2/1/23

Automata theory

Mod 2

Basic machine

1) Finite state machines

Unlike basic machine, in FSM we are concerned with machine behaviour, regarding in which internal state machine is and how machine behaves upon receiving different intermediate inputs in the current state.

3 states $\rightarrow \underbrace{0, 1, 2}_{\text{input slgs}}$ Decimal nos - $0 \rightarrow 9$.

Q1. Construct a FSM for Binary adder.

Possible inputs 2 bits = $I = \{(0,0), (0,1), (1,0), (1,1)\}$
(0 & 1 combinations)

A finite set of states S: $S = \{\text{carry}, \text{no carry}\}$

Output $O = \{0, 1\}$

Machine Function MAF = $I \times S \rightarrow O$.

State Transition Function = $I \times S \rightarrow S$

MAF

I \ S	carry	No carry
(0,0)	1	0
(0,1)	0	1
(1,0)	0	1
(1,1)	1	0

STF

I \ S	carry	No carry
(0,0)	No carry	No carry
(0,1)	carry	No carry
(1,0)	carry	No carry
(1,1)	carry	carry

ex-2 Binary adder simulation for input 1011 + 1111

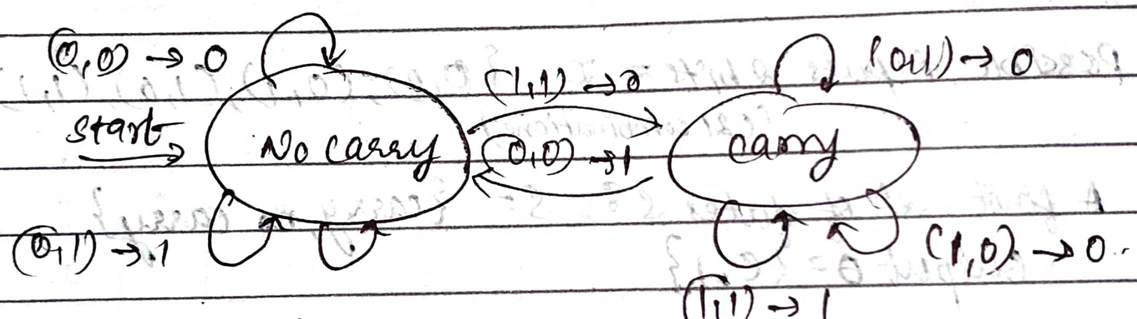
Current state	Input	Next state	output
1. No carry (0)	(1,1)	carry (1)	0
2. carry (1)	(1,1)	carry (1)	1
3. carry (1)	(0,1)	carry (1)	0
4. carry (1)	(1,1)	$\boxed{\text{carry (1)}}$	1

Final state

$\boxed{1010}$ → output

1011
1111
1010
carry
final o/p

Transition graph for binary adder



(1,0) → 1
↓ Input ↓ Output

Transition matrix

Current state \ Next state	No carry	Carry
No carry	(0,1)/1 v (0,0)/0 v (1,0)/1	(1,1)/0
carry	(0,0)/1	(0,1)/0 v (1,0)/0 v (1,1)/1

(Q) Design a FSM for divisibility by 3 tester for a decimal no.

$I = \{0,1,2,3,4,5,6,7,8,9\}$

$= \{(0,3,6,9),(1,4,7),(2,5,8)\}$ → Remainder Reace · divide kiy

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

q_0 → Remainder 0 state

q_1 → Remainder 1 state

q_2 → Remainder 2 state

$O = \{\cancel{1}, 0\}$

1 → div. by 3 ;

0 → Not div. by 3

$\{1, 0\} \xrightarrow{\text{becomes zero}} \{0, 1, 2\}$ Remainder

MAPS = I × S → O

I \ S	q_0	q_1	q_2
(0,3,6,9)	0	1	2
(1,4,7)	1	2	0
(2,5,8)	2	0	1

STP = I × S → S

I \ S	q_0	q_1	q_2
(0,3,6,9)	q_0	q_1	q_2
(1,4,7)	q_1	q_2	q_0
(2,5,8)	q_2	q_0	q_1

HW Transition graph & matrix

Page No.
Date

Q) Design FSM for divisibility by 3 tester for a binary no.

→

$I = \{0, 1\}$
$S = \{q_0, q_1, q_2\}$
$O = \{0, 1\}$

MAF | STF

I \ S	q_0	q_1	q_2
0 (000, 010, 100)	0	(2) (10) Binary form	01 (1)
1 (100, 011, 101)	01 (1)	00 (0)	10 (2)

I \ S	q_0	q_1	q_2
0	q_0	q_2	q_1
1	q_1	q_0	q_2

(Q) Design FSM for divisibility by 5 tester for a decimal no.

