ans: c

Q Consider the NPDA ⟨Q = {q0, q1, q2}, Σ = {0, 1}, Γ = {0, 1, ⊥}, δ, q0, ⊥, F = {q2}⟩, where (as per usual convention) Q is the set of states, Σ is the input alphabet, Γ is stack alphabet, δ is the state transition function, q0 is the initial state, ⊥ is the initial stack symbol, and F is the set of



accepting states, The state transition is as follows:
Which one of the following sequences must follow the string 101100 so that the overall string is accepted by the automaton? **(GATE – 2015) (1 Marks)**

a) 10110 b) 10010 c) 01010 d) 01001

ANSWER B

Q Consider the pushdown automaton (PDA) below which runs over the input alphabet (a, b, c). It has the stack alphabet $\{Z_0, X\}$ where Z_0 is the bottom of – stack marker. The set of states of the PDA is {s, t, u, f} where s is the start state and f is the final state. The transition of the PDA given below is depicted in a standard manner. For example, the transition. For example the transition (s, b, X) → (t, X, Z_0) means that if the PDA is in state s and the symbol on the top of the stack is X, then it can read b from the input and move to state after popping the top of stack and pushing the symbol Z_0 and X (in that order) on the stack.(GATE-2003) (2 Marks)

(s, a, Z_0) → (s, XXZ_0)

(s, ε, Z_0) → (f, ε)

(s, a, X) → (s, XXX)

(t, b, X) → (t, ε)

(t, b, X) → (t, ε)

(t, c, X) → (u, ε)

(u, c, X) → (u, ε)

(u, ε, Z_0) → (f, ε)

The language accepted by the PDA is

a) $\{a^l b^m c^n \mid l = m = n\}$ **b)** $\{a^l b^m c^n \mid l = m\}$

c) $\{a^l b^m c^n \mid 21 = m + n\}$ **d)** $\{a^l b^m c^n \mid m = n\}$

Ans: c

Q Given the following statements: (NET-JUNE-2012)

(i) The power of deterministic finite state machine and non- deterministic finite state machine are same.

(ii) The power of deterministic pushdown automaton and non- deterministic pushdown automaton are same.

Which of the above is the correct statement(s)?

(A) Both (i) and (ii)

(B) Only (i)

(C) Only (ii)

(D) Neither (i) nor (ii)

Ans: b

Introduction

Language usually contains infinite number of strings (string length is finite), we cannot tabulate each and every string to represent the language, therefore like automata, grammar is also a mathematical model of representing a language, using which we can generate the entire language. Therefore, a grammar is usually thought of as a language generator.

Formal Grammar

BASIC DEFINITIONS

A phrase-structure grammar (or simply a grammar) is a 4-tuple (V_N, ∑, P, S), where

1. V_N is a finite nonempty set whose elements are called variables,
2. ∑ is a finite nonempty set whose elements are called terminals, $V_N \cap \Sigma = \Phi$.
3. S is a special variable (i.e., an element of V_N ($S \in V_n$)) called the start symbol. Like every automaton has exactly one initial state, similarly every grammar has exactly one start symbol.
4. P is a finite set whose elements are $\alpha \rightarrow \beta$. where α and β are strings on $V_N \cup \Sigma$. α has at least one symbol from V_N, the element of P are called productions or production rules or rewriting rules. $\{\Sigma \cup V_n\}^*$ some writer refers it as total alphabet

For a formal valid production,

$$\alpha \rightarrow \beta$$

$$\alpha \in \{\Sigma \cup V_n\}^* Vn \{\Sigma \cup V_n\}^*$$

$$\beta \in \{\Sigma \cup V_n\}^*$$

Some points to note about productions

1. Reverse substitution is not permitted. For example, if S → AB is a production, then we can replace S by AB but we cannot replace AB by S.
2. No inversion operation is permitted. For example, if S → AB is a production, it is not necessary that AB →S is a production.
3. To generate a string in the language, one begins with a string consisting of only a single *start symbol*. The *production rules* are then applied in any order, until a string that contains neither the start symbol nor designated *nonterminal symbols* is produced. A sequence of rule applications is called a *derivation*.
4. A production rule is applied to a string by replacing one occurrence of the production rule's left-hand side in the string by that production rule's right-hand side.

