7/2/23

FLAT

FORMAL LANGUAGES AND AUTOMATA THEORY

Alphabet : letters, digits, symbols (everything on keyboard)
: $\Sigma = \{0, 1\}$ (input symbols)
: symbol used to solve problem.

String : $\{0, 1, 00, 01, 10, \ldots\ldots\}$

Language : L = string ends with 0

*	+
Kleene's closure (0 or more occurence)	positive closure (1 or more occurence)

$L^* = \{\epsilon, 0, 1, \ldots\ldots\}$ $\qquad$ $L^+ = \{0, 1, \ldots\ldots\}$

↓ string doesn't exist

$L_1 = \{ab\}$ $\qquad$ $L_2 = \{ba\}$

→ Union : $L_1 \cup L_2 = \{abba, baab, ab, ba\}$

→ Concatenation : $L_1 \cdot L_2 = \{abba\}$

↓ preference

$L' = \{ab\} \Rightarrow L^2 = \{abab\}$

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

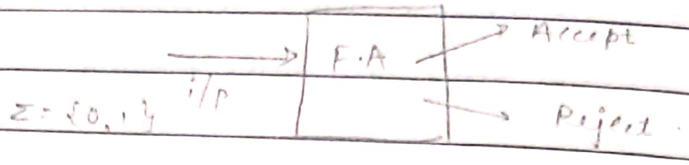
Machine Table : MT

PAGE No.
DATE / /

7/2/23

FSM / FA

i.e. finite state machine / finite automata.



PROBLEMS

→ Type I : ends with

eg: 1] Design FSM for the strings ending with "101" over the input $\Sigma = \{0, 1\}$

Keyword

soln :-

Step 1] : Definition of FSM. (It is defined using this 5 tuples)

$q_0 = \{q_0\}$

$F = \{q_{101}\}$

$Q = \{q_0, q_1, q_{10}, q_{101}\}$

$F.A = q_0, \Sigma, F, \delta, Q$ → total state

initial state, input, final state, transition

Step 2] : Logic :

→ initial state

q_0 ending with '0'

q_1 ending with '1'

q_{10} ending with '10'

* q_{101} ending with '101'

final state

Total state = Q = 4

Scanned with CamScanner

PAGE No. DATE / /

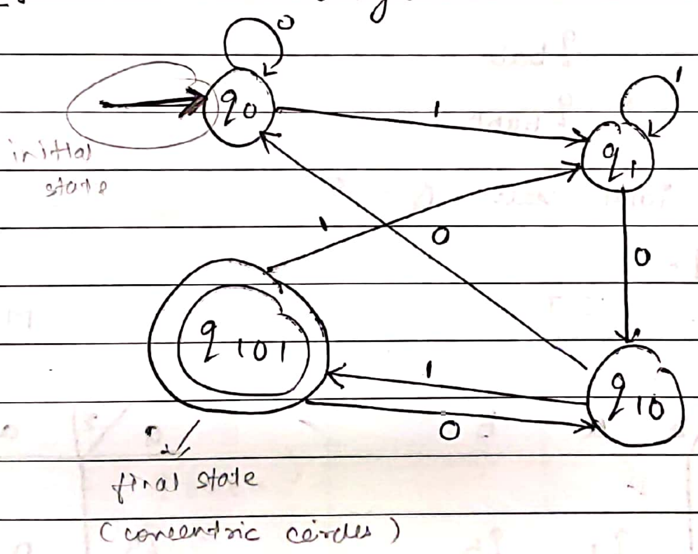
step 3] :-

S.T	M.T (FSM)
state table	Machine table

$Q \backslash \Sigma$	0	1
q_0	q_0	q_1
q_1	q_{10}	q_1
q_{10}	q_0	q_{101}
$^{*}q_{101}$	q_{10}	q_1

