Code — At Lexical Analyzer state

TAFL

* NFA → DFA → L_1 & L_2 will be same for NFA & DFA

eg: M = ({q0, q1, q2, q3}, (0,1), δ, q0, {q3});
Where,

$\delta(q_0, 0) = \{q_1\}$, $\delta(q_0, 1) = \phi$
$(q_1, 0) = \{q_1\}$ $(q_1, 1) = \{q_1, q_2\}$
$(q_2, 0) = \phi$ $(q_2, 1) = \{q_3\}$
$(q_3, 0) = \{q_3\}$ $(q_3, 1) = \{q_3\}$

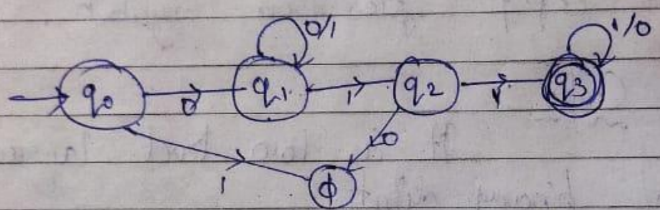


fig: NFA

State	0	1
q0	q1	ϕ
q1	q1	q1, q2
q2	ϕ	q3
q3	q3	q3

fig: NFA

DFA New state

old state	0	1
{q0}	{q1}	ϕ
{q1}	{q1}	{q1, q2}
{q1, q2}	{q1}	{q1, q2, q3}
q1, q2, q3		

∴ to be contd.

→ Start with same initial state i.e q0.
→ Write down new state as next 2nd state i.e q1 & ϕ
→ Next new state is {q1, q2} so we'll take it as new state.

New name	old state	New state: 0	New state: 1
A →	$\{q_0\}$	$\{q_1\}$	ϕ
B	$\{q_1\}$	$\{q_1\}$	$\{q_1, q_2\}$
C	$\{q_1, q_2\}$	$\{q_1\}$	$\{q_1, q_2, q_3\}$
D *	$\{q_1, q_2, q_3\}$	$\{q_1, q_3\}$	$\{q_1, q_2, q_3\}$
E *	$\{q_1, q_3\}$	$\{q_1, q_3\}$	$\{q_1, q_2, q_3\}$
	ϕ	ϕ	ϕ

fig: DFA

$$\delta(\{q_1, q_2\}, 0) = \delta(q_1, 0) \cup \delta(q_2, 0)$$
$$= \{q_1\} \cup \{\phi\}$$
$$= \{q_1\}$$

$$\delta(\{q_1, q_2, q_3\}, 0) = \delta(q_1, 0) \cup \delta(q_2, 0) \cup \delta(q_3, 0)$$
$$= q_1 \cup \phi \cup q_3$$
$$\delta(\text{—}\;\text{''}\;\text{—}, 1) = \delta(q_1, 1) \cup \delta(q_2, 1) \cup \delta(q_3, 1)$$
$$= q_1, q_2 \cup q_3 \cup q_3$$
$$= q_1, q_2, q_3$$

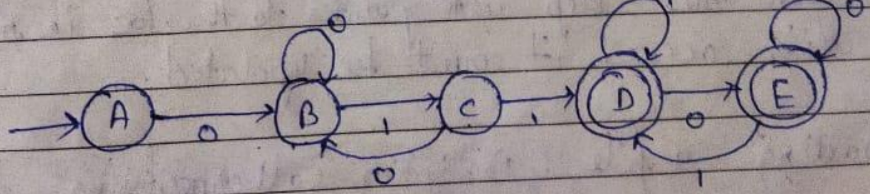


fig:- Converted DFA

11 | 1 | 22

$\delta_1(q, a) = \epsilon$ - closure of $(\delta(\epsilon\text{-closure}(q), a))$

(1) eg.