$I = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
$I = \{(0, 5) [(1, 6), (2, 7), (3, 8), (4, 9)]\}$
$S = \{q_0, q_1, \cancel{q_2, q_3, q_4}\}$
$O = \{0, 1, 2, 3, 4\}$
(zero state) (non-zero state)

MAP

I \ S	q_0	q_1
(0,5)	0	0
(1,2,3,4,6,7,8)	(1,2,3,4)	(1,2,3,4)

STF

I \ S	q_0	q_1
(0,5)	q_0	q_0
(1,2,3,4,6,7,8)	q_1	q_1

FSM

- Unary } by 3
- Ternary

17/1/23

AT

(Q) Divisibility by 3 tester for a ternary no.

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

$O = \{0, 1\}$ 0 = div by 3, 1 = not div-by 3

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

MAF $I \times S \to O$

I \ S	q_0	q_1	q_2
0	1	1	1
1	0	0	0
2	0	0	0

0, 1, 2 in ternary form in decimal

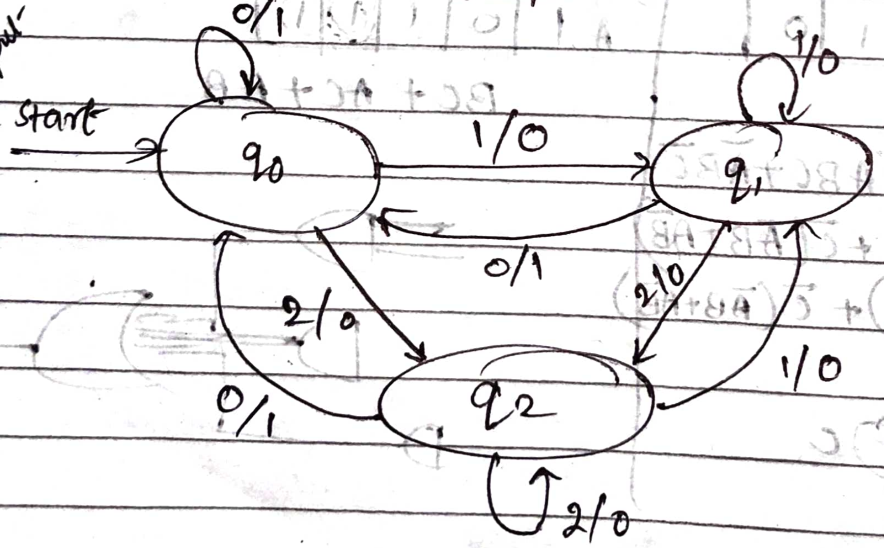
STF $I \times S \to S$

I \ S	q_0	q_1	q_2
0	q_0	q_0	q_0
1	q_1	q_1	q_1
2	q_2	q_2	q_2

10 in ternary
3 in decimal
11 in ternary
4 in decimal

Transition graph :

left | right → output
↓
state nos.



SDS Page No.
Date / /

$(210)_3 \rightarrow (21)_{10}$ (Decimal)

Current State	Current i/p symbol	Next state	o/p symbol
q_0	2	q_2	0
q_2	1	q_1	0
q_1	0	q_0	1

★ Finite Automata

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

$Q \rightarrow$ finite set of states for the machine

$\Sigma \rightarrow$ finite input alphabet

$\delta \rightarrow$ Transition function that maps $Q \times \Sigma \rightarrow Q$

$q_0 \rightarrow$ initial state for machine

$F \rightarrow$ finite state of machine

(Q) Design FA that reads the strings made up of letters in the word "CHARIOT" and recognizes those strings that contain the word "CAT" as a sub-string.

