$\boxed{\text{TOC}}$ (Theory of Computation)

Syllabus of TOC :- (1) Regular language and finite automata
(2) Context free language and pushdown automata
(3) REC, RE and Turning machine
(4) closure properties and undecidability.

1) Language
2) Grammer
3) Automata (Machine)

1) Tables
2) Language
3) Automata

Q. Why we study TOC? – To understand the computation of machine theoritically by using mathematics.

Three pillars :- L A G
Language, Automata, Grammer

① Symbols : {a, b, c, 0, 1, ...}

Alphabets : finite set of symbols. It is denoted by $\Sigma(a,b)$.
(Like Alphabets of English)

Exp – $\Sigma(a,b)$ = a, b – Length 1
aa, ab, ba, bb – Length 2
aaa, aba, ... – Length 3

(Like words in English)

String : sequence of alphabets

Language : Callection of strings.

★ Language can be finite or infinite.

Exp – L_1 : String of length 3
finite = {aaa, aab, aba, baa, bba, bab, abb, ~~aab~~, bbb}

L_2 : String starts with a and end with a.
Infinite = {a, aba, aa, aaa, ..., abbbaa.., ...}

L_3 : String with length 0. It is denoted by Σ^0
$\Sigma^0 = \varepsilon$ – string with length 0.

② Automata :- Automata is a machine / mathematical model that use to determine whether a string is a part of a language or not.

- Types of automata :- (1.) finite automata (FA)
  (2.) Push down automata (PDA)
  (3.) Linear bound automata (LBA)
  (4.) Turing machine (TM)

- Power of Σ :- $\Sigma = \{a, b\}$

Σ^0 = set of all strings with length 0 (Denoted by ε)

Σ^1 = set of all strings with length 1 = $\{a, b\}$

Σ^2 = set of all strings with length 2 = $\{aa, ab, ba, bb\}$

$\vdots$

(Kleene closure) $\rightarrow \Sigma^*$ = set of all strings $\quad \Sigma^* = \Sigma^+ \cup \Sigma^0$

$$\boxed{\text{no. of strings in } \Sigma^n = 2^n}$$

- (Positive closure) $\Sigma^+ = \Sigma^* \setminus \Sigma^0$
- Σ^0 is an identity element in Σ^* but not in Σ^+.

  $\bigcup_{i \geq 1} \Sigma^i$ - set of all finite strings.

(3.) Grammer :- A grammer 'G' is defined as quartruple.

$G = \{V, T, P, S\}$

V → variable, T → Terminal, P → Production Rule, S → Start.

Exp :- $S \rightarrow aSb \mid \varepsilon$

ε, aSb, $aaSbb$, $aaaSbbb$

ab, a^2b^2, a^3b^3 ... a^nb^n

(where $n \geq 0$)

★ By the help of grammer we can make strings that obey the rules and make a language finally.

(Alphabets) Σ $\xrightarrow{\text{Grammer}}$ strings $\longrightarrow$ language.

(Alphabets) Σ $\xrightarrow{\text{No Grammer}}$ strings.

Deterministic finite automata (DFA)

$DFA(Q, \Sigma, \delta, q_0, F)$

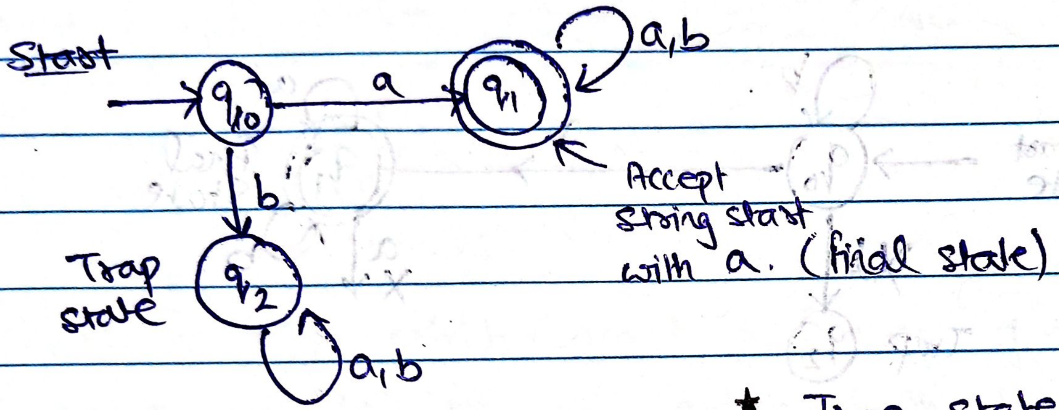
- Q → Set of finite states
- Σ → Alphabets
- δ → Transition
- q_0 → Start State
- F → Set of final states.

$\boxed{F \subseteq Q}$

Exp:- $\Sigma(a, b)$ – Alphabet

Strings start with a – Grammer

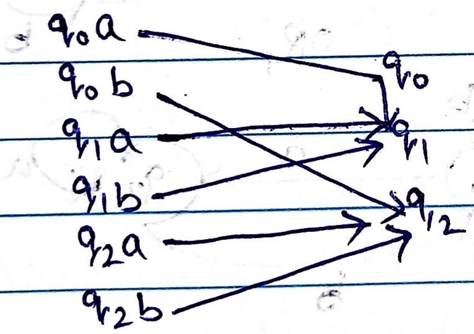
{a, aa, ab, aba, aaa, ... } – Language



★ Trap state always maps to itself for any string input.

