June 2014 Paper II

by Aditi Ma'am…

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)

(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.

June 2014 Paper II

by Aditi Ma'am...

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)

(A) p→aq | br | λ, q→bs | ap, r→as | bp, s→ar | bq, p and s are initial and final states.

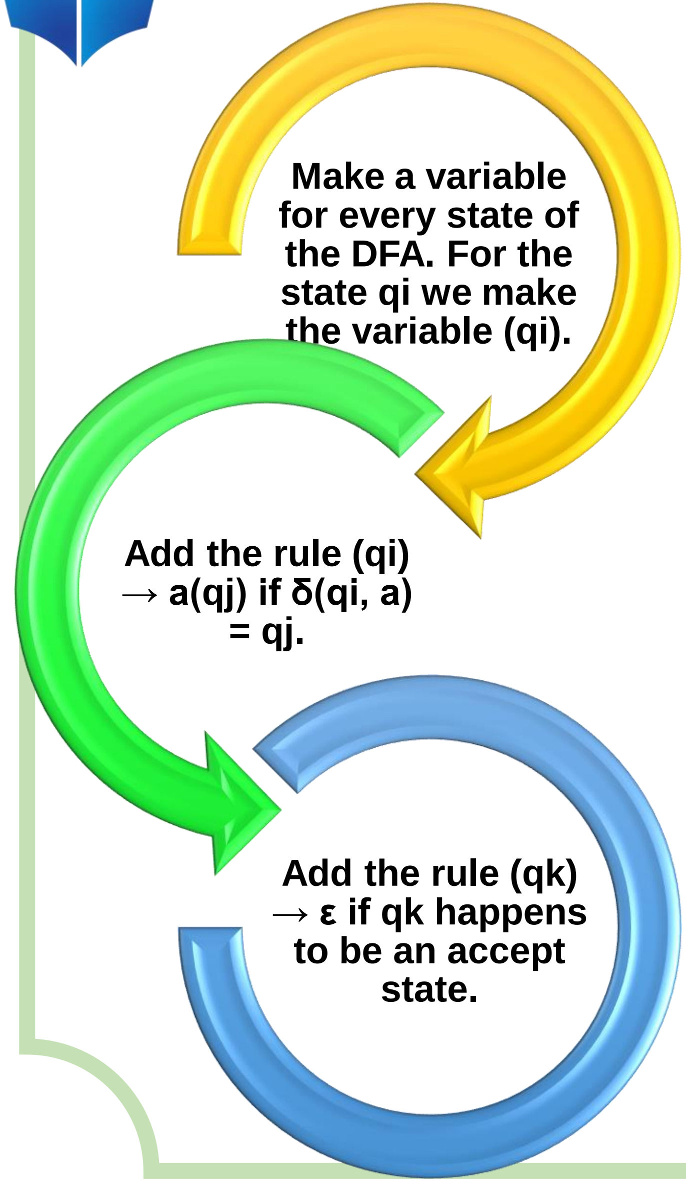
(B) p→aq | br, q→bs | ap, r→as | bp, s→ar | bq, p and s are initial and final states

(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.