$\rightarrow$ $FA = \{Q, \Sigma, \delta, q_0, F\}$

$\Sigma = \{C, H, A, R, I, O, T\}$

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

q_s → initial state

q_0 → remembering string ending with C

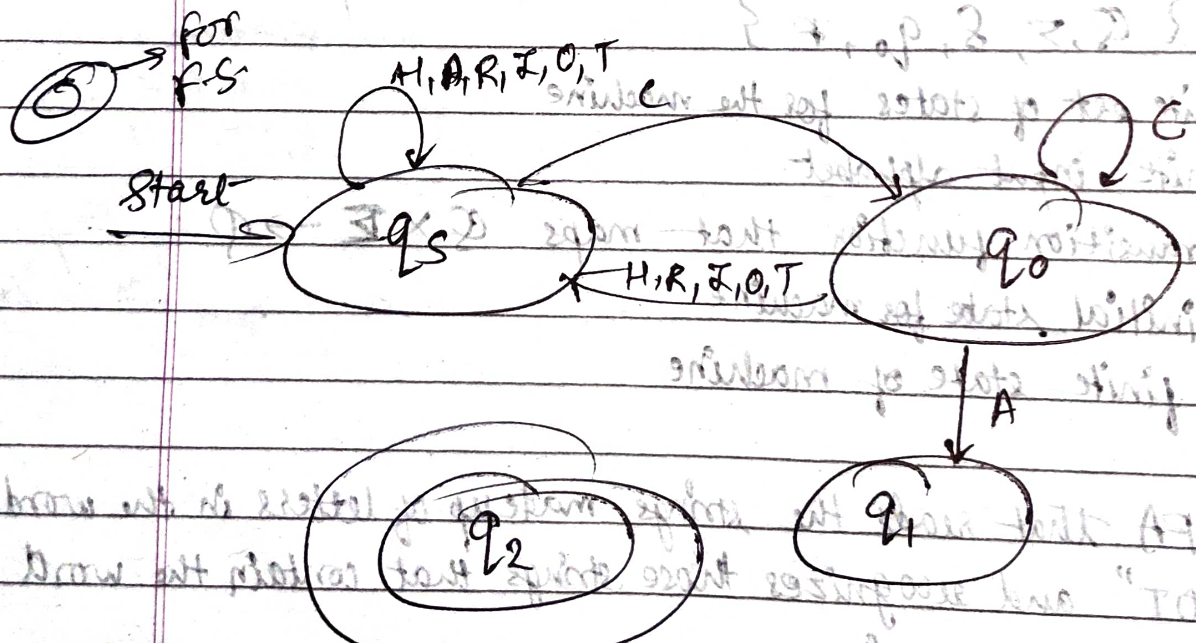
q_1 → ending with CA

q_2 → ending with CAT

	$Q \backslash \Sigma$	C	H	A	R	I	O	T
	q_s	q_0	q_s	q_s	q_s	q_s	q_s	q_s
end. C	q_0	q_0	q_s	q_1	q_0	q_s	q_s	q_s
end CA	q_1	q_0	q_s	q_s	q_s	q_s	q_s	q_2
end CAT	q_2	q_2	q_2	q_2	q_2	q_2	q_2	q_2

$I.S = q_s$

$F.S = \{q_2\}$



11/1/23

AT

SDS Page No.
Date / /

(Q) Design a FA that have all the strings over $\Sigma = \{0,1\}$ not containing 010 as a substring.

→ $FA = \{Q, \Sigma, \delta, q_0, F\}$

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

$Q = \{q_s, q_1, q_0, q_{01}, q_{010}\}$

(q_0 → remember state read 0; q_{010} → 010)

states acc. to which strings you need to remember.

Compulsory state acc. to

Better to use 1 state to remember 1 string ending with read.

