UNIT-I. Basic concepts and Formal Language Theory

Syllabus:

- Languages in abstract, defining languages, Kleene closure, symbol/alphabets, string/word, Formal Introduction, Mathematical foundation. Mathematical formal language theory.

- Representation of formal languages: sets, Logic, functions, relations, graphs.

- Proof Techniques: Formal proofs, Inductive proofs.

- strings and languages, examples,

- Basic machine, functionality & Limitations.

- Importance of Automata Theory, Automata, Automata formal definition & Designing Finite Automata examples, simplified notations, Nondeterminism, formal definition & Designing nondeterministic Automata, computability & complexity, Pattern matching.

- Language acceptor: concept, Machine as a language acceptor, example, Machine as a string processor.

- Finite Automata: Formal definition & Designing finite automata - basic examples, simplified notation.

- Regular Expressions & Languages:
  Recursive definition of regular expression, regular set, identities of regular expressions, regular expressions, examples and FA. Equivalence of RE, examples. Identity rules & algebraic laws of RE. Regular languages and examples. Pumping lemma for regular languages, Limitations of RE.

BASIC MATHEMATICAL OBJECTS :

- some of the fundamental mathematical objects are
  1. sets
  2. Functions
  3. Relations
  4. Logic

- Language is represented by set.

1. SETS :

- Set is a collection of things.

- A set can be ~~de~~ represented by listing it's elements within braces and separated by comma.

- A set is a collection of objects.

  Example : $L = \{a, b, c, d\}$ ---------- (i)

- The objects which are part of set are called as elements or members.

- As per (i) we say that b is a element of set L and x is not element of set L. This can be represented as follows :

  $b \in L$

  $x \notin L$

- In a set we do not distinguish repetitions of the elements.

  Example : A = { Ajay, vidya, Ajay } is the same set as {Ajay, Vidya}

- The order of elements in the set is immaterial or no impotance or irrelevant

example:

$A = \{1, 2, 3\}$, $\{3, 2, 1\}$, $\{1, 3, 2\}$ are the same set.

order of elements does't carray any meaning.

- Two sets are equal if and only if they have the same elements.

- The elements of a set need not be related in any way.

Example:

$A = \{3, \text{red}, \{x, \text{white}\}, 10.20\}$ is a set with four elements, one of which is itself a set.

- Singleton set:

A set may have only one element such set is called as singleton set.

Example:

$A = \{10\}$ is the set with 10 as it's only element.

- There is also set with no element at all such set is called as empty set. and is denoted by ϕ.

- A set can be finite and infinite in nature.

- A set is determined by its elements.

- An easy way to describe or specify a finite set is to list all it's elements.

$A = \{11, 12, 21, 22, 23, 27, 1\}$

- Writting the elements more than once doesnot change the set.

$A_1 = \{A, B, C, D\}$ and $A_2 = \{A, B, C, B, A, D, C\}$

both sets A_1 & A_2 are same.

- Infinite set

$A = \{3, 5, 7, 9, \ldots\}$

set A describes the set of odd integers greater than or equal to 3.

$A = \{x \mid x \text{ is an odd integer greater than } 1\}$

The notation "$\{x|$" at the beginning of ~~both~~ formula is usually read " the set of all x such that".

Equality of sets :

- Two sets A and B are equal, if and only if A & B contains the same elements or both are empty.

- Equality of set A & B is represented by "=" symbol.

$A = B$

- Example :

$A = \{1, 2, 3\}$

$B = \{2, 1, 3\}$

Here,

$A = B$

Scanned by CamScanner

Empty set:

- A set which contains no element is called an empty set.

- Empty set can be represented by ϕ

- Example:

$A = \phi$ It means, set A is empty.

Subset:

- A set 'A' is subset of set 'B', if and only if every element of set 'A' is an element of set 'B'.

- It is represented by symbol $\subseteq$

- Example:

$A = \{1, 2, 3, 4\}$ $B = \{1, 2, 3, 4, 5, 6\}$

Now,

$\boxed{A \subseteq B}$

Proper subset:

- If $A \subseteq B$ and $A \neq B$, then A is called proper subset of B and it is written as $A \subset B$.
- It is denoted by '$\subset$'
- Example:

$A = \{1, 2, 3\}$
$B = \{1, 2, 3, 4\}$

$\boxed{A \subset B \text{ because } A \subseteq B \text{ and } A \neq B}$

- Proper subset means set A is subset of B and set A & B are not equal sets.

Singleton set:

- A set having only one element in it is called singleton set.

- Example:

$A = \{2\}$ $B = \{1\}$

Power set:

- power set of set 'A' is the set/collection of all subsets of set 'A'.

- It is denoted by $P(A)$.

- Example:

$A = \{1, 2, 3\}$

$P(A) = \{\phi, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$

Representation of sets:

set can be represented in two ways

1. listing the elements of sets separated by comma & braces.

2. other than listing elements explicitly (Using Notations)

1. <u>Listing elements:</u>

$A = \{1, 2, 3, 4\}$

$B = \{1, 2, 3, \ldots\}$

2. other than explicitly listing elements (Using notations)

$A = \{x \mid x \text{ is an odd integer}, x > 1\}$

1a) It can ~~bea~~ be read as A is a set of all x such that x is an odd integer and x is greater than 1.

<u>Example:</u>

1. $A = \{1, 3, 5, 7, \ldots\}$ we can write equivalent set by specifying set A as,

$A = \{x \mid x \text{ is an odd integer}, x \geq 1\}$

2. $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ can be specify as,

$A = \{x \mid x \text{ is an integer}, x \geq 1 \text{ and } x \leq 10\}$

<u>OPERATIONS ON SETS:</u>

1. complement of set
2. Union
3. Intersection
4. Set difference
5. Symmetric difference

1. Complement of a set :

- Complement of set 'A' is a set of all elements of universal set which are not in set 'A'.

- Complement of set A is represented as A'.

- **Example:**

$A = \{1, 2, 3\}$

$B = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$A' = \{4, 5, 6, 7, 8, 9\}$

It can also be represented as,

$A' = \{x \mid x \in U \text{ and } x \notin A\}$

OR It can also be represented as

$A' = \{x \in U \mid x \notin A\}$ where 'U' is universal set.

2. Union of two sets :

- Union of two sets A & B is the set of elements in A or in B.

- It is denoted as $A \cup B$

- **Example:**

$A = \{4, 5, 6\}$

$B = \{1, 2, 3, 4, 6, 7, 8\}$

$A \cup B = \{1, 2, 3, 4, 5, 6, 7, 8\}$

It can also be represented as,

$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$

3. Intersection of two sets :

- Intersection of two sets A and B is the set of elements which belong to both A & B.

- Intersection of two sets are represented as $A \cap B$.

- **Example:** $A = \{1, 3, 4\}$ $B = \{1, 2, 3, 5\}$

  $A \cap B = \{1, 3\}$

  It can also be represented as,

  $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$

4. Set difference :

- The set difference of two sets 'A' & 'B' is represented as $A - B$.

- It is the set of those elements of A which are not in B.

- **Example**

  $A = \{1, 2, 4, 5\}$

  $B = \{1, 3, 2, 5, 6\}$

  $A - B = \{\cancel{1}, \cancel{2}, 4, \cancel{5}\} - \{\cancel{1}, 3, \cancel{2}, \cancel{5}, 6\}$

$A-B = \{4\}$

It can also be represented as,

$A-B = \{x \mid x \in A \text{ and } x \notin B\}$

5. Symmetric Difference :

- Symmetric Difference of two sets A & B is written as $A \oplus B$.

- It consists of those elements which are either in A or in B but not in both.

- Example:

$A = \{1, 2, 3, 4\}$

$B = \{3, 4, 5, 6\}$

i. $A \oplus B = (A-B) \cup (B-A)$

ii. $A \oplus B = (A \cup B) - (A \cap B)$

$A \oplus B = \{1,2\} \cup \{5,6\} \Rightarrow \{1,2,5,6\}$

$A \oplus B = \{x \mid (x \in A \text{ and } x \notin B) \text{ or } (x \in B \text{ and } x \notin A)\}$

6. Cartesian product :

The cartesian product of two sets A & B is the set

$A \times B = \{(a,b) \mid a \in A \text{ and } b \in B\}$

Example:

$A = \{a, b, c\}$

$B = \{x, y\}$

$A \times B = \{(a,x), (a,y), (b,x), (b,y), (c,x), (c,y)\}$

Cardinality of set: If a set 'S' has 'n' distinct elements, then cardinality of that set is 'n'. Example: $A = \{1,2,3,4\}$ cardinality of set A i.e. $|A| = 4$

It indicates total number of distinct elements in the set

2. RELATIONS:

- A binary relation from set 'A' to 'B' is a subset of $A \times B$

- A binary relation from set 'A' to 'A' is a subset of $A \times A$

- Example:

$A = \{a, b\}$

$B = \{x, y, z\}$

$A \times B = \{(a,x), (a,y), (a,z), (b,x), (b,y), (b,z)\}$

Let 'R' be Relation ~~R~~ on set A & B then R can be defined as

$R =$ subset of $A \times B$

$$\boxed{R = \{(a,x), (b,x), (b,z)\}}$$

Here, 'R' be relation on set A and B!

Properties of Relation:

Relations have three main properties.

1. Reflexive Relation
2. Symmetric Relation
3. Transitive Relation.

1. Reflexive Relation:

- A relation 'R' in set A is reflexive if xRx for every x in A.

- Example:

$A = \{1, 2, 3, 4\}$

~~AXA~~ $R = \{(1,1), (2,2), (3,3), (4,4)\}$

2. Symmetric Relation:

- A relation 'R' in set A is symmetric, ~~if aRb~~ if $a, b \in A$ and aRb then bRa

- Example:

$A = \{1, 2, 3\}$

Then symmetric relation R,

R = subset of A X A

$A \times A = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)\}$

$R = \{(1,2), (2,1), (1,3), (3,1), (2,3), (3,2)\}$

3. Transitive Relation :

A binary relation 'R' on set 'A' is transitive if and only if a, b, c ∈ A and both aRb and bRc then aRc

$A = \{1, 2, 3\}$

$A \times A = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)\}$

$R = \{(1,2), (2,3), (1,3)\}, (3,2), (3,1), (2,1)\}$

Equivalence of relation : when relation is reflexive, symmetric and transitive then it is called equivalent relation.

closure of relation :

- If a relation R in set A is not reflexive then it can be made reflexive by adding ordered pairs (x,y) to R which are not already there in R.