For the conversion of DFA to CFG the basic theory is as follows

by Aditi Ma'am…

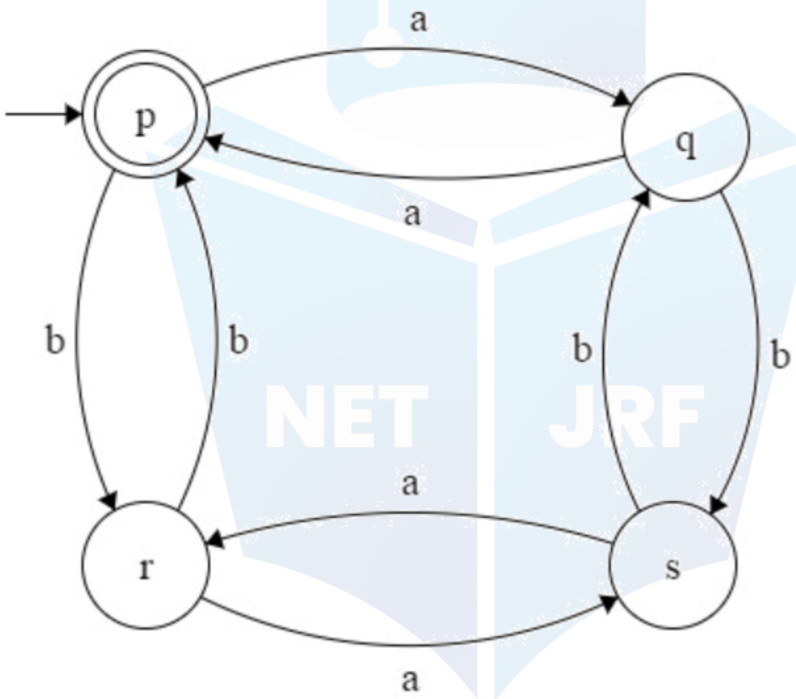


Make a variable for every state of the DFA. For the state qi we make the variable (qi).

Add the rule (qi) → a(qj) if δ(qi, a) = qj.

Add the rule (qk) → ε if qk happens to be an accept state.

$L = \{w \mid n_a(w) \text{ and } n_b(w) \text{ are both even}, w \in \{a, b\}^*\}$



$p \rightarrow aq \mid br \mid \lambda$

$q \rightarrow ap \mid bs$

$r \rightarrow as \mid bp$

$s \rightarrow ar \mid bq$

p is both initial and final states

by Aditi Ma'am...

Paper III December 2014

Given two languages :

L1 = $\{(ab)^n a^k \mid n > k, k \geq 0\}$

L2 = $\{a^n b^m \mid n \neq m\}$

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

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

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

(C) L1 is regular and L2 is regular.

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

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

Given language L1 = $\{(ab)^n\ a^k \mid n > k,\ k \geq 0\}$ here there is comparison between n and k i.e. n should be greater than k so we need stack to perform any push and pop operation it can solve by PDA and as we know that comparison is not allowed in finite automata. So, it not regular language.

Similarly language L2 = $\{a^n\ b^m \mid n \neq m\}$ here we need compare between n and m that is m should not be equal to n so again we need here a stack to solve this PDA. And here is also comparison between n and m so it is not regular language.

by Aditi Ma’am…

The term Pumping Lemma is made up of two words:-

- Pumping: The word pumping refers to generating many input strings by pushing a symbol in an input string repeatedly.
- Lemma: The word Lemma refers to the intermediate theorem in a proof.
- There are two Pumping Lemmas, that are defined for
- Regular Languages
- Context-Free Languages

Pumping Lemma For Regular Languages

Theorem: If A is a Regular Language, then A has a Pumping Length ‘P’ such that any string ‘S’ where |S ≥ P may be divided into three parts S = xyz such that the following conditions must be true:

1.) $xy^iz \in A$ for every $i \geq 0$

2.) $|y| > 0$

3.) $|xy| \leq P$

Pumping Lemma for Context-free Languages

Suppose a language L is context free, then there will exist any integer p≥1, also called pumping length such that every string S in the context free language L which has a length of p or more can be written as : s = uvwxy with the substrings following the following property:

1.) $uv^nwx^ny \in L$ for every $n \geq 0$

2.) $|vx| \geq 1$

3.) $|vwx| \leq P$

NOTE:

- ✓ **Pumping Lemma is used as a proof for irregularity of a language.**
- ✓ **Thus, if a language is regular, it always satisfies pumping lemma.**
- ✓ **If there exists at least one string made from pumping which is not in L, then L is surely not regular.**
- ✓ **The opposite of this may not always be true. That is, if Pumping Lemma holds, it does not mean that the language is regular.**