Defining a language by grammar

The concept of defining a language using grammar is, starting from a start symbol using the production rules of the grammar any time, deriving the string. Here every time during derivation a production is used as its LHS is replaced by its RHS, all the intermediate stages(strings) are called sentential forms. The language formed by the grammar consists of all distinct strings that can be generated in this manner.

$$L(G) = \{w \mid w \in \Sigma^*, S \rightarrow^* W\}$$

- $\rightarrow^*$(reflexive, transitive closure) means from s we can derive w in zero or more steps
- Using same idea, we do processing of natural languages in computers, Actively used in compliers
- L (G) is the set of all terminal strings derived from the start symbol S.
- G_1 and G_2 are equivalent if $L(G_1) = L(G_2)$.

Let us take some examples to understand how to find L (G).

Example:

If $G = (\{S\}, \{0, 1\}, \{S \rightarrow 0S1, S \rightarrow \Lambda\}, S)$, find $L(G)$.

Ans. $\mathbf{S \rightarrow 0S1 \rightarrow 0^2S1^2 \rightarrow \ldots \rightarrow 0^n1^n}$, The string will end using S→^

Therefore,

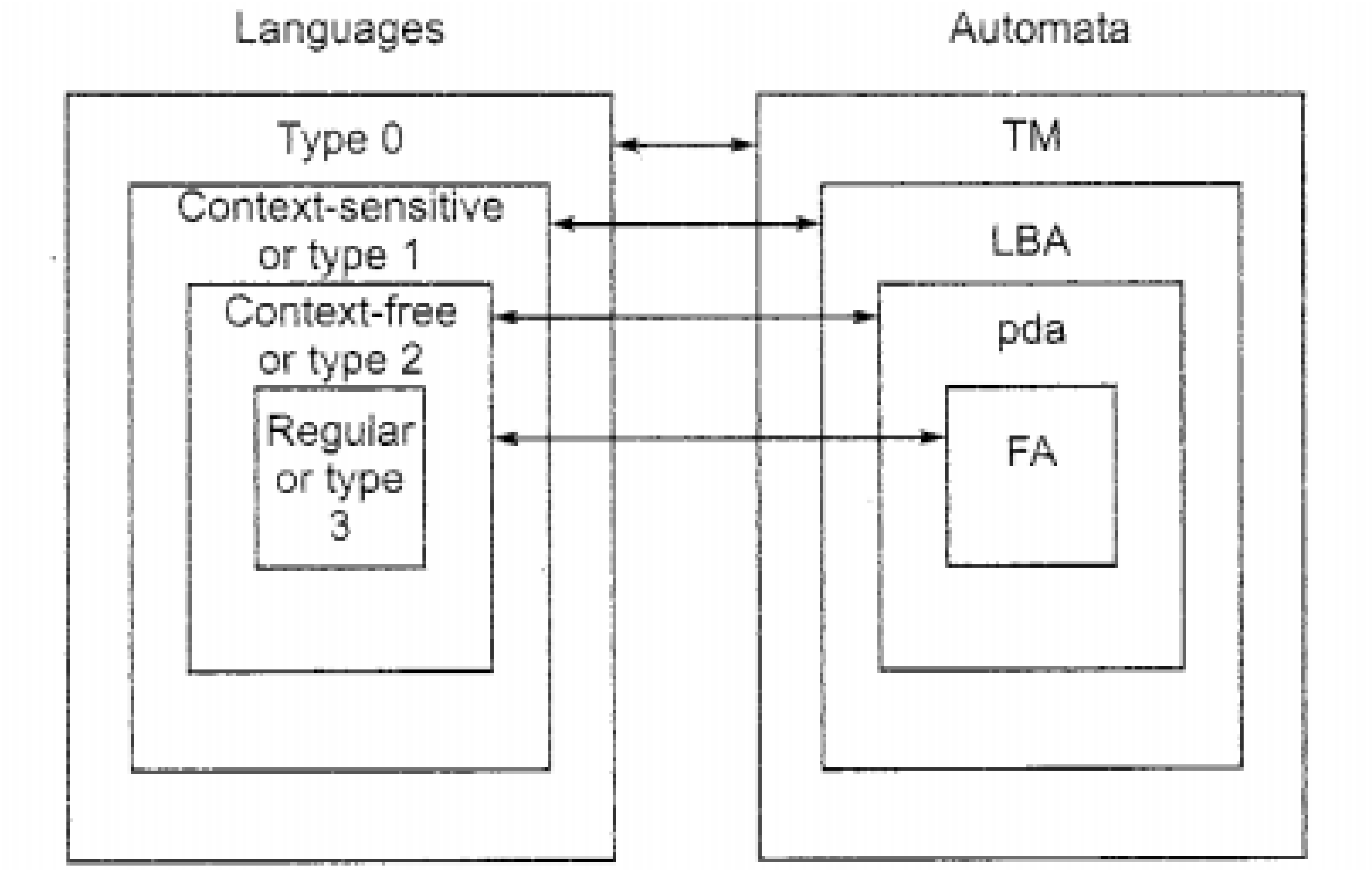
$0^n1^n \in L(G)$

Chomsky Classification of Languages

- Chomsky classified the grammars into four types in terms of productions (types 0-3).
- This hierarchy of grammars was described by Noam Chomsky in 1956. From type 0 to type 3, we will be putting more and more restrictions. We will see that more restrictive is grammar easy will be the language and more liberal is the grammar difficult will be the language. Based on the production rules of the grammar, we can classify the formal grammar into four types, based on which we generate different languages.

LANGUAGES AND AUTOMATA

Following are the machines that accepts the following grammars.



LANGUAGES AND THEIR RELATION

Let L_0, L_{CS}, L_{CF} and L_R be the languages such that they denote families of languages of type 0, context free languages, context sensitive languages and regular languages.

- By definition, $\mathbf{L_R \subseteq L_{CF}, L_{CS} \subseteq L_0, L_{CF} \subseteq L_0}$.
- $\mathbf{L_{CF} \subseteq L_{CS}}$, The inclusion relation is not immediate as we allow $A \rightarrow \Lambda$ in context-free grammars even when $A \neq S$ (start symbol), but not in context-sensitive grammars (we allow only $S \rightarrow \Lambda$ in context-sensitive grammars).
