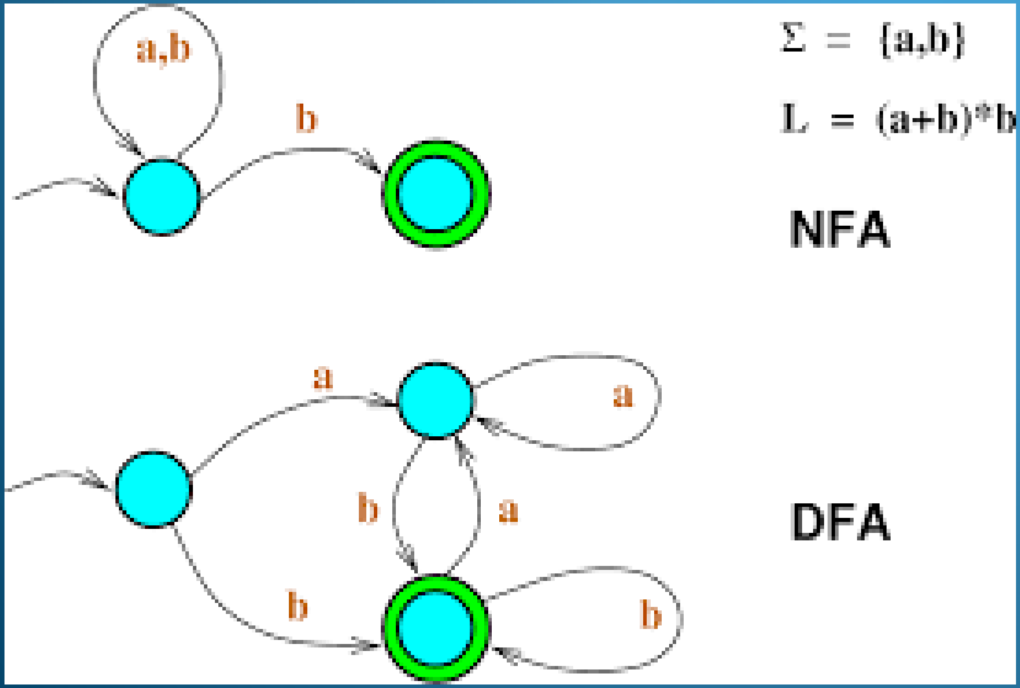
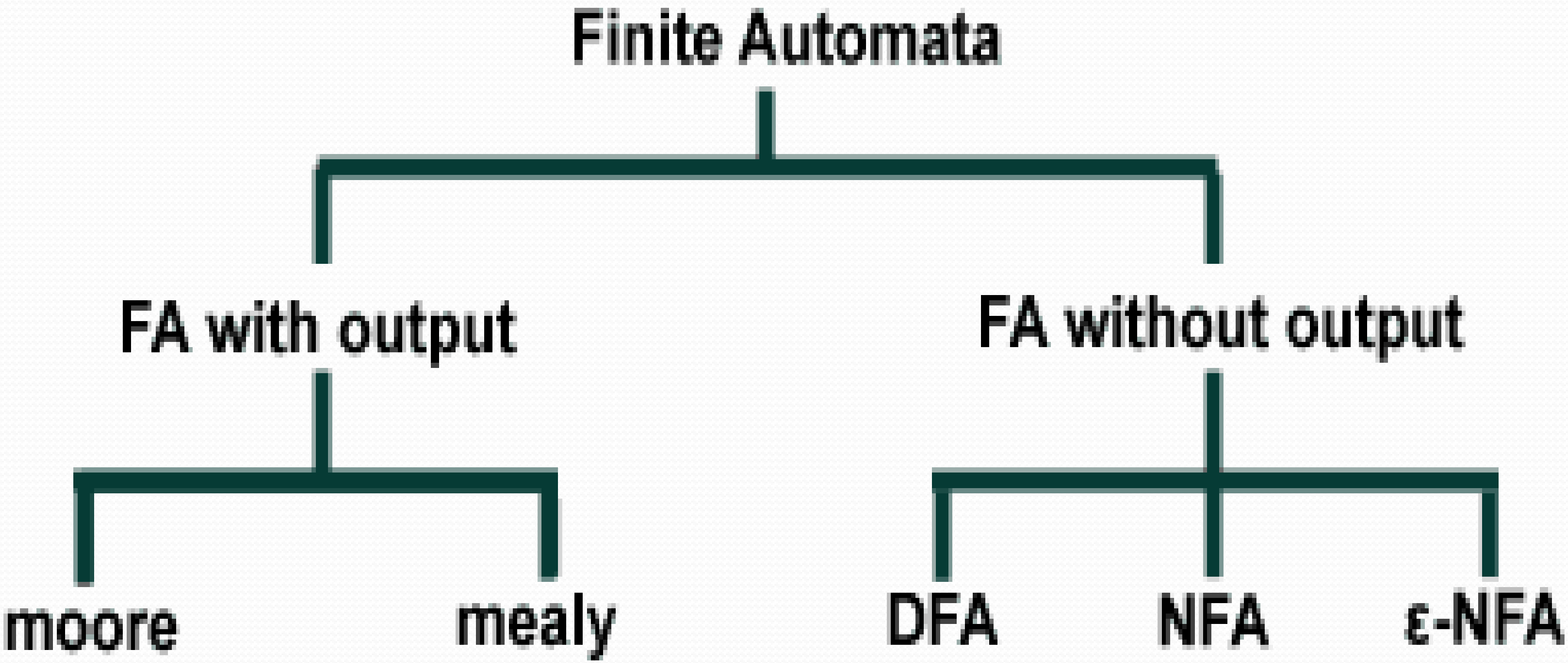
Non Deterministic Finite Automation



---- Sakshi Surve


Finite Automata
FA with output
FA without output
moore
mealy
DFA
NFA
ε-NFA

Non - Deterministic Finite Automaton (NDFA/NFA)

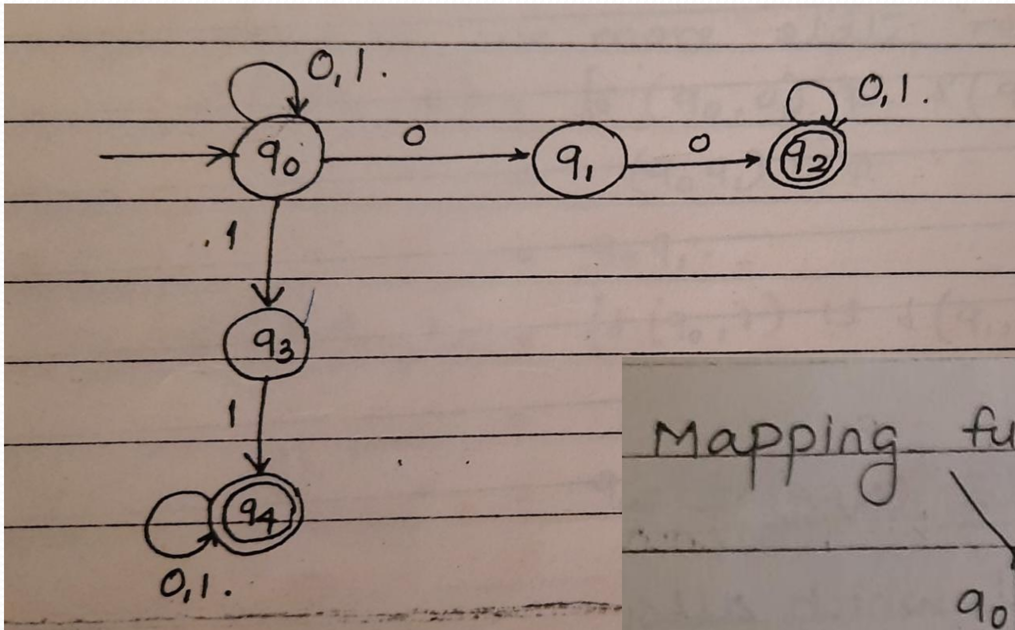
- In NDFA, for a particular input symbol, the machine can move to any combination of the states in the machine.
- In other words, the exact state to which the machine moves cannot be determined. Hence, it is called **Non-deterministic Automaton.**
- As it has finite number of states, the machine is called **Non-deterministic Finite Machine** or **Non-deterministic Finite Automaton**.