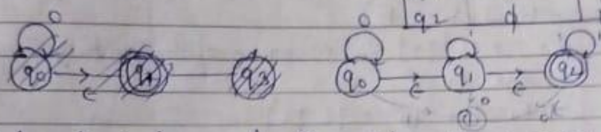
$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$

where,

δ	0	1	ϵ
q_0	$\{q_0\}$	ϕ	$\{q_1\}$
q_1	ϕ	$\{q_1\}$	$\{q_2\}$
q_2	ϕ	$\{q_2\}$	ϕ

It's equivalent NFA without ϵ-moves

δ_1	0	1
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$
q_1	ϕ	$\{q_1, q_2\}$
q_2	ϕ	$\{q_2\}$



1 Step :- Calculate all ϵ-closure of all states with ϵ-moves in NFA with ϵ-moves

S	ϵ - closure
q_0	q_0, q_1, q_2
q_1	q_1, q_2
q_2	q_2

2 Step The formula for calculating transitions is as follow -
(transitions for NFA without ϵ-moves)

$\delta_1(q, a) = \epsilon$-closure of $(\delta(\epsilon\text{-closure}(q), a))$

* Here we have to write down all possible states while NFA→DFA [illegible]

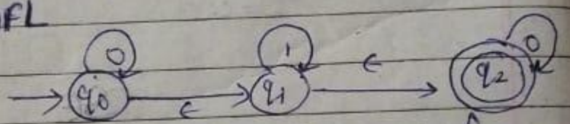
δ_1	0	1
→ q_0	q_0, q_1, q_2	q_1, q_2
q_1	ϕ	q_1, q_2
q_2	ϕ	q_2

$\delta(q_0, 0) = \epsilon$ closure of $(\delta(\epsilon\text{-closure}(q_0), 0))$
$= \epsilon$ - closure of $(\delta(\{q_0, q_1, q_2\}, 0))$
$= \epsilon$ - closure of q_0
$= q_0, q_1, q_2$

- Collection - (Mechanism) ...
  to derive formulated metrics.
- Analysis - Computation of metrics & the appl^n of mathematical tools.
- Interpretation - Evaluation of metrics resulting in insight into the quality of representation
- Feedback - Recommendation derived from interpretation of prod metrics transmitted to s/w team.

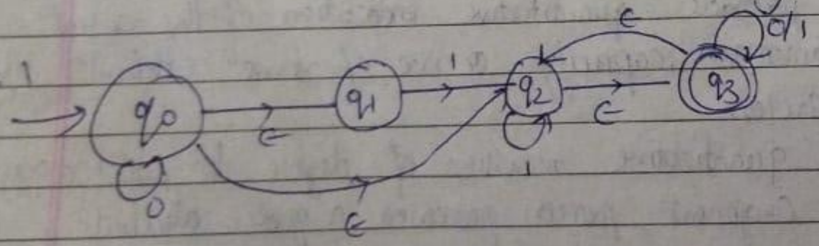
TAFL

* NFA with ∈-Moves



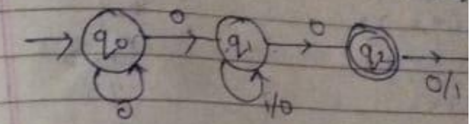
∈-closure(q): set of all state which can be reach from q to on path labeled by ∈

eg: 1



q_1	q_1, ...	...
q_2	q_2, q_1	q_1
q_3	q_3, q_2, q_1	q_2, q_1...