$Q \backslash \Sigma$	0	1
q_0	n	n
q_1	n	n
q_{10}	n	y
$^{*}q_{101}$	n	n

step 4] :- Transition diagram



step 5] :-

string :- 110101

$\delta(q_0, 110101)$

$\delta(q_1, 10101)$

$\delta(q_1, 0101)$

$\delta(q_{10}, 101)$

$\delta(q_{101}, 01)$

$\delta(q_{10}, 1)$

$\delta(q_{101})$

Accept

eg: 2] ends with "babb" & $\Sigma = \{a, b\}$

step 1] :- $q_0 : \{q_0\}$

$F : \{q_{babb}\}$

$Q = \{q_a, q_b, q_{ba}, q_{bab}, {}^{*}q_{babb}\}$

step 2] :- Logic :-

$\rightarrow q_a$

q_b

q_{ba}

q_{bab}

$* \; q_{babb}$

Total state : Q = 5

step 3] :-

S.T

Q \ Σ	a	b
q_a	q_a	q_b
q_b	q_{ba}	q_b
q_{ba}	q_a	q_{bab}
q_{bab}	q_{ba}	q_{babb}
$*q_{babb}$	q_{ba}	q_b

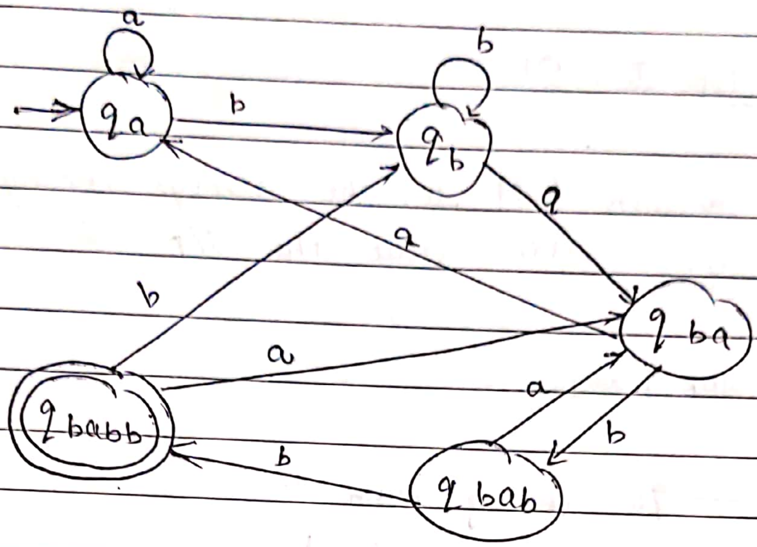
M.T

Q \ Σ	a	b
$\rightarrow q_a$	n	n
q_b	n	n
q_{ba}	n	n
q_{bab}	n	y
$* q_{bab}$	n	n

PAGE No

DATE / /

step 4]



step 5]

string: ~~ab~~ab babb

δ (qa , abbabb)

δ (qa , bbabb)

δ (qb , babb)

δ (qb , abb)

δ (qba , (bb))

δ (qbab , b)

δ (qbabb)

9/2/23

PAGE No.
DATE / /

* Type II : OR

eg 1]: Design FSM for the strings ending with "101" Or "110" over the i/p $\Sigma = \{0,1\}$

→

step 2] Logic :-

→ q_0 ending with 0
q_1 ending with 1
q_{10} ending with "10"
q_{11} ending with "11"
* q_{101} — " — "101"
* q_{110} — " — "110"

step 3] Transition table :-

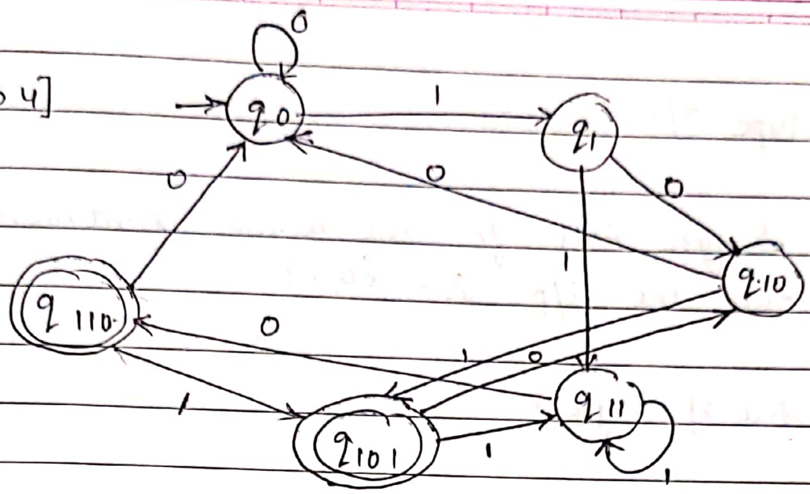
S.T.