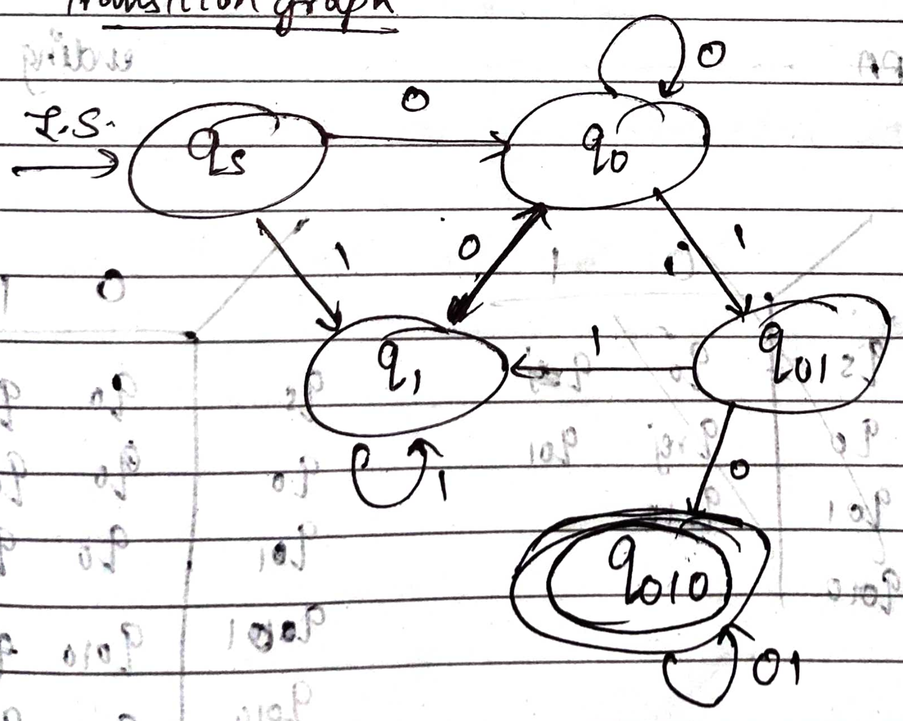
Initial state $= q_s$

$F = \{q_{010}\}$

$\delta = Q \times \Sigma \to Q$

Σ	0	1
q_s	q_0	q_1
q_0	q_0	q_{01}
q_1	q_0	q_1 (or q_s)
q_{01}	q_{010}	q_1
q_{010}	q_{010}	q_{010}

Transition graph



Remains in final state → accepted by machine

Do not " → rejected.

TA

Q) Design FA to accept all the strings over $\Sigma = \{0, 1\}$ beginning with 010.

→ $FA = \{Q, \Sigma, \delta, q_0, F\}$

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

$Q = \{q_s, q_0, q_1, q_{01}, q_{010}, q_{rej}\}$

$I.S = q_s$

$F.S = \{q_{010}\}$

	0	1
q_s	q_0	q_{rej}
q_0	q_{rej}	q_{01}
q_{01}	q_{010}	q_{rej}
q_{010}	q_{010}	q_{010}

Q) Design FA . . . . . . ending with 010.

	0	1
~~q_s~~	~~q_0~~	~~q_{rej}~~
~~q_0~~	~~q_{rej}~~	~~q_{01}~~
~~q_{01}~~	~~q_{010}~~	
~~q_{010}~~		

	0	1
q_s	q_0	q_1
q_0	q_0	q_{01}
q_{01}	q_0	q_1
q_{001}	q_{010}	q_1
q_{010}	q_0	q_{01}

F.S as N.F.S if we have to ~~take not a substring~~

(Q) Design FA to accept all the strings over $\Sigma = \{a,b\}$ containing exactly 3 a's

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

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

$Q = \{q_0, q_1, q_2, q_3, q_4\}$ (1a, 2a, 3a, for more than 3 a's.)

$I.S = \{q_0\}$

$F = \{q_3\}$

Q \ Σ	a	b
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

OR

Q \ Σ	a	b
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_2
* q_3	q_3	q_3

* → final state.

atleast 3 a's me (q_3, q_4) → final states all other NFS.

atmost 3 a's me yehi table.

Q) Design a FA over $\Sigma = \{0,1\}$ containing all the strings ending in 00 or 11 (all other strings will be rejected)

→ $FA = \{Q, \Sigma, \delta, q_0, F\}$

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

$Q = \{q_s, q_0, q_1, q_{00}, q_{11}\}$

I.S = q_s

$F = \{q_{00}, q_{11}\}$

Q \ Σ	0	1
q_s	q_0	q_1
q_0	q_{00}	q_1
q_1	q_0	q_{11}
q_{00}	q_{00}	q_1
q_{11}	q_0	q_{11}

not ending in 00 & 11

(Q) Design a FA for language over $\Sigma = \{a\}$ containing the string which contains number of a's as multiple of 3.

20/1/23

AT

Types of FA

DFA NFA

* FA is said to be 'deterministic' if it does not contain more than one transitions on the same input symbol from the same state.

* FA is said to 'non-deterministic' if it contains atleast one state from which there are more than one transitions on the same input symbols.

* Every NFA can be converted to its equivalent DFA.

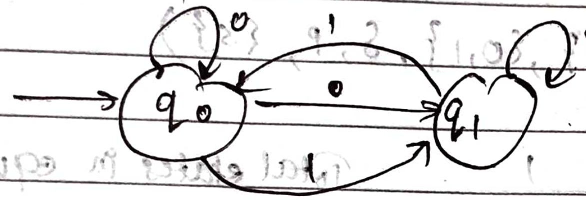
* NFA - DFA conversion

* Given NFA $M = (Q, \Sigma, \delta, q_0, F)$ can be converted to its equivalent DFA $M' = (Q', \Sigma, \delta', q_0, F')$ where
  - → Q' is finite set of states for power set of Q (2^Q) except the null set (ϕ).
  - → δ' is a transition function that maps $Q' \times \Sigma \rightarrow 2^Q$ except null set (ϕ).
  - → F' is a set of final states from the set that belong to Q' containing atleast one final state of given NFA.

eg). Convert foll. NFA to DFA.