Example :

1. consider a Relation $R = \{(1,2), (3,2), (1,1)\}$ on set $A = \{1,2,3\}$

Sol^n :

R is not reflexive but it can be made reflexive by adding ordered pairs (2,2), (3,3)

$R = \{(1,2), (3,2), (1,1), (2,2), (3,3)\}$

∴ Reflexive closure of $R = \{(1,2), (3,2), (1,1), (2,2), (3,3)\}$

2. If a relation R in set A is not symmetric then it can be made symmetric by adding ordered pairs (y, x) to R if $(x,y) \in R$ and $(y,x) \notin R$.

Page No.
Date :

Example:

$R = \{(1,2), (3,2), (1,1)\}$
$A = \{1, 2, 3\}$
R is not symmetric but it can be made symmetric by adding $(2,3)$, $(2,1)$

$\therefore$ symmetric closure of $R = \{(1,2), (3,2), (1,1), (2,3), (2,1)\}$

3. If a relation R in set A is not transitive then it can be made transitive by adding ordered pairs (x,z) to R if $(x,y) \in R$ and $(y,z) \in R$ but $(x,z) \notin R$.

Example: consider relation $R = \{(1,2), (2,3), (1,1)\}$
R is not transitive as $1R2$ and $2R3$ then add $1R3$.

Transitive closure of $R = \{(1,2), (2,3), (1,1), (1,3)\}$

Example 1: Find the reflexive, transitive ~~closure~~ and the symmetric closure of the relation
$R = \{(1,2), (2,3), (3,4), (5,4)\}$

Soln:⇒

Reflexive closure R:
~~1R2 but~~ Add $(1,1)$, $(2,2)$, $(3,3)$, $(4,4)$, $(5,5)$ to Relation
$R = \{(1,2), (2,3), (3,4), (5,4), (1,1), (2,2), (3,3), (4,4), (5,5)\}$

Transitive closure:
$R = \{(1,2), (2,3), (3,4), (5,4)\}$
Here,
$1R2$ ~~but~~ and $2R3$ but $1R3$ not there so add $1R3$
$2R3$ and $3R4$ but $2R4$ not there add $2R4$

Scanned by CamScanner

$R = \{(1,2), (2,3), (3,4), (5,4), (1,3), (2,4)\}$

Symmetric closure:

$R = \{(1,2), (2,3), (3,4), (5,4)\}$

Here,

1R2 but 2R1 not there so add 2R1 in R

2R3 but 3R2 not there so add 3R2 in R

3R4 but 4R3 not there so add 4R3 in R

5R4 but 4R5 not there so add 4R5 in R

After adding R,

$R = \{(1,2), (2,3), (3,4), (5,4), (2,1), (3,2), (4,3), (4,5)\}$

Example 2: consider a relation $R = \{(1,1), (2,2), (3,4), (4,3)\}$ defined on set $A = \{1,2,3,4\}$ obtain R^* (Reflexive and transitive closure of R).

Sol^n: $\Rightarrow$ $R = \{(1,1), (2,2), (3,4), (4,3)\}$

Reflexive closure = add pairs (3,3), (4,4) to Relation.

$R = \{(1,1), (2,2), (3,3), (4,4), (3,4), (4,3)\}$

New Relation already transitive.

$R^* = \{(1,1), (2,2), (3,3), (4,4), (3,4), (4,3)\}$

Scanned by CamScanner

Page No.
Date :

Proof Techniques :

A proof involves a statement of the form $p \rightarrow q$.
There are several methods to find proof of statements.

Types of proof Techniques :

1. Direct proof (proof by construction)
2. Proof by contradiction
3. Proof by Induction
4. proof by contrapositive

1. DIRECT PROOF

- In direct proof method, suppose we have to prove $p \rightarrow q$ statement.
- It is also called as proof by construction

- steps of finding direct proof are as follows

step 1 : Assume p is true

step 2 : Use p to show that q must be true.

- working principle :

The direct proof method, if we have to prove that statement $p \rightarrow q$, then a direct proof method assumes p is true and uses this assumption to show that 'q' is true

Example 1 : Prove for any integer a & b, if a & b are odd then ab is odd.

Sol :⇒

Here, statements are,

a & b are odd ....(i)

ab is odd ....(ii)

Scanned by CamScanner

~~Positive~~ Even number is the number which is either multiple or divisible by 2.

We say number 'm' is even when there exists,

$m = 2x$ where x is any integer number.

We say number 'm' is odd when there exists,

$m = 2x + 1$ where, x is any integer number.

a & b are odd it can be represented as,

$a = \underbrace{2x}_{\text{even number}} + 1$ [$\because$ a is odd]

$b = 2y + 1$ [$\because$ b is odd]

To prove ab is odd,

$ab = (2x+1)(2y+1)$

$ab = 4xy + 2x + 2y + 1$

$\boxed{ab = 2(2xy + x + y) + 1}$

put $x = 2$ and $y = 3$

$ab = 2(2(2*3) + 2 + 3) + 1$

$ab = 2(12 + 5) + 1$

$ab = 2(17) + 1$

$ab = 34 + 1$

$\boxed{ab = 35}$

Here ab is odd. Hence proved.

Page No.
Date :

Example 2: If a & b are consecutive integers, then the sum a+b is odd.

Sol:⇒

Two integers a & b are consecutive if and only if $b = a+1$.

statements:
a and b are consecutive - - (i)
a+b is odd - - (ii)

Assume, a & b are consecutive then values of $a = a$ & $b = a+1$

Use this statement to prove a+b is odd.
put values of a & b

$a+b \Rightarrow a + a+1$

$$\boxed{a+b = 2a+1}$$

~~Suppose~~

suppose $a=1$

~~a+b~~ $a+b = 2\times1+1$

$a+b = 2+1$

$$\boxed{a+b = 3}$$

3 is odd number. Hence proved.

2. Proof by contradiction:

If we have to prove that $p \rightarrow q$, then method of contradiction assumes that p does not implies q and ~~th~~ try to derive some contradiction.

Page No.
Date :

i.e. $p \rightarrow q$ - - given statement

As per contradiction assume,

$p \rightarrow \neg q$

- The proof by contradiction is gronded in the fact that any proposition must be either true or false, but not both true and false at the same time.

We can use this to demonstrate $p \rightarrow Q$ by assiming both p and $\neg Q$ are simultaneasly true and deriving a contradiction. when we derive this contradi-ction it means that one of our assumption was untenable.

The method of contradiction,

$P \rightarrow Q$

1. Assume P is true.
2. Assume that $\neg Q$ is true.
3. Use P and $\neg Q$ to demonstrate a contradiction.

Example 1: prove for any integer a & b if a & b are odd then ab is odd.

Soln:⇒

Method of contradiction says that,

a & b are odd ... (i)

ab is odd ... (ii)

As per method of contradiction,

i. assume a & b are odd

ii. Assume ab is even

iii. Use statement (i) & (ii) to demonstrate a contradiction.

we can write,

a & b are odd as,

$a = 2x + 1$ ∴ (a is odd & x be any integer)

$b = 2y + 1$ ∴ (b is odd & y be any integer)

Contradiction step: ab is even.

Now, use eq^n (i) and (ii) to prove a & b are odd and ab is even.

$ab = (2x+1)\ (2y+1)$ —— (iv)

∴ so ab is even can be written as,

$ab = 2z$ —(v) (where 'z' be any integer & ab is even)

∴

$(2x+1)(2y+1) = 2z$ . . . . (vi)

$4xy + 2x + 2y + 1 = 2z$

$2(2xy + x + y) + 1 = 2z$

$$\boxed{z = (2xy + x + y) + \frac{1}{2}}$$

suppose, $x = 1$ & $y = 2$

$z = (2 \times 1 \times 2 + 1 + 2) + \frac{1}{2}$

$z = (4 + 1 + 2) + \frac{1}{2}$

$z = 7 + \frac{1}{2}$

$z = \frac{14+1}{2} = \frac{15}{2} = 7.5$

$\therefore \boxed{z = 7.5}$ Here, z is a float value: so result contradicts.

we expect 'z' as integer.

Hence, proved.

3. Method of Proof by Induction:

- Proof by induction method consists of three basic steps:

1. Basis step
2. Induction Hypothesis
3. Induction step.

The steps in details:

Step 1: Prove $P(n)$ for $n = 0$ or 1
It is called proof for basis.

Step 2: Assume that the result/properties for $P(n)$.
It is called induction Hypothesis.

Step 3: Prove $P(n+1)$ using induction hypothesis.

If a statement 'p' is to be proved then,

Basis step:
show that the statement 'p' is true for particular integer $n = 0$ or 1.

Induction Hypothesis:
Assume p is true for ~~some~~ Particular integer $k \geq n$.

Induction step: Prove that 'p' is true for $n = k+1$.

Page No.
Date :

Example 1: Show for any $n \geq 0$

$$1+2+3+\cdots+n = (n^2+n)/2$$

Sol^n: $\Rightarrow$ $P(n) = 1+2+3+\cdots+n = (n^2+n)/2$ ---- (i)

The proof by induction works in three steps.