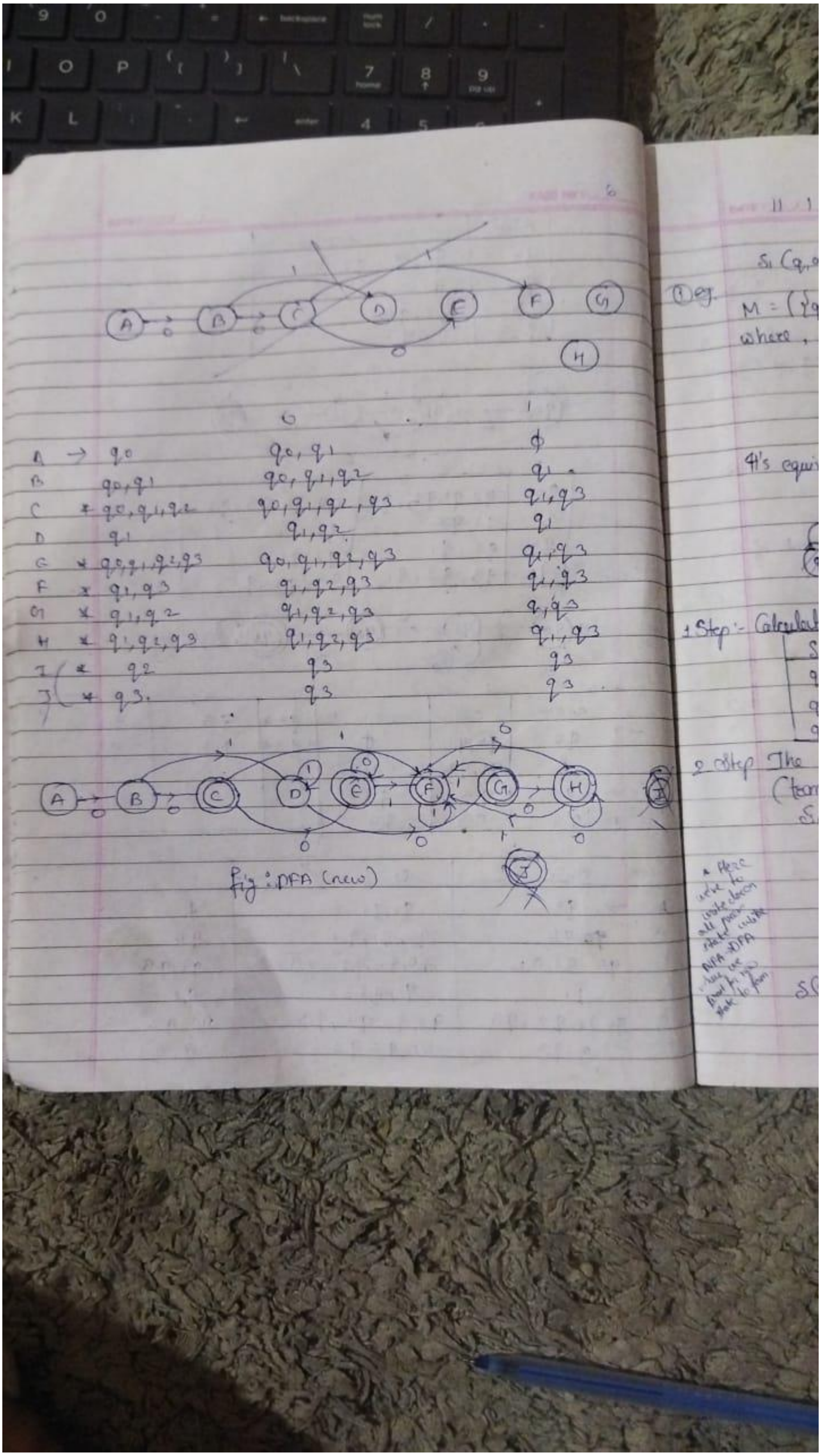
eg:



	state	0	1
→	q_0	q_0, q_1	ϕ
	q_1	q_1, q_2	q_1
*	q_2	q_3	q_3
*	q_3	q_3	q_3

	State	0
A	→ q_0	q_0, q_1
B	q_0, q_1	q_0, q_1, ...
* C	q_0, q_1, q_2	q_0, q_1, ...
D	q_1	q_1, q...
* G	q_0, q_1, q_2, q_3	q_0, q_1, ...
* F	q_1, q_3	q_1, q...

