Formal Languages & Automata Theory

Chapter 1:

Introduction to finite Automata

RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

Objectives of FLAT:

① To familiarize students with the foundations & principles of computer science.

② To teach material which is useful in subsequent courses.

③ To strengthen students ability to carry out formal and regorous mathematical arguments.

This subject is also called as Theory of computation. In this subject, we will study many theoritical machines which are mathematical models for actual physical processes.

Rajeev. Bilagi
M.Tech
Dept of CSE

1.1 Formal Languages:

→ It is an abstraction of general characterstics of programming languages. It is a set of symbols & simple rules of formulation by which the entities called sentences are obtained.

Hence formal language is set of all sentences permitted by the rules of formulation.

Scanned with CamScanner

→ The word Formal refers to the fact that all the rules for the language are explicitly stated in terms of what strings of symbols can occur.

[Formal Language is an abstraction of the characterstics of programming languages].

Bajeev Bilagi
M.Tech
Dept of CSE

1.2 Automata Theory:

To model the b/w of computer, we introduce the notion of automation.

→ Automation is a construct that possess all the indispensible features of a digital computer.

→ It accepts input, produces o/p, may have temporary storage & it can make decisions in transforming the input into output.

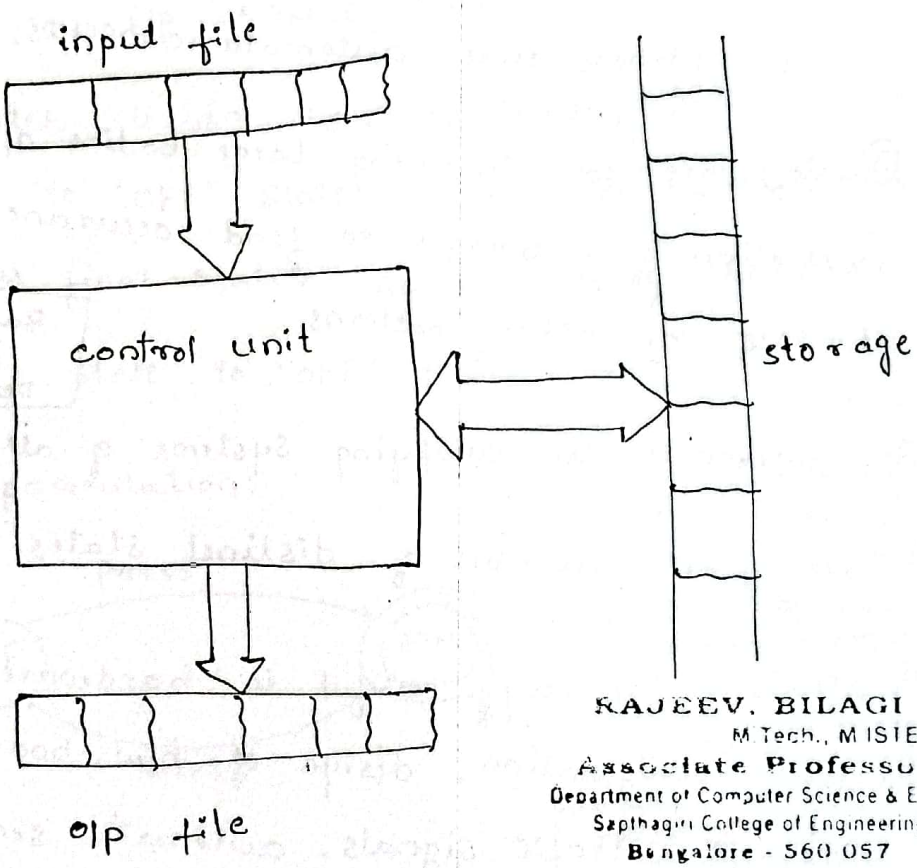
→ It is an abstract model of a digital computer. It has a mechanisam for reading input, written on an input file. I/p file is divided into cells, each of which can hold 1 symbol, it scans from left to right.

→ It can produce o/p of some form. It may have a temporary storage device, consisting of unlimited no of cells, each capable of holding a single symbol from alphabet.

Scanned with CamScanner

→ The automaton has a control unit, which can be any one of the finite number of states. The state of control unit at the next time step is determined by a next state or transition function.

→ General structure of a automaton is shown in figure 1.



RAJEEV. BILAGI
M.Tech., M.ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

Figure 1: General structure of Automaton

Rajeev Bilagi
M.Tech
Dept of CSE

Scanned with CamScanner

1.3 Applications of finite Automata:

① To develop softwares for designing and checking the behavior of digital circuits.

② To develop softwares for different applications

Ex: Compilers.

In compiler design, lexical analyzer uses finite automata to break the i/p text into different tokens. Ully other phases uses automata concepts.

③ Software for scanning Large bodies of text such as collection of web pages to find occurances of words, phrases or other patterns.

Rajeev Bilagi
M.Tech,
Dept of CSE

④ Softwares for varifying systems of all types that have finite number of distinct states.

⑤ This is a very useful in hardware design such as circuit varification, design of h/w board (mother Board), automatic trafic signals, automatic sensors.

⑥ Finite automation is very usful in recognizing difficult problems. i.e some times it is very essential to solve an un-decidable problem. Even though there is no general solution exists for the specified problem, using theory of computation, we can find approximate solutions.

Scanned with CamScanner

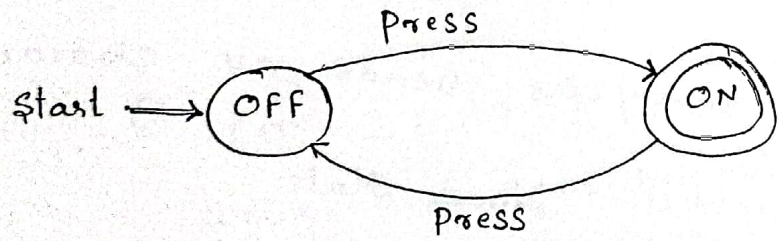
Finite Automata:

This is a type of automata which is having finite number of states & There is no memory for it.

Ex: Simplest finite automata is an ON|OFF Switch. This device remembers only 'ON' state and "OFF" state, on i/p "PRESS" switch changes its state.

If switch is in "ON" condition, pressing the button changes it to "OFF" state, if that is the desired state, it becomes final state. Again pressing the same button changes its state to "OFF" state.

Model Representation:



Rajeev Bilagi
M.Tech
Dept of CSE

RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

Notations used to build finite automation model are:

1) States are represented by circles.
(2) Edges will represent connections betn nodes on which i/p will be passed.
(3) Start state is indicated by an arrow labelled "Start" to that state

Scanned with CamScanner

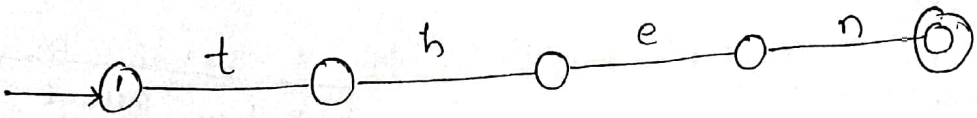
④ Final state are indicated by double circle.

~~The central concepts~~

Ex:

Finite Automation that could be part of lexical analyzer.

→ If we want to recognize keyword "then". It needs 5 states, each state represents a different position in word "then"



→ when lexical analyzers reads all characters of then, it reaches final state.

Bajeev . Bilagi
M.Tech
Dept of CSE

Scanned with CamScanner

The Central Concepts of Automata Theory :

① Alphabet :

An alphabet is a finite, non empty set of symbols. We use 'Σ' for an alphabet.