- **So, in general,** $\mathbf{L_R \subseteq L_{CF} \subseteq L_{CS} \subseteq L_0}$

Important Remark

- Two grammars of different types may generate the same language.

Type 0 Grammar

- also known as Unrestricted Grammar, phase structured grammar, recursively enumerable grammar used to generate recursive enumerable language which is accepted by a Turing machine.
- A type 0 grammar is without any restrictions.
- No restriction on the production rule, that is if there is a production from

$$\alpha \rightarrow \beta$$

$$\alpha \in \{\Sigma \cup V_n\}^* Vn \{\Sigma \cup V_n\}^*$$

$$\beta \in \{\Sigma \cup V_n\}^*$$

Type 1 Grammar

- Also known as case sensitive language, length increasing grammar, non-contracting grammar, used to generate context sensitive language which is accepted by a linear bounded automaton.

$$\alpha\ A\ \beta \rightarrow \alpha\ \delta\ \beta$$

$$\alpha\ ,\ \beta \in \{\Sigma \cup V_n\}^* \quad A \in V_n \quad \delta \in \{\Sigma \cup V_n\}^+$$

or

$$\alpha \rightarrow \beta$$

$$\alpha \in \{\Sigma \cup V_n\}^* Vn \{\Sigma \cup V_n\}^*$$

$$\beta \in \{\Sigma \cup V_n\}^+$$

$$|\alpha| <= |\beta|$$

- As from the rule we can understand that we cannot have null production, in order to solve that problem, Production $S \rightarrow \in$, is allowed if S do not appear on the right-hand side of the production.
- A grammar is called type 1 or context-sensitive or context dependent if all its productions are type 1 productions.
- Very difficult to have a parse tree
- FORTRAN, PL1 with CGF we use STD

Type 2 Grammar

- Also known as Context Free Grammar, which will generate context free language that will be accepted by push down automata. (NPDA default case)
- if there is a production, from

$$\alpha \rightarrow \beta$$

$$\alpha \in V_n \qquad |\alpha| = 1$$

$$\beta \in \{\Sigma \cup V_n\}^*$$

- In other words, the L.H.S. has no left context or right context.
- A grammar is called a type 2 grammar if it contains only type 2 productions.
- Eg ALGOL 60, PASCAL

Type 3 Grammar

Used to generate Regular Grammar which will generate regular language which will be accepted by finite machine.