AUTOMATA



		0	1
A →	q_0	q_0, q_1	ϕ
B	q_0, q_1	q_0, q_1, q_2	q_1
C	* q_0, q_1, q_2	q_0, q_1, q_2, q_3	q_1, q_3
D	q_1	q_1, q_2	q_1
E	* q_0, q_1, q_2, q_3	q_0, q_1, q_2, q_3	q_1, q_3
F	* q_1, q_3	q_1, q_2, q_3	q_1, q_3
G	* q_1, q_2	q_1, q_2, q_3	q_1, q_3
H	* q_1, q_2, q_3	q_1, q_2, q_3	q_1, q_3
I	* q_2	q_3	q_3
J	* q_3	q_3	q_3

fig : DFA (new)

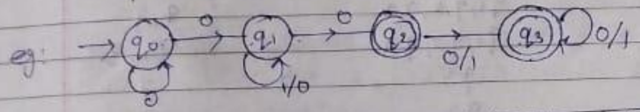
5

State	ϵ-moves reach to
q_0	q_0, q_1, q_2, q_3
q_1	q_1
q_2	q_2, q_3
q_3	q_3, q_2

eg:- 2

q_0	q_0, q_1, q_3, q_2	q_0, q_1, q_3, q_2
q_1	q_1, ~~q_2~~	q_1
q_2	q_2, q_1	q_2, q_1
q_3	q_3, q_2, q_1	q_3, q_2, q_1

eg:



state	0	1			
$\rightarrow q_0$	q_0, q_1	ϕ	q_2-x	~~q_3~~	~~q_3~~
q_1	q_1, q_2	q_1	q_3*	~~q_3~~	~~q_2~~
* q_2	q_3	q_3			
* q_3	q_3	q_3			

	State	0	1
A	$\rightarrow q_0$	q_0, q_1	ϕ
B	q_0, q_1	q_0, q_1, q_2	q_1
*C	q_0, q_1, q_2	q_0, q_1, q_2, q_3	q_1, q_3
D	q_1	q_1, q_2	q_1
*G	q_0, q_1, q_2, q_3	q_0, q_1, q_2, q_3	q_1, q_3
*F	q_1, q_3	q_1, q_2, q_3	q_1, q_3

$\delta(q_0,1) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure of } q_0),1)$
$= \epsilon\text{-closure}(\delta(\{q_0,q_1,q_2\},1)) = \epsilon\text{closure of } q_1,q_2$
$= q_1,q_2$
$\delta(q_1,0) = \epsilon\text{-closure}(\delta(\epsilon\text{ closure of } q_1),0) = \epsilon\text{ closure of } \phi = \phi$
$\delta(q_1,1) = \epsilon\text{-closure}(\delta(\epsilon\text{ clos }(q_1),1)) = \epsilon\text{-closure of } q_1,q_2 = q_1,q_2$
$\delta(q_2,0) = \epsilon\text{ clo}(\delta(\epsilon c(q_2),0)) = \epsilon\text{ clo } \phi = \phi$
$\delta(q_2,1) = \epsilon\text{ clo}(\delta(\epsilon cl(q_2),1)) = \epsilon\text{-clo } q_2 = q_2$

* NFA without ϵ-moves initial state will be same as NFA with ϵ-moves
  Final state (F)
* $F_1 = F \cup \{q_0\}$ if ϵ-closure of (q_0) contains member of F.
  else $F_1 = F$

$\therefore F_1 = F \cup \{q_0\} = \{q_2, q_0\}$

1) Boots...
(s.nc...
the...
to...
of...
Comp...
* When...
it...
for...
* Comp...
2) To...
wi...
to...
eg...
Ho...

14 1 22 | 35

TAFL

* Moore & Mealy Machines

Mealy → output on trans^n $\quad A \xrightarrow{a/o/p} B$

a'll get o/p during transi^n from 1 state to other.