NFA, $M = (\{q_0, q_1\}, \{0,1\}, \delta, q_0, \{q_1\})$
where

δ Q \ Σ	0	1
q_0	$\{q_0, q_1\}$	q_1
q_1	ϕ	$\{q_0, q_1\}$



NFA.

$Q' = \{[q_0], [q_1], [q_0, q_1]\}$
$F' = \{[q_1], [q_0, q_1]\}$
$\delta' = Q' \times \Sigma \rightarrow Q'$

$\delta' = Q' \times \Sigma \rightarrow Q'$

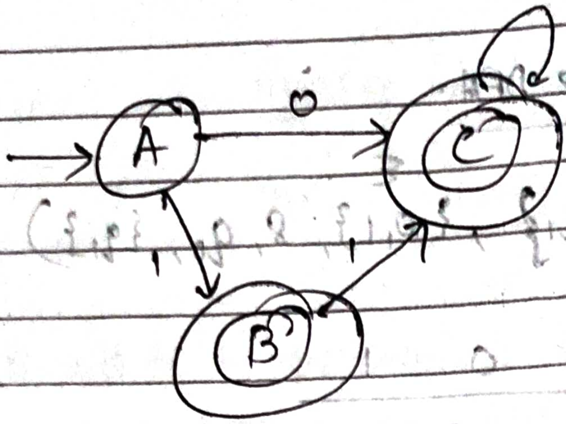
Q \ Σ	0	1
A $[q_0]$	$[q_0, q_1]$	$[q_1]$
B $[q_1]$	–	$[q_0, q_1]$
C $[q_0, q_1]$	$[q_0, q_1]$	$[q_0, q_1]$

$\delta'([q_0], 0) = \delta(q_0, 0)$
$= [q_0, q_1]$

$\delta'([q_0], 1] = [q_1]$

$\delta'([q_0, q_1], 0) = \delta(q_0, 0) \cup \delta(q_1, 0)$
$= \{q_0, q$

[ ] → one transition.



(Q) NFA $(\{p,q,r,s\}, \{0,1\}, \delta, p, \{s\})$

1st method

δ \ Σ	0	1
p	p,q	p
q	r	r
r	s	–
s	s	s

Total states in equi. DFA $= 2^Q = 2^4$
$= 16$ including ϕ

Here we are not considering ϕ
'x' states $= 15$

$Q' = \{p, q, r, s, pq, pr, ps, qs, qr, rs, pqr, pqs, qrs, prs, pqrs\}$

$F' = \{s, ps, qs, rs, pqs, qrs, prs, pqrs\}$

$\delta' : Q' \times \Sigma \to Q'$

Q \ Σ	0	1
p	pq	p
q	r	r
r	s	–
* s	s	s
pq	pqr	pr
pr	pqs ⇒ ps	p
* ps	pqs → ps	ps
* qs	rs	rs
qr	rs → s (replace)	r
* rs	s	s
pqr	pqrs → pqs (ps)	pr
* pqs	pqs	prs
* qrs	rs	rs
* prs	pqs	ps
* pqrs	pqrs	prs → ps

transition on i/p s/g on both s/g should be same → equivalent states.

qs = qrs (equivalent)

s = rs (equivalent)

ps = prs (equi.)

1. Draw table & remove states.
2. Draw T.G.

pqs = pqrs (equi.)

Unreachable state

dead state

ps = pqs

Q' \ Σ	0	1
p	pq	p
q	r	r
r	s	–
* s	s	s
pq	pr	pr
pr	ps	pr
* ps	ps	ps
qr	s	r
~~pqr~~	~~ps~~	~~pr~~