Regular grammar can be of two types either left linear or right linear.

- Left linear grammar, support two types of production

$$A \rightarrow a \ / \ Ba$$

$$A, B \in V_n \quad |A| = |B| = 1$$

$$a \in \Sigma^*$$

- right regular grammar

$$A \rightarrow a \ / \ aB$$

$$A, B \in V_n \quad |A| = |B| = 1$$

$$a \in \Sigma^*$$

– however, if left-linear rules and right-linear rules are combined, the language need no longer be regular.

Examples:

Find the highest type number which can be applied to the following productions:

(a) $S \rightarrow Aa, \quad A \rightarrow c \mid Ba, \quad B \rightarrow abc$
(b) $S \rightarrow ASB \mid d, \quad A \rightarrow aA$
(c) $S \rightarrow aS \mid ab$

(a) $S \rightarrow Aa$, $A \rightarrow Ba$, $B \rightarrow abc$ are type 2 and $A \rightarrow c$ is type 3. So the highest type number is 2.
(b) $S \rightarrow ASB$ is type 2. $S \rightarrow d$, $A \rightarrow aA$ are type 3. Therefore, the highest type number is 2.
(c) $S \rightarrow aS$ is type 3 and $S \rightarrow ab$ is type 2. Hence the highest type number is 2.

Q Consider the grammar

S → aaaS | a | aa

L(G) =?

a) L(G) = {w: |w| mod 3 = 0}

b) L(G) = {w: |w| mod 3 = 1 or 2}

c) L(G)= L(a*)

d) d) L(G) = L(a*)- (λ)

Q The basic limitation of FSM is that

a) It cannot remember arbitrary large amount of information

b) It sometimes fails to recognize grammars that are not regular

c) It sometimes fails to recognize grammars that are regular

d) All of these

Q Finite state machine can recognize language generated by __________. **(NET-NOV-2017)**

a) Only context free grammar

b) Only context sensitive grammar

c) Only regular grammar

d) any unambiguous grammar

Ans: c

Q The regular grammar for the language L = {$a^n b^m$ | n + m is even} is given by **(NET-AUG-2016)**

a) S → S1 | S2
S1 → a S1| A1
A1 → b A1| λ
S2 → aaS2| A2
A2 → b A2| λ

b) S → S1 | S2
S1 → a S1| aA1
S2 → aaS2| A2
A1 → b A1| λ
A2 → b A2| λ

c) S → S1 | S2
S1 → aaa S1| aA1
S2 → aaS2| A2
A1 → b A1| λ
A2 → b A2| λ

d) S → S1 | S2

S1 → aa S1| A1
S2 → aaS2| A2
A1 → bb A1| λ
A2 → bb A2| λ
Ans: d

Q Let Σ = {a, b} and language L = {aa, bb}. Then, the complement of L is **(NET-AUG-2016)**
a) {λ, a, b, ab, ba} ∪ {w ∈ {a, b}* | |w| > 3}
b) {a, b, ab, ba} ∪ {w ∈ {a, b}* | |w| > 3}
c) {w ∈ {a, b}* | |w| > 3} ∪ {a, b, ab, ba}
d) {λ, a, b, ab, ba} ∪ {w ∈ {a, b}* | |w| ≥ 3}
Ans: d

Q The regular grammar for the language L = {w | n_a (w) and n_b (w) are both even, w ∈ {a, b}*} is given by : (Assume, p, q, r and s are states) **(NET-DEC-2014)**
(A) p → aq | br | λ
q → bs | ap
r → as | bp
s → ar | bq
p and s are initial and final states.

(B) p → aq | br
q → bs | ap
r → as | bp
s → ar | bq
p and s are initial and final states.

