Start

classmate Date ____ Page ____

Extended Transition Function:-

$\delta : Q \times \Sigma \longrightarrow Q$ for DFA

$\delta : Q \times \Sigma \longrightarrow P(Q)$ for NFA.

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

$\hat{\delta} : Q \times \Sigma^* \longrightarrow P(Q)$ for NFA.

example

$\delta(q_0, a) = q_1$

$\hat{\delta}(q_0, abaab) = q_3$

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

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

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

$\hat{\delta}(q, \epsilon) = q$ ($\because$ not moved anywhere string of zero length)

$\hat{\delta}(q, w) = \bigcup_{p \text{ in } q} \hat{\delta}(p, w)$

Here, If q represents set of states q_0, q_1, q_2.

w is String

Σ is alphabet

Σ^* is string

(ϵ-NFA)

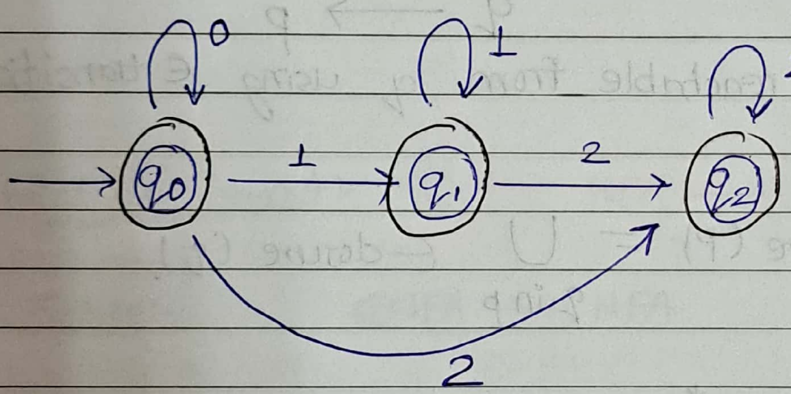
A more powerful machine :- (NFA with ϵ transitions)

- You do not take any character as input but still get to new state.

$$(p) \xrightarrow{\epsilon} (q)$$

ex :- $0^i \; 1^j \; 2^k \; ; \quad i, j, k \geq 0$

string like 00 11111 222



~~N~~DFA

Let find ϵ-NFA for above DFA :-

$$\epsilon\text{-NFA} = \langle Q, \Sigma, \delta, q_0, F \rangle$$

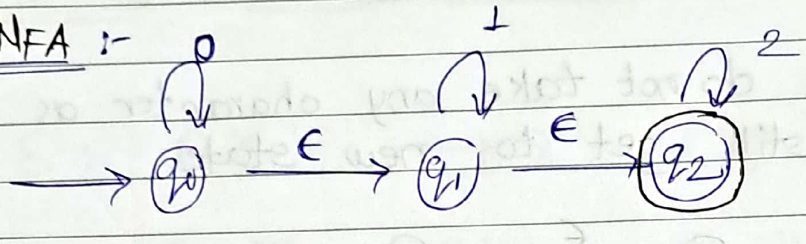
Transition function :- $\delta : Q \times \{\Sigma \cup \epsilon\} \longrightarrow \underbrace{P(Q)}_{\text{set of states}}$

$\epsilon\epsilon 0 \epsilon 1 \epsilon\epsilon\epsilon 2$
ex :- $\epsilon 0 1 2$] Just working
ex :- $0 \epsilon 1 \epsilon 2$

$\Downarrow$

$\boxed{012}$] Reality

ε-NFA :-



ε-closure Rules

Rule 1 ε-closure (q) denotes all the states p such that there is a path from q to p, labelled by

$$q \xrightarrow{\epsilon} p$$

ie. p is reachable from q using ε transition.

Rule 2

$$\epsilon\text{-Closure}(P) = \bigcup_{q \text{ in } p} \epsilon\text{-closure}(q)$$

Rule 3 For $\omega \in \Sigma^*$ and $a \in \Sigma$
(ω is string) (a is alphabet)

then

$$\hat{\delta}(q, \omega, a) = \epsilon\text{-closure}\left(\delta\left(\hat{\delta}(q, \omega), a\right)\right)$$

ex:- ε-closure(q_0) = $\{q_0, q_1, q_2\}$
ε-closure(q_1) = $\{q_1, q_2\}$
ε-closure(q_2) = $\{q_2\}$

Q \ $(\Sigma \cup \epsilon)$	0	1	2	ϵ
q_0	q_0	ϕ	ϕ	q_1
q_1	ϕ	q_1	ϕ	q_2
q_2	ϕ	ϕ	q_2	ϕ

dead end.

What we learnt :- NFA $\longrightarrow$ DFA

Future :- ϵNFA $\longrightarrow$ NFA

$$NFA = \langle Q, \Sigma, \delta_N, q_0, F_N \rangle$$

$$\epsilon NFA = \langle Q, \Sigma, \delta_\epsilon, q_0, F_\epsilon \rangle$$

Steps :-

- Find ϵ-closure of a state
- Read an alphabet on ϵ-closure & take union.

ex) - $q_0 \xrightarrow{0} q_0$

$q_0 \xrightarrow{0} \xrightarrow{\epsilon} q_1$

$q_0 \xrightarrow{0} \xrightarrow{\epsilon} \xrightarrow{\epsilon} q_2$

$(q_0, q_1, q_2) \xrightarrow{0} [(q_0, q_1, q_2), \phi, \phi] \xrightarrow{\text{union}} [q_0, q_1, q_2]$