Paper III June 2015

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

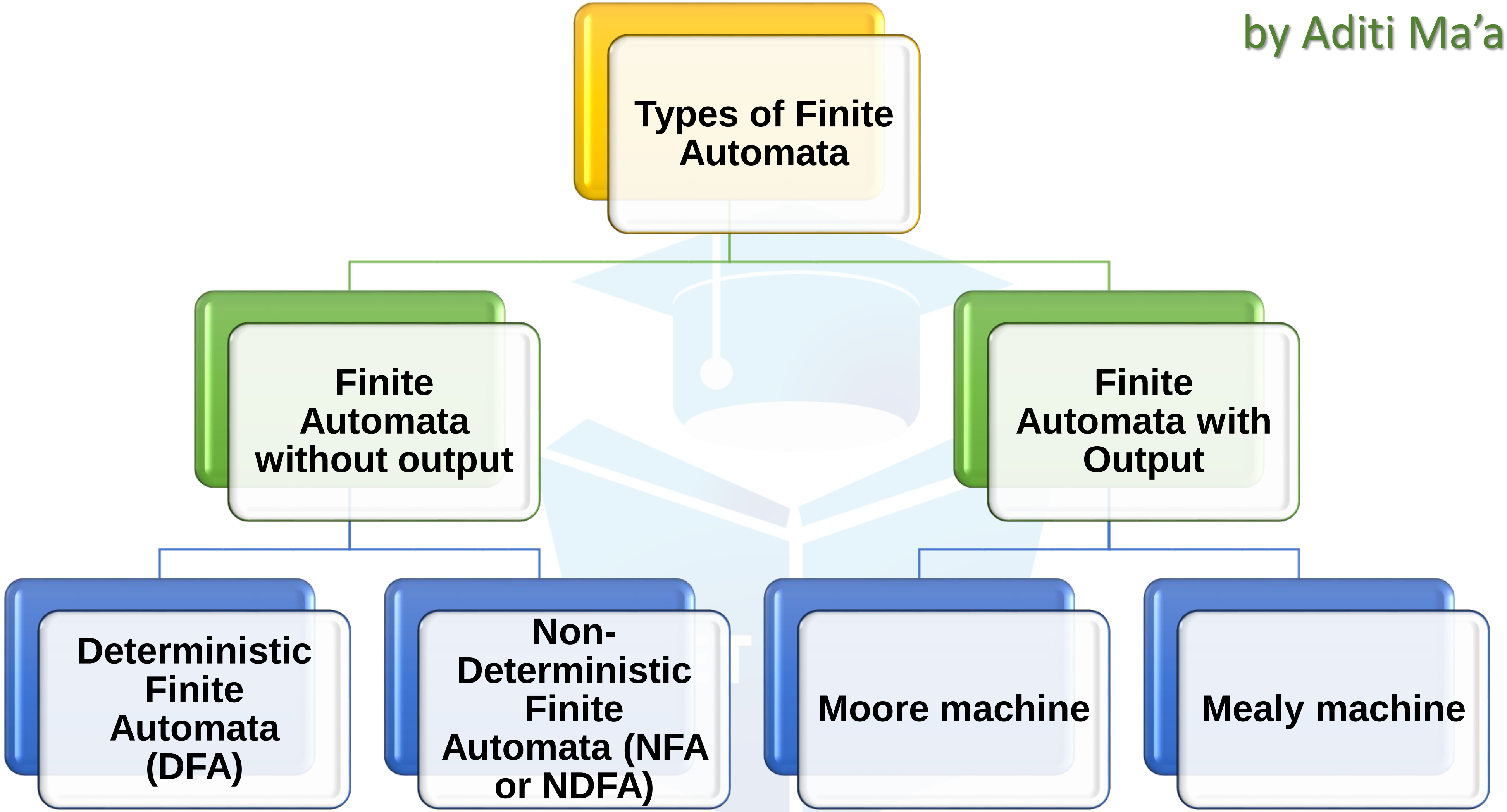
δ(q0, a)=q1	**;**	**δ(q0, b)=q2**
δ(q1, a)=q0	**;**	**δ(q1, b)=q3**
δ(q2, a)=q3	**;**	**δ(q2, b)=q0**
δ(q3, a)=q2	**;**	**δ(q3, b)=q1**

the initial and final states of the automata are:

(A) q0 and q0 respectively **(B) q0 and q1 respectively**

(C) q0 and q2 respectively **(D) q0 and q3 respectively**

by Aditi Ma’am...