1. **Basis step:** prove that given statement $P(n)$ is true for $n=0$. Put value $n=0$ in equation (i)

$LHS = 1+2+3+\cdots+n$

$\boxed{LHS = 0}$

Put value of $n=0$ in RHS of eq^n (i)

$R.H.S = (n^2+n)/2$

$R.H.S = ((0)^2+0)/2$

$R.H.S = 0/2$

$\boxed{R.H.S = 0}$

Hence, $\boxed{L.HS = RHS}$

2. **Induction Hypothesis:**

Assume that given statement is true for $n=k$.

i.e. $\boxed{1+2+3+\cdots+n = (n^2+n)/2}$ ---- (ii)

Put value of $n=k$ in eq^n (ii)

$L.H.S = 1+2+3+\cdots+n$

$= 1+2+3+\cdots+K$

$R.H.S = (n^2+n)/2$

$R.H.S = (K^2+K)/2$

Scanned by CamScanner

$$\boxed{1+2+3+\cdots+k = (k^2+k)/2} \quad \text{(iii)}$$

3. <u>Induction step:</u>

Prove that the statement 'p' is true for $n=k+1$

$$\therefore 1+2+3+\cdots+n = (n^2+n)/2 \quad \text{(iv)}$$

Put value of $n=k+1$ in eq^n. (iv)

$$\therefore 1+2+3+\cdots+n = (n^2+n)/2$$

$$\therefore 1+2+3+\cdots+k+1 = \frac{(k+1)^2+(k+1)}{2}$$

$$\therefore \underbrace{1+2+3+\cdots+k}+k+1 = \frac{(k+1)^2+(k+1)}{2}$$

As per induction Hypothesis in eq^n. (iii)

$$\therefore \frac{(k^2+k)}{2} + (k+1) = \frac{(k+1)^2+(k+1)}{2}$$

$$\therefore \frac{(k^2+k)+2k+2}{2} = \frac{(k+1)^2+(k+1)}{2}$$

$$\therefore \frac{k^2+k+2k+2}{2} = \frac{(k+1)^2+(k+1)}{2}$$

$$\therefore \frac{k^2+2k+1+k+1}{2} = \frac{(k+1)^2+(k+1)}{2}$$

$$\therefore \frac{(k+1)^2+(k+1)}{2} = \frac{(k+1)^2+(k+1)}{2}$$

Put $k+1=n$

$$\therefore \boxed{\frac{n^2+n}{2} = \frac{n^2+h}{2}} \qquad \therefore \boxed{L.H.S = R.H.S} \quad \text{Hence Proved}$$

Need of FA / Applications of Finite Automata:

1. Digital Circuit:
   For developing automated software for designing and checking behavior of digital circuits.

2. Compiler Design:

   Finite automata is used to implement lexical analyzer.

3. Searching:

   For finding occurrences of words, phrases or other patterns.

4. Security Protocols:

   To verify systems of all types such as communication protocols or protocol for secure exchange of information.

5. Natural Language Processing: →

   - Automata theory i.e. RE, CFG is used to perform language processing.

Notations used in DFA and NFA :

Sr.	Notation	Meaning
1.	$\rightarrow q_0$	Start/Initial state of FA
2.	q_1	Intermediate state of FA
3.	◎	Final state or Accepting state.
4.	$\xrightarrow{1}$	Transition with label 1.

- A String is said to be accepted by FA if and only if that string is processed completely using transition function and it ends up with final state of the said machine.

- Regular language :

It is a language 'L' for which there exists a deterministic finite automata i.e DFA.

Steps to follow for solving/designing any finite automata,

Step 1 : Define machine mathematically.

Step 2 : Explain the logic in brief

Step 3 : Draw transition Diagram

Step 4 : Draw transition Table

Step 5 : Example.

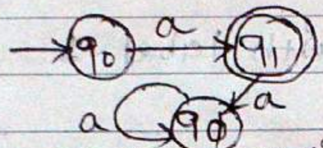
Example 1: Design a DFA that accepts strings ~~containing exactly one~~ 'a' symbol over alphabet $\Sigma = \{a\}$

Sol^n:→

Step 1: Logic:

$L = \{a\}$

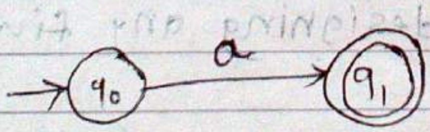
Step 2: Transition Diagram:



Step 3: State Transition Table

Q \ Σ	Input alphabet (Σ) a
→q_0	q_1
q_1*	q_ϕ
q_ϕ	q_ϕ

Modified transition diagram,



Q \ Σ	Input a
→q_0	q_1
q_1*	ϕ

Step 4: $M = (Q, \Sigma, \delta, q_0, F)$

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

Step 5: suppose string 'w'.

$w = a$

$\delta(q_0, a) \Rightarrow \delta(q_0, a)$

$\Rightarrow q_1$ It is a final state. It accepts the string "a".

Example 2: Draw DFA for the following languages over {0,1}

i. All strings of length at most five

ii. All strings with exactly two 1's

iii. All string containing at least two 0's.

iv. All strings containing at most two 0's.

v. All strings starting with 1 and length of the string is divisible by 3.

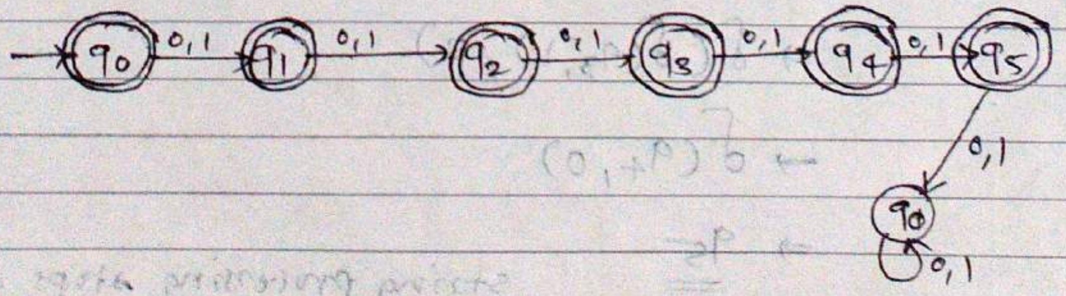
Soln:→

i. All strings of length at most five.

Step 1: Logic:

$L = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111,$ ---- { upto 5 length strings } $\}$

Step 2: State Transition Diagram



Step 3: Transition diagram

Q \ Σ	Input 0	1
$\rightarrow q_0^*$	q_1	q_1
q_1^*	q_2	q_2
$*q_2$	q_3	q_3
$*q_3$	q_4	q_4
$*q_4$	q_5	q_5
$*q_5$	q_Φ	q_Φ

Page No.
Date :

step 4:

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

$\therefore Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}.$

$\Sigma = \{0, 1\}$

$F = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

$M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{0,1\}, \delta, q_0, \{q_0, q_1, q_2, q_3, q_4, q_5\})$

Step 5: Example: string $w = 10110$

~~$\delta(w, input) \Rightarrow \delta(10110,$~~

$\delta(q_0, w) \Rightarrow \delta(q_0, 10110)$

$= \delta(\delta(q_0, 1), 0110)$

$= \delta(q_1, 0110)$

$\Rightarrow \delta(\delta(q_1, 0), 110)$

$\Rightarrow \delta(q_2, 110)$

$\Rightarrow \delta(\delta(q_2, 1), 10)$

$\Rightarrow \delta(q_3, 10)$

$\Rightarrow \delta(\delta(q_3, 1), 0)$

$\rightarrow \delta(q_4, 0)$

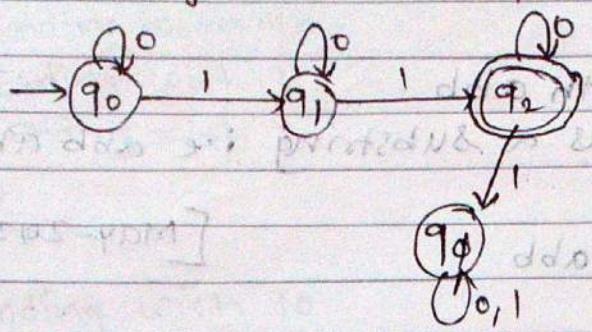
$\Rightarrow \underline{q_5}$

string processing stops at q_5 state which is final state. so ~~the~~ the string gets accepted by finite Automata.

Scanned by CamScanner

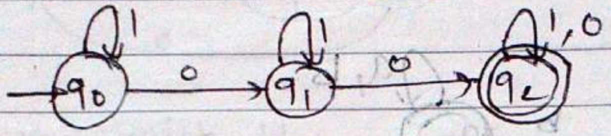
ii. All strings with exactly two 1's

$L = \{11, 011, 110, 101, \ldots\}$



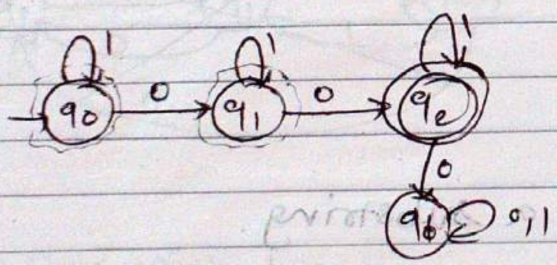
iii. All string containing at least two 0's.

$L = \{00, 100, 010, 001, 0101 \ldots\}$



iv. All strings containing at most two 0's

$L = \{\varepsilon, 1, 0, 01, 10, 11, 00, 101, 111, 001, \cdots\}$



v. All strings starting with 1 and length of the string is divisible by 3

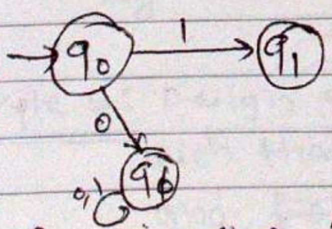


fig. string start with 1

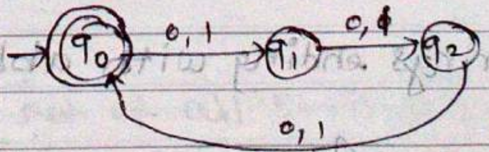
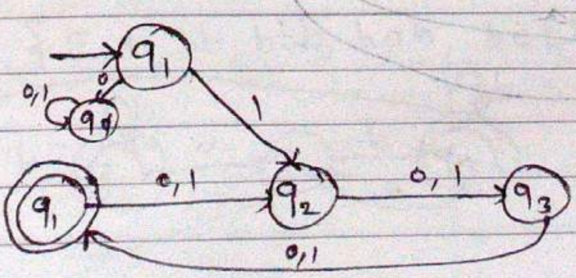


fig. string divisible by 3.