(C) p → aq | br | λ
q → bs | ap
r → as | bp
s → ar | bq
p is both initial and final states.

(D) p → aq | br
q → bs | ap
r → as | bp
s → ar | bq
p is both initial and final states.
Ans: c

Q A regular grammar for the language L = {$a^n b^m$ | n is even and m is even}is given by **(NET-SEP-2013)**
(A) S → aSb | S1
S1 → bS1a | λ

(B) S → aaS | S1
S1 → bSb | λ

(C) S → aSb | S1
S1 → S1ab | λ

(D) S → aaS | S1
S1 → bbS1 | λ
Ans: d

Q Given the following productions of a grammar: **(NET-SEP-2013)**
S → aA | aBB
A → aaA |λ
B → bB | bbC
C→ B
Which of the following is true?
(A) The language corresponding to the given grammar is a set of even number of a's.
(B) The language corresponding to the given grammar is a set of odd number of a's.
(C) The language corresponding to the given grammar is a set of even number of a's followed by odd number of b's.
(D) The language corresponding to the given grammar is a set of odd number of a's followed by even number of b's.
Ans: b

Q If G is a grammar with productions
S → SaS | aSb | bSa | SS | ϵ
where S is the start variable, then which one of the following strings is not generated by G? (GATE-2017) (1 Marks)
(A) abab **(B)** aaab **(C)** abba **(D)** babba
ANSWER D

Q Consider the following grammar (**GATE – 2004) (2 Marks)**

S → bS | aB | b

A → bA | aB|

B → bB | aS| a

Let N_a (w) and N_b (w) denote the number of a's and b's in a string w respectively. The language L(G)I {a, b}* generated by G is

a) {w | N_a(w) > 3 N_b(w)} **b)** {w | N_b(w) > 3 N_a(w)}

c) w | N_b(w) = 3k, k $\hat{I}$ {0, 1, 2,…….}} **d)** w | N_a(w) = 3k, k $\hat{I}$ {0, 1, 2,…….}}

Ans: c

Q Consider the regular grammar: **(GATE-2005) (1 Marks)**

S → X a | Ya

X → Za

Z → Sa | B

Y → Wa

W → Sa

Where S is the starting symbol, the set of terminals is {a} and the set of non – terminals is {S, W, X, Y, Z). We wish to construct a deterministic is (DFA) to recognize the same language. What is the minimum number of states required for the DFA?

a) 2 **b)** 3 **c)** 4 **d)** 5

Ans: b

Q What is the regular expression for the language generated by

S → aS | bA A → d | ccA

a) a*bd **b)** a* (bd)(bcc)* d **c)** a* b(cc)* d **d)** None of these

Q Write the grammar for the regular expression a*b*

a) S → AB, A → aA | bB | ϵ, B → bB |aA | ϵ

b) S→ AB, A → aA | ϵ, B → bB | ϵ

c) S → ab | ϵ, A → aA | ϵ, B → bB | ϵ

d) None of these

Q $\alpha \rightarrow \beta$ and $|\alpha| \le |\beta|$ is a production rule for _________ grammar.

Q Language L1 is defined by the grammar: S1 -> aS1b | ε

Language L2 is defined by the grammar: S2 -> abS2 | ε

Consider the following statements **(GATE-2016) (2 Marks)**

P: L1 is regular

Q: L2 is regular

Which one of the following is TRUE?

(A) Both P and Q are true

(B) P is true and Q is false

(C) P is false and Q is true

(D) Both P and Q are false

Answer: (C)

Q Match the following: **(NET-JUNE-2012)**

(i) Regular Grammar	(a) Pushdown automaton
(ii) Context free Grammar	(b) Linear bounded automaton
(iii) Unrestricted Grammar	(c) Deterministic finite automaton
(iv) Context Sensitive Grammar	(d) Turing machine

	(i)	(ii)	(iii)	(iv)
(A)	(c)	(a)	(b)	(d)
(B)	(c)	(a)	(d)	(b)
(C)	(c)	(b)	(a)	(d)
(D)	(c)	(b)	(d)	(a)