NFA :

- A Finite Automaton model which allows zero or one or more transitions from a state on the same input symbol is called as Non- Deterministic Finite Automaton

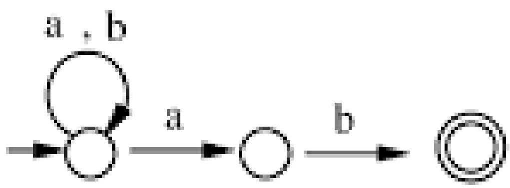
- DFA is a special case of NFA where from each state, there is a unique transition over each input symbol

- Design a NFA for accepting all binary strings with either two consecutive zeroes or two consecutive 1's



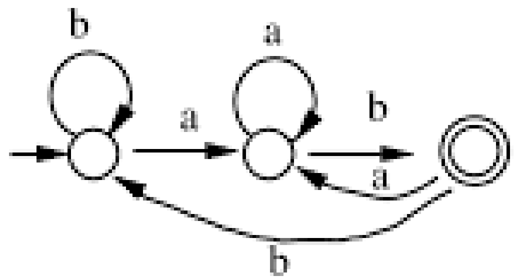
Mapping function δ for NFA is :-

	1	0
q_0	$\{q_0, q_3\}$	$\{q_0, q_1\}$
q_1	ϕ	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_4\}$	ϕ
q_4	$\{q_4\}$	$\{q_4\}$.



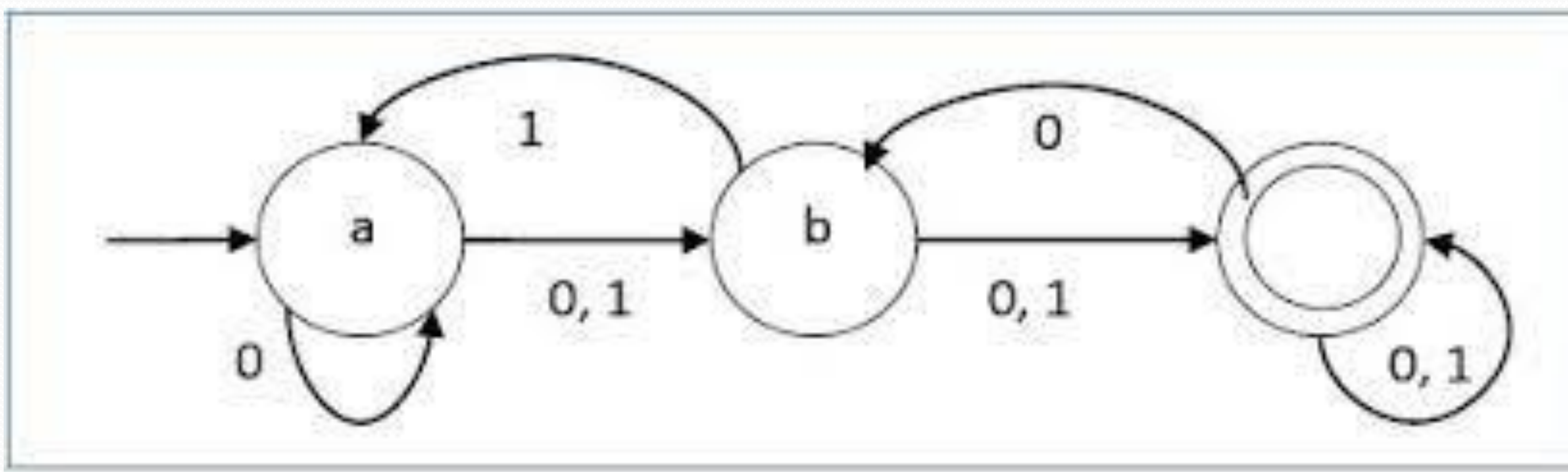
NFA

for $M[p]$



DFA

for $M[p]$



Formal Definition of Nondeterministic Finite Automata

- An NFA is a five-tuple: $N = (Q, \Sigma, \delta, q_0, F)$

Q A finite set of states

Σ A finite input alphabet

q_0 The initial/starting state, q_0 is in Q

F A set of final/accepting states, which is a subset of Q

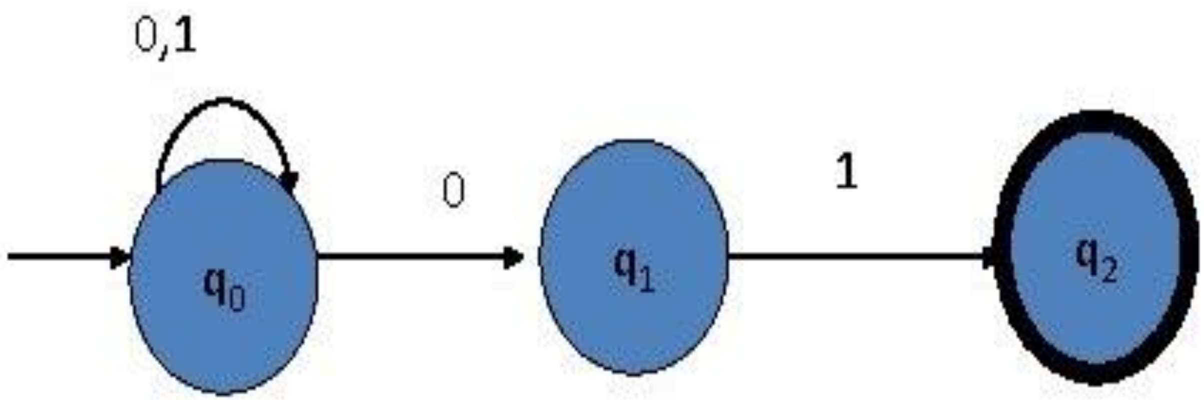
δ A transition function, which is a total function from $Q \times \Sigma$ to 2^Q

$\delta: (Q \times \Sigma) \rightarrow P(Q)$ -$P(Q)$ is the power set of Q, the set of all subsets of Q

$\delta(q,s)$ -The set of all states p such that there is a transition labeled s from q to p

$\delta(q,s)$ is a function from $Q \times S$ to $P(Q)$ (but not to Q)