Example 3 : Draw DFA for the following languages over $\{a, b\}$

i. All strings starting with abb

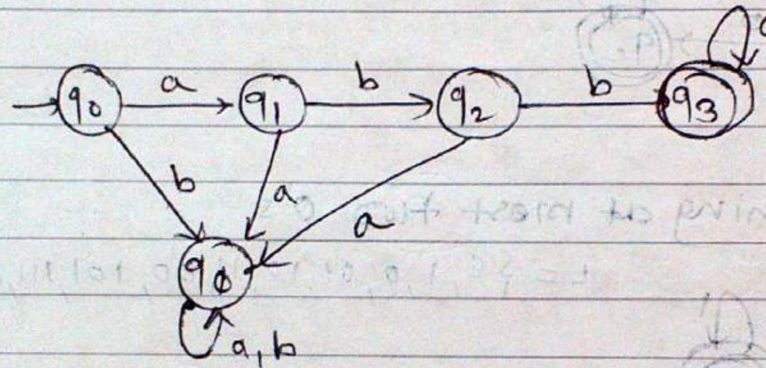
ii. All strings with abb as a substring i.e abb anywhere in the string.

[May-2012]

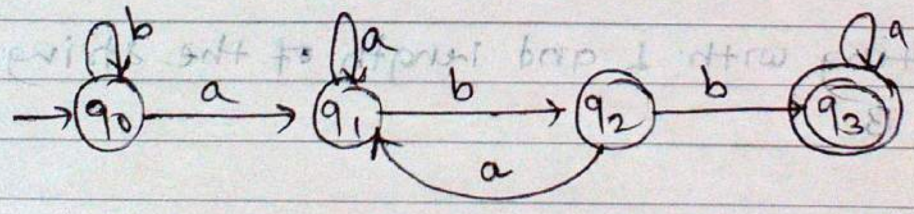
iii. All strings ending in abb

Sol^n :→

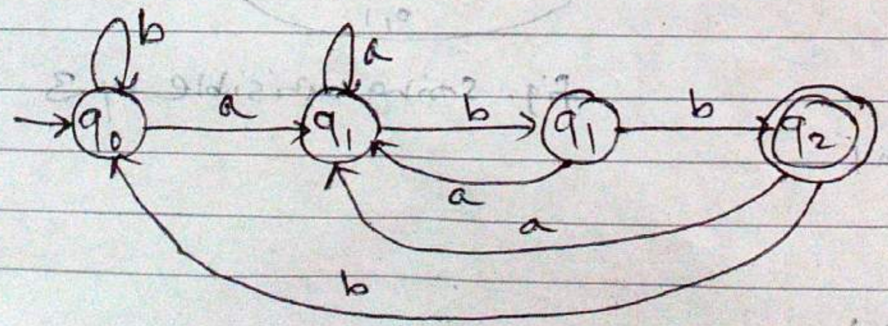
i. All strings starting with abb



ii. All strings with abb as a substring.



iii. All strings ending with abb.

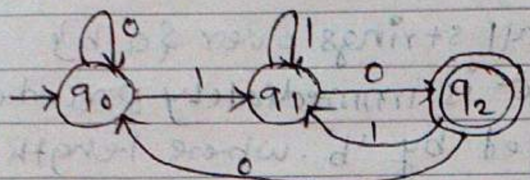


Example 4: Design DFA for a language ~~of~~ string 0 & 1
i. Ending with 10
ii. Ending with 11
iii. Ending with 1

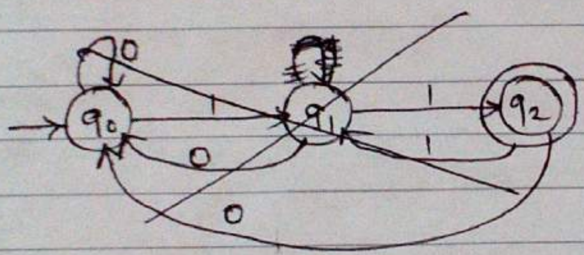
[May-2010]

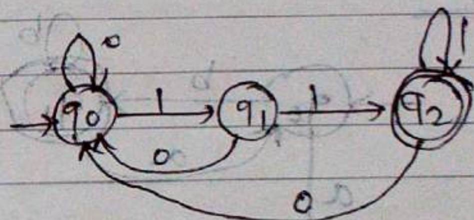
Solⁿ:⇒

i. Ending with 10

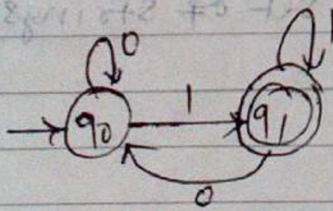


ii. Ending with 11



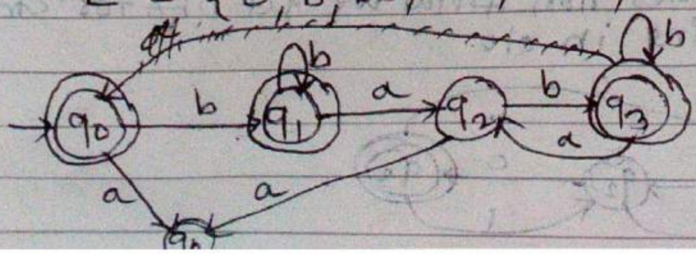


iii. Ending with 1

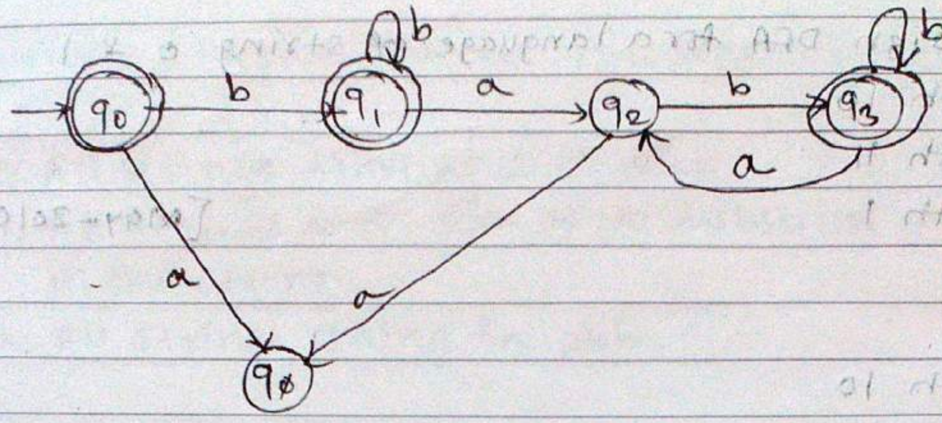


Example 5: Design a DFA for set of all strings over {a,b} such that each 'a' in w is immediately preceded and ~~fo~~ immediately followed by b.

Solⁿ:⇒ $L = \{\varepsilon, b, bb, bbb, bab, babb, bbab, \ldots\}$.



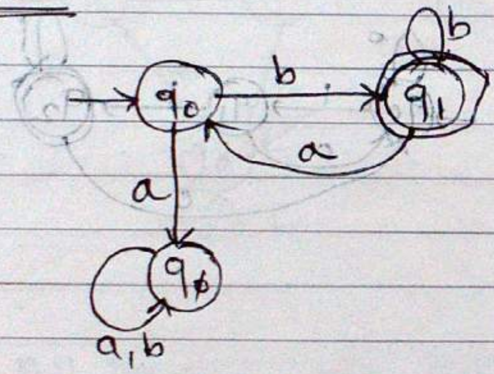
i. bababbbabb
ii. bbababbbbab



Example 6: Design DFA for set of all strings over {a, b} such a that 'a' in w is immediately preceded and immediately followed by 'b'. where length of string should be greater than or equal to 1.

Soln:⇒

L = {b, bb, bab, babab, ...}



Example 7: construct DFA for accepting a set of strings over Σ{0, 1} not ending in 010.

Soln:⇒

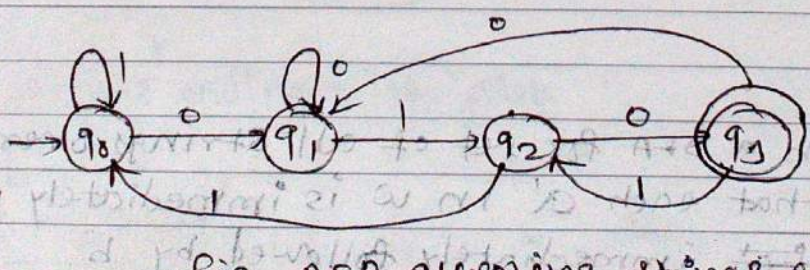


fig. DFA accepting strings ending in 010.

complementing the DFA by making all non final states final & final states non final we get DFA for accepting strings which are not ending in 010.