Q \ Σ	0	1
→ q_0	q_0	q_1
q_1	q_{10}	q_{11}
q_{10}	q_0	q_{101}
q_{11}	q_{110}	q_{11}
* q_{101}	q_{10}	q_{11}
* q_{110}	q_0	q_{101}

M.T.

Q \ Σ	0	1
q_0	n	n
q_1	n	n
q_{10}	n	y
q_{11}	y	n
* q_{101}	n	n
* q_{110}	n	y

step 4]



step 5]

String :- 10101

$\delta(q_0, 10101)$

$\delta(q_1, 0101)$

$\delta(q_{10}, 101)$

$\delta(q_{101}, 01)$

$\delta(q_{10}, 1)$

$\delta(q_{101})$

Accept //

H.W

eg 2] ending with "aba" or "bab"

done backside.

* Type 3]:- Contains.

eg 1] Design FSM for the string containing "1101" over the i/p $\Sigma = \{0, 1\}$

step 2] Logic :-

→ q_0	contains	'0'
q_1	contains	'1'
q_{11}	—	'11'
q_{110}	—	'110'
* q_{1101}	—	'1101'

step 3] Transition table :-

S.T.

Q \ Σ	0	1
→ q_0	q_0	q_1
q_1	q_0	q_{11}
q_{11}	q_{110}	q_{11}
q_{110}	q_0	q_{1101}
* q_{1101}	q_{1101}	q_{1101}

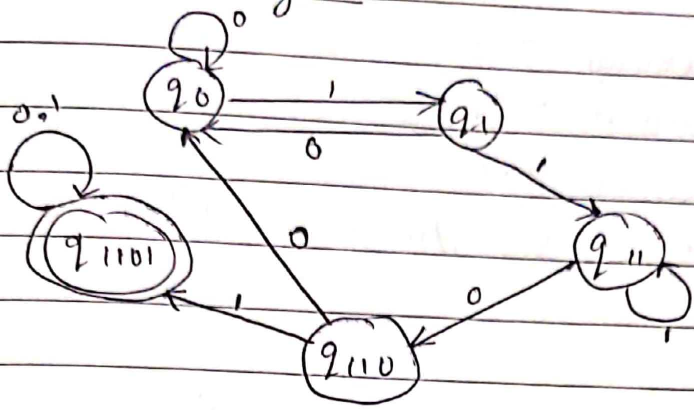
same state.

M.T.

Q \ Σ	0	1
→ q_0	n	n
q_1	n	n
q_{11}	n	n
q_{110}	n	y
* q_{1101}	y	y

so! no transitions!
remain in same state

step 4] Transition diagram :



step 5] String : 1101

$\delta(q_0, 1101)$
$\delta(q_1, 101)$
$\delta(q_{11}, 01)$
$\delta(q_{110}, 1)$
$\delta(q_{1101})$

Accept //

* Type 4] : Does not contain.

First you have to follow whole ~~pro~~ procedure upti! step 3 of 'Contain'.
string which is not contained should not be Final state other than that every states becomes final state.
Even initial state is final state.

suppose eg 1) of Type 3]

step 3] M.T. :

Q \ Σ	0	1
* q_0	y	y
* q_1	y	y
* q_{11}	y	y
* q_{110}	y	n
q_{1101}	n	n

S.T remains same which q_{1101} [illegible] state final [illegible]

PAGE No.
DATE / /

* ~~Typ~~
eg: Contains exactly 2a's
atleast 2a's
atmost 2a's
over the i/p $\Sigma = \{a, b\}$

Type II OR
eg 2] ending with "aba" or "bab"

step 2] Logic :-

if 2 final state then 2 initial state

q_a
q_b
q_{ab}
* q_{aba}
q_{ba}
* q_{bab}

step 3] Transition table :-

S.T.