Eg: $\Sigma = \{0, 1\}$ Binary alphabet.

$\Sigma = \{a, b \ldots\ldots z\}$ set of lower case letters.

Bajeev Bilagi
M.Tech
Dept of CSE

② Strings :

It is finite sequence of symbols choosen from some alphabet.

Ex: 001101 is a string from alphabet $\Sigma = \{0, 1\}$

Let $\Sigma = \{0, 1\}$ is set of alphabets. The various strings that can be obtained from Σ are

$\{0, 1, 00, 01, 0110, 01110 \ldots\ldots\}$

RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

③ Empty string :

It is a string with '0' occurances of symbols. It is denoted by 'ϵ' which is a string that may choosen from any alphabet.

Scanned with CamScanner

④ Length of string :

The no of positions for symbols in the strings.

Eg: 01101 has length 5.

Standard Notation for string w is $|w|$

Eg: $|011| = 3$ & $|\epsilon| = 0$

⑤ The lower case letters such as u, v, w, x, ... are used to indicate strings

Ex: w = 010011.

⑥ Powers of an alphabet :

→ If 'Σ' is an alphabet. Set of all strings of a certain length from given alphabet is called as powers of alphabet.

→ We define Σ^k to be set of strings of length k.

Eg: $\Sigma^0 = \{\epsilon\}$ is only one string whose length is 0.

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

then

Bajeev Bilagi
M.Tech
Dept of CSE

Scanned with CamScanner

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

$\Sigma^2 = \{00, 01, 10, 11\}$

$\Sigma^3 = \{000, 001, 010, 011, 100, 110, 101, 111\}$

$\vdots$

Then

Σ^* is set of words of any length. Each string is made up of symbols only from Σ.

$\therefore \Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots\ldots \cup \Sigma^n$

Where Σ^* is called <u>star clousure</u>.

Bajeev Bilagi
M.Tech
Dept of CSE

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

Then $\Sigma^* = \{\epsilon, 0, 1, 00, 01, 000 \ldots\ldots\}$

<u>positive closure</u> Σ^+:

$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \ldots\ldots$ which is set of all words of any length except the null string <u>i.e</u> ϵ.

RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

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

$\Sigma^+ = \{0, 1, 00, 01 \ldots\ldots\}$ is a set of strings of 0's & 1's of any length except the null string.

Scanned with CamScanner

(7) Concatenation of strings:

Let x & y be two strings.

Then xy denotes concatenation of x and y.

i.e the string formed by making a copy of x & following it by a copy of y.

Eg: $u = 01101$

$v = 110$

Then $xy = 01101110$

Bajeev Bilagi
M.Tech
Dept of CSE

Now $\epsilon w = w \epsilon = w$

'ϵ' is the identity for concatenation.

(8) Languages:

→ A Language can be defined as a set of strings from the alphabet.

→ Generally, a language can be defined as a subset of Σ^*. The set may be empty, finite or infinite.

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

$L = \{\epsilon, 01, 0011, 000111\} ---\}$

is a language of all strings of 'n' number of '0' is followed by 'n' number of 1's.

Scanned with CamScanner

Deterministic Finite Automata: (DFA)

There are 3 types of finite automata.

① Deterministic Finite Automata. (DFA)

② Non-Deterministic finite Automata (NFA)

③ NFA with ε-transitions (ε-NFA)

The word deterministic refers to the fact that on each i/p, there is exactly one and one transition an automation can make, from its current state.

Formally, DFA is defined as 5-tuple or Quintuple automata

$$M = \{ Q, \Sigma, \delta, q_0, F \}$$

Rajeev Bilagi
M.Tech
Dept of CSE

Where

i. Q is a finite set of states.

ii. Σ is a finite set of input symbols.

iii. δ is a transition function which is a mapping from $Q \times \Sigma$ to Q.

RAJEEV. BILAGI
M.Tech, MISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

iv. q_0 is initial state & $q_0 \in Q$.

v. F is finial state or states & $A \subseteq Q$
or
(Accepting states)

Scanned with CamScanner

Notations for DFA:

For describing DFA, two types of notations are used.

i) Transition Diagram.

ii) Transition table.

Transition Diagram:

→ It is a graph of a DFA

$D = \{Q, \Sigma, \delta, q_0, F\}$ which is defined as follows.

i) For each state in Q, There is node.

Nodes are represented with circles.

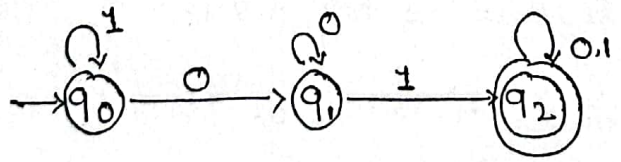
ii) For each state 'q' in Q, & each i/p symbol a in Σ

Let $\delta(q, a) = p$. Then transition diagram has an arc from node q to node p, labelled a.

iii) There should be an arrow into start state q_0, & should be labelled as Start state.

iv) Nodes corresponding to accept or final states (F) are marked by double circle.

Eg:



∵ Transition dig for DFA accepting 01 as substring.

q_0 → initial state

q_2 → final state , q_1 → intermidiate states.

Scanned with CamScanner

Transition tables:

→ A transition table is tabular representation of a function like δ that makes two arguments & returns a value.

→ Rows of table corresponds to states & columns corresponds to inputs or input symbols.

→ The entry for row corresponds to state q & column corresponding to input "a" is state

$\delta(q, a)$

For above DFA, Transition table will be

δ	0	1
→ q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_2	q_2

Bajeev Bilagi
M.Tech
Dept of CSE

RAJEEV. BILAGI
M.Tech., MISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

General stratargy to solve DFA problems:

1. List all the valid strings in hierarchy order. (L).
2. Take first or minimum valid string expect ϵ.
3. path this into path to reach final state.
4. Remaining transitions have to be taken care.

Scanned with CamScanner

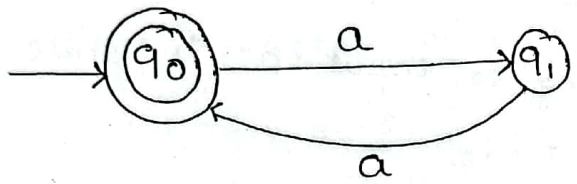
Examples: ⑥

① obtain DFA which accepts even no of a's on $\Sigma = \{a\}$.

Soln:

$L = \{\epsilon, aa, aaaa \ldots\ldots\}$

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



Transition Digram. for L.

Now

- $Q = \{q_0, q_1\}$
- $\Sigma = \{a\}$
- Transition f^n or transition table

δ	a
$\rightarrow q_0$	q_1
q_1	q_0

- q_0 is final state $\quad q_0 = \{q_0\}$
- q_0 is initial state $\quad F = \{q_0\}$.

Scanned with CamScanner

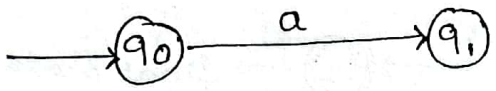
Ex ② Obtain a DFA which accepts odd no of a's over $\Sigma = \{a\}$

Soln:

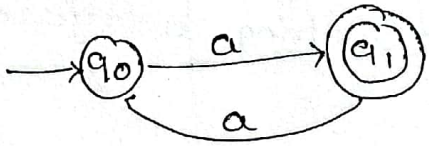
$L = \{a, aaa, \ldots\}$

$\bar{L} = \{\epsilon, aa, \ldots\}$

Minimum 'a' should reach to final state & hence to pass 'a' we need only two states.