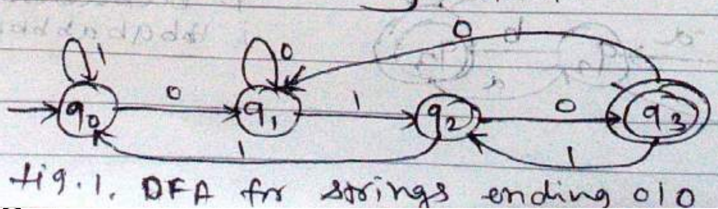


fig. 1. DFA for strings ending 010

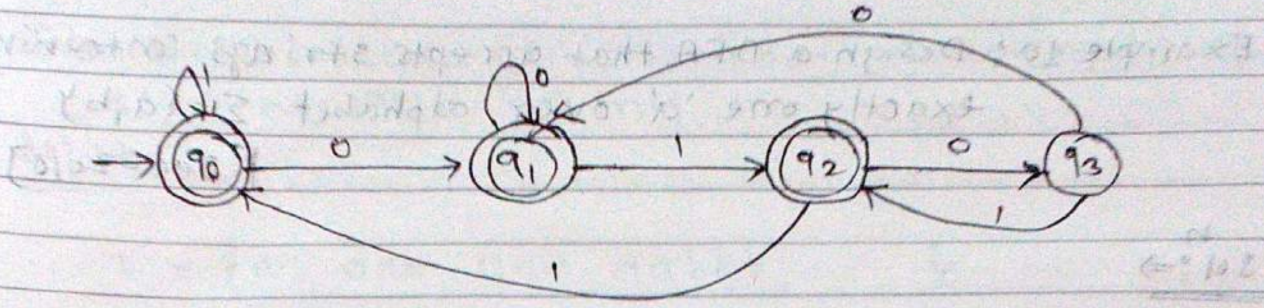
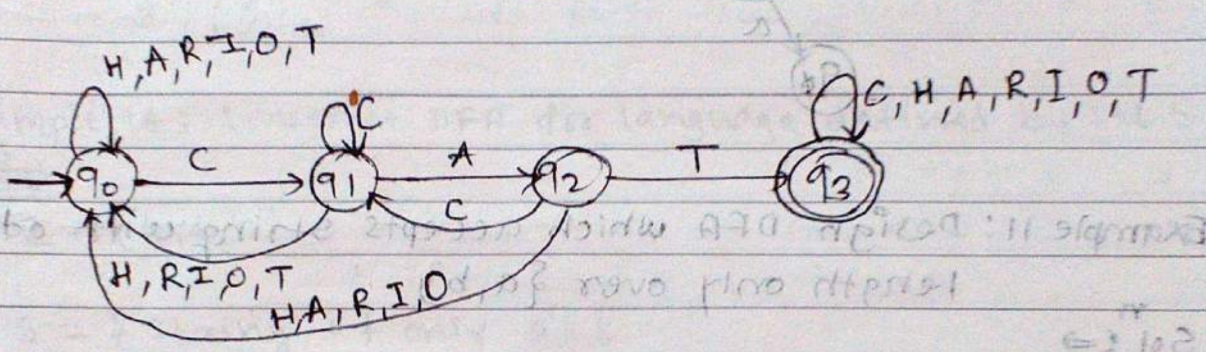


Fig. 2. DFA for strings not ending in 010.

Example 8: Design DFA that reads strings made up of letters in the word 'CHARIOT' and recognizes these strings that contain the word 'CAT' as a substring.

Sol^n:→

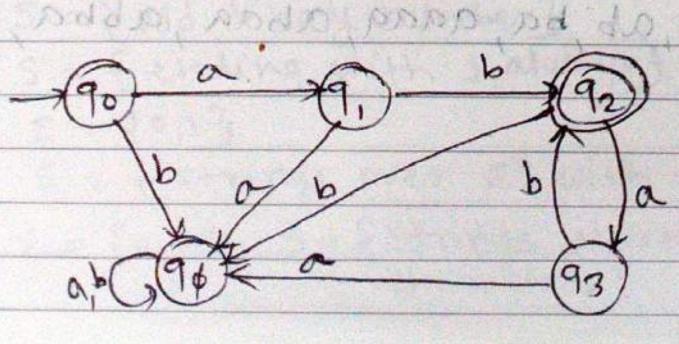


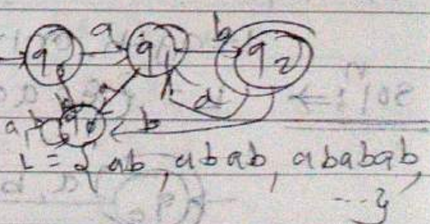
Example 9: Design finite Automata to accept following formal language specification. Justify your design.

$$L = \{ (a.b)^n \mid n >= 1 \}$$ [May - 2011]

Sol^n:⇒

L = { ab, abab, ababab, abababab, ... }



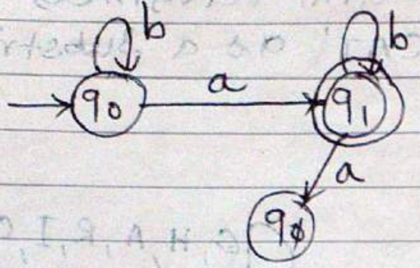


Page No.
Date :

Example 10 : Design a DFA that accepts strings containing exactly one 'a' over alphabet $\Sigma = \{a, b\}$ [Dec-2010]

Sol^n :⇒

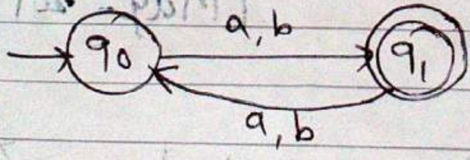
$L = \{a, ab, ba, abb, bab, bba, abbb, babb, bbab, bbba, \ldots\}$



Example 11: Design DFA which accepts string with odd length only over $\{a, b\}$

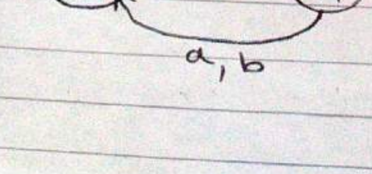
Sol^n :⇒

$L = \{a, b, aaa, aba, baa, aab, bbb, bba, bab, \ldots\}$



Example 12 : Design DFA ~~for~~ which accepts strings of even length only over $\{a, b\}$

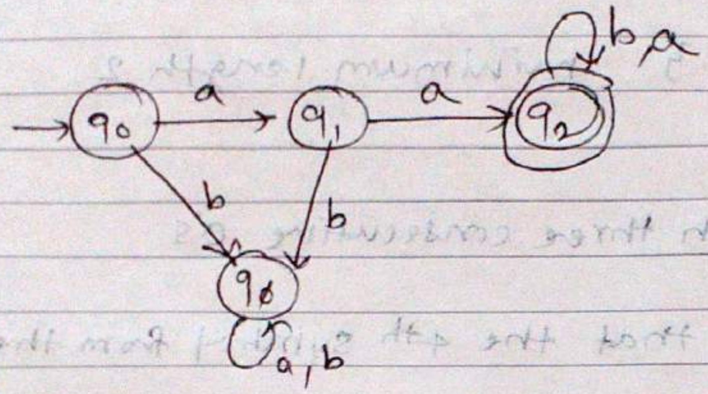
Sol^n :→ $L = \{\epsilon, aa, bb, ab, ba, aaaa, abaa, abba, \ldots\}$



Example 13: Design DFA over {a, b} accepting string starting with 'aa' only.

sol:→

$L = \{aa, aab, aaa, aabbb, \ldots\}$



Example 14: Construct DFA for language defined by set S.

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

a. S = { strings of only a's }
b. S = { string of only b's }
c. S = { strings of only a's or b's.
d. S = { strings with zero or more occurrences of a's followed by zero or more occurrences of b's.
e. S = { ba, baa }
f. S = { strings starting with aa only }
g. S = { strings without substring aa }
h. S = { a is always doubled }
i. S = { set of strings ending with 0 always } & $\Sigma = \{0, 1\}$
j. S = { Even binary number } & $\Sigma = \{0, 1\}$
k. S = { odd binary number } & $\Sigma = \{0, 1\}$
l. S = { ending with 1 always } & $\Sigma = \{0, 1\}$
m. $\Sigma = \{0, 1\}$
S = { strings over Σ^* with total number of 0's even.
n. $\Sigma = \{a, b\}$, S = { strings starting & ending with different characters }

O. S = { starting and ending with same letter} , $\Sigma = \{a, b\}$

P. $\Sigma = \{a, b\}$
S = { strings with starting character a}

Q. $\Sigma = \{a, b\}$
S = { starting & ending with 'a' always.

R. S = { aa, aba, ... } minimum length 2.

S. $\Sigma = \{0, 1\}$
S = { strings with three consecutive 0's.

T. S = { strings such that the 4th symbol from the right is 1 .

U. $\Sigma = \{0, 1\}$
S = { strings such a that each 0 is immediately preceded and followed by 1}

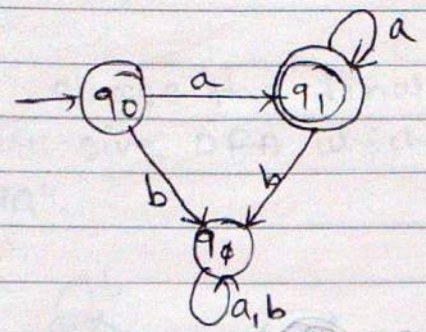
V. $\Sigma = \{0, 1\}$
S = { strings that contain substring 101}

15. Construct DFA for language defined by set 'S' over $\Sigma = \{a, b\}$

$S = \{$ strings of only a's $\}$

Soln:→

$L = \{a, aa, aaa, \ldots\}$

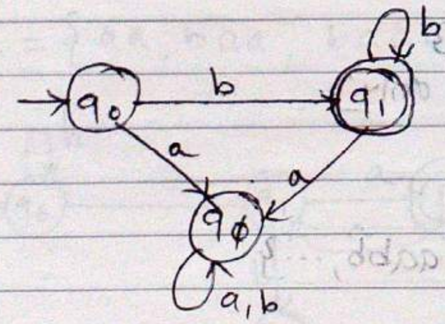


16. Construct DFA for language defined by set 'S' over $\Sigma = \{a, b\}$

$S = \{$ strings of only b's $\}$

Soln:→

$L = \{b, bb, bbb, \ldots\}$

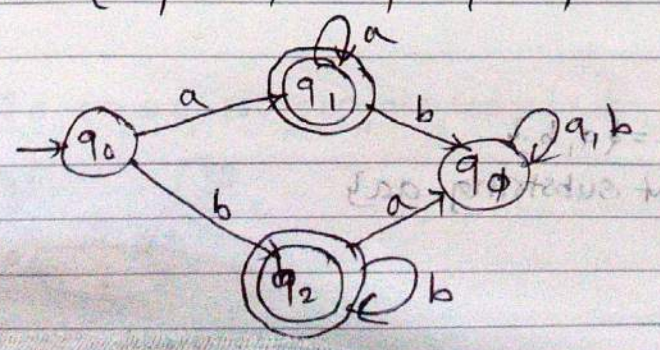


17. Construct DFA for language defined by set 'S' over $\Sigma = \{a, b\}$

$S = \{$ string of only a's or b's $\}$

Soln:→

$L = \{a, aa, aaa, b, bb, bbb, \ldots\}$

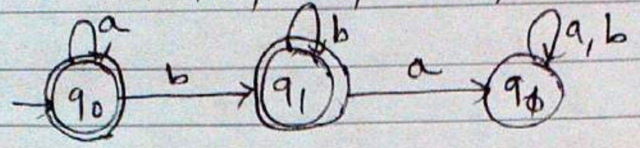


Page No.
Date :

18. construct DFA over $\Sigma = \{a, b\}$
S = { strings with zero or more occurrences of a's followed by zero or more occurrences of b's.