24/1/23

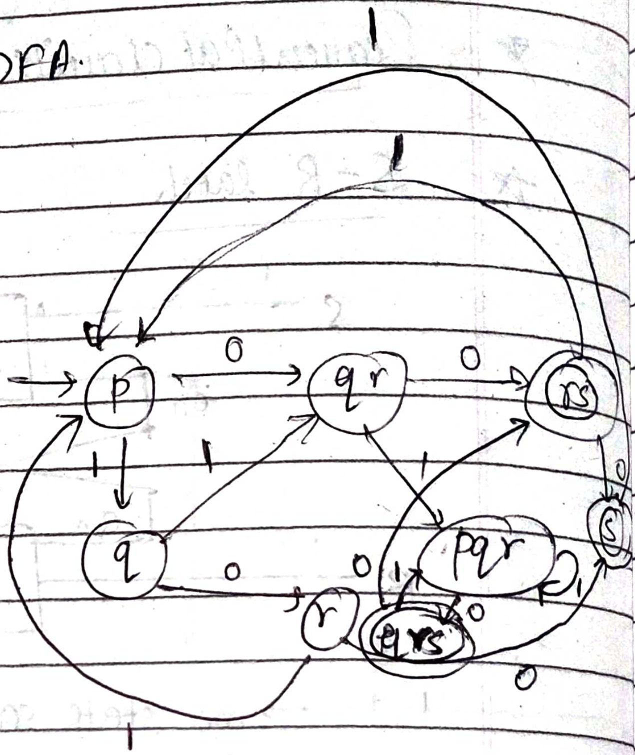
AT

(Q) Convert foll. NFA to equivalent DFA.

NFA = ({p, q, r, s}, {0, 1}

Q \ Σ	0	1
I.S ← p	q, r	q
q	r	q, r
r	s	p
* s	–	p

2nd method



Q \ Σ	0	1
p	qr	q
q	r	qr
r	s	p
* s	–	p
qr	rs	pqr
* rs	s	p
pqr	qrs	pqr
* qrs	rs	pqr

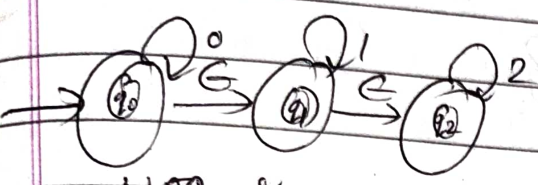
Both should be same states.

- ϵ

* NFA with empty transitions /moves

$$A \xrightarrow{\epsilon} B$$

$0^* 1^* 2^*$



NFA with empty moves

Only NFA can have empty transition.
DFA → only one transition.
↳ can never have empty transitions

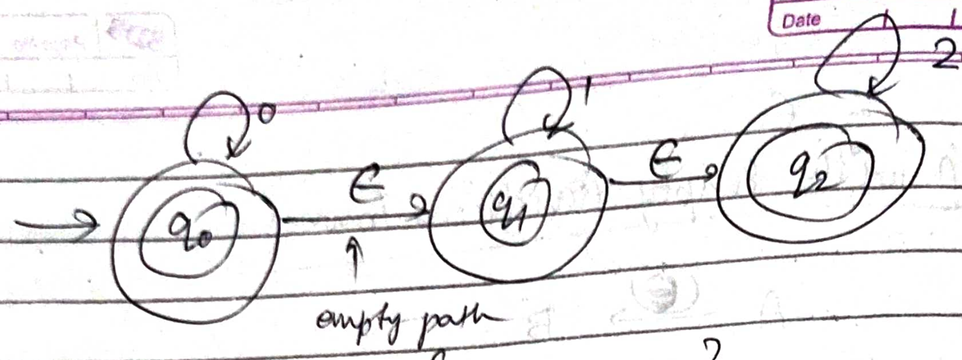
- Every NFA with empty moves can be converted to equi. NFA.

- NFA - ϵ : $\delta : Q \times \Sigma \cup \{\phi\} \longrightarrow 2^Q$

- The NFA with empty moves $M = (Q, \Sigma, \delta, q_0, F)$, where $\delta : Q \times \Sigma \cup \{\phi\} \to 2^Q$, can be converted to its equivalent NFA without empty moves $N = (Q', \Sigma, \delta', q_0, F')$ where δ' is $Q \times \Sigma \rightarrow 2^Q$.

- Here, Q, Σ & initial state remains same.
- F and F' for M and N might not be same.
- The transition function is determined using the formula $\delta'(q, a) = \epsilon\text{-closure}(\delta(\hat{\delta}(q, \epsilon), a))$ where δ closure $\hat{\delta}(q, \epsilon) = \epsilon\text{-closure}(q)$.

- ϵ-closure of a state is a set of all the states having the distance 0 from the state q.



ϵ_closure$(q_0) = \{q_0, q_1, q_2\}$

ϵ_closure$(q_1) = \{q_1, q_2\}$

ϵ_closure$(q_2) = \{q_2\}$

NFA $\epsilon \rightarrow$ NFA

$\delta'(q,a) = \epsilon$_closure$(\delta(\delta\hat{}(q,\epsilon), a))$

$Q = \{q_0, q_1, q_2\}$ (same set of states)

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

I.S $= q_0$

~~F = q_2~~

$\delta'(q_0, 0) = \epsilon$_closure $(\delta(\underbrace{\delta\hat{}(q_0,\epsilon)}_{\epsilon\text{_closure of } q_0}, 0))$