Types of Finite Automata

Finite Automata without output

Finite Automata with Output

Deterministic Finite Automata (DFA)

Non-Deterministic Finite Automata (NFA or NDFA)

Moore machine

Mealy machine

by Aditi Ma'am…

Formal definition of Finite Automata

Finite automata is defined as a 5-tuples

M=(Q, Σ, δ,q0,F)

Where,

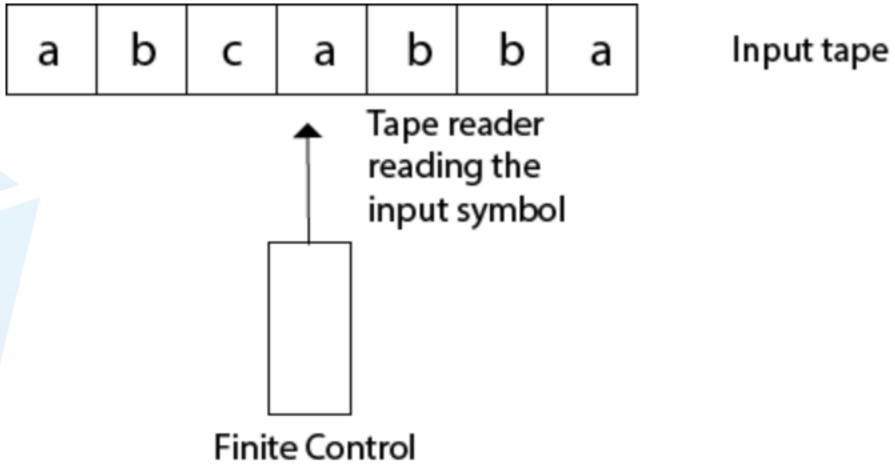
Q: Finite set called states.

Σ: Finite set called alphabets.

δ: Q × Σ → Q is the transition function.

q0 ∊ Q is the start or initial state.

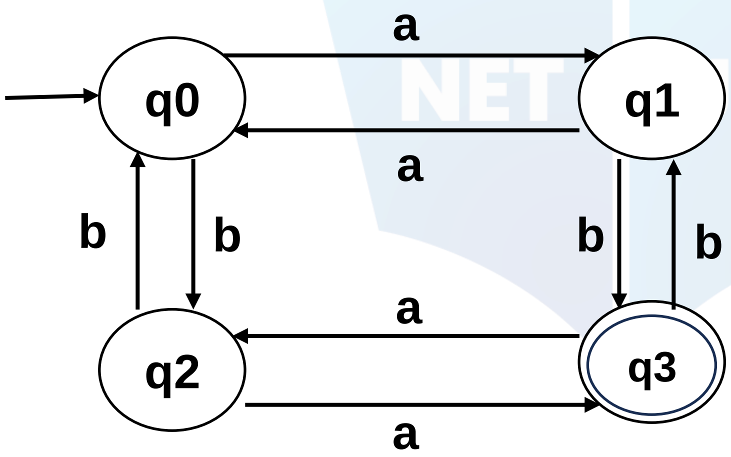
F: Final or accept state.



(D) q0 and q3 respectively

	a	b
q0	q1	q2
q1	q0	q3
q2	q3	q0
q3	q2	q1

q0 is given as initial state in all four options



by Aditi Ma’am…

The symmetric difference of two sets S1 and S2 is defined as

S1 ⊕ S2 = {x|x ∈ S1 or x ∈ S2, but x is not in both S1 and S2}
The nor of two languages is defined as
nor (L1, L2) = {w|w ∉ L1 and w ∉ L2}.