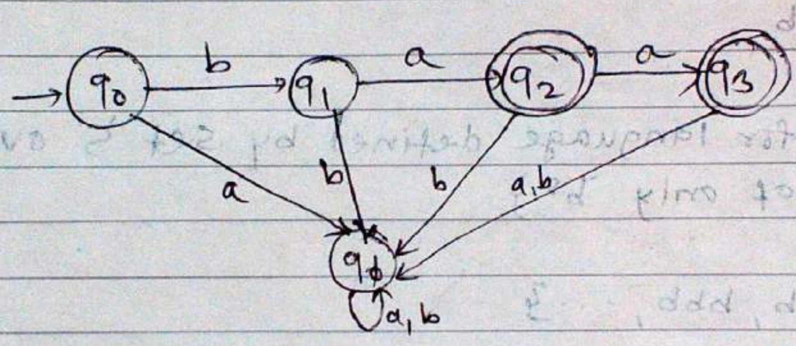
Sol^n :⇒

L = {ε, a, b, ab, abb, aabb, aab, ...}



19. construct DFA over $\Sigma = \{a, b\}$
S = { ba, baa }

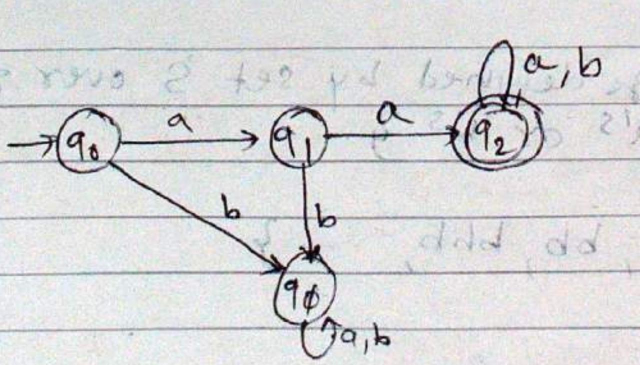
Sol^n :⇒



20. construct DFA over $\Sigma = \{a, b\}$
S = { strings starting with aa only }

Sol^n :⇒

L = { aa, aaa, aab, aaaa, aabb, ... }



21. construct DFA over $\Sigma = \{a, b\}$
S = { strings without substring aa }

Sol^n :⇒

strings with substring 'aa'.

L = {aa, aaab, baaa, aab, ...}

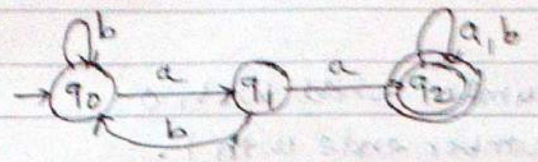
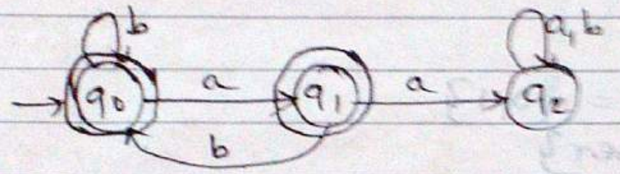


fig. DFA for substring "aa"

change the final & non final state of above DFA, will give DFA which accepts strings not having substring "aa".

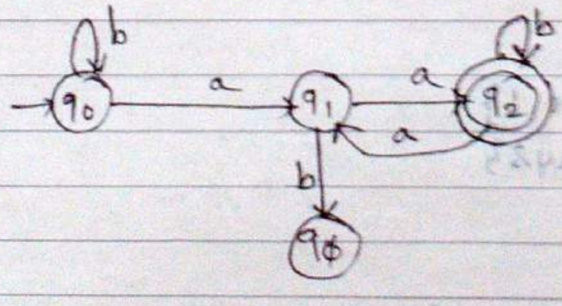


22. construct DFA over $\Sigma = \{a, b\}$

S = { 'a' is always doubled }

sol^n :⇒

L = {aa, baa, baab, baabb, bbaa, ...}

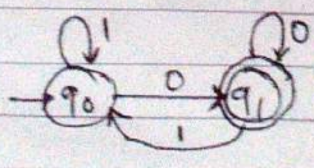


23. construct DFA over $\Sigma = \{0, 1\}$.

S = { set of strings ending with '0' always }

sol^n :⇒

L = {0, 10, 00, 100, 000, 010, 110, ...}

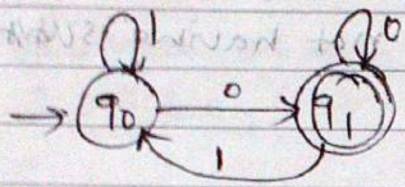


24. construct DFA over $\Sigma = \{0,1\}$
$S = \{$even binary number$\}$

sol^n:⇒

Even binary number: A number ends with 0
odd binary number: A number ends with 1.

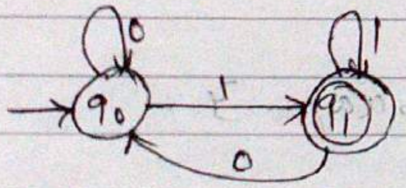
$L = \{0, 10, 010, 110, 0110, \ldots\}$



25. construct DFA over $\Sigma = \{0,1\}$
$S = \{$odd binary number$\}$

sol^n:⇒

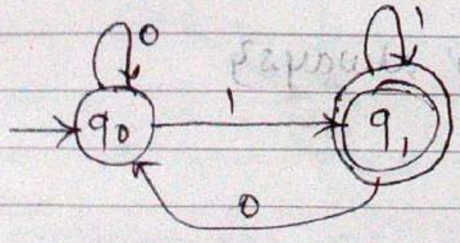
$L = \{1, 01, 11, 001, 111, 101, \ldots\}$



26. construct DFA over $\Sigma = \{0,1\}$
$S = \{$ending with 1 always$\}$

sol^n:⇒

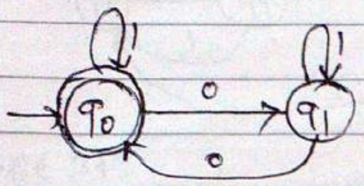
$L = \{1, 01, 11, 001, 111, 101, \ldots\}$



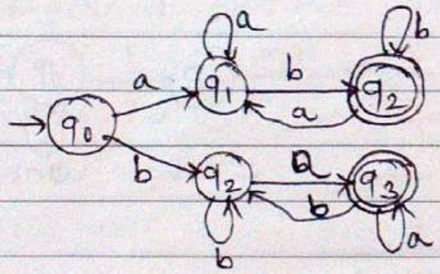
27. Construct DFA over $\Sigma = \{0,1\}$
$S = \{$ strings over Σ^* with total numbers of 0's even $\}$

Sol^n:⇒

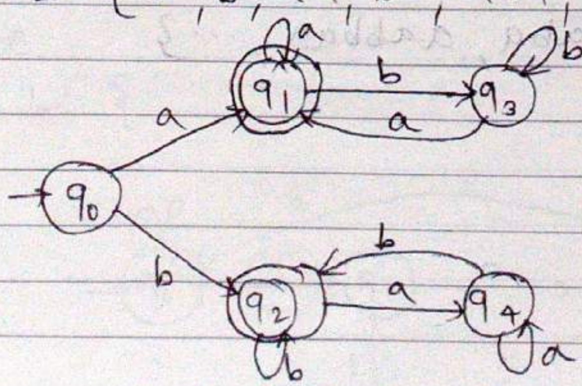
$L = \{\varepsilon, 00, 001, 10D, 010, \ldots\}$



28. Construct DFA over $\Sigma = \{a,b\}$
$S = \{$ strings starting & ending with different characters $\}$
Sol^n:⇒
$L = \{ab, ba, aab, bba, abb, \ldots\}$



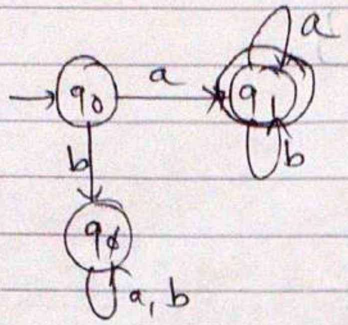
29. Construct DFA over $\Sigma = \{a,b\}$
$S = \{$ strings starting and ending with ~~different~~ Same ~~character~~
characters $\}$
Sol^n:⇒
$L = \{a, b, aa, bb, aba, bab, aaa, bbb, \ldots\}$



Example 30: $\Sigma = \{a, b\}$
$S = \{$strings with starting character $a\}$

Sol^n :→

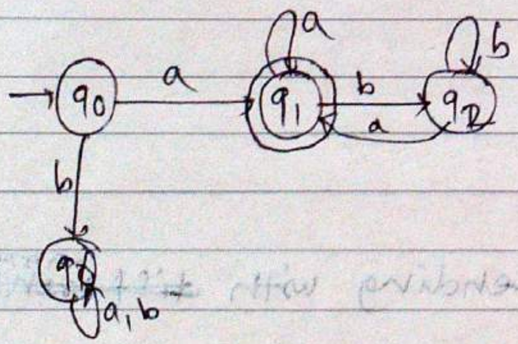
$L = \{a, ab, aa, aba, aaa, aab, \dots\}$



Example 31: $\Sigma = \{a, b\}$
$S = \{$starting and ending with a always$\}$

Sol^n :→

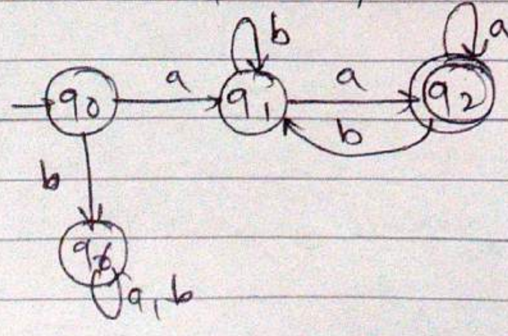
$L = \{a, aaa, aba, aa, \dots\}$



Example 32: $S = \{aa, aba, \dots\}$ minimum length 2 !