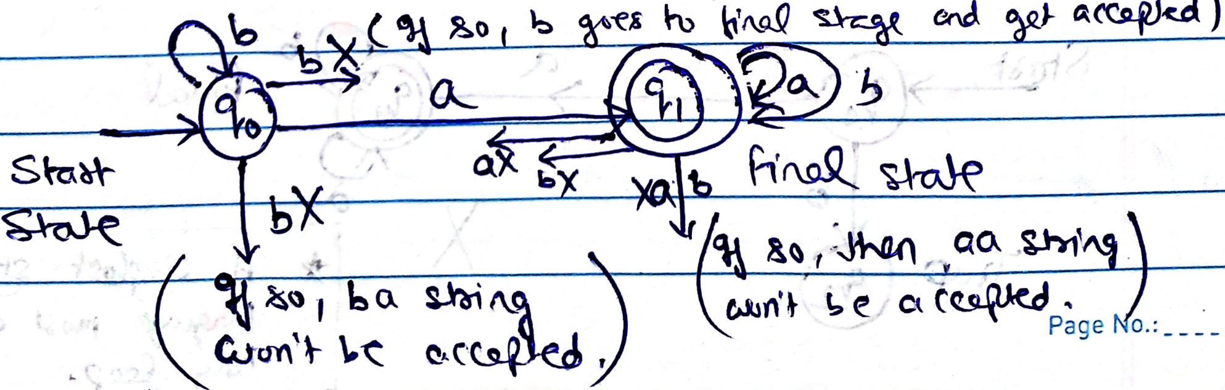
$\delta : Q \times \Sigma \rightarrow Q$

$Q = \{q_0, q_1, q_2\}$, $\Sigma = (a, b)$

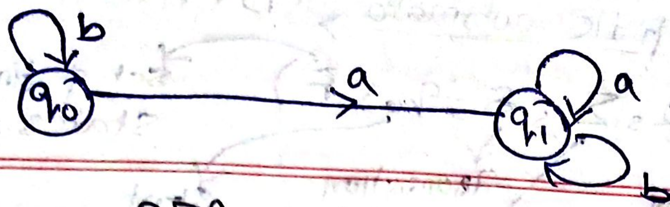


Q. Construct a DFA which accept a language of all strings containing 'a' $\Sigma(a, b)$.

{ a, aa, aaa, ba, ab, abab, ... }



Ans :

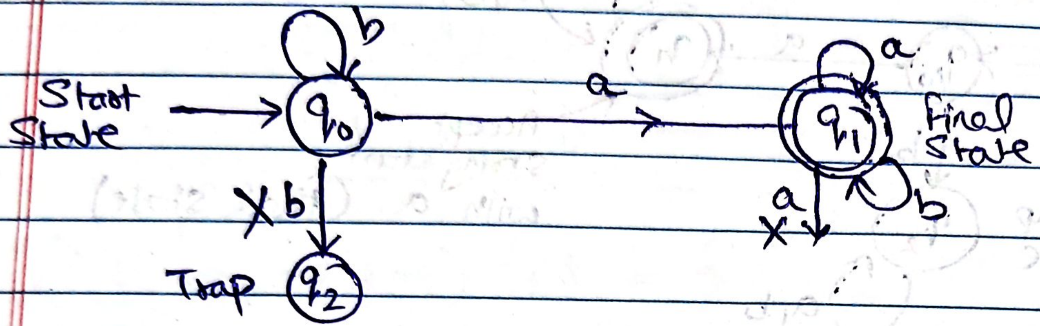


Q. Construct a DFA which accept a language of all strings that end with 'a'.

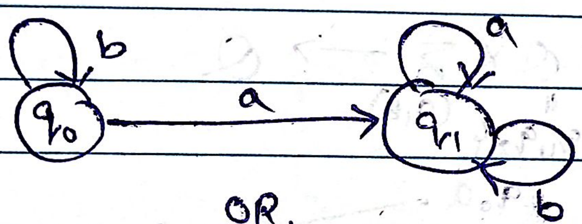
$\Sigma(a,b)$ – Alphabet

Grammar – string end with 'a'.

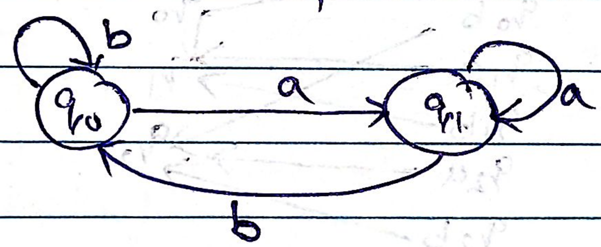
Language = {a, aa, aaa, ..., ba, aba, baa, ...} Infinite



Ans:

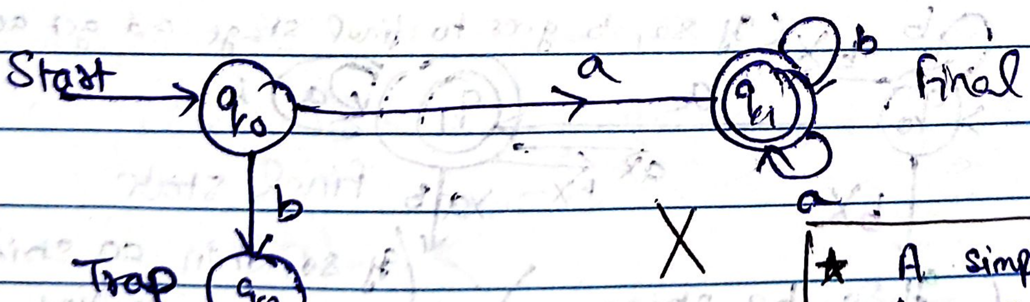


OR,

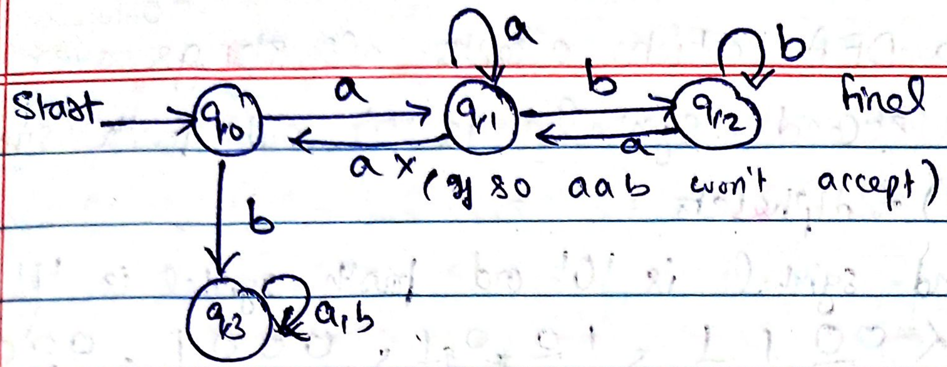


Q. Construct a DFA which accept a language of all strings start with 'a' and ending with 'b'.

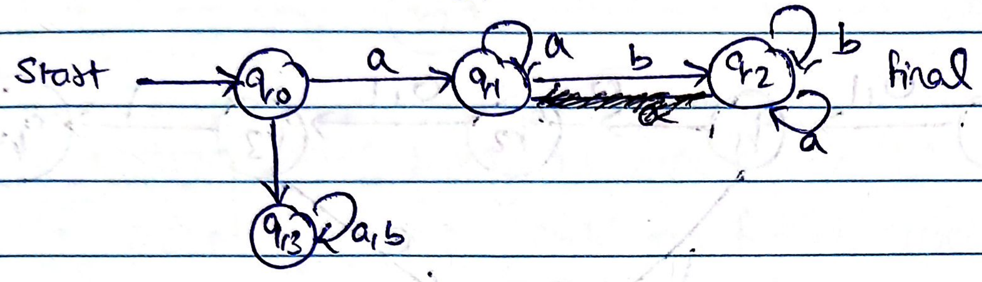
L = {ab, abb, aab, aabb, ... }



★ A simplest string of the Language must accept without any loop.



OR



Q. Construct a DFA which accept a language of all strings not starting with 'a' or not ending with 'b'.

Let A: Strings not starting with a

B: Strings not ~~starting~~ ending with b.

$(A \cup B) = (A^c \cap B^c)^c = \{\epsilon, a, b, aa, baa, \dots\}$

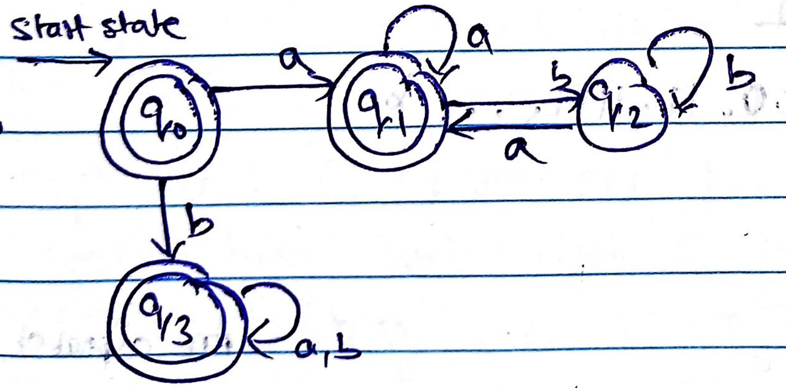
$A^c \cap B^c$ = Starting with a and ending with b (Last problem)

we need to take Complement of the last problem's automata.

Hint:- final → non final

non-final → final

Ans:

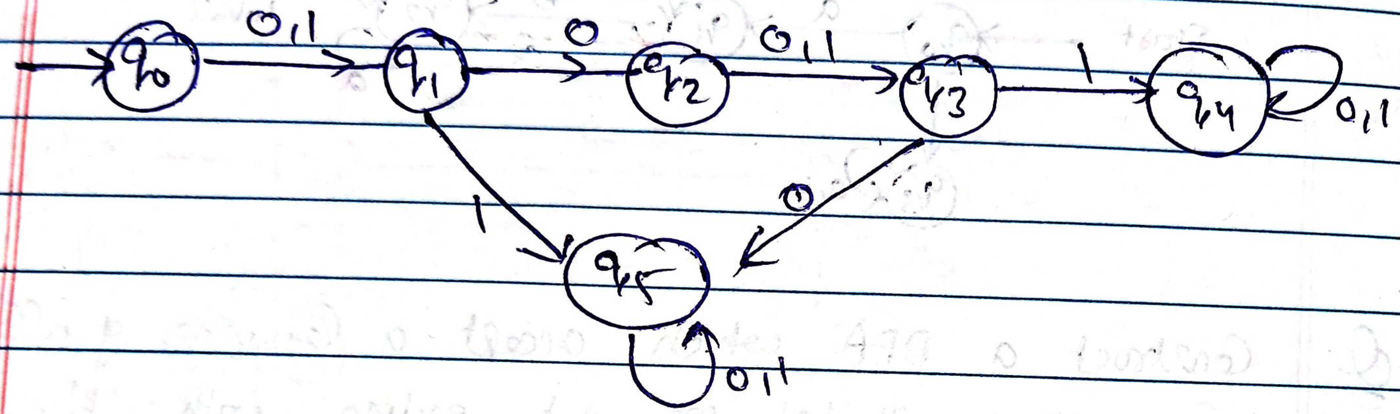


Q. Design a DFA which accepts all strings over (0,1) in which second symbol is '0' and fourth symbol is '1'.

$\Sigma(0,1)$ – Alphabets

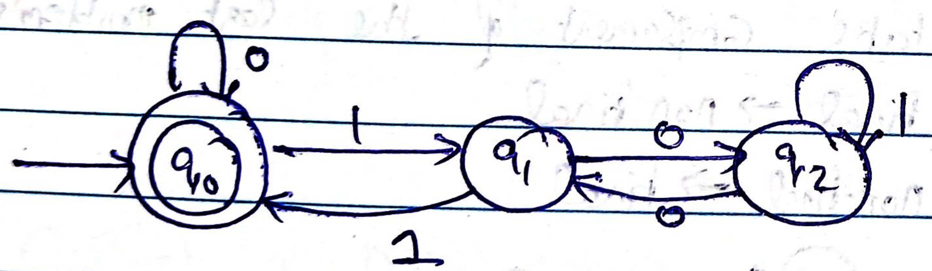
- second symbol is '0' and fourth symbol is '1' – Grammer

– L = { 0011, 1001, 0001, 00011, ... }



★ no. of states ≥ n+1, where n = length of the simplest string in language.