Which of the following is correct?

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.

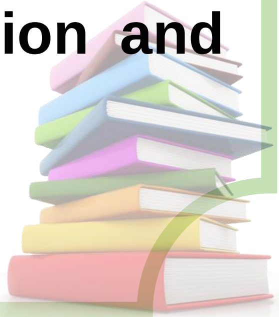
c) The family of regular languages are closed under both symmetric difference and nor.

$A \oplus B = (A^c \cap B) \cup (B^c \cap A)$

Regular languages are closed under intersection, complementation and union and hence they are closed under set difference

nor $(L1 \cup L2) = L1^c \cap L2^c$

Regular languages are closed under complementation and intersection and hence they are closed under nor.



by Aditi Ma’am…

Closure Properties

OPERATIONS	REG	DCFL	CFL	CSL	REC	REL
Union	Y	N	Y	Y	Y	Y
Intersection	Y	N	N	Y	Y	Y
Set Difference	Y	N	N	Y	Y	N
Complement	Y	Y	N	Y	Y	N
Intersection with a Regular Language	Y	Y	Y	Y	Y	Y
Union with a Regular Language	Y	Y	Y	Y	Y	Y
Concatenation	Y	N	Y	Y	Y	Y
Kleene Star	Y	N	Y	Y	Y	Y
Kleene Plus	Y	N	Y	Y	Y	Y
Reversal	Y	Y	Y	Y	Y	Y

by Aditi Ma’am…

OPERATIONS	REG	DCFL	CFL	CSL	REC	REL
Epsilon-free Homomorphism	Y	N	Y	Y	Y	Y
Homomorphism	Y	N	Y	N	N	Y
Inverse Homomorphism	Y	Y	Y	Y	Y	Y
Epsilon-free Substitution	Y	N	Y	Y	Y	Y
Substitution	Y	N	Y	N	N	Y
Subset	N	N	N	N	N	N
Left Difference with a Regular Language (L-Regular)	Y	Y	Y	Y	Y	Y
Right Difference with a Regular Language (Regular-R)	Y	Y	N	Y	Y	N
Left Quotient with a Regular Language	Y	Y	Y	N	Y	Y
Right Quotient with a Regular Language	Y	Y	Y	N	Y	Y

Paper III July 2016

Consider the following two languages:

L1 = {0^i1^j | gcd(i,j)=1}

L2 is any subset of 0*.

Which of the following is correct?

(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

(B) L1 is not regular and L2* is regular

L1 = $\{0^i1^j \mid gcd(i,j)=1\}$

There are infinite pair having gcd 1. We can’t design FA for such language. That’s why L1 is not regular.

L2 is any subset of 0*, we can construct FA for any subset of single alphabet. That’s why L2 is regular.

by Aditi Ma’am…

Paper III August 2016 (Re-test)

The regular grammar for the language L = $\{a^n b^m \mid n + m \text{ is even}\}$ is given by

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 | a A1, S2 → aa S2 | A2, A1 → bA1 | λ, A2 → bA2 | λ

C. S → S1 | S2, S1 → aaa S1 | aA1, S2 → aaS2 | A2, A1 → bA1 | λ, A2 → bA2 | λ

D. S → S1 | S2, S1 → aa S1 | A1, S2 → aaS2 | aA2, A1 → bbA1 | λ, A2 → bbA2 | b

(D) S → S1 | S2, S1 → aa S1 | A1, S2 → aaS2 | aA2, A1 → bbA1 | λ, A2 → bbA2 | b

For n+m to be even n and m should be even or n and m should be odd S→S1|S2

S1→aaS1|A1------ as A1 derives even number of b's so no problem, S1 derives even number of a's

S2→aaS2|aA2---- as A2 derives odd number of b's and aA2 will have even number of a's and b's

A1→bbA1|λ------- this derives even number of b's

A2→bbA2|b -----this derives odd number of b's

by Aditi Ma'am…

Paper III August 2016 (Re-test)

Given the following two languages:

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

$L2 = \{uww^Rv \mid u, v, w \in \{a, b\}^+, |u| \geq |v|\}$

Which of the following is correct ?