Sol^n :→

$L = \{aa, aba, abba, aaba, aabbaa, \dots\}$

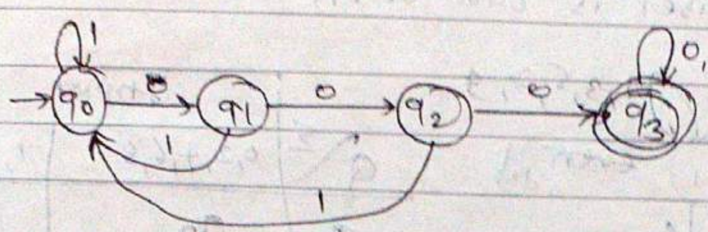


Example 33 : $\Sigma = \{0,1\}$

$S = \{$ strings with three consecutive 0's $\}$ [May-2008 8 M.

Soln :→

$L = \{000, 10001, 10000, 00010, \ldots\}$

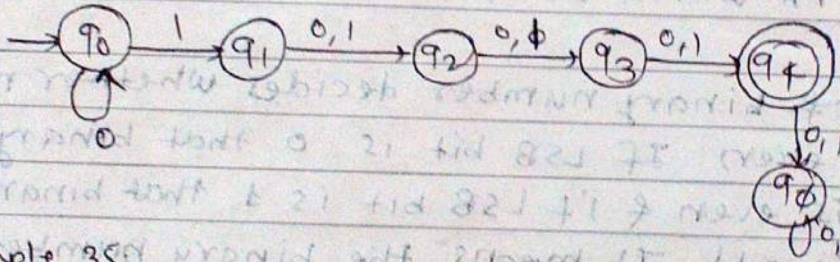


Example 34 :

$S = \{$ strings such that 4th symbol from the right is 1 $\}$

Soln :→

$L = \{01000, 1111, 1010, \ldots\}$



Example 35 :

$\Sigma = \{0,1\}$

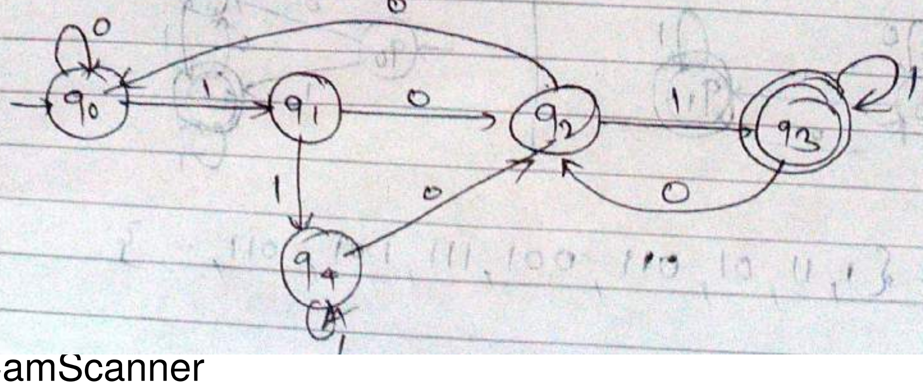
$S = \{$ string such that each 0 is immediately preceded & followed by 1 $\}$

i.e string that contains substring 101

[Dec-2008, 8 marks]

Soln :→

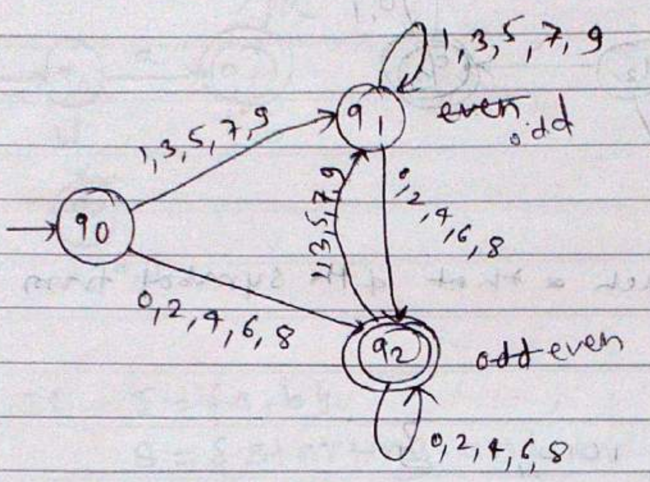
$L = \{$



Example 36: Design FA which accepts a decimal ~~odd~~ even number

Sol^n :⇒

Logic : Decimal Number is even when it ends with 0,2,4,6,8 & decimal number is odd when it ends with 1,3,5,7,9



Q \ Σ	0,2,4,6,8	1,3,5,7,9
→q_0	q_2	q_1
q_1	q_2	q_1
q_2*	q_2	q_1

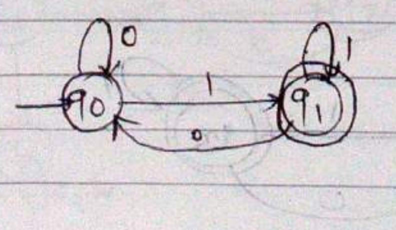
Example 37: Design FA which accepts an odd binary number.

Sol^n :→

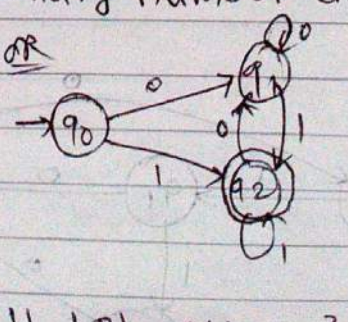
Logic: LSB bit of binary number decides whether number is odd or even. If LSB bit is 0 that binary number is even & if LSB bit is 1 that binary number is odd. It means, the binary number which ends with '0' is even & ~~that~~ ends with 1 is odd.

Example: 00 (even, ends with 0), 01 (odd, ends with 1), 001 (odd), 101 (odd), 111 (odd), 100 (even) and so on.

Automata that accepts binary number ends with '1' is odd binary number.



OR



$L = \{1, 11, 01, 011, 001, 111, 101, 011, \dots\}$

Example 38: Design FA to test divisibility by 3 for a unary number.

Sol^n:→

Unary number: In a unary number there is only single bit used to represent the number. Assume single bit or digit used is 1.

Decimal number	unary equivalent number
1	1
2	11
3	111
4	1111
5	11111

L = ~~{1, 11, 111, 1111,~~ ...

L = {1, 111, 111111, ...}

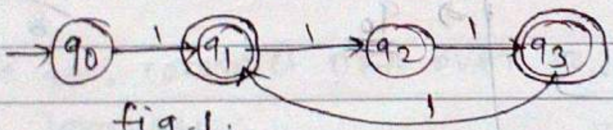


fig 1:

L = { ε, 1, 111, 111111, ...}

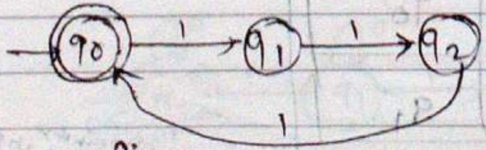


fig 2:

Example 39: Design FA to test divisibility by 3 tester for a given decimal number.

Sol^n:→

For divisibility by 3, the decimal input numbers can be divided into 3 groups as:

0, 3, 6, 9 ⇒ Remainder 0 after dividing by 3.

1, 4, 7 ⇒ Remainder 1

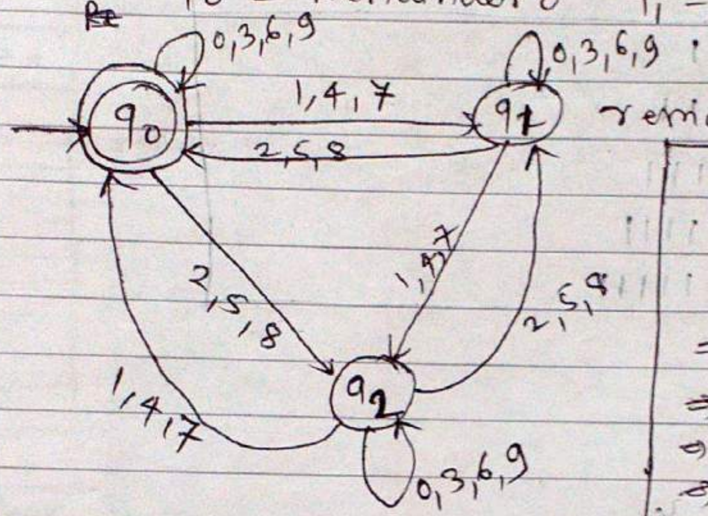
2, 5, 8 ⇒ Remainder 2

decimal number

for divisibility by 3 possible remainders are 0, 1, 2.

Here, we have to form number as follows
previous remainder 1 & current input is 3 then the number formed is 13 which on division by 3 gives remainder 1.

q_0 = Remainder 0 q_1 = Remainder 1 q_2 = Remainder 2.



Here, trace 123, w = 123

$\delta(q_0, 123) \Rightarrow \delta(\delta(q_0,1),23)$ (current i/p)

Here, number formed 01

01 % 3 = 1 next state is q_1.