So, finaly



DFA for L

Bajeev Bilagi
M.Tech
Dept of CSE

So, DFA will be

$Q = \{q_0, q_1\}$

$\Sigma = \{a\}$

δ is given by

δ	a
$\rightarrow q_0$	q_1
q_1	q_0

RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

q_0 is initial state $q_0 = \{q_0\}$

q_1 is final state $F = \{q_1\}$.

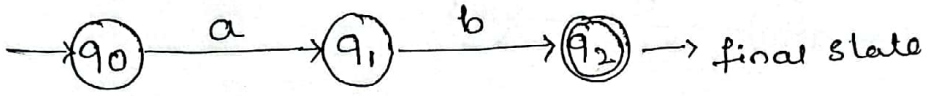
Scanned with CamScanner

Example 3:

Design DFA to accept strings of a's & b's start with string "ab".

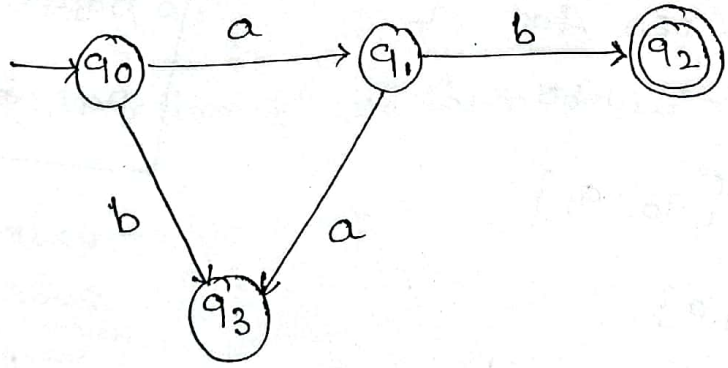
Soln: $L = \{ ab, abab, \ldots\ldots \}$

From problem, it is clear that 'ab' is the minimum string that can be accepted by machine. To accept string "ab", we need three states & can be written as.

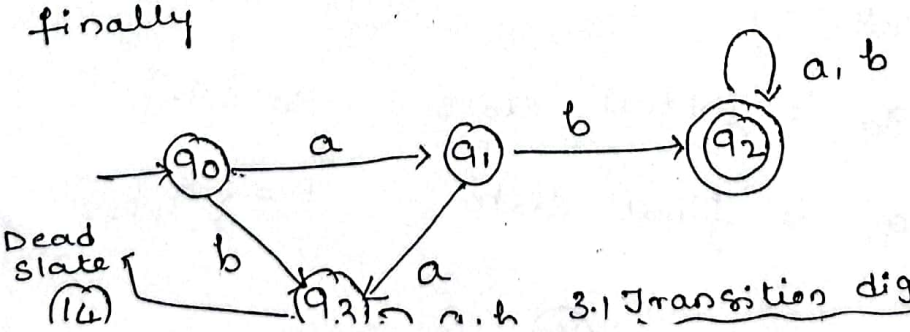


Now

if it starts with 'b', string is rejected.



Now, for each state, we need to take symbols of a & b & also DFA cond^n should be satisfied finally



3.1 Transition dig: $ab(a+b)^*$

Scanned with CamScanner

DFA can be given as

$$M = \{ Q, \Sigma, \delta, q_0, F \}$$

where

$$Q = \{ q_0, q_1, q_2, q_3 \}$$

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

$q_0 = \{ q_0 \}$ – initial state

$F = \{ q_2 \}$ – final state

δ is given as follows.

Σ

δ	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_3	q_2
q_2	q_2	q_2
q_3	q_3	q_3

RAJEEV. BILAGI
M.Tech., MISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

The state q_3 is called trap state or dead state.

Bajeev Bilagi
M.Tech
Dept of CSE

Scanned with CamScanner

Example 4:

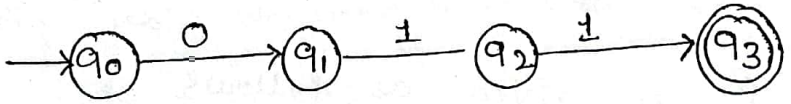
Design a DFA to accept string of 0's & 1's ending with string 011

Soln:

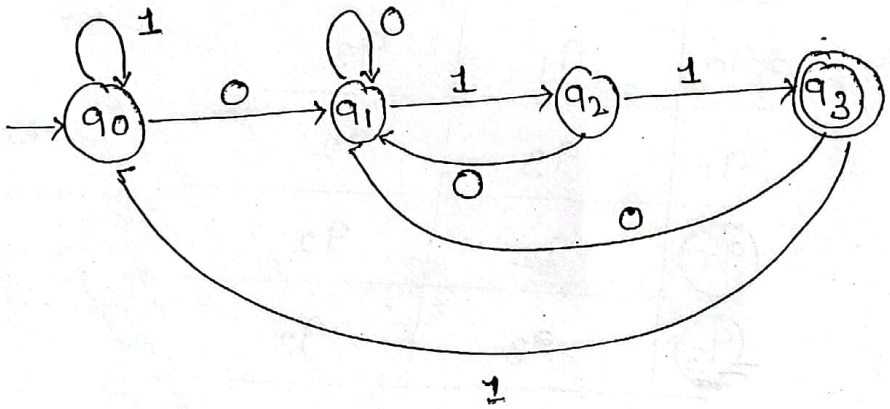
$L = \{011, 0011, 1011, \ldots\}$

$L = \{(0+1)^* 011\}$

The minimum string accepted is 011. It requires four states with q_0 is start & q_3 is final state.



Now final DFA can be written as



Transition Dig for accepting $(0+1)^* 011$

The DFA can be given as

$M = \{Q, \Sigma, \delta, q_0, A\}$

Where

$Q = \{q_0, q_1, q_2, q_3\}$

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

q_0 is initial state

$A = \{q_3\}$

δ is given as follows.

δ (states ↑)	0 (← Σ →)	1
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_1	q_0

Scanned with CamScanner

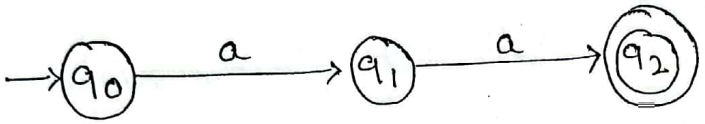
Example 5:

Obtain DFA to accept strings of a's & b's having a substring aa

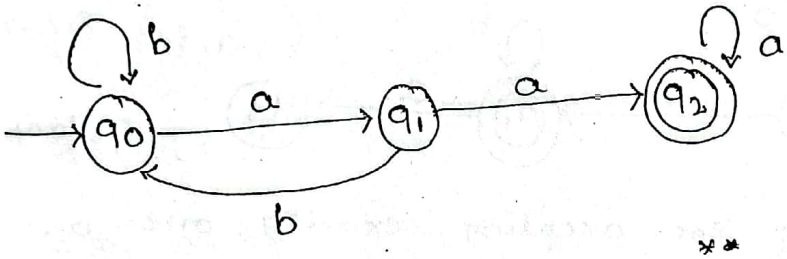
Soln: $L = \{aa, baa \ldots\}$.

$L = \{(a+b)^* aa (a+b)^*\}$

Minimum string is 'aa' & it requires 3 states



Now final DFA can be written as



RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Eng
Sapthagiri College of Engineering
Bangalore - 560 057

Transition dig for $(a+b)^* aa (a+b)^*$.