Ans: b

Q Consider a language A defined over the alphabet sum = {0, 1} as The expression A = left { 0^{left lfloor n / 2 right rfloor}1^{n}: n> = 0 right } The expression left lfloor n / 2 right rfloor means the floor of n/2, or what you get by rounding n/2 down to the nearest integer. Which of the following is not an example of a string in A? **(NET-DEC-2015)**

a) 011 b) 0111 c) 0011 d) 001111

Ans: c

Q Consider the following context-free grammars:

$$G_1:\ S \rightarrow aS|B,\ B \rightarrow b|bB$$

$$G_2:\ S \rightarrow aA|bB,\ A \rightarrow aA|B|\varepsilon,\ B \rightarrow bB|\varepsilon$$

Which one of the following pairs of languages is generated by G1 and G2, respectively?

(A) $\{a^m b^n | m > 0 \text{ or } n > 0\}$ and $\{a^m b^n | m > 0 \text{ and } n > 0\}$

(B) $\{a^m b^n | m > 0 \text{ and } n > 0\}$ and $\{a^m b^n | m > 0 \text{ or } n \geq 0\}$

(C) $\{a^m b^n | m \geq 0 \text{ or } n > 0\}$ and $\{a^m b^n | m > 0 \text{ and } n > 0\}$

(D) $\{a^m b^n | m \geq 0 \text{ and } n > 0\}$ and $\{a^m b^n | m > 0 \text{ or } n > 0\}$

Answer: (D)

Q Match the following: (NET-JUNE-2013)

a. Context sensitive language	i. Deterministic finite automation
b. Regular grammar	ii. Recursive enumerable
c. Context free grammar	iii. Recursive language
d. Unrestricted grammar	iv. Pushdown automation

Codes :

	a	b	c	d
(A)	ii	i	iv	iii
(B)	iii	iv	i	ii
(C)	iii	i	iv	ii
(D)	ii	iv	i	iii

Ans: c

Q If all the production rules have single non - terminal symbol on the left side, the grammar defined is: (NET-JUNE-2015)

a) context free grammar

b) context sensitive grammar

c) unrestricted grammar

d) phrase grammar

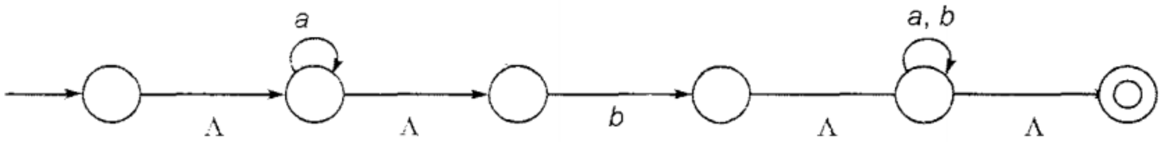
Ans: a

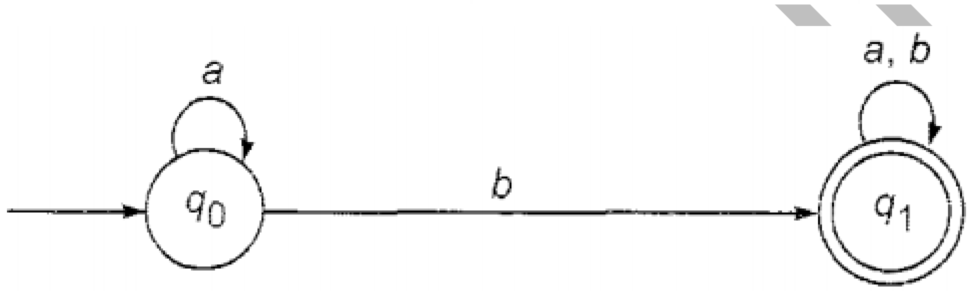
CONSTRUCTION OF A REGULAR GRAMMAR

Let us understand this with the help of an example:

Construct a regular grammar G generating the regular set represented by P = **a*b(a + b)***.

- We draw the corresponding DFA to the given problem.





Now since we have the DFA it becomes fairly easy to generate the grammar.

Let G = ({A0, A1}, {a, b}, P, A0), A0 and A1 are representing the states q0 and q1 respectively.

where P is given by:

A0 → aA0, A0→ bA1, A0→ b and

A1→ aA1, A1→ bA1, A1→ a, A1→ b

G is the required regular grammar.