Q. Construct a DFA which accept a language of all binary strings divisible by 3 over $\Sigma(0,1)$.



L = { 0, 1100, 11, 1111, ... }

An alphabet Σ is a special whose elements are recogizable by 'machine'

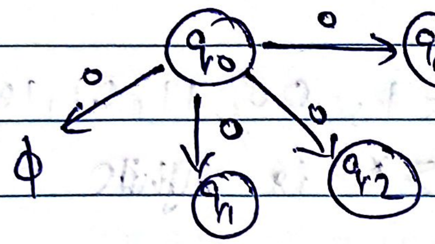
$\Sigma^* = \{x_0 x_1 .. x_k \mid k \in \mathbb{N}, x_i \in \Sigma\}$

Non-Deterministic finite Automata (NDFA)

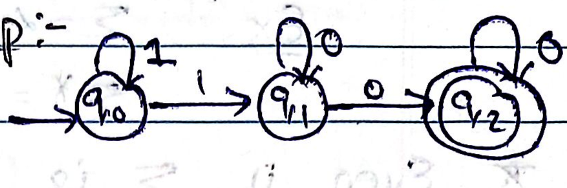
Date:

- When we have multiple choices to move from one state to another, it is called non-deterministic finite automata.

NDFA$(Q, \Sigma, q_0, F, \delta)$ $F \subseteq Q$



Exp:-



Introduction :- A function with a 2 bit out is called decision problem. For a decision problem one must specify the sets. (1.) A of all inputs
(2.) B of all inputs which give "yes" outputs.

Example :- To decide when a graph is connected.
A: all graphs
B: all connected graphs.

- Central theme → How to decide whether a given decision problem is algorithmically solvable
An example which is not :- To decide any given mathematical statement is a correct thm or not.

- Alphabet : Σ = finite set from which out decision problems will take their input strings. Exp- $\{0,1\}$
- String : A string over Σ is any finite length sequence of elements of Σ

Exp-(a) $\Sigma = \{0,1\}$
Strings = 011001, 00111, 01, ...
(b) $\Sigma = \{a, b, c, ...\}$
strings = abaca, acbad, ...

Page No.:

- Length of any string x is denoted by $|x|$.

  Ex $x = ababa \longrightarrow |x| = 5$

  If length = 0, then string is called 0 length string which denoted by ε.

- The set of all strings over Σ is denoted by Σ^*

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

  $\Sigma^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 001, 110, \ldots\}$

★ Even if Σ is finite, Σ^* is infinite

- $\phi^* \stackrel{\text{defn}}{=} \varepsilon$

<u>Operators on strings</u>

(1) Cocatenation : $x \cdot y = xy$

$(01)\cdot(11) = 0111$

- ε sets as an identity for cocatenation.

  $x \cdot \varepsilon = \varepsilon \cdot x = x$

- Cocatenation is associative. i.e., $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
- Cocatenation is not-commutative.

Thm: Σ^* is a monoid under cocatenation where ε as an identity.

monoid = closed + Associative + Identity

closed: $x, y \in \Sigma^*$, $x \cdot y \in \Sigma^*$

Associative: $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

Identity: $x \cdot \varepsilon = \varepsilon \cdot x = x$

Set Cocatenation : A, B - sets

$A \cdot B \stackrel{\text{defn}}{=} \{xy \mid x \in A, y \in B\}$

$A^0 \stackrel{\text{defn}}{=} \{\varepsilon\}$

Taking $A^0 = \{\varepsilon\}$ makes $A^m \cdot A^n = A^{m+n}$ holds

Asterate A^* of A is defined by $A^* = \bigcup_{n \geq 0} A^n$ — all finite strings of elem of A.

$\boxed{A^n = \text{set of all finite string of length } n, \text{ of elements of } A.}$

· **Properties** :- 1. $A \cup (B \cup C) = (A \cup B) \cup C$
2. $(A \cap B) \cap C = A \cap (B \cap C)$
3. $A(BC) = (AB)C$

} (Associative)

apsara

Date: __.__.__

$A \cup B = B \cup A$

$A \cap B = B \cap A$

$A \cdot B \neq B \cdot A$

$\cup \longrightarrow \phi$ is identity

$\cap \longrightarrow$ universal set is identity.

Note / Convention :- $A \cdot \phi = \phi \cdot A = \phi$

- $A \cup (B \cap C) = (A \cup B) \cap (A \cap C)$
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- $A(B \cup C) = AB \cup AC$
- $(A \cup B)C = AC \cup BC$
- $A(\bigcup_i B_i) = \bigcup_i AB_i$
- $(\bigcup_i B_i) A = \bigcup_i B_i A$

★ Concatenation doesn't distributing over $\cap$.

$A = \{a, ab\}$, $B = \{b\}$, $C = \{\varepsilon\}$

$A \cdot (B \cap C) \neq AB \cap AC$

LHS, $B \cap C = \phi \Rightarrow A \cdot (B \cap C) = \phi$

RHS, $AB = \{ab, abb\}$

$AC = \{a, ab\} \Rightarrow AB \cap AC = \{ab\}$

- $(A \cup B)^c = A^c \cap B^c$
- $(A \cap B)^c = A^c \cup B^c$
- $\phi^* = \{\varepsilon\}$
- $A^* A^* = A^*$
- $A^{**} = A^*$
- $A^* = \{\varepsilon\} \cup AA^* = \{\varepsilon\} \cup A^* A$

$\varepsilon \notin AA^*$

States and Transistion of a system

A state is one of the conditions of a system. A transistion of a system is its change from one state to another.

Ex :- ① for a bulb there are two states on and off.
② For a fan there are 6 states.