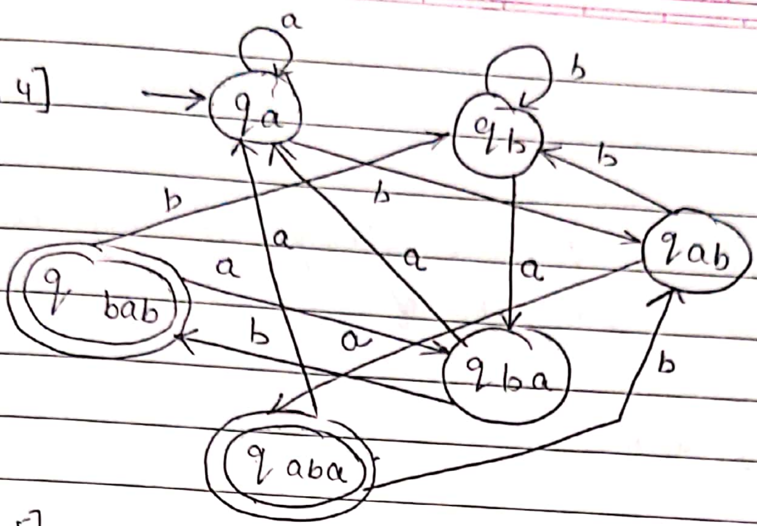
Q \ Σ	a	b
q_a	q_a	q_{ab}
q_b	q_{ba}	q_b
q_{ab}	q_{aba}	q_b
q_{ba}	q_a	q_{bab}
* q_{aba}	q_a	q_{ab}
* q_{bab}	q_{ba}	q_b

M.T.

Q \ Σ	a	b
q_a	n	n
q_b	n	n
q_{ab}	y	n
q_{ba}	n	y
* q_{aba}	n	n
* q_{bab}	n	n

Scanned with CamScanner

step 4]



step 5]

string : bbaba

$\delta(q_a, bbaba)$

$\delta(q_{ab}, baba)$

$\delta(q_b, aba)$

$\delta(q_{ba}, ba)$

$\delta(q_{bab}, a)$

$\delta(q_{aba})$

Accept //

13|2|23

Type V]: And

eg.1] Design FSM for the strings having even no. of 0's and odd no. of 1's.

step 2] Logic

q_s

	0	1
→ q	o	o
q	o	e
q	e	o
q	e	e

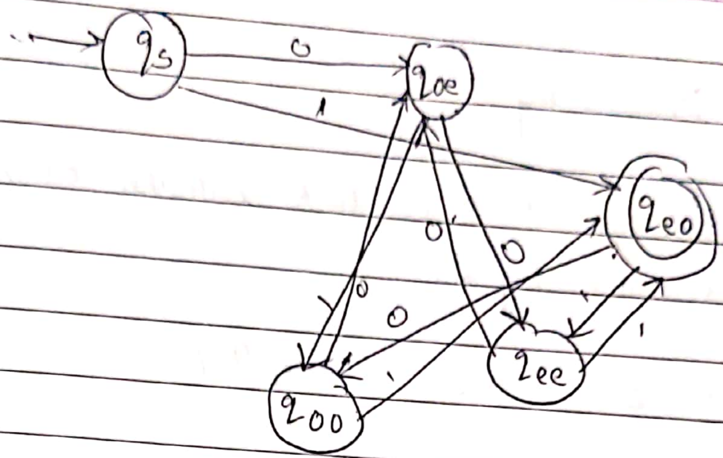
Step 3]

S.T.

Q \ Σ	0	1
→ q_s	q_{oe}	q_{eo}
q_{oe}	q_{ee}	q_{oo}
* q_{eo}	q_{oo}	q_{ee}
q_{ee}	q_{oe}	q_{eo}
q_{oo}	q_{eo}	q_{oe}

Q \ Σ	0	1
q_s	n	y
q_{oe}	n	n
q_{eo}	n	n
q_{ee}	n	y
q_{oo}	y	n

Step 4]



Step 5] String : 0101010

$\delta(q_s, 0101010)$

$\delta(q_{oe}, 101010)$

Type VI] :- Divisible by

eg 1] Design FSM to check whether decimal no. is divisible by 3.