Now DFA $M = \{Q, \Sigma, \delta, q_0, F\}$ can be given as

$Q = \{q_0, q_1, q_2\}$

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

q_0 is initial state

$F = \{q_2\}$ is final state

Bajeev Bilagi
M.Tech
Dept of CSE

δ can be given as

δ	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_2

Scanned with CamScanner

Example-6

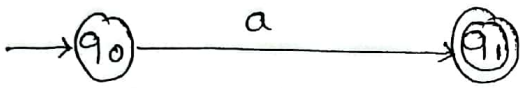
Design a DFA to accept strings of a's & b's having (i) exactly one a, (ii) at least one a, (iii) not more than three a.

Soln:

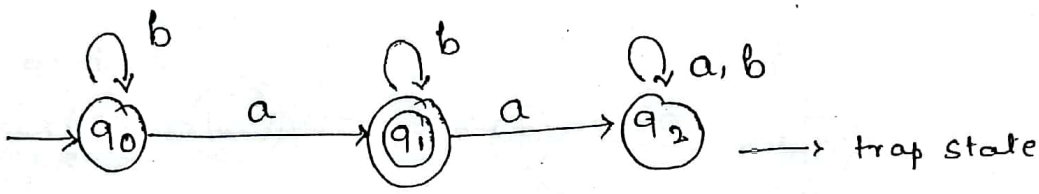
i) To accept exactly one a.

$L = \{a, ab, ba \ldots\}$ or $L = \{b^m a b^n \mid m, n \geq 0\}$

To accept exactly one 'a', we need two states q_0 & q_1.



Now



DFA for accepting exactly one a.

DFA can be given as

$M = \{Q, \Sigma, \delta, q_0, F\}$

$Q = \{q_0, q_1, q_2\}$

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

q_0 is initial state

$F = \{q_1\}$ is final state

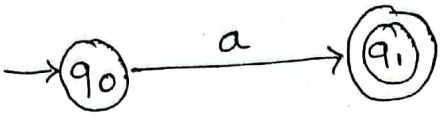
δ can be given as

δ	a	b
$\rightarrow q_0$	q_1	q_0
$\leftarrow q_1$	q_2	q_1
q_2	q_2	q_2

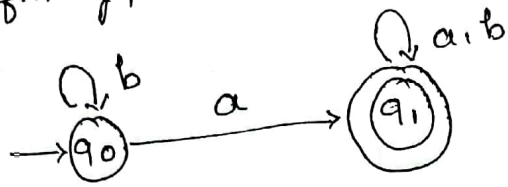
Scanned with CamScanner

ii) To machine to accept at least a.

Minimum string is a, which requires 2 states



finally, DFA can be given as



transition dig for DFA accepting $L = \{b^m a (a+b)^n\}$

DFA can be given as

$M = \{Q, \Sigma, \delta, q_0, F\}$

$Q = \{q_0, q_1\}$

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

q_0 is initial state

q_1 is final state $F = \{q_1\}$

δ can be given as

δ	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_1

RAJEEV. BILAGI
M.Tech., M.ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

Rajeev Bilagi
M.Tech
Dept of CSE

Scanned with CamScanner

iii> Not more than 3' ~~zeros~~ a's.

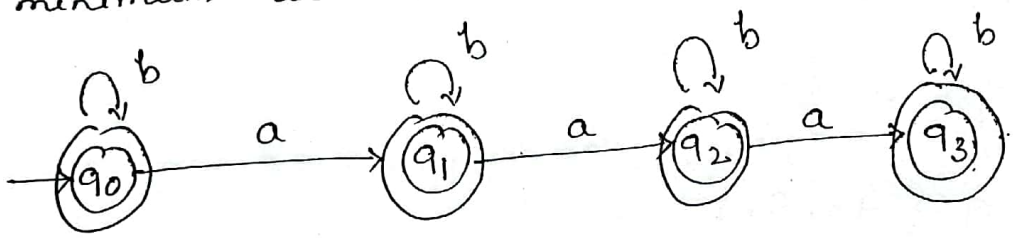
<u>soln</u>

Machine can accept

- It can accept zero a's
- " " " one a
- " " " two a's
- " " " three a's.
- But not accept 3' a's.

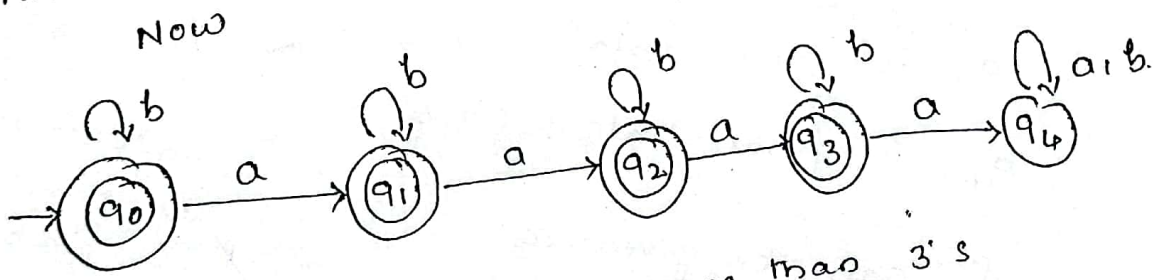
Now fulfil conditions of i, ii & iii.

minimum we can have.



Now

Now



Not accepting more than 3's

$L = \{ b^* a b^* a b^* a b^* \}$

$m = \{ Q, \Sigma, \delta, q_0, F \}$

where

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

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

q_0 is initial state

$F = \{ q_0, q_1, q_2, q_3 \}$ - final states

δ is

δ	a,	b
$\rightarrow q_0$	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

Scanned with CamScanner

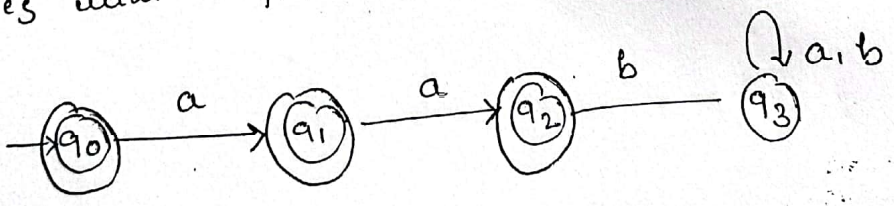
Example. 7

Design a DFA to accept strings of a's & b's except those containg the substring 'aab'

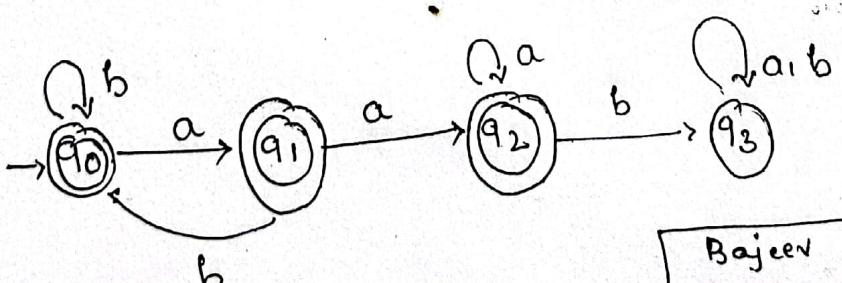
Soln:

Soln ① Draw DFA to accept the strings of a's & b's having substring abb. Then change the final states to non final states & non final states to final states. The resulting machine accepts strings those containing "abb" as substring.

Soln ② The minimum string that can be rejected is abb. for this we need four states. Still aab is rejected, q_3 will not be final state. The rest of states will be final states



Now



Dig: DFA

Bajeev Bilagi
M.Tech
Dept of CSE

Scanned with CamScanner

So, DFA. $m = \{ Q, \Sigma, \delta, q_0, A \}$

$Q = \{ q_0, q_1, q_2, q_3 \}$

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

q_0 is the start state.

δ is shown below.