- Finite Automata :- A deterministic finite automata is a structure (DFA) $M = (Q, \Sigma, \delta, S, F)$

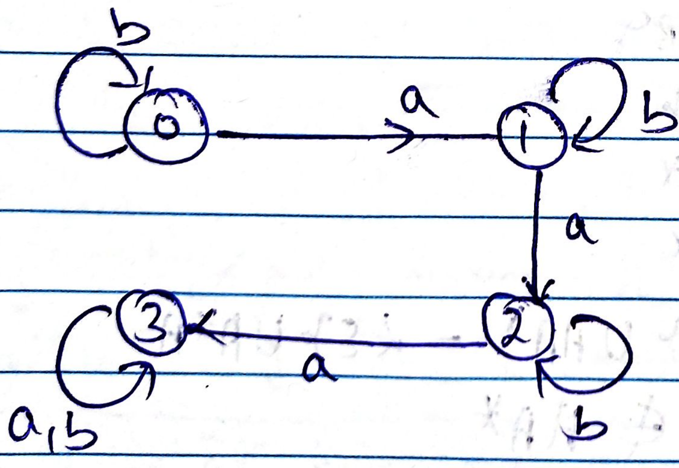
Q = finite set of states
Σ = finite set of input alphabets
$\delta : Q \times \Sigma \rightarrow Q$ is Transistion function
S - start state
F - accept or final state

Ex A_4 state finite automata

$Q = \{0, 1, 2, 3\}$ $\Sigma = \{a, b\}$ $S = \{0\}$ $F = \{3\}$

δ: $(0, a) = 1$
$(1, a) = 2$
$(2, a) = (3, a) = 3$
$(q, b) = q$

	a	b
0	1	0
1	2	1
2	3	2
3	3	3



L : Starts from aa

Date:

★ Informally, an input can be any string $x \in \Sigma^*$ scan the input from left to right. Now, 1st element of the string is x_1 (say) move to $\delta(s, x_1)$. Now let the 2nd element in the string is x_2 (say). Move to $\delta(\delta(s,x_1), x_2)$

Next go to $\delta(\delta(\delta(s,x_1), x_2), x_3)$ and so on.

Here $\delta(s,x_1)$, $\delta(\delta(s,x_1), x_2) \in Q$

Defn: $\hat{\delta} : Q \times \Sigma^* \longrightarrow Q$

Defined by induction : $\hat{\delta}(q, \varepsilon) = q$

$\hat{\delta}(q, xa) = \delta(\hat{\delta}(q,x), a)$

The function $\hat{\delta}$ maps a state q to the ~~string~~ state $\hat{\delta}(q,x)$, $\forall$ string x. $\hat{\delta}$ is just a 'multistep version of q'.

Note :- $\hat{\delta}(q,a) = \hat{\delta}(q, \varepsilon a)$

$= \delta(\hat{\delta}(q,\varepsilon), a) = \delta(q,a)$

$\forall a \in \Sigma$.

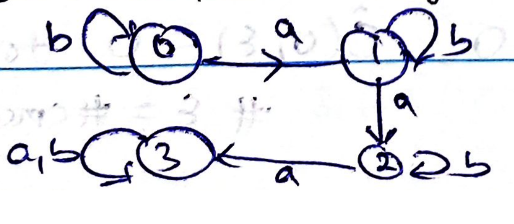
Defn: A string is said to be accepted by the automation M if $\hat{\delta}(s,x) \in F$ and rejected if $\hat{\delta}(s,x) \notin F$.

★ The set or language accepted by M is L(M).

$L(M) = \{x \in \Sigma^* \mid \hat{\delta}(s,x) \in F\}$

• Regular set :- A subset $A \subseteq \Sigma^*$ is called regular if $A = L(M)$ for some finite.

Ex - In previous page example of A_4.

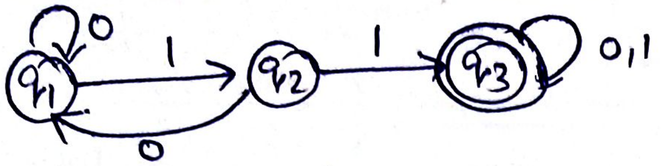


$L(M) = \{0,1,2,3\} = A_4$

(Like range of automata L(M) is called a regular set.)

Page No.:

Ex—

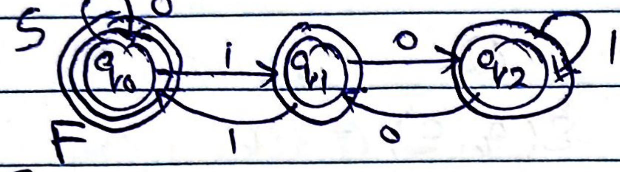


$L(M) = \{011, 11, 0011, 01011, \dots\}$

= Any string with 2 Consecutive 1's.

Def^n: for any alphabet Σ, a set $A \subseteq \Sigma^*$ is called regular set if $\exists$ an automata M s.t. $L(M) = A$.

Ex— $A = \{x \in \{0,1\}^* \mid x$ represents a multiple of 3 in binary$\}$
$= \{0, 11, 1111, 1100, \dots\}$



Define Automata in such a way.

	0	1
$q_0 \equiv 0$	0	1
$q_1 \equiv 1$	2	0
$q_2 \equiv 2$	1	2

$\delta(q, c) = (2q + c) \bmod 3$

$\#(x0) = 2(\#x) + 0$

$\#(x1) = 2(\#x) + 1$

Hence $\#(xc) = 2(\#x) + c$

$x = \epsilon \Rightarrow \hat{\delta}(0, \epsilon) = 0$
$= \# \epsilon \bmod 3$

$$\hat{\delta}(0, xc) = \delta(\hat{\delta}(0, x), c)$$
$$= \delta(\#x \bmod 3, c)$$
$$= (2(\#x) + c) \bmod 3$$
$$= \#xc \bmod 3$$

Claim : $A = L(M)$

Observe: $\delta(q, c) = (2q + c) \bmod 3$ (check)

we shall show $\forall x \in \{0,1\}^* = \Sigma^*$

$$\hat{\delta}(0, x) = \begin{cases} 0 & \text{if } \#x = 0 \bmod 3 \\ 1 & \text{if } \#x = 1 \bmod 3 \\ 2 & \text{if } \#x = 2 \bmod 3 \end{cases}$$

This will do!!