Moore → output on state

Mealy Machine :-
It is an FSM whose o/p depends on the present state as well as present i/p.
In Mealy every transition for a particular i/p symbol has fixed o/p

Can be described by 6 tuple $(Q, \Sigma, O, \delta, X, q_0)$

Q - Finite set of states

Σ - Finite set of symbol $\quad \Sigma = \{0, 1\}$ called i/p alphabet

O - — " — called o/p alphabet $\quad O = K$

δ - i/p trans^n func^n $\delta : Q \times \Sigma \to Q$

X - X is the o/p tran^n func^n where $X : Q \to O$.

q_0 - initial state from where any i/p is processed ($q_0 \in Q$)

19.1.22

Present State	Next State a=0 State	a=0 Output	Next State a=1 State	a=1 Output
→q_0	q_3	0	q_1	1
q_1	q_0	1	q_3	0
q_2	q_2	1	q_2	0
q_3	q_1	0	q_0	1

Table for Mealy M.

<u>Moore Machine</u>

Present State	Nxt State a=0	a=1	Output
→q_0	q_3	q_1	1
q_1	q_0	q_3	0
q_2	q_2	q_2	0
q_3	q_1	q_0	1

Transⁿ Table for Moore

* o/p of Moore is only depend upon its present states whereas,
  o/p of Mealy is depend on present state as well as input alphabets i.e 0/1

* Moore ⟶ Mealy
1. Given ~~Mealy~~ Moore table.
2. Draw Mealy table format from given.
3. U'll get blank 2 o/p col.
4. Now check o/p for present state.
   for eg: in above moore table o/p for →q_0 is 1
5. Write 1 as o/p of every state in Mealy having q_0

Eg.

Prese

Preser

→ In ab
eg:
Write
wher
for

Eg.

Present State	Next State		Output
	a = 0	a = 1	
a	e	d	1
c	f	a	0
d	a	c	1
e	c	f	1
f	d	e	1

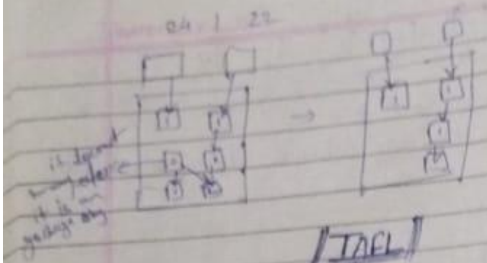
Moore Tranⁿ Table (a)

Present State	Next State			
	a = 0		a = 1	
	State	Output	State	Output
a	e	1	d	1
c	f	1	a	1
d	a	1	c	0
e	c	0	f	1
f	d	1	e	1

Mealy Tranⁿ Table (b)

→ In above table (a) we check o/p for each p.s
eg: a = 1 ; c = 0 ; d = 1, etc.
Write these o/p as state o/p in mealy machine
wherever we see a as state write 1 as o/p
for c write 0 and so on.

04.1.22



TAFL

* Mealy → Moore

eg: ① Mealy machine table -

Present state	Next state			
	a = 0		a = 1	
	Next state	o/p	Next State	o/p
→ a	d	0	b	1
b	a	1	d	0
c	c	1	c	0
d	b	0	a	1

a) check o/p of each state here, a = 1, d = 0 but, c = 1/0
b = 0/1

b) We've to divide b = b_0, b_1 & c = c_0, c_1 now, b_0 = 0, b_1 = 1, c_0 = 0, c_1

Present state	Next state		O/P
	a = 0	a = 1	
→ a	d	b_1	1
b_0	a	d	0
b_1	a	d	1
c_0	c_1	c_0	0
c_1	c_1	c_0	1
d	b_0	a	0