$\leftarrow \Sigma \rightarrow$

δ	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_3
q_3	q_3	q_3

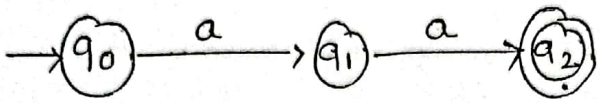
$\uparrow$ Q $\downarrow$

Scanned with CamScanner

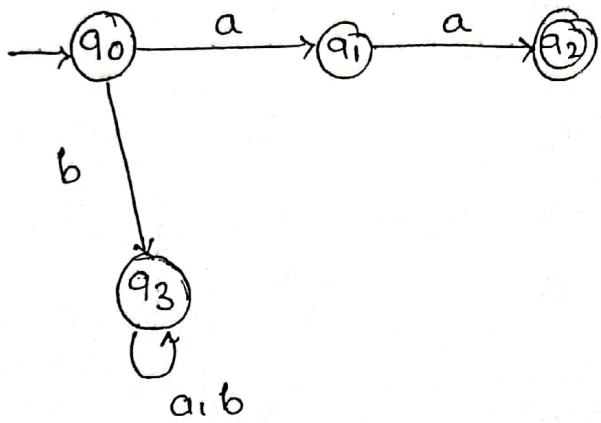
Example: 8

Design a DFA for $L = \{ awa \mid w \in (a+b)^* \}$

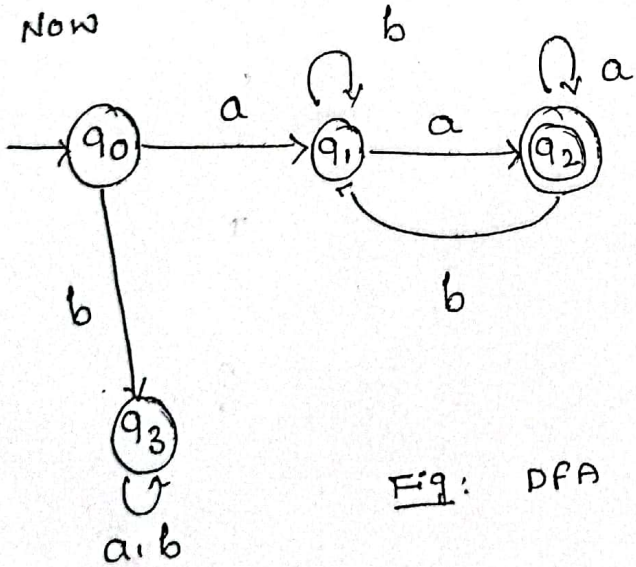
Soln: The string should start with 'a' & end with 'a'. Minimum string accepted aa



Now



Now



Rajeev Bilagi
M.Tech
Dept of CSE

RAJEEV. BILAGI
M.Tech., MISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

Fig: DFA to accept awa

Mention $M = \{ Q, \Sigma, \delta, q_0, A \}$

Scanned with CamScanner

So,

$Q = \{ q_0, q_1, q_2, q_3 \}$

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

q_0 is initial state

$A = \{q_2\}$

δ is shown given below.

$\longleftarrow \Sigma \longrightarrow$

$\uparrow$ Q $\downarrow$

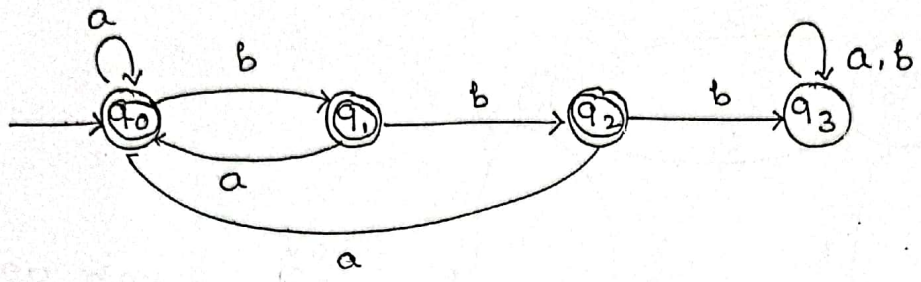
δ	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_2	q_1
q_2	q_2	q_1
q_3	q_3	q_3

Scanned with CamScanner

Obtain a DFA to accept strings of a's & b's with at most two consecutive b's.

Soln:

The machine accepts at most two consequtive b's

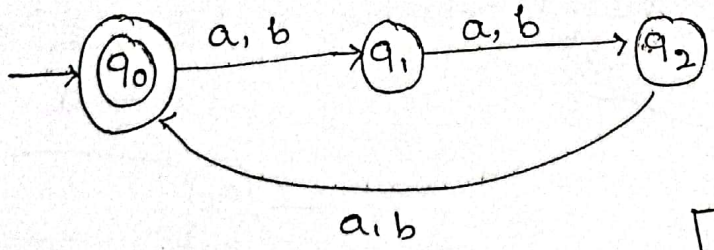


Example 10:

Obtain DFA to accept language $L = \{w : |w| \bmod 3 = 1$ on $\Sigma = \{a, b\}$

Soln: $L = \{\epsilon, aaa, aaaaaa, \ldots\}$

The DFA is given by



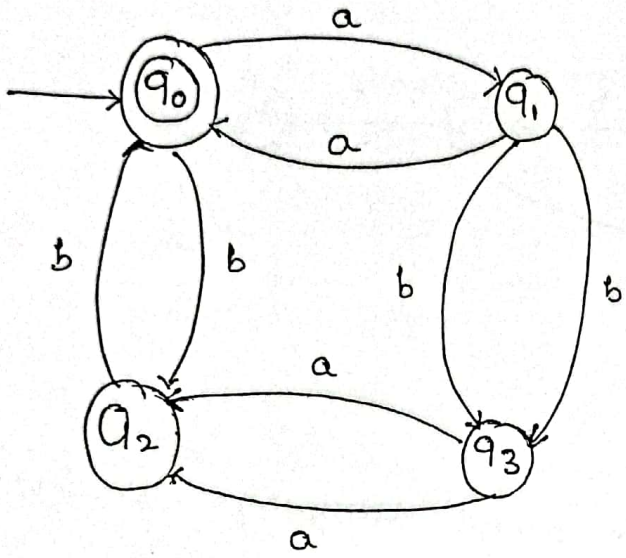
RAJEEV. BILAGI
M.Tech., MISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

Rajeev Bilagi
M.Tech
Dept of CSE

Scanned with CamScanner

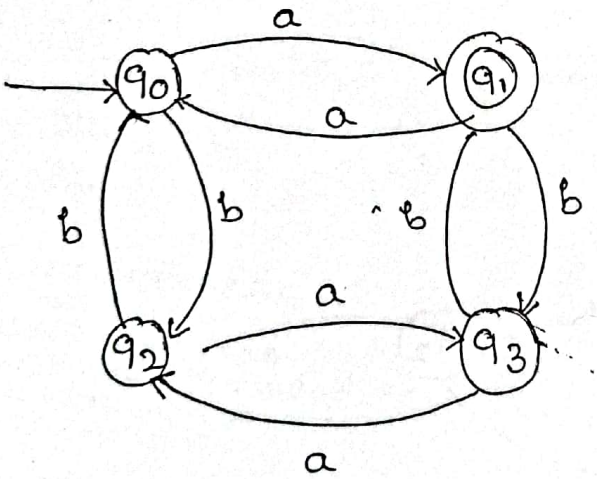
DFA which accepts even no of a's & even no on $\Sigma = \{a, b\}$

Soln: There is a std soln with 4 states & it is for 4 diff conditions.