(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 L2 are not regular languages.

by Aditi Ma’am…

Paper III August 2016 (Re-test)

Given the following two languages:

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

L2 = $\{uww^Rv \mid u, v, w \in \{a, b\}^+, |u| \geq |v|\}$

Which of the following is correct ?

(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 L2 are not regular languages.

by Aditi Ma’am…

Paper III January 2017

Which of the following are not regular?

(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.

(D) Strings of a’s whose length is a perfect square.

(1) (A) and (B) only

(2) (A), (B) and (C) only

(3) (B), (C) and (D) only

(4) (B) and (D) only

(3) (B), (C) and (D) only

- **Strings of even number of a’s is Regular because we can draw Finite acceptor(FA) for this.**
- **Strings of a’s, whose length is a prime number – There are infinite No. which are prime and we can’t design FA for infinite language. This is not Regular**
- **Set of all palindromes made up of a’s and b’s. This is not Regular because we can’t design FA for infinite language.**
- **Strings of a’s whose length is a perfect square. This is not Regular because we can’t design FA for infinite language.**

Paper III November 2017

Pumping lemma for regular language is generally used for proving:

(1) whether two given regular expressions are equivalent

(2) a given grammar is ambiguous

(3) a given grammar is regular

(4) a given grammar is not regular

(4) a given grammar is not regular

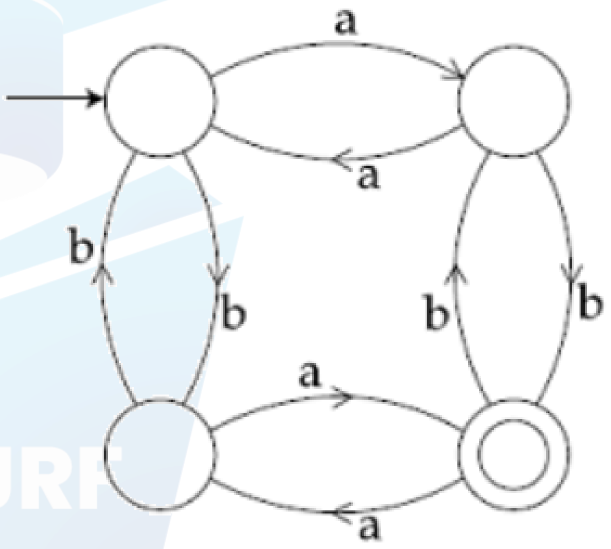
NOTE:

- ✓ **Pumping Lemma is used as a proof for irregularity of a language.**
- ✓ **Thus, if a language is regular, it always satisfies pumping lemma.**
- ✓ **If there exists at least one string made from pumping which is not in L, then L is surely not regular.**
- ✓ **The opposite of this may not always be true. That is, if Pumping Lemma holds, it does not mean that the language is regular.**

by Aditi Ma'am...

Paper II July 2018

The finite state machine given in figure below recognizes:



(1) any string of odd number of a's

(2) any string of odd number of b's

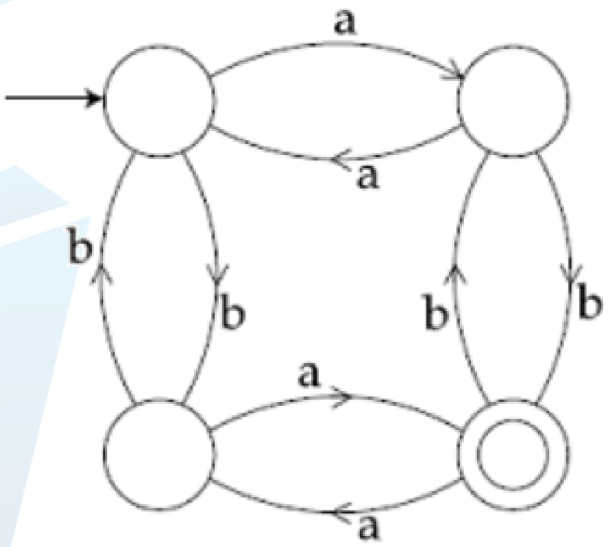
(3) any string of even number of a's and odd number of b's