- NFA that recognizes the language of strings that end in 01



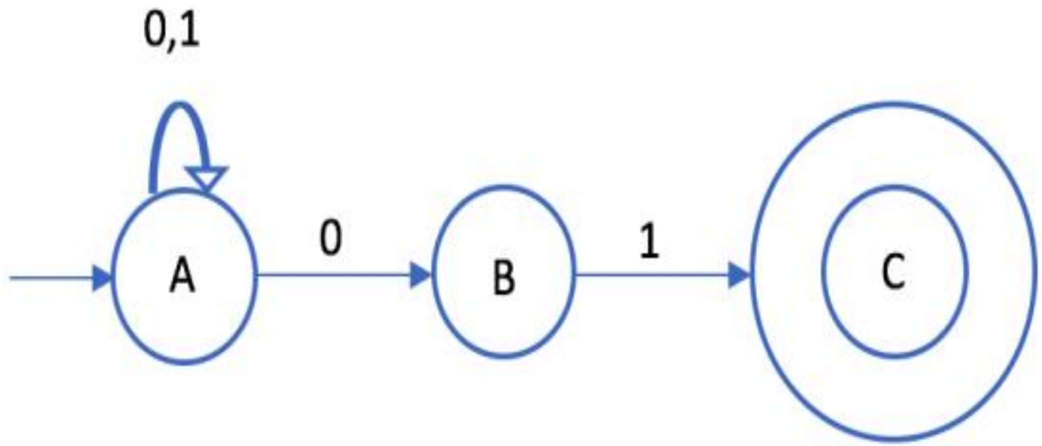
note: $\delta(q_0,0) = \{q_0,q_1\}$

$\delta(q_1,0) = \{\}$

NFA

State Transition Diagram for NFA

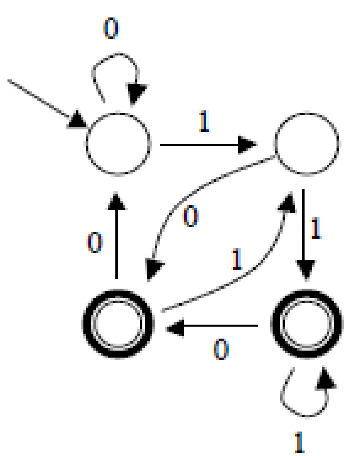
	0	1
→ A	{A, B}	{A }
B	Ø	{C}
* C	Ø	Ø



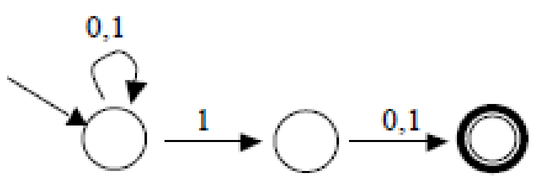
NFA Advantage

- An NFA for a language can be smaller and easier to construct than a DFA
- Let L={x ∈ {0,1}*|where x is a string whose next-to-last symbol is 1}
- Construct both a DFA and NFA for recognizing L.

DFA:



NFA:



Differences: DFA vs. NFA

- **DFA**

1. All transitions are deterministic
   - Each transition leads to exactly one state
2. For each state, transition on all possible symbols (alphabet) should be defined
3. Accepts input if the last state is in F
4. Sometimes harder to construct because of the number of states
5. Practical implementation is feasible

- **NFA**

1. Transitions could be non-deterministic
   - A transition could lead to a subset of states
2. For each state, not all symbols necessarily have to be defined in the transition function
3. Accepts input if *one of* the last states is in F
4. Generally easier than a DFA to construct
5. Practical implementation has to be deterministic (so needs conversion to DFA)

But, DFAs and NFAs are equivalent (in their power) !!

NFA to DFA Conversion :

Q. Convert given NFA to DFA.

PAGE No.
DATE / /

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

$q \setminus \Sigma$	0	1
q_0	$\{q_0, q_1\}$	$\{q_1\}$
q_1	$\{\phi\}$	$\{q_0, q_1\}$

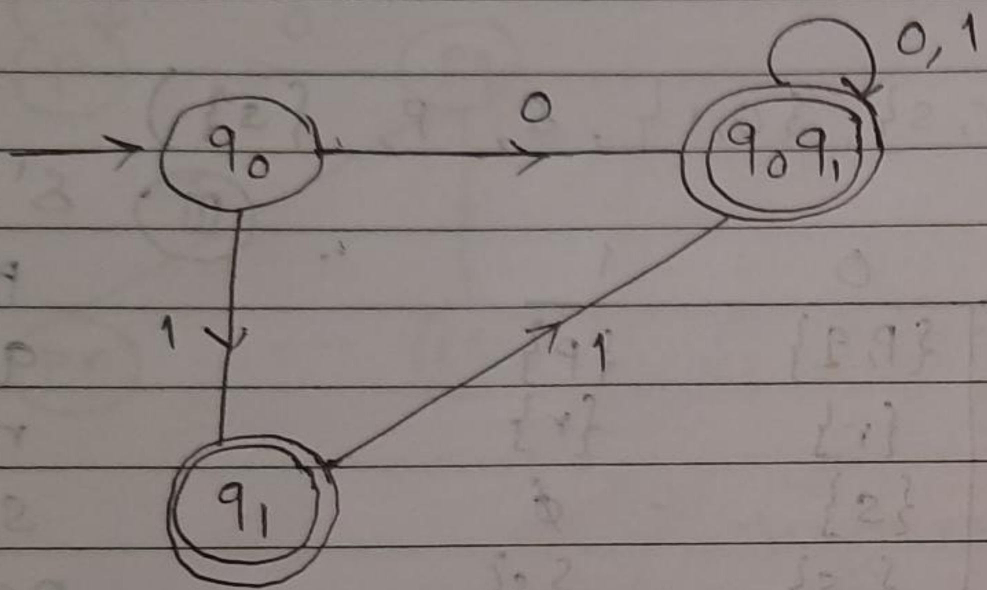
The converted DFA will be -

$$M' = (Q', \Sigma, q_0, \delta', F')$$

Constructing δ' from given δ as follows -

δ'	0	1
q_0	q_0q_1	q_1
* q_1	–	q_0q_1
* q_0q_1	q_0q_1	q_0q_1

This δ' can not be reduced further. $\therefore$ converting it to Transition Diagram.



The tuples of the converted DFA are:-

$Q' = \{q_0, q_1, q_0q_1\}$

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

$q_0 = q_0$

$F' = \{q_1, q_0q_1\}$

δ' - Mentioned above.

Rules Of Replacement :

- A Non Final state can be replaced by it’s equivalent Non Final state only
- A Final state can be replaced by another Final state only
- The Initial state can not be replaced by any other state
- Replacing state ‘A’ by ‘B’ means deleting entries related to ‘A’ and whenever we find symbol ‘A’ in the Transition Mapping Function, replace it with ‘B’

- Convert the following NFA to DFA

$M = (\{p, q, r, s\}, \{0, 1\}, \delta, p, \{q, s\})$

	0	1
p	{q, r}	{q}
q	{r}	{q, r}
r	{s}	{p}
s	{ϕ}	{p}

The converted DFA will be -

$$M' = (Q', \Sigma, q_0, F', \delta')$$

Constructing δ' from given δ as follows -

	δ'	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

From δ',

$r \equiv rs$

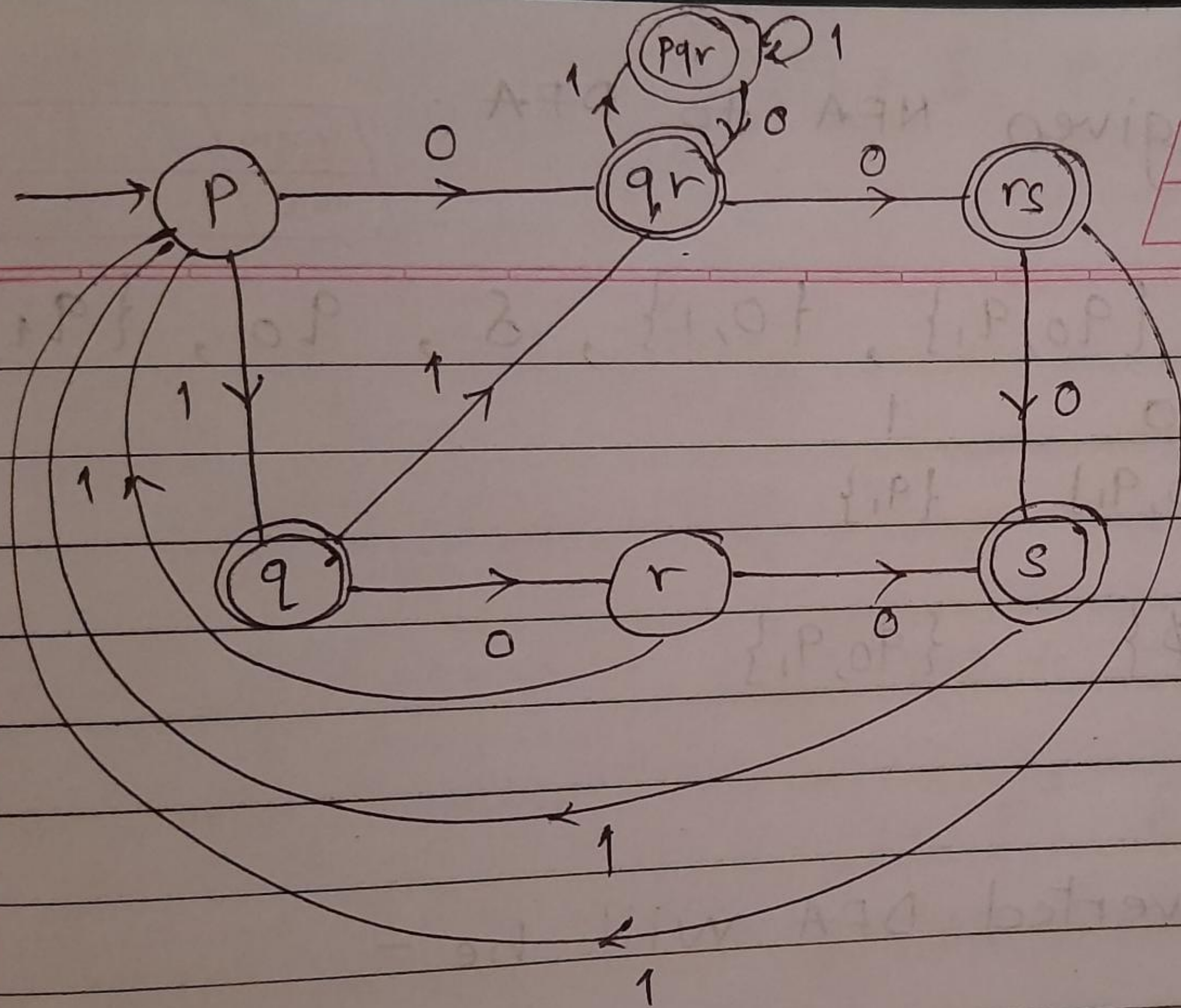
$qr \equiv qrs$.