fig: Moore machine table

eg: 2 Present

→ a
b
c
d

Step 2) a = 0

Pres

eg: 2

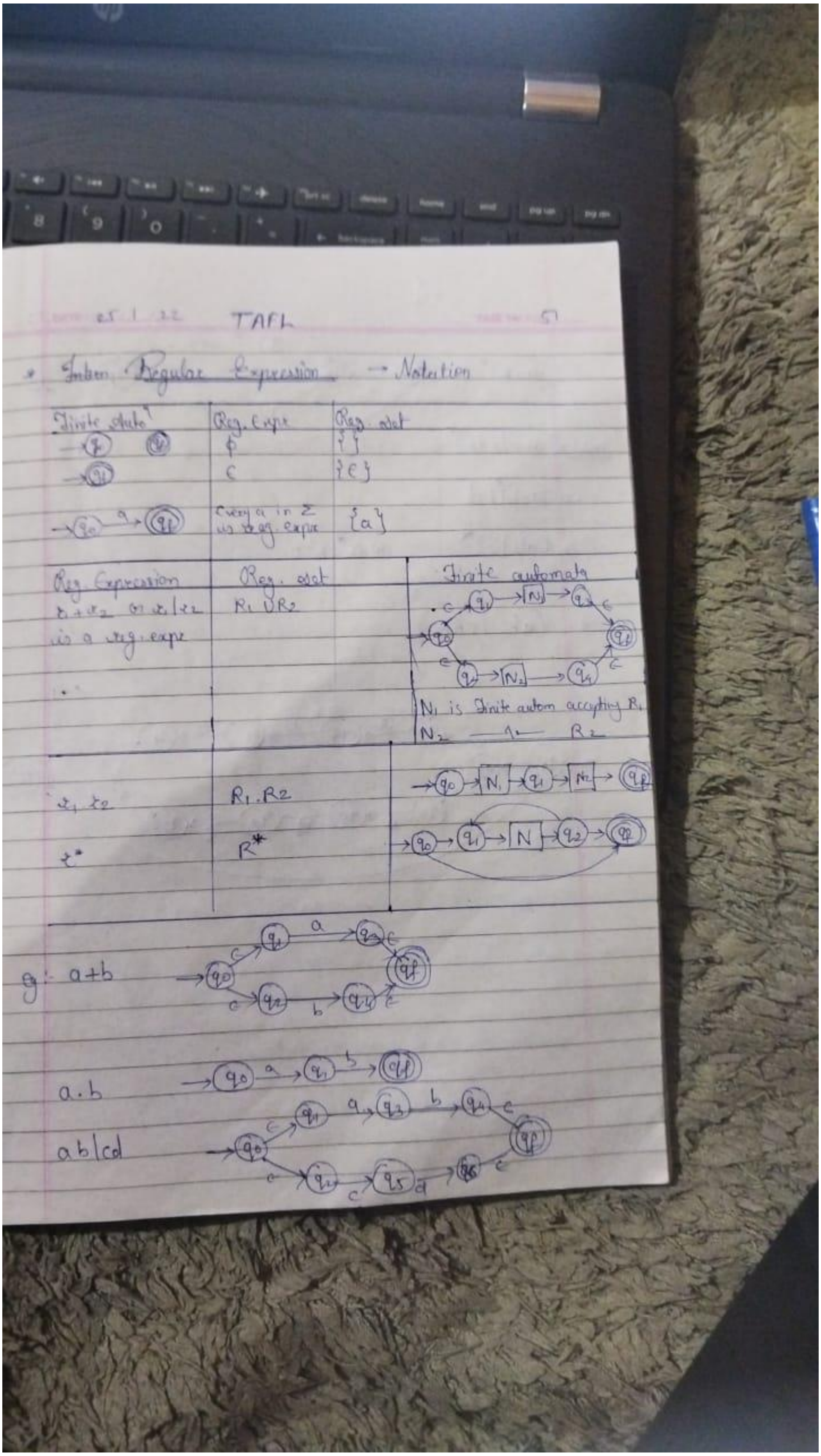
Present state	Next state			
	a = 0		a = 1	
	Next state	o/p	Next state	o/p
→a	d	0	b	1
b	c	1	a	0
c	a	0	c	1
d	b	1	d	1
e	e	0	e	1

Step ① a = 0 , b = 1 , c = 1 , d = 0/1, e = 0/1 ∴ $e_0 = 0, e_1 = 1, d_0 = 0, d_1 = 1$

Present state	Next state		Output
	a = 0	a = 1	
→a	d_0	b	0
b	c	a	1
c	a	c	1
d_0	b	d_1	0
d_1	b	d_1	1
e_0	e_0	e_1	0
e_1	e_0	e_1	1