soln :-

1) $\Sigma = \{0, 1, \ldots\ldots 9\}$

2] Logic :-

$q_0 \rightarrow$ remainder 0

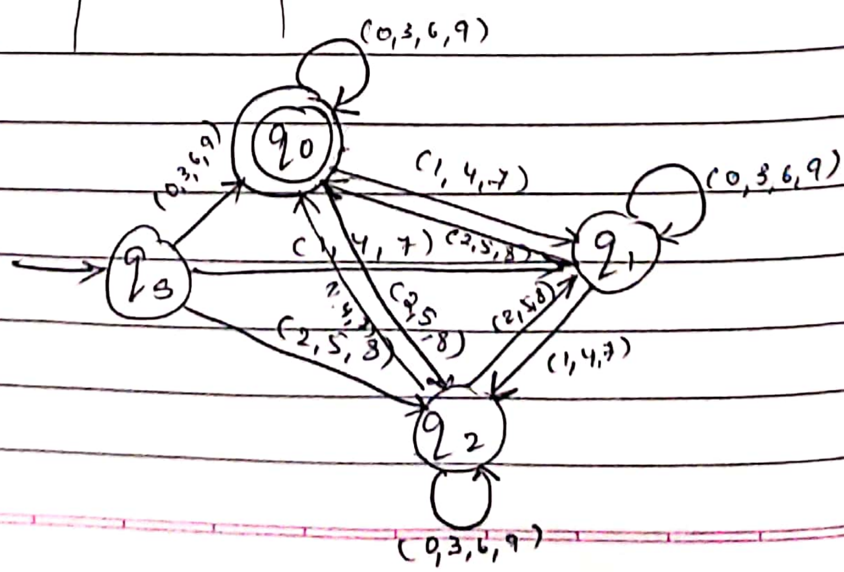
$q_1 \rightarrow$ — " — 1

$q_2 \rightarrow$ — " — 2

3] S.T.

$Q \backslash \Sigma$	{0,3,6,9}	{1,4,7}	{2,5,8}
q_s	q_0	q_1	q_2
* q_0	q_0	q_1	q_2
q_1	q_1	q_2	q_0
q_2	q_2	q_0	q_1

4]



5] eg: 234

$\delta(q_s, 234)$

$\delta(q_2, 34)$

$\delta(q_2, 4)$

$\delta(q_0)$

Accept

PAGE No.
DATE / /

pe. VII] Starts with

eg:- Design FSM for string starting with 3 consecutive a's over $\Sigma = \{a, b\}$.

soln:-

step 2] Logic :-

$\rightarrow q_s$ — string

q_a — string starting with 'a'

q_{aa} — 'aa'

q_{aaa}^* — 'aaa'

$q_D \rightarrow$ trap / dead state.

step 3]

S. T.

$Q \backslash \Sigma$	a	b
$\rightarrow q_s$	q_a	q_D
q_a	q_{aa}	q_D
q_{aa}	q_{aaa}	q_D
$*q_{aaa}$	q_{aaa}	q_{aaa}
q_D	q_D	q_D

2/23

PAGE NO.
DATE / /

Type VIII] :- Divisibility

eg :- Design FSM to check whether binary number is divisible by 2 or not.

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

2) Logic :-

$\rightarrow q_s$

$* q_0$ remainder '0'

q_1 — " — '1'

make list of steps based on remainder.

3)

S.T

$(2*R+0)\%2$ $(2*R+1)\%2$

$Q \backslash \Sigma$	0	1
$\rightarrow q_s$	q_0	q_1
q_0^*	q_0	q_1
q_1	q_0	q_1
~~q_2~~	~~q_1~~	~~q_2~~

~~divisible by~~ 3

$(3*R+0)\%3$ $(3*R+1)\%3$

$Q \backslash \Sigma$	0	1
$\rightarrow q_s$	q_0	q_1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

1011/3.

Num should ...

PAGE No.
DATE / /

Binary

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

Q \ Σ	0	1
$\rightarrow q_s$		
q_0		
q_1		
q_2		

ternary

$\Sigma = \{0, 1, 2\}$

Q \ Σ	0 $(3 * R + 0) \% 3$	1 $(3R + 1) \% 3$	2 $(3R + 2) \% 3$
$\rightarrow q_s$	q_0	q_1	q_2
q_0	q_0	q_1	q_2
q_1	q_0	q_1	q_2
q_2	q_0	q_1	q_2

→ Regular expression.

Reg. exp.	Reg. language.
r	$L(r)$
a	$L(r) = \{a\}$
$a \cdot b$ concatenation	$L(r) = \{ab\}$
$a+b$ or a/b (union, or)	$L(r) = \{a, b\}$
a^*	$L(r) = \{\epsilon, a, aa, \ldots\}$
a^+	$L(r) = \{a, aa, \ldots\}$
$* a^+ = a \cdot a^*$	$L(r) = \{a, aa, \ldots\}$
v.v. Imp → $\boxed{(a+b)^*}$	$L(r) = \{\epsilon, a, b, aa, bb, ab, ba, \ldots\}$

$*$ – 0 or more occurrence of that char.

$+$ – 1 or more occurrence