Applying the rules of replacements, the new δ' becomes -

$\delta' \Rightarrow$		0	1
	p	qr	q
*	q	r	qr
	r	s	p
*	s	–	p
*	qr	rs	pqr
*	rs	s	p
*	pqr	qr	pqr

This δ' cannot be reduced further. The transitio

Diagram becomes -



PAGE No.
DATE

Now, tuples of ~~minimized~~ (converted) DFA can be mentioned as :-

$Q' = \{p, q, r, s, qr, rs, pqr\}$

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

$F' = \{q, s, rs, qr, pqr\}$

$q_0 = p$

δ' - Mentioned before.

Homework

- Convert the given NFA to DFA

$$M = (\{p, q, r, s\}, \{0, 1\}, \delta, p, \{s\})$$

III

$\delta \Rightarrow$	0	1
p	$\{p, q\}$	$\{p\}$
q	$\{r\}$	$\{r\}$
r	$\{s\}$	ϕ
s	$\{s\}$	$\{s\}$

PAGE No.

DATE / /

$\delta' \Rightarrow$

	0	1
p	pq	p
q	r	r
r	s	—
s	s	s
pq	pqr	pr
pqr	pqrs	pr
pr	pqs	pr
* pqrs	pqrs	prs
* pqs	pqrs	prs
* prs	pqs	ps
* ps	pqs	ps

From δ', the Equivalent states are :-

pqrs ≡ pqs

prs ≡ ps

The Reduction is possible as all of them are final states.

Applying the rules of replacement, the new δ' becomes -

$\delta' \Rightarrow$

	0	1
p	pq	p
q	r	r
r	s	–
s	s	s
pq	pqr	pr
pqr	~~pqrs~~ pqs	pr
pr	pqs	pr
* pqs	pqs	ps
* ps	pqs	ps

From new δ', the equivalent states are :-

$pqs \equiv ps$.

Reduction is possible.

By Applying rules of replacement, the new δ' becomes -

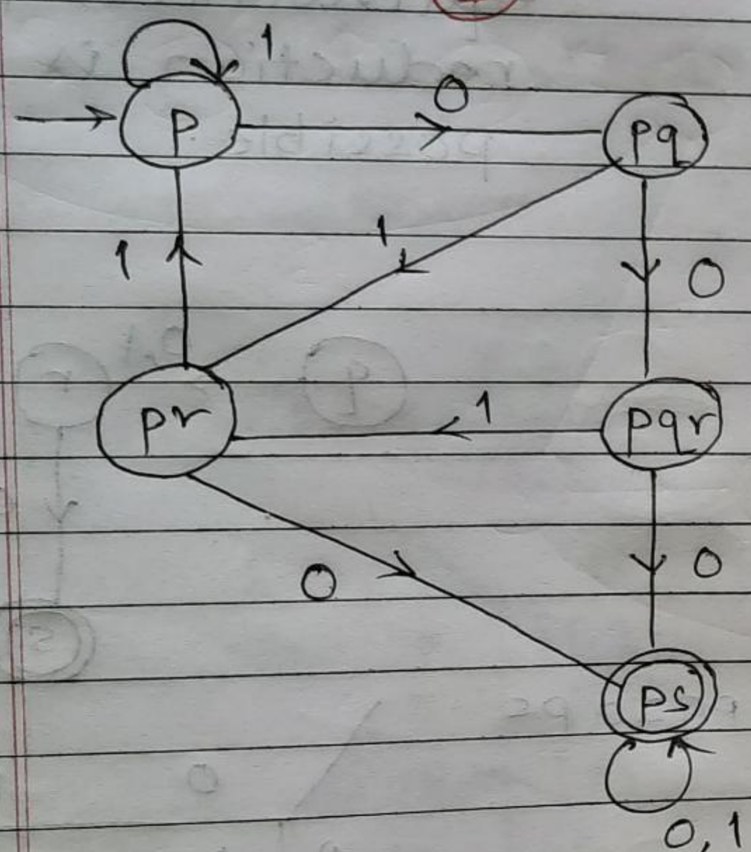
δ' ⇒

	0	1
p	pq	p
q	r	r
r	s	—
s	s	s
pq	pqr	pr
pqr	ps	pr
pr	ps	p
* ps	ps	ps

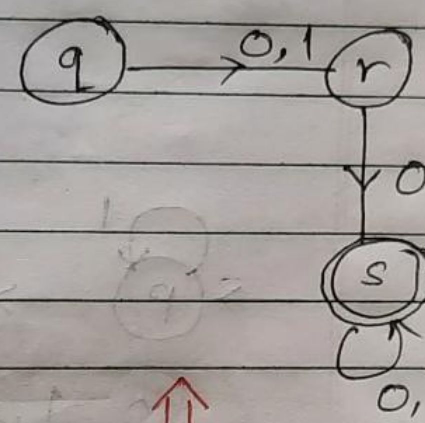
No more equivalence found in δ'.

Converting the δ' to Transition Diagram -

I



II



This is Unreachable
Also, the start state is missing.

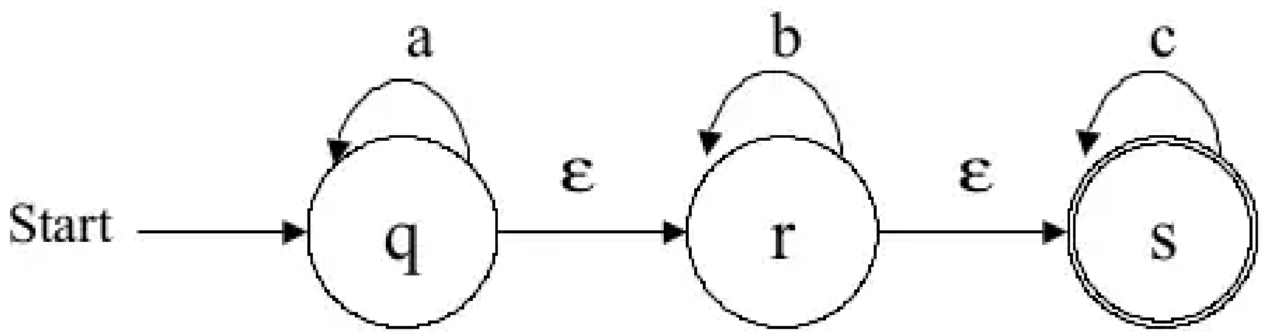
∴ Final Answer - Write Tuples of part I.