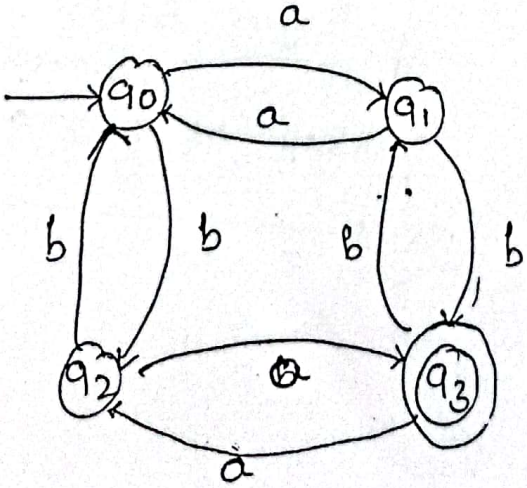


$L = \{ \epsilon, abab, aabb \ldots \}$

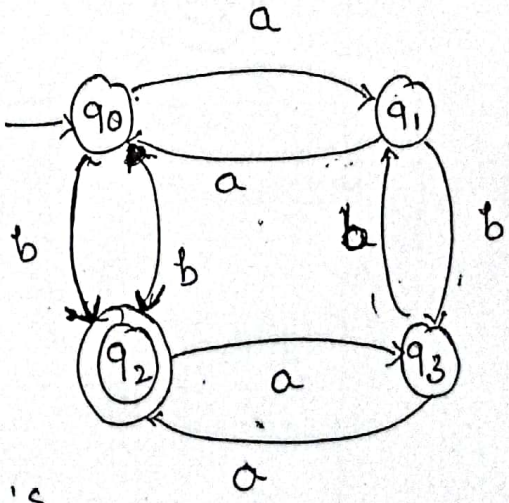
Now other 3 conditions



odd no's of a's & even no of b's



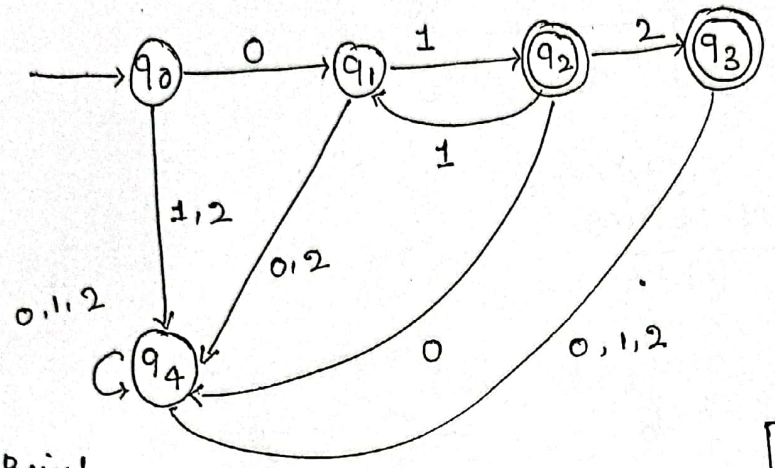
odd of no' a's & odd no of b's.



even of No of a's & ~~even~~ odd no of b's.

Obtain DFA to accept strings of 0's, 1's & 2's begining with a '0' followed by odd number of 1's, followed by a 2.

Soln:



Bajeev state

Bajeev Bilagi
M.Tech
Dept of CSE

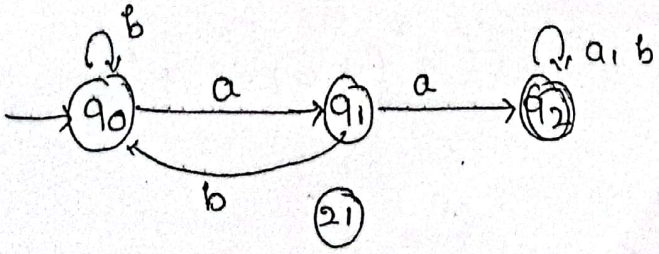
Example 17:

Design a DFA which accepts the strings having aa as its substring. Check whether string abaabb is accepted or not. using ① pictorical representation

② properties of transition function

Soln;

DFA can be given by



RAJEEV. BILAGI
M.Tech, MISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

Scanned with CamScanner

Thus the DFA M is given by

$M = \{\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, q_2\}$

& δ is given by

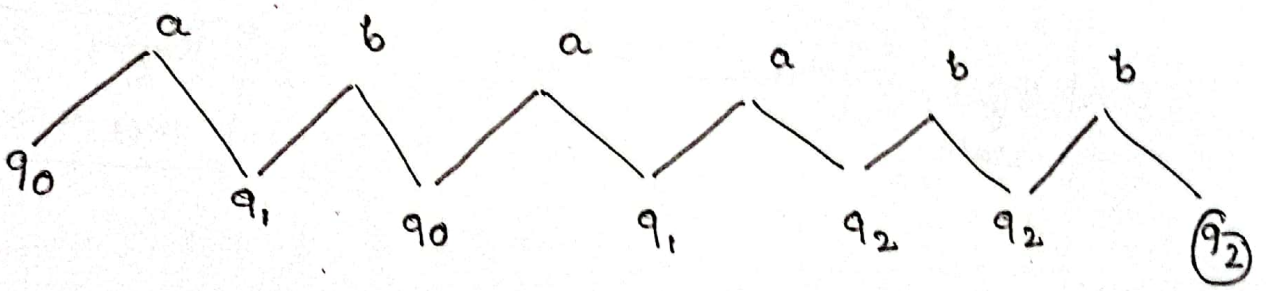
$\longleftarrow \Sigma \longrightarrow$

δ	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_2

$\uparrow$ Q $\downarrow$

Acceptance of string

①



final state

∴ The string abaabb is accepted.

② **Transition function properties**

(i) $\delta(q_0, \epsilon) = \delta^*(q_0, \epsilon) = q_0$ —— (i)

ii) $\delta(q_0, wa) = \delta(\delta^*(q, w), a) = P_j$ —— (ii)

iii) $\delta(q_0, aw) = \delta(\delta^*(q, a) w) = P_q$ —— (iii)

Where $q \in Q$ $\quad w \in \Sigma^*$

$a \in \Sigma$

& P_i, P_q are all states which are reachable from q_0.

Scanned with CamScanner

Now for strings abaabb

Now

$\delta(q_0, abaabb) = \delta(\delta^*(q_0, a), baabb)$ [∵ apply rule iii)

$= \delta(q_1, baabb)$ (∵ ")

$= \delta(\delta^*(q_1, b)\, aabb)$ (∵ ")

$= \delta(q_0, aabb)$ (∵ ")

$= \delta(\delta^*(q_0, a), abb)$ (∵ ")

$= \delta[(q_1, abb)]$ (∵ ")

$= \delta[\delta^*(q_1, a), bb]$ (∵ ")

$= \delta[\delta(q_2, bb]$ (∵ ")

$= \delta[\delta^*(q_2, b), b]$ (∵ ")

$= \delta[q_2, b]$ (∵ ")

$= \delta[\delta^*(q_2, b), \epsilon]$ (∵ ")

$= \delta[q_2, \epsilon]$ [∵ Apply rule i]