Induction on $|x|$: $|x| = 1$ done ✓

$|x| = 0$: $\hat{\delta}(0, \epsilon) = 0$ done

$\# \epsilon = \# \epsilon \bmod 3$

(2.) Assume result hold for x and try to show for xc whose $c \in \{0,1\}$

$\hat{\delta}(0, xc) = \delta(\hat{\delta}(0,x), c) = \delta(\#x \bmod 3, c)$
$= (2(\#x \bmod 3) + c) \bmod 3$
$= (2(\#x) + c) \bmod 3$ [mod property]
$= \#xc$ (property of binary)
$xc = 2x + c$

Deterministic finite automata (DFA) →

$M_1 = (Q_1, \Sigma, \delta_1, S_1, F_1)$ $L(M_1) = A$

$M_2 = (Q_2, \Sigma, \delta_2, S_2, F_2)$ $L(M_2) = B$

Define $M_3 = (Q_3, \Sigma, \delta_3, S_3, F_3)$ ← Product of M_1 & M_2

$Q_3 = Q_1 \times Q_2$

$F_3 = F_1 \times F_2$

$S_3 = (S_1, S_2)$ and $\delta_3 : Q_3 \times \Sigma \rightarrow Q_3$

$\delta_3((p,q), a) = (\delta_1(p,a), \delta_2(q,a))$

Recall : $\hat{\delta}_3((p,q), \varepsilon) = (p,q)$
$\hat{\delta}_3((p,q), xa) = \delta_3(\hat{\delta}_3((p,q), x), a)$

Thm : $L(M_3) = L(M_1) \cap L(M_2)$

Lemma : $\forall x \in \Sigma^*$, $\delta_3((p,q), x) = (\hat{\delta}_1(p,x), \hat{\delta}_2(q,x))$

Pf Induction on length of x, $|x|$

$|x| = 0 \Rightarrow x = \varepsilon$

$\hat{\delta}_3((p,q), \varepsilon) = (p,q)$
$= (\hat{\delta}_1(p,\varepsilon), \hat{\delta}_2(q,\varepsilon))$

$|x| = 1$, Definition of δ_3

Induction step ÷ Assume lemma holds for $x \in \Sigma^*$.
we show it for xa where $a \in \Sigma$

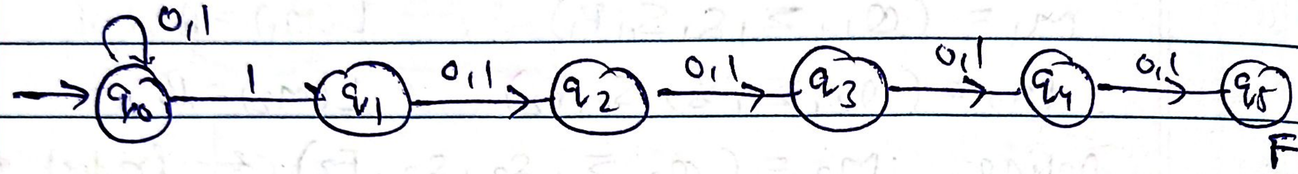
$\hat{\delta}_3((p,q), xa) = \delta_3(\hat{\delta}_3((p,q), x), a)$ [Definition]
$= \delta_1(\hat{\delta}_1(p,x), a), \delta_2(\hat{\delta}_2(q,x), a))$ (Definition of δ_3)
$= (\hat{\delta}_1(p, xa), \hat{\delta}_2(q, xa))$ (Defn of $\hat{\delta}_1$ & $\hat{\delta}_2$)

apsara

Date: __.__.____

Remark: Let $M = (Q, \Sigma, \delta, s, F)$ be a DFA with $A = L(M)$. Then $A^c = L(M')$ where M' is the DFA $(Q, \Sigma, \delta, s, F^c)$. Hence the property of being regular is closed under taking complement of a set and together with remark this shows that it is closed under union ($\cup$).

<Non-deterministic finite automata (NDFA)>



$A = \{x \in \{0,1\}^* \mid$ fifth symbol from right is $1\}$

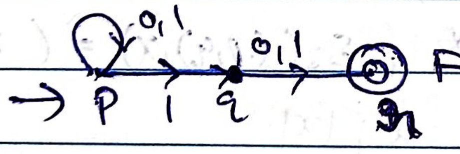
M accepts A.

—×—×—×—×—

24/8/22 Non-deterministic finite automata (NFA):

$\forall$ NFA N with $A = L(N)$, $\exists$ a DFA M s.t. $L(M) = A$.
(& both N & M has same alphabet Σ)

Ex:–



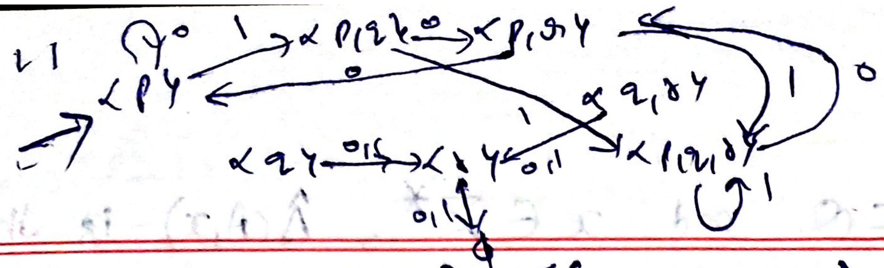
$L(N) = \{x \mid$ second symbol from right is $1\}$