(4) any string of odd number of a's and odd number of b's

by Aditi Ma'am…

Paper II July 2018

The finite state machine given in figure below recognizes:



(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

by Aditi Ma'am…

Paper II December 2018

Consider the language L given by

L = {2^{nk} | k>0, and n is non-negative integer number}

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

(1) n

(2) n+1

(3) (n(n+1))/2

(4) 2^n

by Aditi Ma’am…

(2) n+1

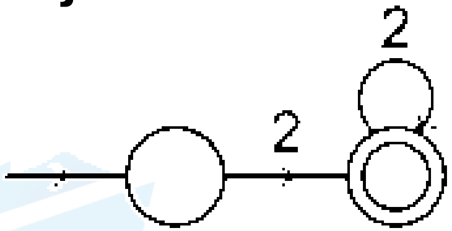
Given language is :

L = {2^{nk} | k>0, and n is non-negative integer number}

Case 1: Consider n = 1, Then k can be 1, 2 ,3,….

Strings possible in this case : {2, 2^2, 2^3, 2^4,…………}

DFA for this:

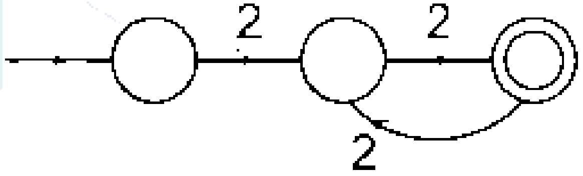


States possible with n = 1 are 2. Case 2:

When n = 2 Then k can be 1, 2, 3….

Strings possible in this case : {2^2, 2^4, 2^6, 2^8,……}

DFA for this:



States possible when n = 2 are 3.

So, in general we are getting n + 1 states in DFA

by Aditi Ma’am…

Paper II December 2018

Consider the following two languages:

L1 = {x | for some y with |y| = $2^{|x|}$,xy ∊ L and L is regular language}

L2 = {x | for some y such that |x| = |y|, xy ∊ L and L is regular language}

Which one of the following is correct?

Code:

(1) Only L1 is regular language

(2) Only L2 is regular language

(3) Both L1 and L2 are regular languages

(4) Both L1 and L2 are not regular languages

(3) Both L1 and L2 are regular languages

If y = 8 and |y| = $2^{|x|}$ then |x| = 3 since it finite it will be finite automaton.

If y = 8 and |y| = |x| then |x| = 8 since it finite it will be finite automaton.

Statement says for some y and hence L1 and L2 both are accepted by finite automata. Therefore, they are regular language

by Aditi Ma’am…

Paper II November 2020

Consider the following languages:

L1 = $\{a\grave{Z}^{Z} \mid \grave{Z} \text{ is an integer}\}$

L2 = $\{a^{Z\grave{Z}} \mid \grave{Z} \geq 0\}$

L3 = $\{\omega\omega \mid \omega \in \{a,b\}^*\}$

Which of the languages is (are) regular?

Choose the correct answer from the options given below:

(1) L1 and L2 only

(2) L1 and L3 only

(3) L1 only

(4) L2 only

by Aditi Ma’am…

(4) L2 only

L1 = {a^1, a^4, a^{27}, a^{256}, …..}

L2 = {ε, a^{11}, a^{22}, a^{33}, ……}

L3 = set of all strings starting and ending with the same word.

L1 does not contain any pattern to form a loop. Hence it's not possible to construct a DFA for L1.

L2 is a regular language since a DFA can be constructed using a loop that after every 11 symbols, the automata reaches the final state.

In L2, once the word 'ω' is scanned, there is no way to compare the next word with the starting symbol of the first word.