Epsilon Transitions

- Extension to NFA – a "feature" called epsilon transitions, denoted by ε, the empty string
- The ε transition lets us spontaneously take a transition, without receiving an input symbol
- Another mechanism that allows our NFA to be in multiple states at once.
  - Whenever we take an ε edge, we must fork off a new "thread" for the NFA starting in the destination state.
- While sometimes convenient, has no more power than a normal NFA
  - Just as a NFA has no more power than a DFA

Epsilon Elimination Exercise

- Exercise: Here is the ε-NFA for the language consisting of zero or more a's followed by zero or more b's followed by zero or more c's.
- Eliminate the epsilon transitions.



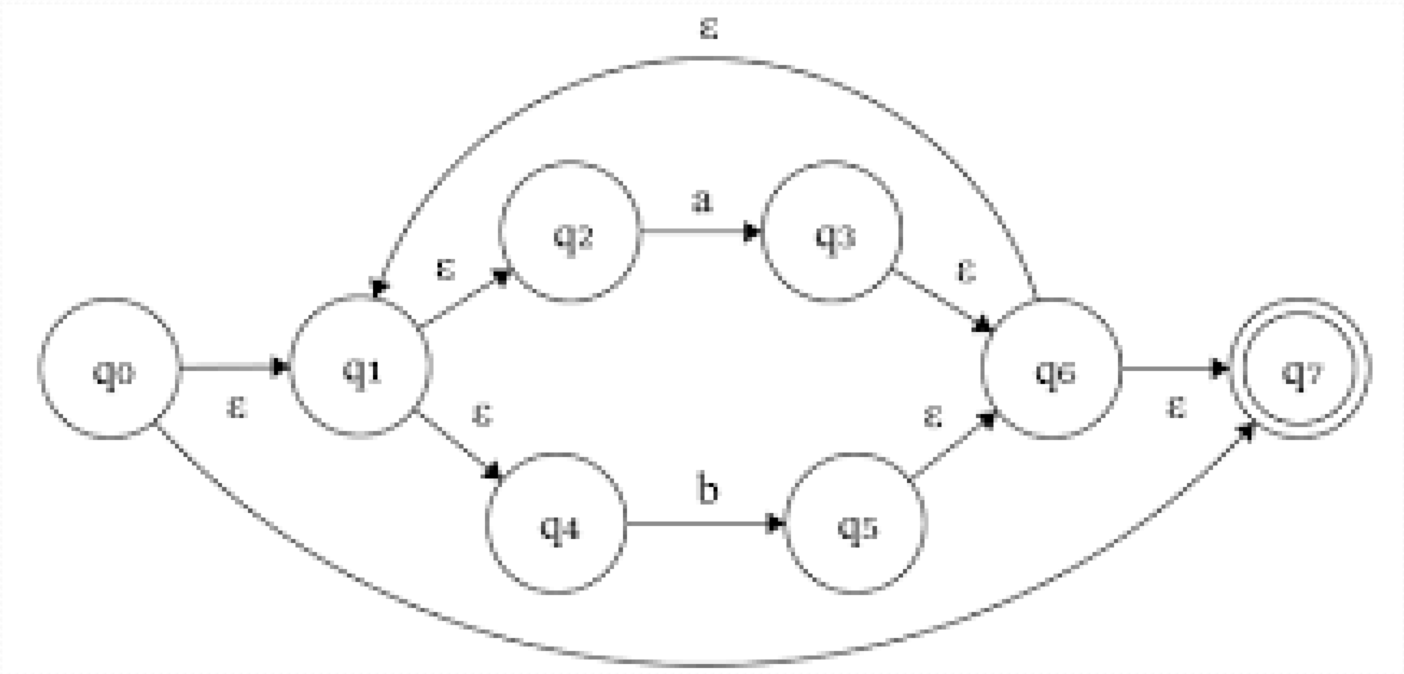
Formal Definition of NFAs with ε Moves

- An NFA-ε is a five-tuple: $N = (Q, \Sigma, \delta, q_0, F)$

Q	A finite set of states
Σ	A finite input alphabet
q_0	The initial/starting state, q_0 is in Q
F	A set of final/accepting states, which is a subset of Q
δ	A transition function, which is a total function from $Q \times \Sigma \cup \{\varepsilon\}$ to $P(Q)$

$$\delta: (Q \times (\mathbf{\Sigma} \cup \{\varepsilon\})) \rightarrow P(Q)$$

- A String w in Σ^* is *accepted* by NFA iff there exists a path in NFA from q_0 to a state in F labeled by w and zero or more ε transitions.
- Sometimes referred to as an NFA-ε other times, simply as an NFA.

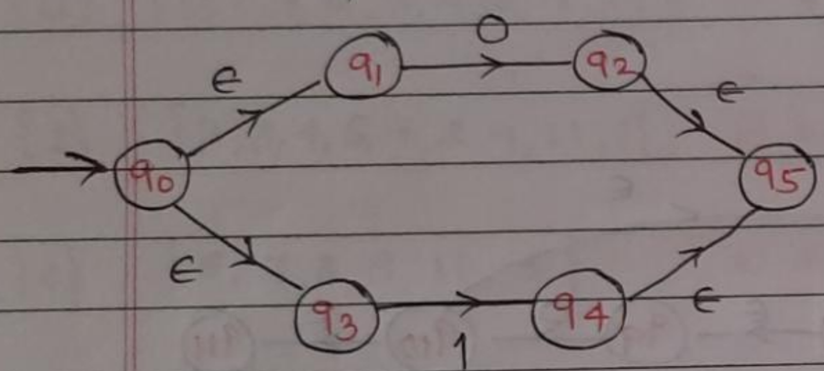

ε
q2
a
q3
ε
ε
q0
q1
q6
q7
ε
ε
ε
ε
q4
b
q5

- **Basic constructs of any RE :**
  - Union (+)
  - Concatenation(.)
  - Kleene Closure (*)
  - Positive Closure (+)

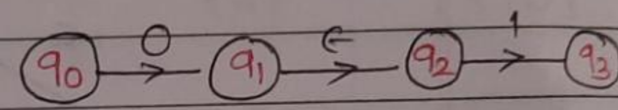
- Thomson's Rules

Basic Constructs using Thomson's Rules :

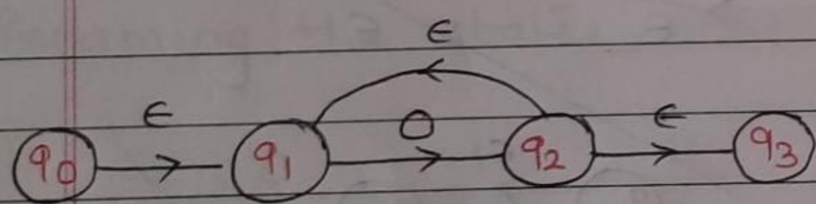
Parallel path, series (concatenation), closures can be represented using Thomson's Rules as -



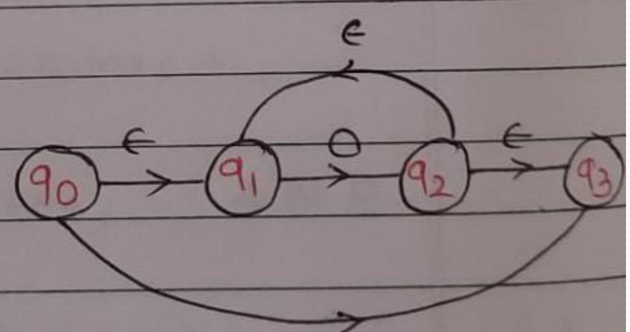
(a+b)