CONSTRUCTION OF A TRANSITION SYSTEM M ACCEPTING L(G) FOR A GIVEN REGULAR GRAMMAR G

Let us take an example to understand:

Let G =({A0, A1}, {a, b}, P, A0), where P consists of

A0 → aA1, Al → bA1, A1 → a, A1→ bA0.

Construct a transition system M accepting L(G).

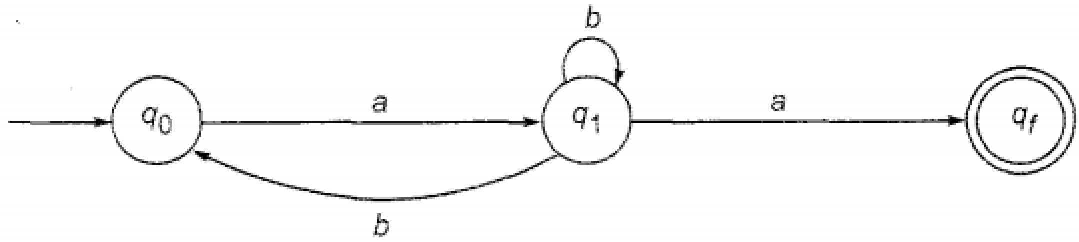
Solution:

Let M = ({ q0, q1, qf}, {a, b}, δ, q0 {qf}), where q0 and q1 correspond to A0 and A1 respectively and qf is the new (final) state introduced.

A0→ aA1 induces a transition from q0 to q1 with label a.

Similarly, A1→ bAl and A1 → bA0 induce transitions from q1 to q1 with label b and from ql to q0 with label b, respectively.

Al→ a induces a transition from q1 to qf with label a. So the resulting transition system is:



Decision properties

- Approximately all the properties are decidable in case a finite automaton. Here we will use machine model to proof decision properties.

i) Emptiness

ii) Non-emptiness

iii) Finiteness

iv) Infiniteness

v) Membership

vi) Equality

- ## Emptiness & Non-emptiness

Step 1: - select the state that cannot be reached from the initial states & delete them (remove unreachable states)

Step 2: - if the resulting machine contains at least one final states, so then the finite automata accepts the non-empty language.

Step 3: - if the resulting machine is free from final state, then finite automata accepts empty language.

- ## Finiteness & Infiniteness

Step 1: - select the state that cannot be reached from the initial state & delete them (remove unreachable states)

Step 2: - select the state from which we cannot reach the final state & delete them (remove dead states)

Step 3: - if the resulting machine contains loops or cycles then the finite automata accepts infinite language

Step 4: - if the resulting machine do not contain loops or cycles then the finite automata accepts infinite language

- ## Membership

  Membership is a property to verify an arbitrary string is accepted by a finite automaton or not i.e. it is a member of the language or not.

  Let M is a finite automata that accepts some strings over an alphabet, and let 'w' be any string defined over the alphabet, if there exist a transition path in M, which starts at initial state & ends in anyone of the final state, then string ‘w’ is a member of M, otherwise ‘w’ is not a member of M.

- **Equality**

Two finite state automata M_1 & M_2 is said to be equal if and only if, they accept the same language.

Minimise the finite state automata and the minimal DFA will be unique.

Closure Properties of Regular Languages

- Regular languages are closed under following operations
  - Kleen Closure
  - Positive closure
  - Complement
  - Reverse Operator
  - Prefix Operator
  - Complement
  - Union
  - Intersection
  - Set Difference operator
  - Prefix operator
  - Symmetric Difference
  - Quotient Operator
  - Substitution
  - Homomorphism
  - Inverse Homomorphism
  - Max
  - Min
  - Cycle

Q Which of the following is TRUE? **(GATE-2007) (2 Marks)**

a) Every subset of a regular set is regular.

b) The union of two non-regular sets is not regular.

c) Every finite subset of a non-regular set is regular.