$= \epsilon$_closure $(\delta(\{q_0, q_1, q_2\}, 0)) = \epsilon$closure$(\delta(q_0,0) \cup \delta(q_1,0) \cup \delta(q_2,0))$

~~$\delta(q_0, 1) = \epsilon$_closure~~

$= \epsilon$_closure $(q_0 \cup \phi \cup \phi)$

~~$\delta'(q_0, 2)$~~

$= \epsilon$_closure (q_0)

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

~~ϵ_closure~~ $\delta'(q_0, 1) = \epsilon$_closure$(\delta(\delta\hat{}(q_0,\epsilon), 1))$

$= \epsilon$_closure $(\delta(\{q_0, q_1, q_2\}, 1))$

$= \epsilon$_closure $(\delta(q_0,1) \cup \delta(q_1,1) \cup \delta(q_2,1))$

$= \epsilon$_closure $(\phi \cup q_1 \cup \phi)$

$= \epsilon$_closure (q_1)

$= \{q_1, q_2\}$

$$\delta'(q_0, 2) = \epsilon\_closure(\delta(\hat{\delta}(q_0, \epsilon), 2))$$
$$= \epsilon\_closure(\delta(q_0, 2) \cup \delta(q_1, 2) \cup \delta(q_2, 2))$$
$$= \epsilon\_closure(\phi \cup \phi \cup q_2)$$
$$= \epsilon\_closure(q_2)$$
$$= q_2$$

$$\delta'(q_1, 0) = \epsilon\_closure(\delta(\hat{\delta}(q_1, \epsilon), 0))$$
$$= \epsilon\_closure(\delta(\{q_1, q_2\}, 0))$$
$$= \epsilon\_closure(\delta(q_1, 0) \cup \delta(q_2, 0))$$
$$= \epsilon\_closure(\phi \cup \phi)$$
$$= \phi$$

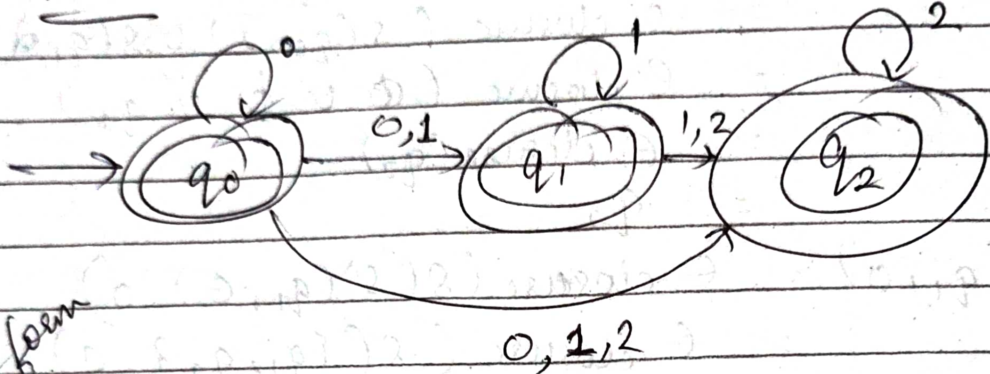
$$\delta'(q_1, 1) = \epsilon\_closure(\delta(\hat{\delta}(q_1, \epsilon), 1))$$
$$= \epsilon\_closure(\delta(q_1, 1) \cup \delta(q_2, 1))$$
$$= \epsilon\_closure(q_1 \cup \phi)$$
$$= \{q_1, q_2\}$$

$$\delta'(q_1, 2) = \epsilon\_closure(\delta(\hat{\delta}(q_1, \epsilon), 2))$$
$$=$$

$Q \backslash \Sigma$	0	1	2
$* q_0$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$* q_1$	—	$\{q_1, q_2\}$	$\{q_2\}$
$* q_2$	—	—	$\{q_2\}$

F' is a set of all the states whose ϵ_closure contains atleast one final state from F

NFA



of the form
0* 1* 2*

Chp 2:

QB 1) Draw 8086 architecture and explain.

2) Diff. b/w computer architecture & organisation.