0.1



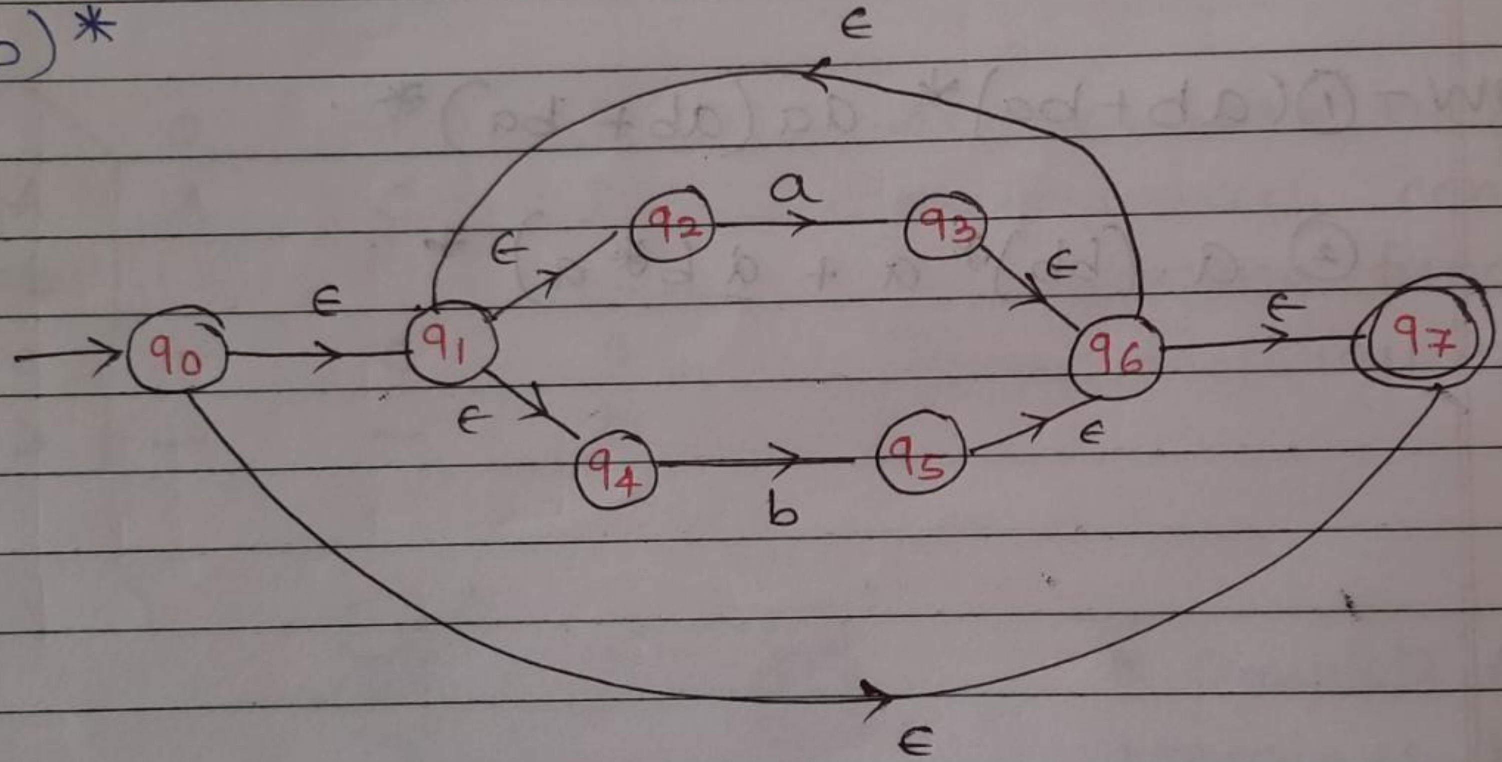
0^+



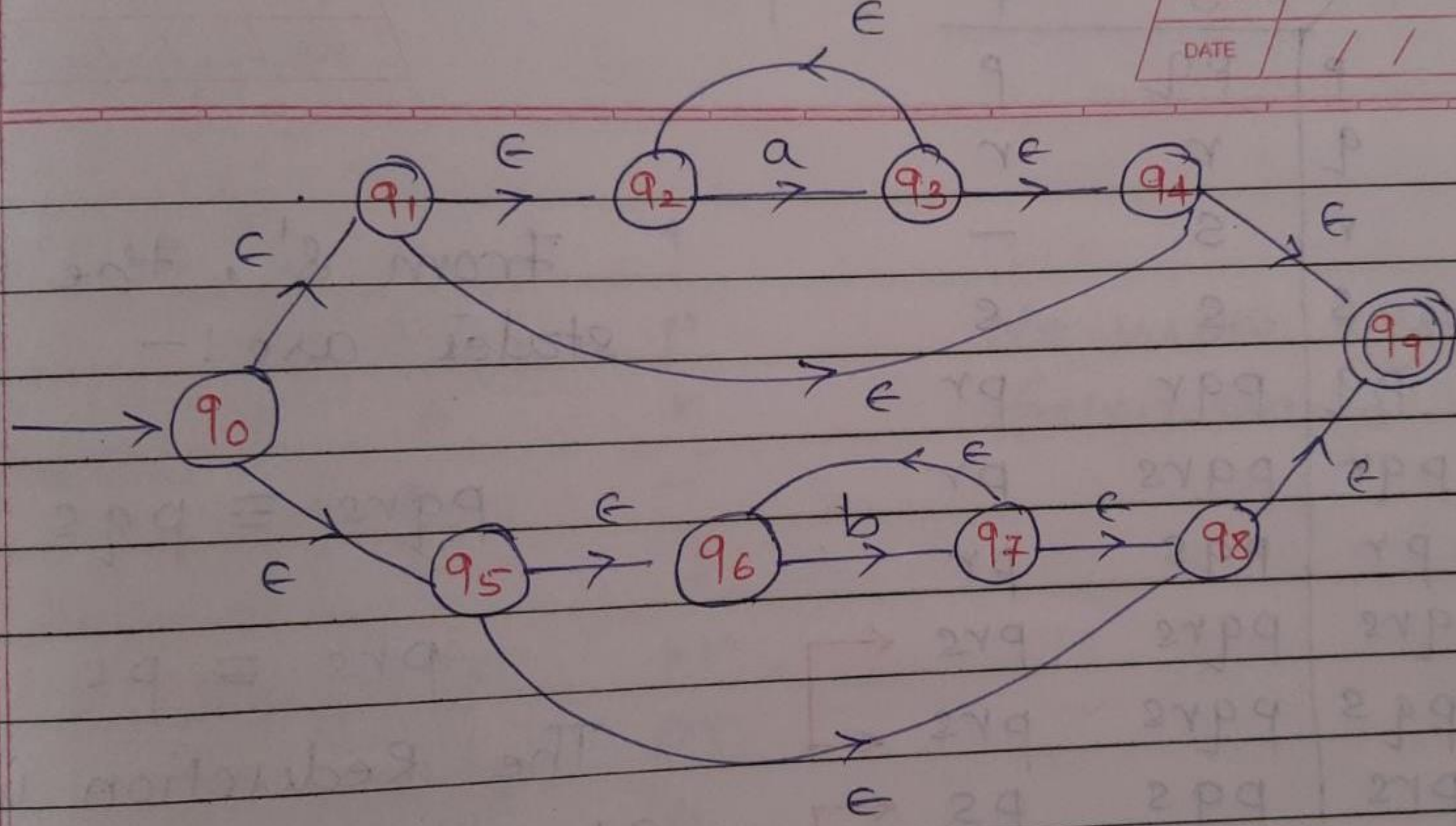
0^*

- Design NFA with € -moves for
  - (a+b)*
  - (a* + b*)
  - 0* 1* 2*
  - (0+ €) (10)* (€+1)