Claim: The following ~~NFA~~ M has same regular set as N

	0	1
ϕ	ϕ	ϕ
$\to \{p\}$	$\{p\}$	$\{p,q\}$
$\{q\}$	$\{r\}$	$\{r\}$
F: $\{r\}$	ϕ	ϕ
$\{p,q\}$	$\{p,r\}$	$\{p,q,r\}$
F: $\{p,r\}$	$\{p\}$	$\{p,q\}$
F: $\{p,r\}$	$\{p\}$	$\{p,q,r\}$
F: $\{q,r\}$	$\{r\}$	$\{r\}$
F: $\{p,q,r\}$	$\{p,r\}$	$\{p,q,r\}$

States of M:
$P(\{p,q,r\})$

Rule:– $T_1 \subseteq \{p,q,r\}$ $T_1 \xrightarrow{\alpha} T_2$
where $\alpha \in \{0,1\}$ and
$T_2 \subseteq \{p,q,r\}$ if $\forall r_2 \in T_2$
$\exists r_1 \in T_1$ s.t. $r_1 \xrightarrow{\alpha} r_2$



$L(N) \subseteq L(M)$ ✓ (Check easily).

Def^n — An NFA is a 5-tuple $N = (Q, \Sigma, \Delta, S, F)$

Q : States

S : set of states that are start states

Σ : Alphabet from where the strings are taken

Transition function

$\Delta : Q \times \Sigma \longrightarrow P(Q)$

Intuitively $\Delta(p, a)$ is the set of all possible states where p can go if it gets 'a' as input

[Notes: $p \xrightarrow{a} q$, if $q \in \Delta(p, a)$]

$\Delta(p, a)$ can be empty set.

Acceptance & Transition in NFA

$\delta \rightsquigarrow \hat{\delta}$ DFA

$\Delta \rightsquigarrow \hat{\Delta}$ NFA $= (Q, \Sigma, \Delta, S, F)$

Def^n: $\hat{\Delta} : P(Q) \times \Sigma \longrightarrow P(Q)$

$$\hat{\Delta}(A, \varepsilon) = A$$

$$\hat{\Delta}(A, xa) = \Delta(\hat{\Delta}(A, x), a)$$

$$= \bigcup_{q \in \hat{\Delta}(A, x)} \Delta(q, a)$$

(where $\Delta : Q \times \Sigma \rightarrow P(Q)$)

apsara

Date: __.__.__

Remark: $\forall A \subseteq Q$ and $x \in \Sigma^*$, $\hat{\Delta}(A,x)$ is the set of all states reachable under input string x from some state in A.

If $a \in \Sigma$

$$\hat{\Delta}(A,a) = \hat{\Delta}(A,\varepsilon a)$$

$$\overset{\text{by defn}}{=} \bigcup_{q \in \hat{\Delta}(A,\varepsilon) = A} (\Delta(q,a)) = \bigcup_{q \in A} \Delta(q,a)$$

Defn:- The automation N is said to accept $x \in \Sigma^*$ if $\hat{\Delta}(S,x) \cap F \neq \phi$.

$L(N) = \{ x \in \Sigma^* \mid N \text{ accepts } x \}$

Remark: Every DFA $(Q, \Sigma, \delta, s, F)$ is equivalent to an NFA $(Q, \Sigma, \Delta, \{s\}, F)$

Lemma 1: $\forall\ x, y \in \Sigma^*$ & $A \subseteq Q$ $\hat{\Delta}(A, xy) = \hat{\Delta}(\hat{\Delta}(A,x), y)$

Pf:- Induction on $|y|$

$|y| = 0, 1$

$|y| = 0$

$$\hat{\Delta}(\hat{\Delta}(A,x), y) = \hat{\Delta}(\hat{\Delta}(A,x), \varepsilon)$$

$$= \hat{\Delta}(A,x)$$

$$= \hat{\Delta}(A, x\varepsilon)$$

$$= \hat{\Delta}(A, xy)$$

$|y| = 1$, $y = a \in \Sigma$

$$\hat{\Delta}(A, xa) \stackrel{\text{defn of } \hat{\Delta}}{=} \bigcup_{q \in \hat{\Delta}(A,x)} \Delta(q, a) \quad \text{--- } (*)$$

$$\hat{\Delta}(\hat{\Delta}(A,x), y) = \hat{\Delta}(\hat{\Delta}(A,x), a) \stackrel{\text{by defn}}{=} \bigcup_{q \in \hat{\Delta}(\hat{\Delta}(A,x), \varepsilon)} \Delta(q, a)$$

$$\hat{\Delta}(\hat{\Delta}(A,x), \varepsilon) = \hat{\Delta}(A,x)$$

$$= \bigcup_{q \in \hat{\Delta}(A,x)} \Delta(q, a) \quad \text{--- } (\#)$$

$$(*) = (\#)$$

So $\hat{\Delta}(A, xy) = \hat{\Delta}(\hat{\Delta}(A,x), y)$ if $|y| = 1$

Let the result be true upto $|y|$. To show the same for xya where $a \in \Sigma$

$$\hat{\Delta}(A, xya) \stackrel{\text{defn}}{=} \bigcup_{q \in \hat{\Delta}(A, xy)} \Delta(q, a)$$

$$\stackrel{\text{Induction}}{=} \bigcup_{q \in \hat{\Delta}(\underbrace{\hat{\Delta}(A,x)}_{B}, y)} \Delta(q, a)$$

$$\stackrel{\text{by defn of } \hat{\Delta}}{=} \hat{\Delta}(\underbrace{\hat{\Delta}(A,x)}_{B}, ya)$$

QED

Lemma 2: The function $\hat{\Delta}$ commutes with set union.

i.e.,

$$\hat{\Delta}\left(\bigcup_i A_i, x\right) = \bigcup_i \hat{\Delta}(A_i, x)$$

Pf Induction

$x = \varepsilon$ $\quad \hat{\Delta}(\bigcup_i A_i, \varepsilon) = \bigcup_i A_i = \bigcup_i \hat{\Delta}(A_i, \varepsilon)$