$\Rightarrow \delta(q_1, 23) \Rightarrow \delta(\delta(q_1,2),3)$

$\Rightarrow$ current remainder 1 & i/p 2

$\Rightarrow$ 12 % 3 = 0 next state 0

$\Rightarrow \delta(q_0, 3)$

$\Rightarrow$ 03 % 3 = 0 next state = q_0

$\Rightarrow \underline{q_0}$

L = {ε, 0, 3, 6, 9, 12, ...}

Q \ Σ	(0,3,6,9)	(1,4,7)	(2,5,8)
→ q_0*	q_0	q_1	q_2
q_1	q_1	q_2	q_0
q_2	q_2	q_0	q_1

q_0 = Remainder 0 ∴ New number can be formed as 01, 02, 03, 04, 05, 06, 07, 08, 09 (previous remainder, next input)

q_1 = Remainder 1 ∴ New number = 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,

q_2 = Remainder 2 ∴ New number = 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,

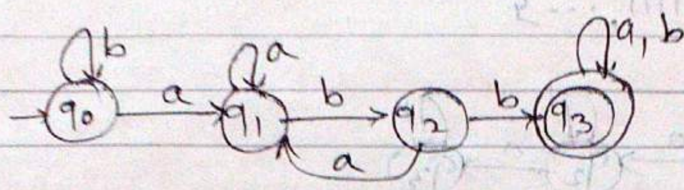
$\delta(q_0, 123) \Rightarrow \delta(\delta(q_0,1),23) \Rightarrow \delta(q_1, 23) \Rightarrow \delta(\delta(q_1,2),3)$

$\Rightarrow \delta(q_0,3) \Rightarrow \underline{q_0}$

Example 40 : Design DFA over $\Sigma = \{a,b\}$ to accept all strings containing abb as a substring.

Soln :⇒

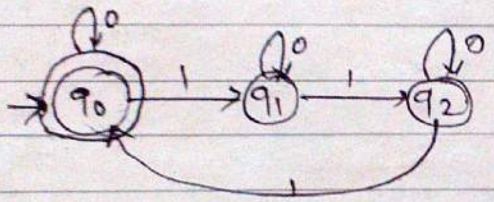
$L = \{abb, aabb, abbb, babb, \ldots\}$



Example 41 : Give DFA accepting following language over the alphabet {0,1} where $L = \{$ Number of 1's multiple of 3 $\}$

Soln :⇒

$L = \{\varepsilon, 111, 111111, \ldots\}$

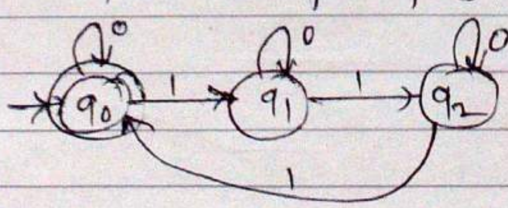


Example 42 : construct DFA over {0,1} which accept following language,

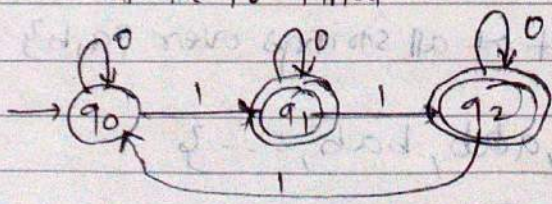
$L = \{$ Number of 1's not multiple of 3 $\}$:

Soln :⇒

step 1 : construct DFA for accepting strings containing number of 1's multiple of 3.



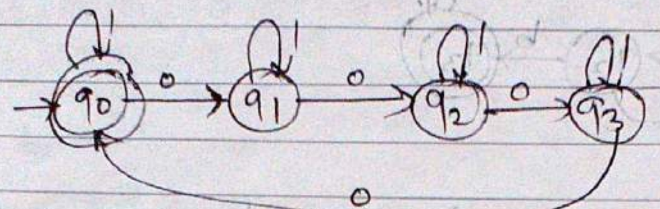
step 2 : change final state to non final & non final state to final.



Example 43: Design DFA over {0,1} where number of zero's are multiple of 4.

Sol^n: ⇒

L = {ε, 1111, 11111111, ...}

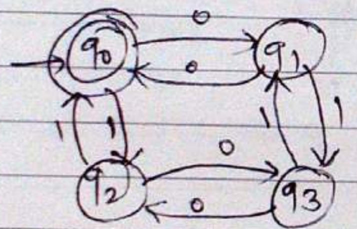


Example 44: Design DFA for language over {0,1}.

L = {Number of 0's and 1's are even}

Sol^n: ⇒

L = {ε, 0011, 1100, 1010, 0101, ...}

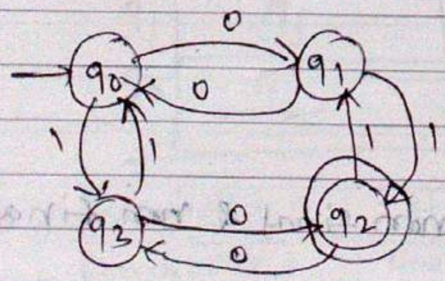


Example 45: Design DFA for language over {0,1}

S = {Number of 0's and 1's are odd}

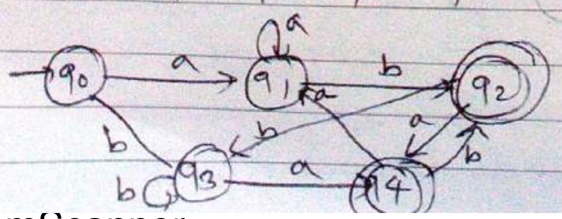
Sol^n: ⇒

L = {01, 000111, 10, 010101, ...}



Example 46: Design DFA for set of all strings over {a,b} ending in either ab or ba.

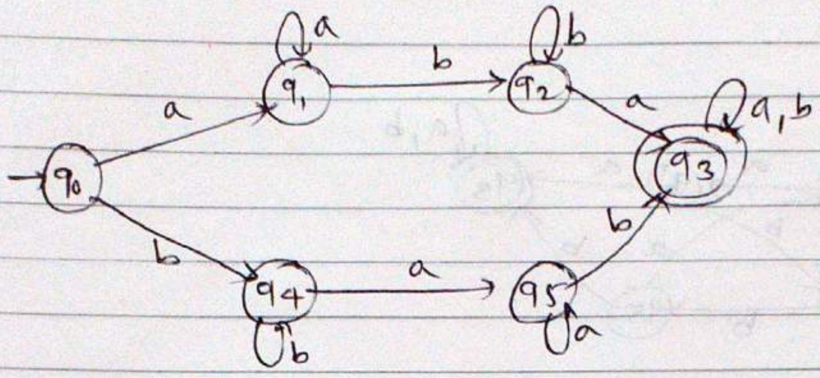
Sol^n: ⇒ L = {ab, ba, aba, bba, aab, bab, ...}



Example 47: Design DFA for set of all strings over {a,b} containing both ab and ba as substrings.

Sol^n: ⇒

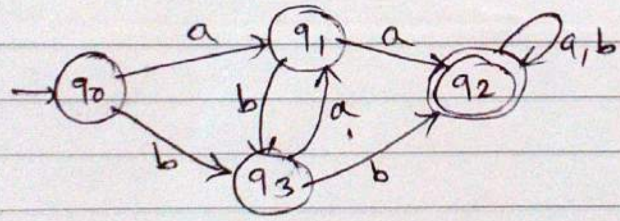
L = {aba, bab, aaba, abba, baba, ...}



Example 48: DFA over a,b containing either aa or bb as a substring

Sol^n: ⇒

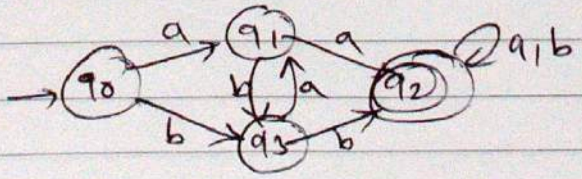
L = {aa, bb, aaa, bbb, abba, baab, ...}



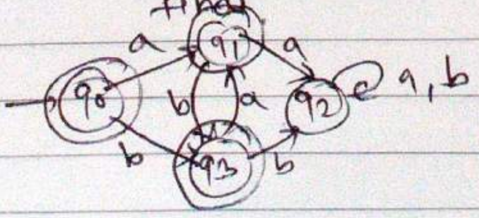
Example 49: DFA over {a,b} containing neither aa nor bb as a substring.

Sol^n: ⇒

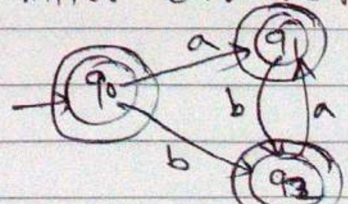
step 1: construct aa or bb as a substring.



step 2: Change all final states to non final & other states to final (non final)



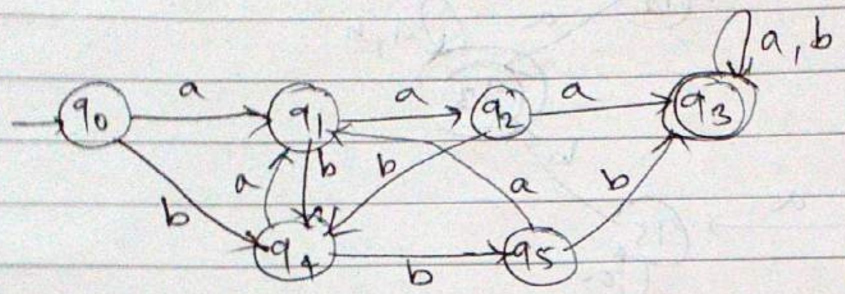
After eliminating dead state.



Example 50: construct a DFA for set of strings containing either the substring "aaa" or "bbb"

Sol^n:→

L = { aaa, bbb, abbb, bbbb, aaaa, baaa, ... }



Scanned by CamScanner