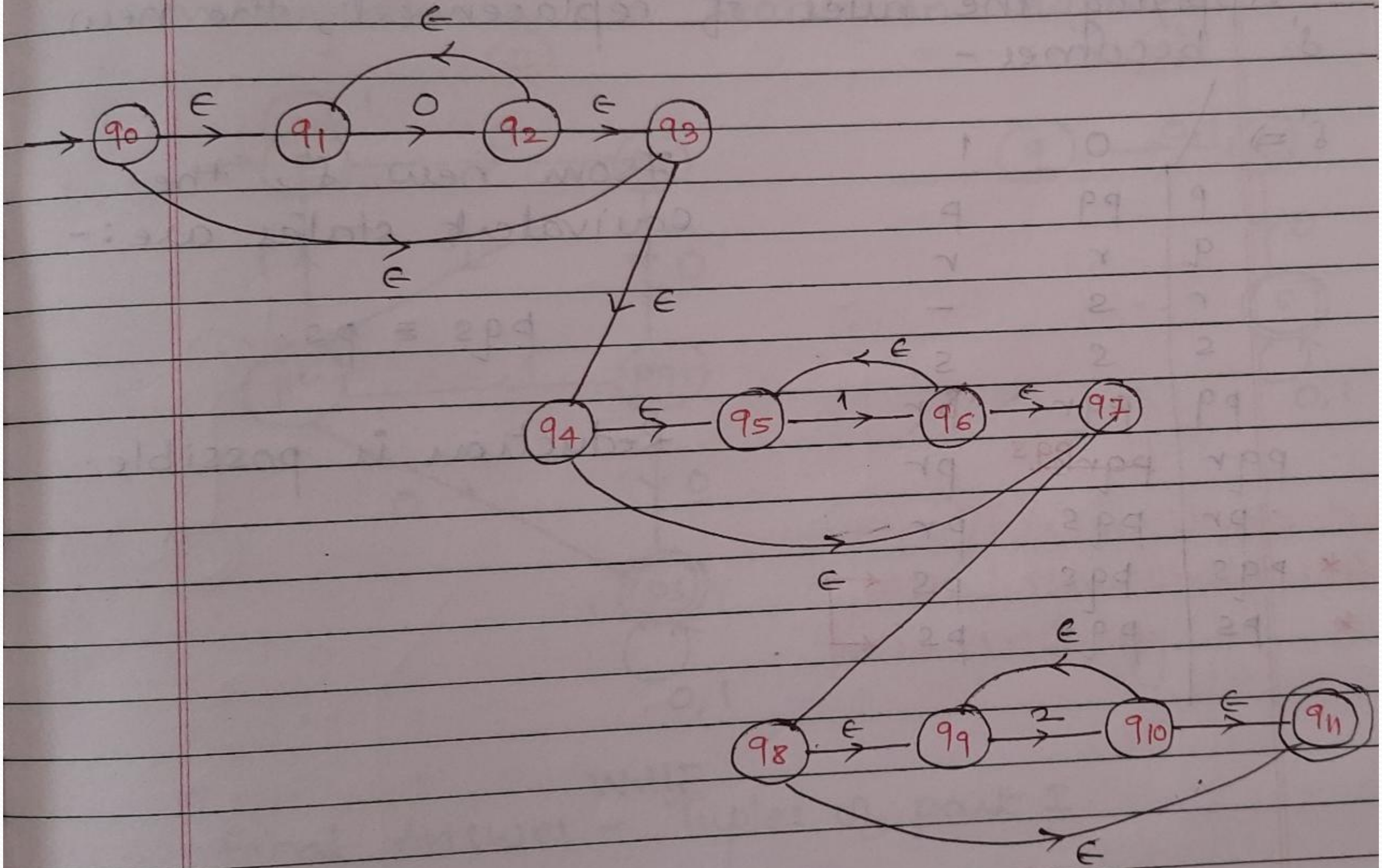
$(a+b)^*$



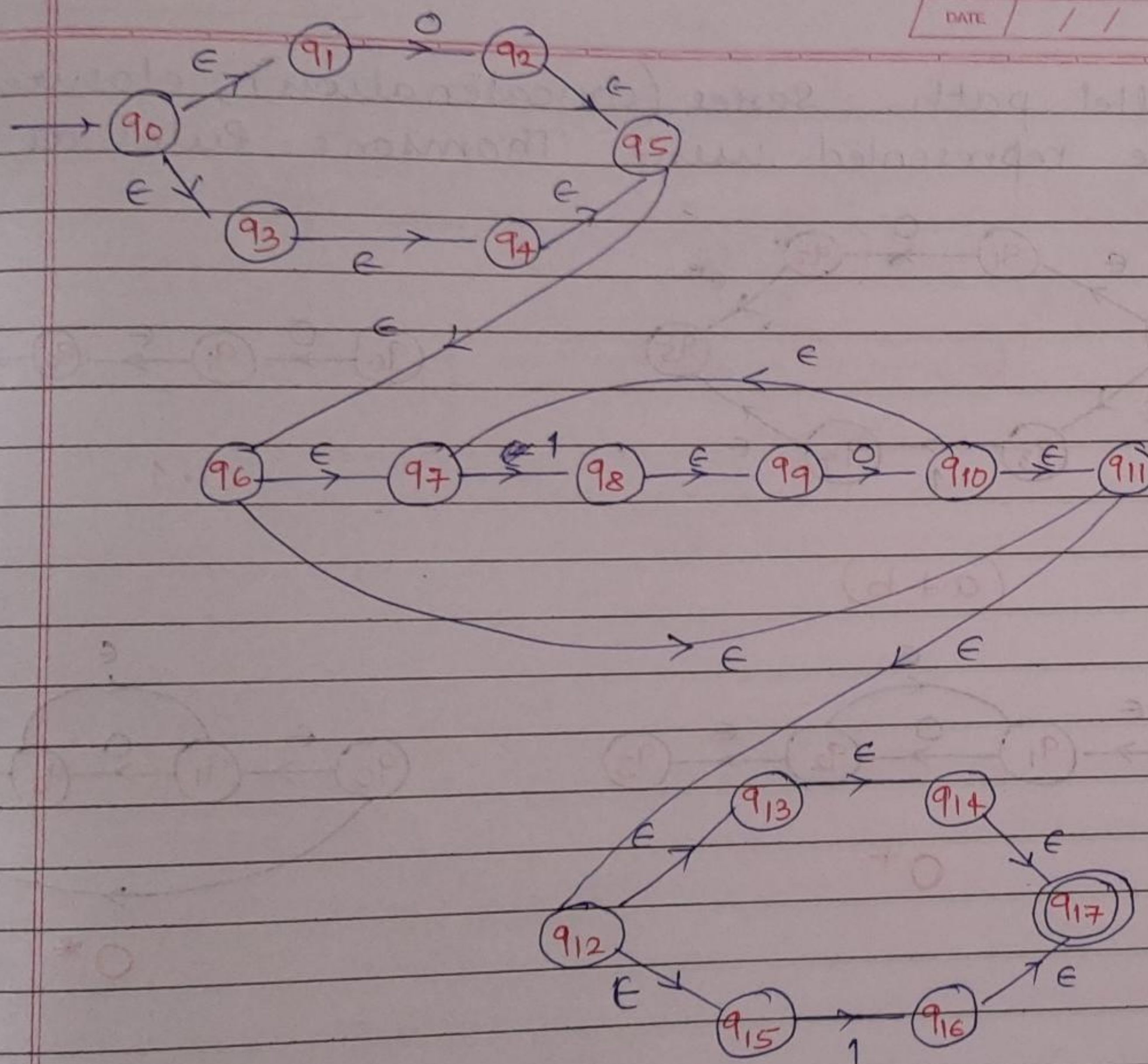
$a^* + b^*$



$0^* 1^* 2^*$



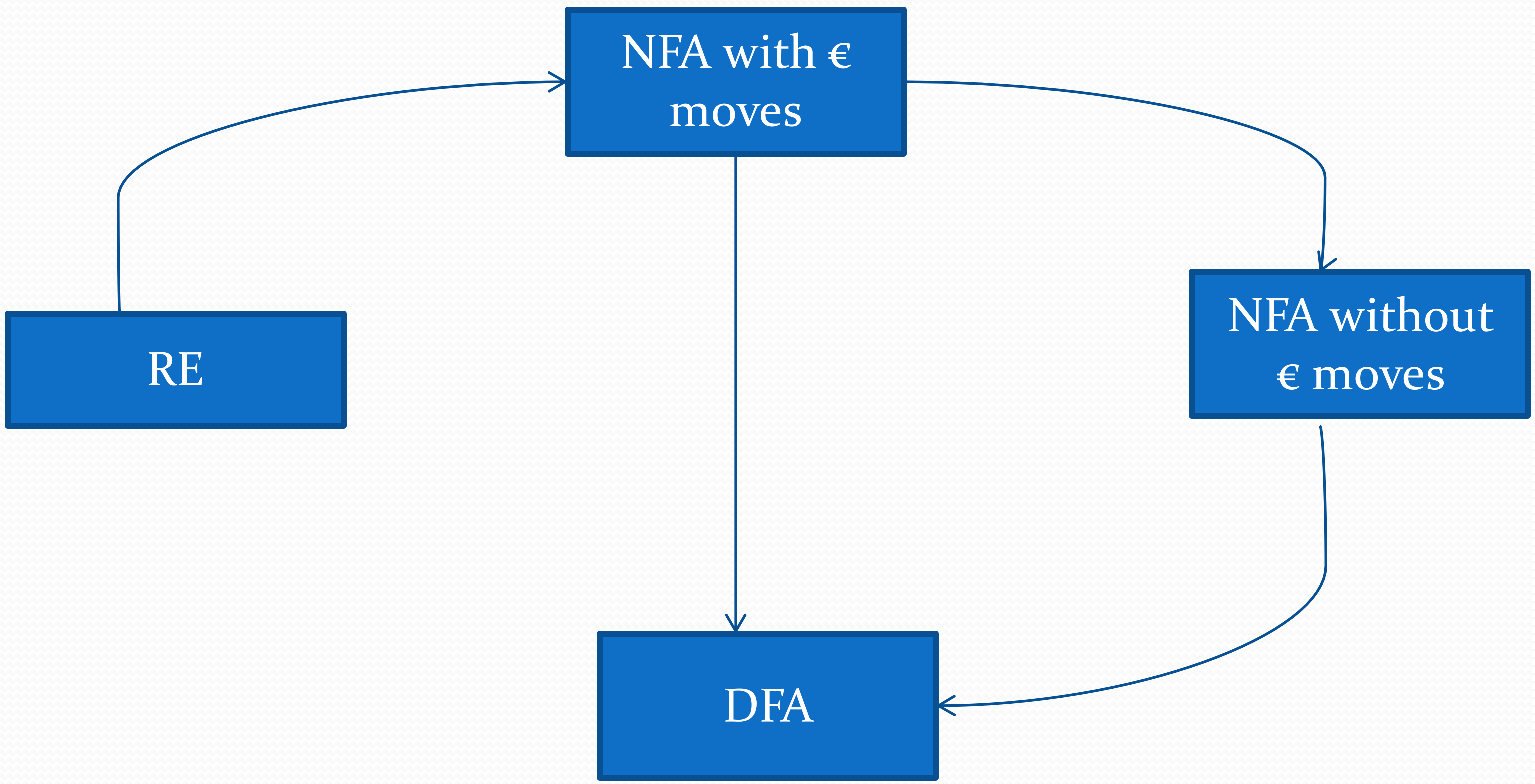
$(0+\epsilon)(10)^*(\epsilon+1)$



Homework :

- Design NFA with ϵ -moves for
  - $(ab + ba)^* \, aa \, (ab + ba)^*$
  - $(a(bb)^* a + a b^* a)^*$

RE to NFA



- Design DFA for the following REs
  - $0^* 1^* 2^*$
  - $(0+ \epsilon) (10)^* (\epsilon+1)$

$0^* 1^* 2^*$

x	$y = \epsilon\text{-closure}(x)$		$\delta'(y,0)$	$\delta'(y,1)$	$\delta'(y,2)$
{0}	{0,1,3,4,5,7,8,9,11}	A *	{2}	{6}	{10}
{2}	{2,3,4,5,7,8,9,11,1}	B *	{2}	{6}	{10}
{6}	{6,7,8,9,11,5}	C *	{ }	{6}	{10}
{10}	{10,9,11}	D *	{ }	{ }	{ }

Renaming the states, the δ' becomes-

	0	1	2
* A	B	C	D
* B	B	C	D
* C	-	C	D