· Converted moore table from mealy table

AUTOMATA



25 | 22 TAFL 51

* Intro. Regular Expression → Notation

Finite Auto.	Reg. Expr	Reg. set
$\rightarrow q_f$ q_f	ϕ	{ }
$\rightarrow q_f$ (final)	ϵ	{ϵ}
$\rightarrow q_0 \xrightarrow{a} q_f$	Every a in Σ is reg. expr	{a}

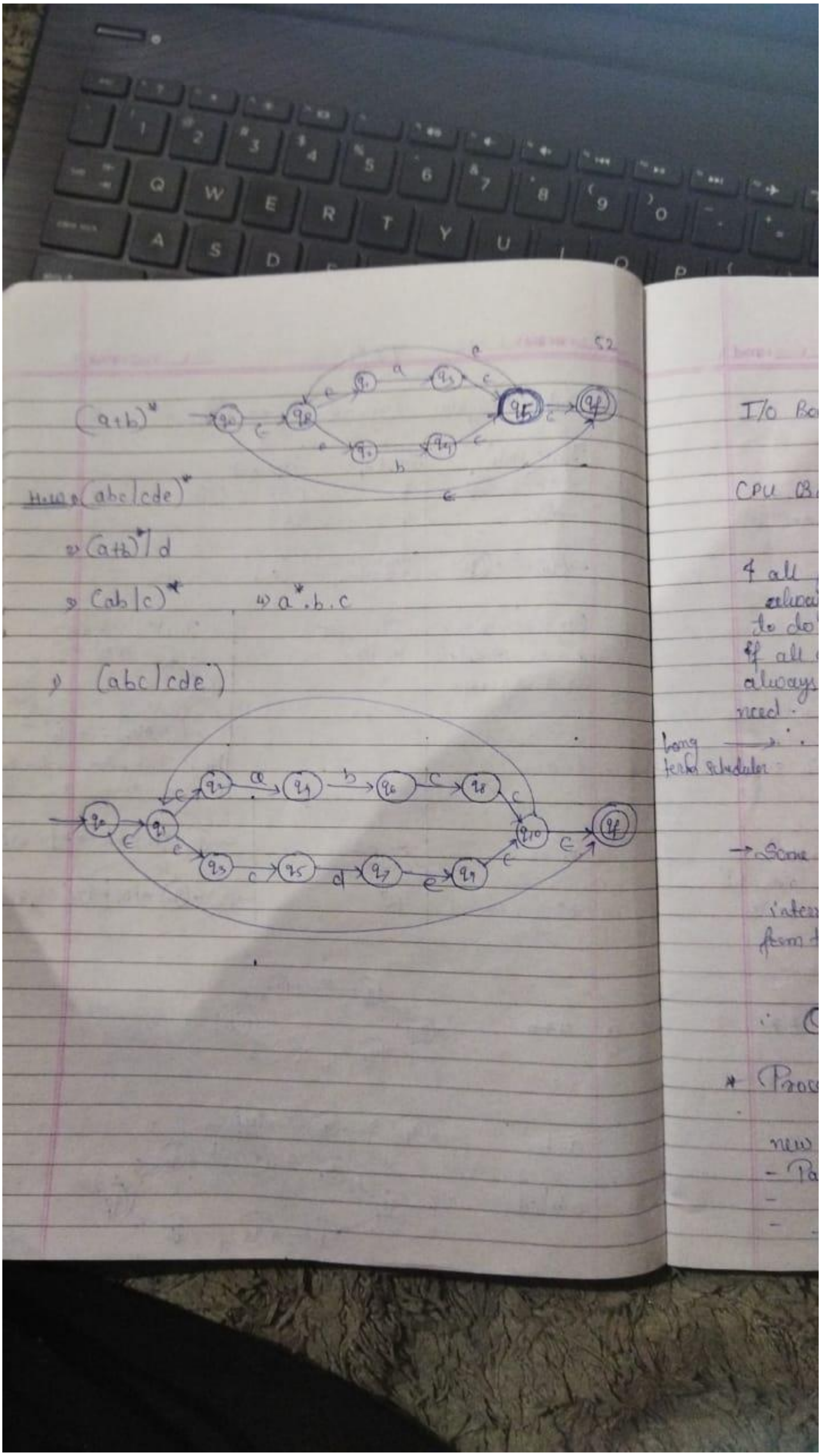
Reg. Expression	Reg. set	Finite automata
$r_1 + r_2$ or $r_1 \mid r_2$ is a reg. expr	$R_1 \cup R_2$	N_1 is finite autom accepting R_1; N_2 — " — R_2
$r_1 . r_2$	$R_1 . R_2$	$\rightarrow q_0 \rightarrow \boxed{N_1} \rightarrow q_1 \rightarrow \boxed{N_2} \rightarrow q_f$
r^*	R^*	$\rightarrow q_0 \rightarrow q_1 \rightarrow \boxed{N} \rightarrow q_2 \rightarrow q_f$