$= q_2$ — Final State

∴ String is Accepted

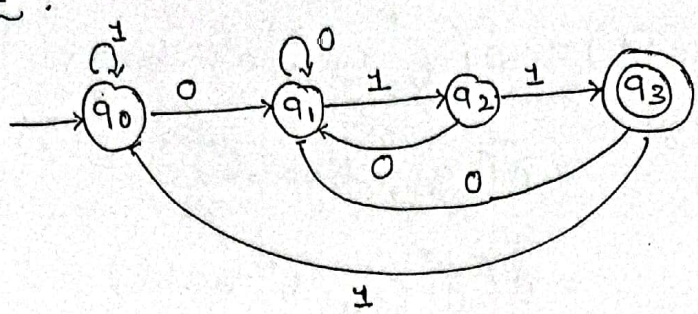
RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

Rajeev Bilagi
M.Tech
Dept of CSE

Scanned with CamScanner

For DFA given below check whether string 010 accepted or rejected. using transition properties

Soln:



Now

$\delta(q_0, 0101) = \delta(\delta^*(q_0, 0), 101)$ (∵ Apply rule iii)

$= \delta(q_1, 101)$ (")

$= \delta(\delta^*(q_1, 1), 01)$ (")

$= \delta(q_2, 01)$ (")

$= \delta(\delta^*(q_2, 0), 1)$ (")

$= \delta(q_1, 1)$ (")

$= \delta(\delta^*(q_1, 1), \epsilon)$ (")

$= \delta(q_1, \epsilon)$ (")

$= q_1$ – which is not final state (∵ Apply rule i)

String <u>0101</u> is rejected by given DFA

Scanned with CamScanner

Divisible by K problems:

We will discuss a method of constructing DFA that divide a number by 'k' For this kind of problems transitions can be obtained using following relation.

$$\delta(q_i, a) = q_j$$

Where $j = (r * i + d) \bmod k$.

* r is the radix of input. For binary $r = 2$.
* i is the reminder obtained after dividing by k.
* d is representation of digits $\Sigma = \{0, 1\}$
* k is divisor.

Rajeev Bilagi
M.Tech
Dept of CS&E

Steps to follow to find FA:

Step ①: identify the radix, input alphabets & Division k

②: compute possible reminders. These reminders represent states of FA.

RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

③: Find transitions using $\delta(q_i, a) = q_j$ where $j = (r * i + d) \bmod$

④: construct DFA using transitions obtained in step 3

Scanned with CamScanner

Example ①:

construct a DFA which accepts 0's & 1's which divisible by 5. & each string is represented as a binary number.

Ex: 0000, 0101, 1010, 1111 should be accepted.

Solution:

In given problem,

$r = 2$

$d = \{0, 1\}$

$i = 0, 1, 2, 3, 4$

$k = 5$

Now $\delta(q_i, a) = q_j$ where $j = (r * i + d) \bmod k$.

Now $r = 2$
$k = 5$

$\delta(q_i, a) = (2 * i + d) \bmod 5$

$j = (2i + d) \bmod 5$

Scanned with CamScanner

reminder	d	$(2*i+d) \bmod k$	$\delta(q_i, d) = q_j$
$i=0$	0	$(2*0+0) \bmod 5 = 0$	$\delta(q_0, 0) = q_0$
	1	$(2*0+1) \bmod 5 = 1$	$\delta(q_0, 1) = q_1$
$i=1$	0	$(2*1+0) \bmod 5 = 2$	$\delta(q_1, 0) = q_2$
	1	$(2*1+1) \bmod 5 = 3$	$\delta(q_1, 1) = q_3$
$i=2$	0	$(2*2+0) \bmod 5 = 4$	$\delta(q_2, 0) = \delta_4$
	1	$(2*2+1) \bmod 5 = 0$	$\delta(q_2, 1) = q_0$
$i=3$	0	$(2*3+0) \bmod 5 = 1$	$\delta(q_3, 0) = q_1$
	1	$(2*3+1) \bmod 5 = 2$	$\delta(q_3, 0) = q_2$
$i=4$	0	$(2*4+0) \bmod 5 = 3$	$\delta(q_4, 0) = q_3$
	1	$(2*4+1) \bmod 5 = 4$	$\delta(q_4, 1) = q_4$

RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057
Rajeev Bilagi
M.Tech
Dept of CSE

Step ④: $M = \{Q, \Sigma, \delta, q_0, F\}$

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

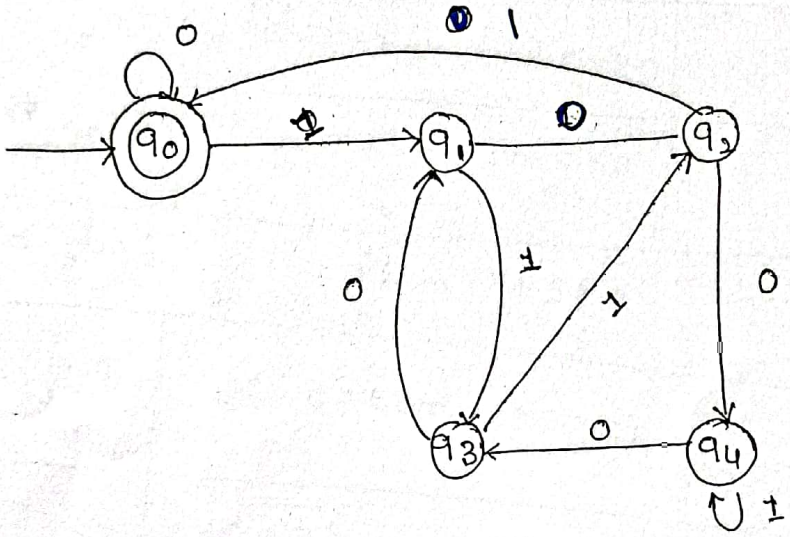
$\Sigma = \{0, 1\}$

q_0 is initial state

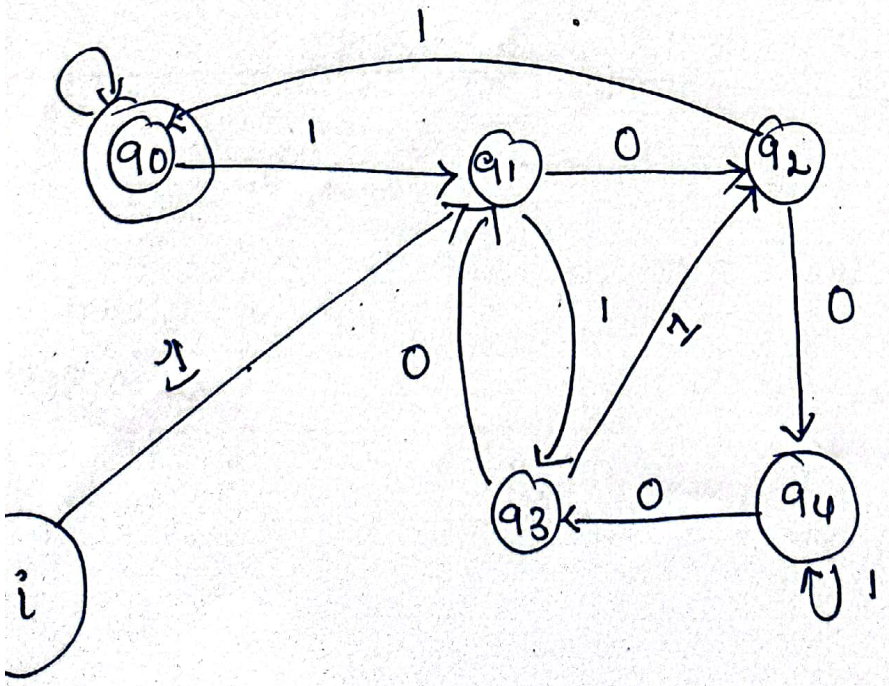
$F = \{q_0\}$

Scanned with CamScanner

δ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4



(*) Divisible by 5, but not starts with 0



Scanned with CamScanner

Modulo 'K' counter problems:

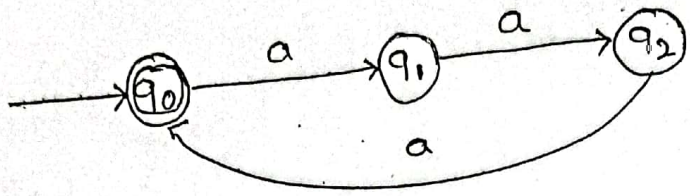
① Obtain DFA to accept the language

$L = \{ W : |W| \bmod 3 = 0 \}$ where $\Sigma = \{a\}$.

or

$L = \{ a^{3n} : n \geq 0 \}$

Soln:



$Q = \{ q_0, q_1, q_2 \}$

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

$F = \{ q_0 \}$

q_0 is initial state

δ:

	a
$\rightarrow q_0$	q_1
q_1	q_2
q_2	q_3

RAJEEV MIRAGI
M.Tech., M.B.A.
Assistant Professor
Department of Computer Science & E
Bangalore College of Engineering
Bangalore - 560 057

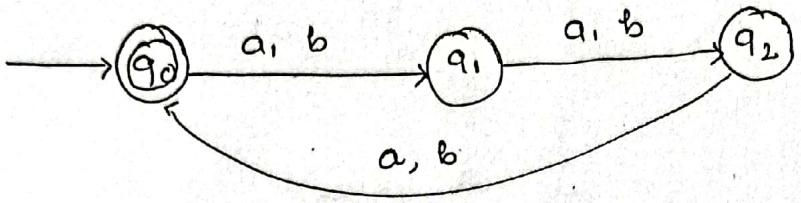
Rajeev Mirajgi
Dept of CS&E

Scanned with CamScanner

(2) $L = \{ w : |w| \bmod 3 = 0$ on $\Sigma = \{a, b\}$

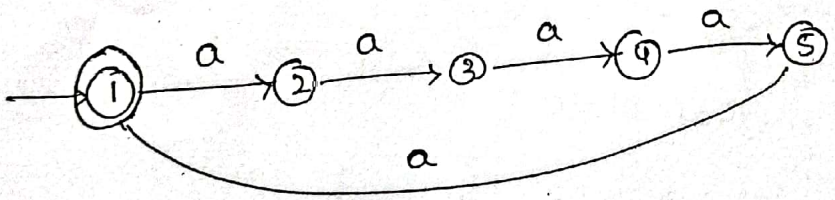
Soln:

$L = \{ \epsilon, aaa, aab, abb, bab, \ldots \}$

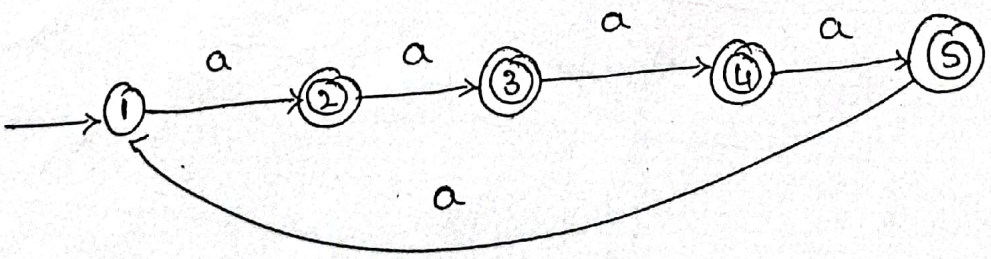


(3) $L = \{ w : |w| \bmod 5 \neq 0 \}$ on $\Sigma = \{a\}$.

$L = \{ \epsilon, aaaaa, \ldots \}$



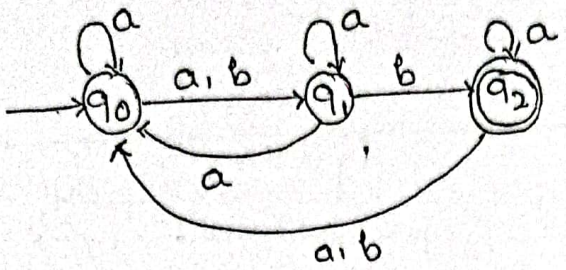
Not containing



Scanned with CamScanner

Non Deterministic finite Automata :-

consider the transition diagram



Here

$$M = \{Q, \Sigma, \delta, q_0, F\}$$

Where

$Q = \{q_0, q_1, q_2\}$

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

q_0 is initial state.

$F = \{q_2\}$ is final state.

δ is given by

δ	a	b
$\rightarrow q_0$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$
q_1	$\{q_0, q_1\}$	q_2
q_2	$\{q_2\}$	–

Rajeev Bilagi
M.Tech
Dept of CSE

Definition :

NFA is a five tuple or quintuple

$$M = \{Q, \Sigma, \delta, q_0, F\}$$

RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

where Q - set of finite states.

Σ - set of finite states

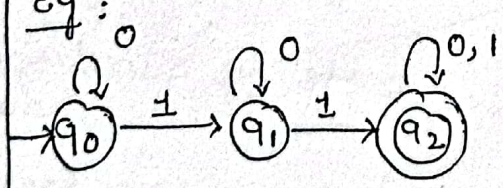
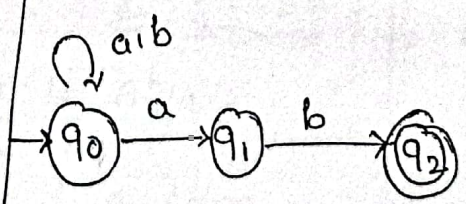
δ - is transition f^n $Q \times \Sigma$ to subsets of 2^Q

Scanned with CamScanner

q_0 — initial state

$F \subseteq Q$ set of final states.

Differances betn NFA & DFA

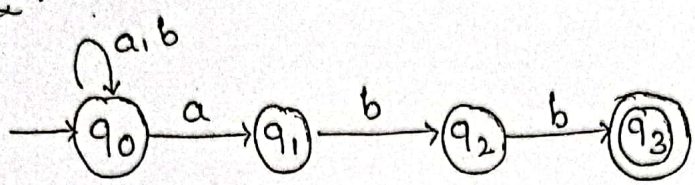
	DFA	NFA
1.	DFA is 5 tuple $M = \{Q, \Sigma, \delta, q_0, F, A\}$ where Q — is set of finite states Σ - is set of finite alphabets δ : transition fn $\delta : Q \times \Sigma$ to Q $A \subseteq Q$ is set of final states	NFA is same as DFA except the definition δ. & δ is given by $\delta : Q \times (\Sigma \cup \epsilon)$ to subsets of 2^Q.
2	No- ϵ - transitions i.e	ϵ- transitions can exist. i.e with out any put there can be transition from one state to other state
3	Difficult to construct	very construct
4	Eg:  	

Scanned with CamScanner

Example 1:

Design NFA to accept strings of a & b ending with strings abb.

Soln:



Here the machine

$$N = \{Q, \Sigma, \delta, q_0, F\}$$

Where

$Q = \{q_0, q_1, q_2, q_3\}$

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

q_0 - is initial state

q_3 - is final state.

δ is given by.

δ	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0
q_1	$\emptyset$	q_2
q_2	$\emptyset$	q_3
q_3	$\emptyset$	$\emptyset$

Rajeev Bilagi
M.Tech
Dept of CSE

RAJEEV. BILAGI
M.Tech., M ISTE
Associate Professor
Department of Computer Science & Engg
Sapthagiri College of Engineering
Bangalore - 560 057

Scanned with CamScanner