* D	-	-	D

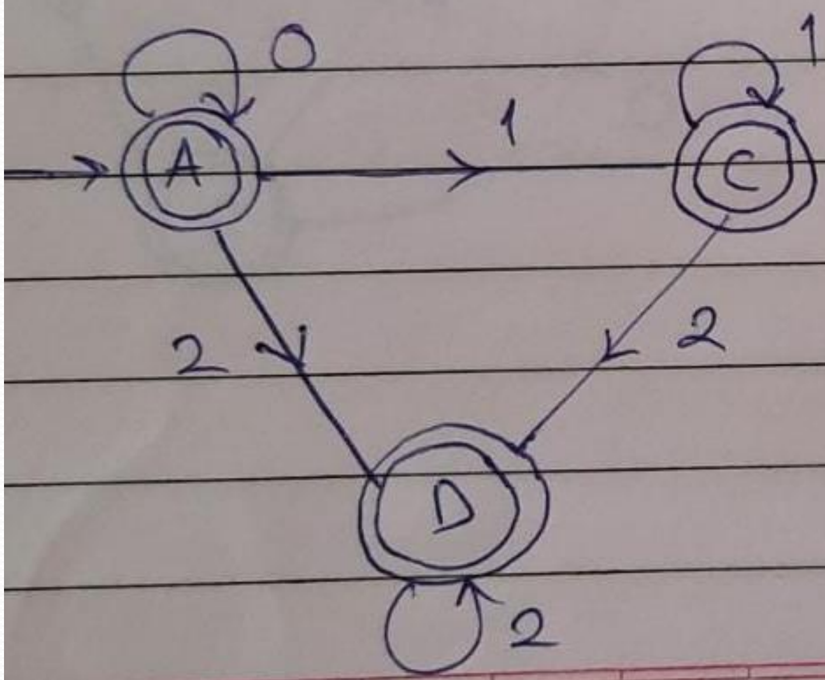
Here, $A \equiv B$.

$\therefore$ By applying Rules of Replacements, the new δ' becomes-

$\delta' \Rightarrow$

	0	1	2
*A	A	C	D
~~*B~~			
*C	–	C	D
*D	–	–	D

As there are no further equivalences, converting the mapping function to Transition Diagram-



* Complete Ans with remaining tuples.

* No examples needed.

- **(0+ €) (10)* (€+1)**

DATE / /

x	$y = \epsilon\text{-closure}(x)$		$\delta(y,0)$	$\delta(y,1)$
{0}	{0, 1, 3, 4, 5, 6, 7, 11, 12, 13, 15, 16, 17}	A *	{2}	{8, 16}
{2}	{2, 5, 6, 7, 11, 12, 13, 15, 16, 17}	B *	{∅}	{8, 16}
{8,16}	{8, 9, 16, 17}	C *	{10}	{∅}
{10}	{10, 11, 12, 13, 15, 16, 17}	D *	{∅}	{8, 16}

	0	1
* A	B	C
* B	—	C
* C	D	—
* D	—	C

B ≡ D

	0	1
* A	B	C
* B	—	C
* C	B	—