eg:- a+b

a.b $\rightarrow q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_f$

ab|cd

AUTOMATA



Intern   Extern

→ keyword - static   -"-

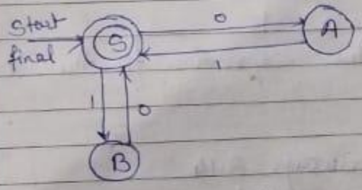
Internal static var has no linkage   External -"- has linkage

Symbol Table

→ Analysis & synthesis

→ determine scope

TAFL



Step 1 :- Associate suitable variables with the state of the automata i,e S, A, B

Step 2 :- Form the ~~stat~~ set of equⁿ using following rules —

a) Check all outgoing transⁿ from each state and write it

eg:- S = 0A   A = 1S   B = 0S

S = 1B

i,e write 0/1 with reaching state.

AUTOMATA

31/1/22

b) Write all equ^n i.e $S = 0A$ $A = 1S$ $B = 0S$
$S = 1B$

c) Take your final state i.e S and associate equ^n with final state as final state $= \epsilon$
i.e $\underline{S = \epsilon}$

if have 2 final state then $S = \epsilon$ and $A = \epsilon$
other f.s.

Step: $S = 0A \mid 1B \mid \epsilon$ —— ①
$A = 1S$ —— ②
$B = 0S$ —— ③

Step: Remember equ^n $S = aS + b$
The regular equation expression for this equ^n:-
$S = a^*b$

As value of π is always 3.14
Similarly, $aS + b = a^*b$

Step: Replace $A = 1S$ and $B = 0S$ in eq ①
as we want equ^n in $S = aS + b$ format

$\therefore$ $S = 0(1S) \mid 1(0S) \mid \epsilon$
$S = 0.1.S \mid 1.0.S \mid \epsilon$ (. means concatenation)
$S = (01 + 10)S + \epsilon$ { . can be ignored, | = + }

$\therefore$ $S = (01 + 10)^* \epsilon$ { $S = a^*b$

$\boxed{\therefore S = (01 + 10)^*}$

Que:

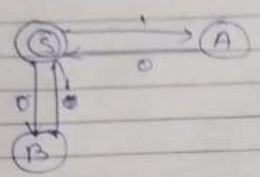
$S = 1A \mid$
$A = 0S$
$B = \cancel{0S} \mid$

$S = 1(0$
$S = 10.S$
$S = (10$

$\therefore$ $S =$

$\therefore$ $S =$

Que:



$S = 1A \mid 0B \mid \epsilon$

$A = 0S$

$B = \cancel{0S}\ 1S$

$S = 1(0S) \mid 1(0S) \mid \epsilon \quad = 1(0S) \mid 0(1S) \mid \epsilon$

$S = 10.S \mid 1.1.S \mid \epsilon$

$S = (10 + 11) S + \epsilon$ i,e $S = aS + b$

$\therefore S = (10+11)^* \epsilon$ i,e $S = a^* b$

$\therefore S = (10+11)^*$