Date:

$$\hat{\Delta}(\bigcup_i A_i, xa) = \bigcup_{q \in \hat{\Delta}(\cup A_i, x)} \Delta(q, a) = \bigcup_{q \in \bigcup_i \hat{\Delta}(A_i, x)} \Delta(q, a)$$

(Induction)

$$= \bigcup_i \bigcup_{q \in \hat{\Delta}(A_i, x)} \Delta(q, a) = \bigcup_i \hat{\Delta}(A_i, xa)$$

The Subset Construction

NFA: $N = (Q_N, \Sigma, \Delta_N, S_N, F_N)$

Let $M = (Q_M, \Sigma, \delta_M, S_M, F_M)$

be the following,

$$\delta_M(A, a) \overset{\text{defn}}{=} \hat{\Delta}_N(A, a) \quad \forall A \subseteq Q_N$$

$$S_M = S_N$$

$$F_M \overset{\text{defn}}{=} \{A \subseteq Q_N \mid A \cap F_N \neq \phi\}$$

Thm: M & N accepts the same set.

To prove this we need a final lemma.

Lemma 3 : $\forall A \subseteq Q_N, x \in \Sigma^*, \hat{\delta}_M(A, x) = \hat{\Delta}(A, x)$

Pf $\forall x \in \Sigma^*, \quad x \in L(M) \iff \hat{\delta}_M(S_M, x) \in F_M$

$$\underset{\text{Lemma 3}}{\iff} \hat{\Delta}_N(S_N, x) \cap F_N \neq \phi$$

$$\iff x \in L(N).$$

Induction $x = \varepsilon$ $\quad \hat{\delta}_M(A, \varepsilon) = \hat{\Delta}_N(A, \varepsilon) = A$

Assume $\hat{\delta}_M(A, x) = \hat{\Delta}_N(A, x)$

$$\hat{\delta}_M(A, xa) = \delta_M(\hat{\delta}_M(A, x), a) = \delta_M(\hat{\Delta}_N(A, x), a)$$

$$\overset{\text{defn}}{=} \hat{\Delta}_N(\hat{\Delta}_N(A, x), a) = \hat{\Delta}_N(A, x)$$

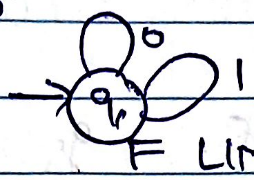
QED

$A = L(M)$

26/8/22

Q.1 Are all subsets of Σ^* regular? — No

Q.2 Is Σ^* itself regular? — Yes



$L(M) = \Sigma^*$

Q. Is $A = \{0^n 1^n \mid n \in \mathbb{N}\} = \{\varepsilon, 01, 0011, 000111, \dots\}$ regular?

$\mathbb{N} = \{0, 1, 2, 3, \dots\}$
$\mathbb{N}_+ = \{1, 2, 3, \dots\}$

Ans No. Pf by contradiction. Let $\exists$ DFA M s.t. $L(M) = A$. Let M has K states.

Take $n >> K$.

Then $\exists$ a state p s.t. the string $\underbrace{0\dots0}_{n\text{-times}}$ visits p twice (Pigeon hole)

$0^n 1^n \equiv 0^x 0^y 0^{n-x-y} 1^n$

$$\underbrace{\underbrace{0\dots0}_{x}\;\underbrace{0\dots0}_{y}\;0\dots0}_{n \text{ times}}\;\underbrace{1\dots1}_{n\text{ times}}$$

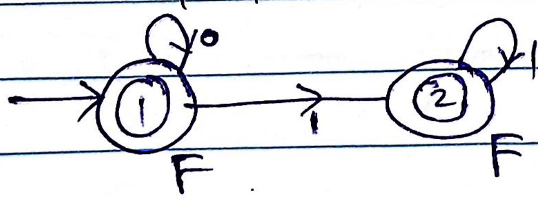
$p \quad p \quad \omega$

$|y| << n$

Then M also accepts $0^x 0^{2y} 0^{n-x-y} 1^n$
$= 0^{n+y-1} 1^n$

but as $|y| \geq 2$ $n+y-1 \neq n$

Q. Is $\{0^n 1^m \mid n, m \geq 0\}$ regular? — Yes



Q. Is $\{0^n 1^m \mid n \geq m\}$ regular? — No

Ans Take $m >> K$ $\quad 0\dots0\underbrace{1\dots1}_{x}\underbrace{1\dots1}_{y}\underbrace{1\dots1}_{z}$

$x + y + z = m$ $\quad$ $p \quad p$ (By Pigeon hole)

$0^n 1^m = 0^n 1^x 1^y 1^z$ get accepted where $|y| \geq 2$

But then $\forall\ i \geq 1$

$0^n 1^x 1^y (1^{y-1})^i 1^z$ as also accepted.

choose i s.t. $x+y+iy-i+z > n$ ✓.

Q: Is $A = \{0^{2^n} \mid n \geq 0\}$ regular? — NO

$n >> 0$

$0 \ldots 0 \ldots 0 \ldots 0 \ldots (2^n \text{ times})$

$\underbrace{p \quad p}_{k}$

$0^{2^n} \cdot 0^k \in L$ (Contradiction)

$2^n < 2^n + k \underset{\neq}{\leq} 2^{n+1}$

$\uparrow \mid k \underset{\neq}{\leq} 2^n$

$0^{2^{k+1}} = 0^{2 \cdot 2^k} = 0^{2^k} \cdot 0^{2^k}$

Thm (Pumping Lemma): Let A be a regular set then the following holds: $\exists\ k \geq 0$ s.t. for any strings x, y, z with $xyz \in A$ & $|y| \geq k$ $\exists$ strings u, v, w with $y = uvw$, $v \neq \varepsilon$ & $\forall\ i \geq 0$ the string $xuv^iwz \in A$.

└ Pf - Exercise. (NEXT PAGE)