× 3) Von Neumann model

8/2/22

AT

$a^+ = \{a, aa, aaa, \ldots\}$

$a^* = \{\epsilon, a, aa, aaa \ldots\}$

ϵ = Empty set

Regular expression

$\phi \rightarrow q_0$

$\epsilon \rightarrow q_0$

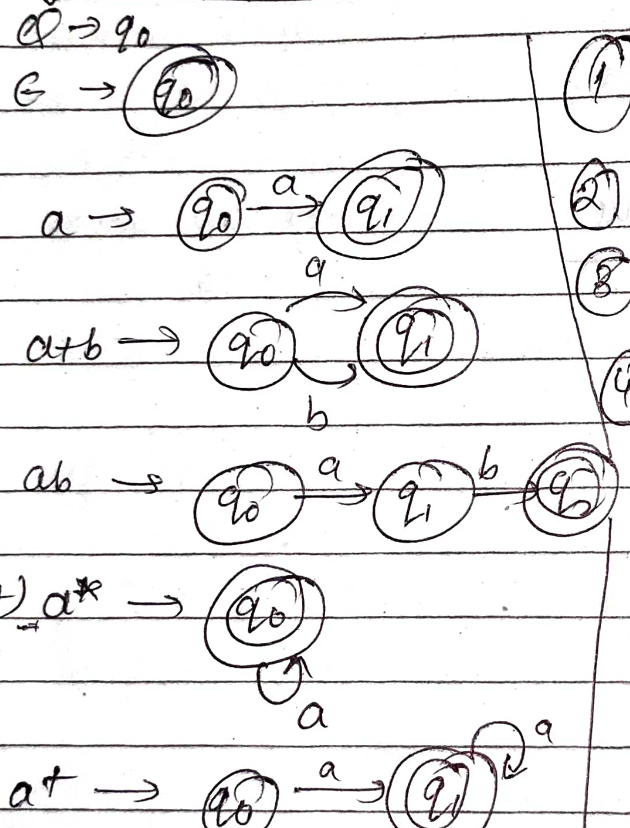
$a \rightarrow$

$a+b \rightarrow$

$ab \rightarrow$

(same way a^+) $a^* \rightarrow$

$a^+ \rightarrow$



1) a^*b

2) ba^*

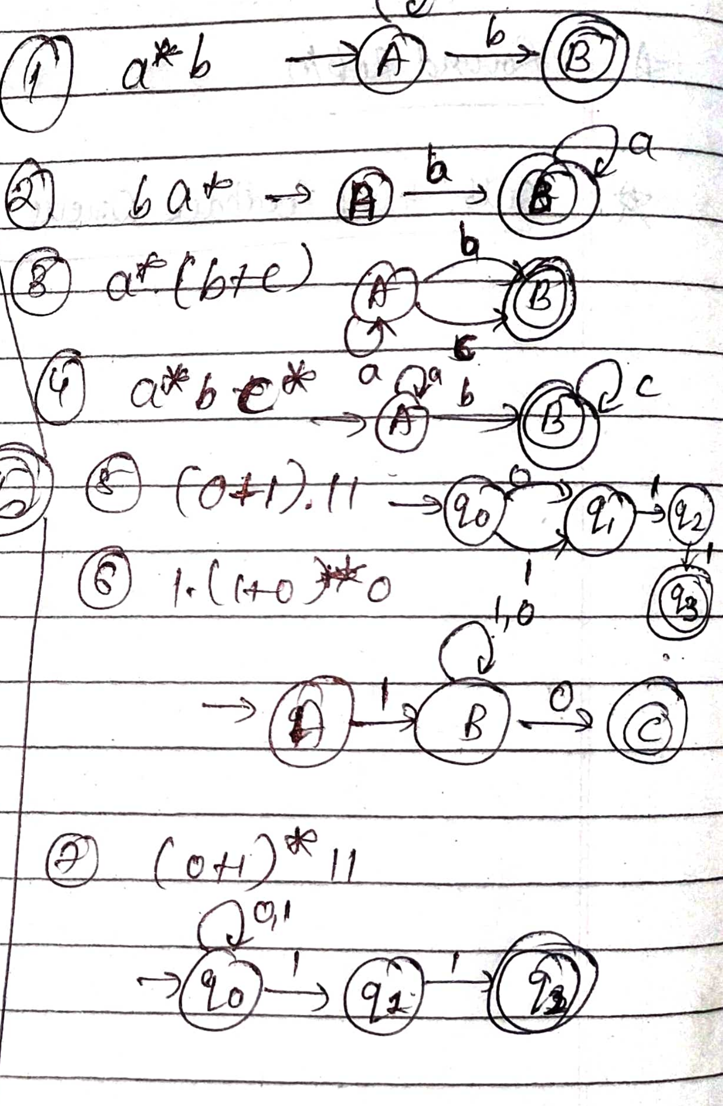
3) $a^*(b+c)$

4) a^*bc^*

5) $(0+1).11$

6) $1.(1+0)^*0$

7) $(0+1)^*11$



$a|b = a+b$

1) $10 + (0+11)0^*1$

2) $(a|b)^*(abb|a^*b)$