d) Infinite union of finite sets is regular

ANSWER C

Q Which of the following statements is false? (**GATE – 1998) (1 Marks)**

a) Every finite subset of a non – regular set is regular

b) Every subset of a regular set is regular

c) Every finite subset of a regular set is regular

d) The intersection of two regular sets in regular

Ans: b

Q Which of the following is TRUE? (**GATE – 2007) (1 Marks)**

a) Every subset of a regular set is regular

b) Every finite subset of a non-regular set is regular

c) The union of two non – regular sets is not regular

d) Infinite union of finite sets is regular

Ans: b

Q Which of the following statements about regular languages is NOT true? **(GATE-2006) (1-Marks)**

a) Every language has a regular superset

b) Every language has a regular subset

c) Every subset of a regular language is regular

d) Every subset of a finite language is regular

Ans: c

Q Which of the following is true? (**GATE – 2007) (1 Marks)**

a) Infinite union of regular set is regular

b) The union of two non-regular net is not regular

c) finite union of infinite set is regular

d) every R.L is also C.F.L

Ans: b

Q Let L_1 and L_2 are regular sets defined over alphabet ∑*. Mark the false statement

a) $L_1 \cup L_2$ is regular

b) $L_1 \cap L_2$ is not regular

c) ∑* -L_1 is regular

d) $L_1 *$ is regular

Q Which of the following statements are true?

i) The complement of a language is always regular

ii) The intersection of regular languages is regular

iii) The complement of a regular language is regular

a) i) and ii) only **b)** ii) and iii) only **c)** i) and iii) only **d)** All of the above

ANSWER: (B)

Q If L is a regular language over ∑ = {a, b}, which one of the following languages is NOT regular? (**GATE – 2019) (1 Marks)**

a) L· LR {xy | x ∈ L, y^R∈ L}

b) Suffix (L) = {y ∈ ∑* | ∃x ∈ ∑* such that xy ∈ L}

c) Prefix (L) = {x ∈ ∑* | ∃y ∈ ∑* such that xy ∈ L}

d) {w w^R | w ∈ L}

Answer: (B)

Q Consider the following two statements **(GATE-2016) (1 Marks)**

I. If all states of an NFA are accepting states then the language accepted by the NFA is Σ*.

II. There exists a regular language A such that for all languages B, A ∩ B is regular.

Which one of the following is CORRECT?

(A) Only I is true

(B) Only II is true

(C) Both I and II are true

(D) Both I and II are false

Answer: (B)

Q The symmetric difference of two sets S_1 and S_2 is defined as
$S_1 \ominus S_2 = \{x \mid x \in S_1$ or $x \in S_2$, but x is not in both S_1 and $S_2\}$
The nor of two languages is defined as nor (L_1, L_2) = {w | w |∈L_1 and w |∈ L_1}. Which of the following is correct? **(NET-JULY-2016)**
a) The family of regular languages is closed under symmetric difference but not closed under nor.
b) The family of regular languages is closed under nor but not closed under symmetric difference.
c) The family of regular languages are closed under both symmetric difference and nor.
d) The family of regular languages are not closed under both symmetric difference and nor.
Ans: c

Q Consider the following two languages:
L1 = $\{0^i1^j \mid \gcd(i, j) = 1\}$
L2 is any subset of 0*
Which of the following is correct? **(NET-JULY-2016)**
a) L1 is regular and L2* is not regular
b) L1 is not regular and L2* is regular
c) Both L1 and L2* are regular languages
d) Both L1 and L2* are not regular languages
Ans: b

Q Given L_1=L(a*baa*) and L_2=L(ab*). The regular expression corresponding to language $L_3 = L_1/L_2$ (right quotient) is given by **(NET-JUNE-2013)**

(A) a*b **(B)** a*baa* **(C)** a*ba* **(D)** None of the above

Q Let L1, L2 are regular languages and L3 & L4 are non-regular languages then which of the following need not be regular?

a) $(L1 \cup L2) / (L3 \cup L4)$

b) $(L1 \cap L2) / (L3 \cap L4)$

c) $(L1 \cup L3) / (L2 \cup L4)$

d) L1 / L3

Q Let X = {0, 1}, L = X* and R = $\{\frac{0^n1^n}{n} > 0\}$ then the language L ∪ R and R respectively

a) Regular, Regular

b) None regular, Regular

c) Regular, Not regular

d) Not regular, Not regular