$(q_0, q_1, q_2) \xrightarrow{1} [(q_1, q_2), (q_1, q_2), (\phi)] \rightarrow [q_1, q_2]$

- conversion Transition table:-

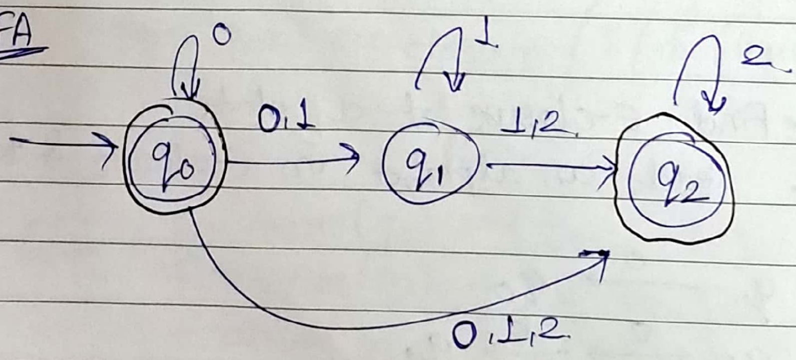
Q \ Σ	0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	ϕ	$\{q_1, q_2\}$	$\{q_2\}$
q_2	ϕ	ϕ	$\{q_2\}$

★ Initial state will be added only under a condition:-

If ε-closure of q_0 (initial) state includes any 1 F-ε states.

ie You can directly jump from q_0 to q_2

NFA



Algorithm to convert ε-NFA → DFA :-

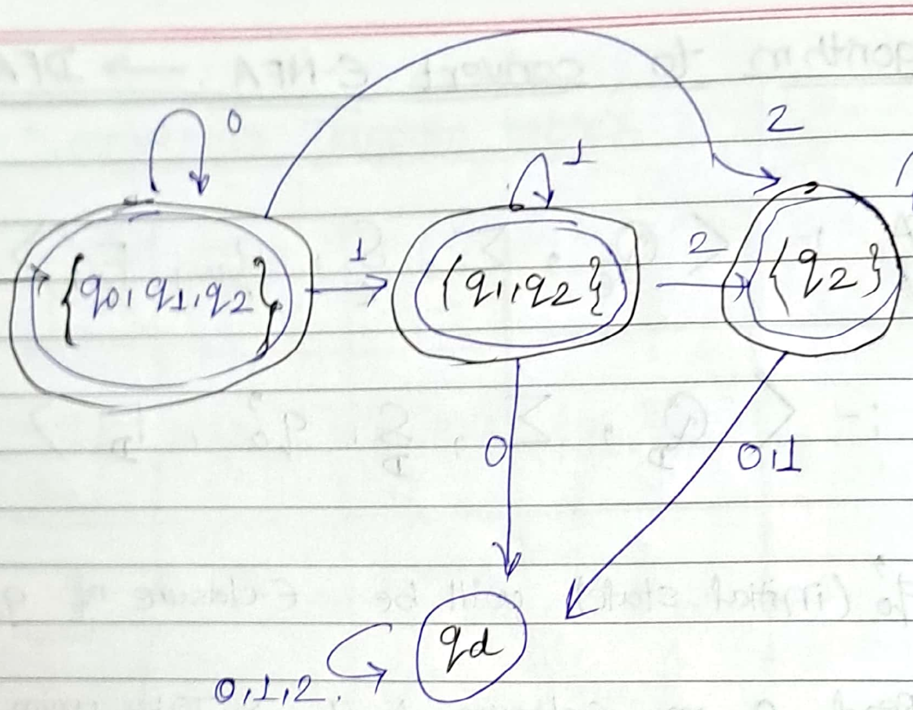
ε-NFA :- $\langle Q_\epsilon, \Sigma, \delta_\epsilon, q_0, F_\epsilon \rangle$

DFA :- $\langle Q_D, \Sigma, \delta_D, q_0', F_D \rangle$

- q_0' (initial state) will be ε-closure of q_0

★ Read 0 on εclosure of q_0 ⇒ Take union
— 11 — 1 on —— 11 ———— ⇒ Take union.
— 11 — 2 on —— 11 ———— ⇒ Take union.

		0	1	2
εclosure q_0	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
ε-closure q_1	$\{q_1, q_2\}$	q_d	$\{q_1, q_2\}$	$\{q_2\}$
εclosure q_2	$\{q_2\}$	q_d	q_d	$\{q_2\}$
Extra	q_d	q_d	q_d	q_d



- ϵ is allowed to be an answer

  Hence {q0, q1, q2} can be final state

- only 1 allowed ⇒ {q1, q2} also Final
- only 2 allowed ⇒ {q2} also final.

- All final states of NFA, if present in DFA, will become final state in DFA.

  ex - q0, q2 was final in NFA

  ∘) If any one of them present in DFA, it becomes final state

- Miley & Murrey machine control output of display] Homework for Today

So far :- 3 automata studied [DFA, NFA, ϵ-NFA]

Can we have algorithm to minimize DFA?

ans :- $\boxed{\text{YES}}$

Minimization of Finite Automata :-

1) 2 states q_1 and q_2 are equivalent $q_1 \equiv q_2$ if

$$\hat{\delta}(q_1, x) \ \& \ \hat{\delta}(q_2, x)$$

(it means reading same string on different states)

are final states or both of them are Non-final states $\forall x \in \Sigma^*$.

2) 2 states q_1 and q_2 are k-equivalent if both $\hat{\delta}(q_1, x)$ & $\hat{\delta}(q_2, x)$ are final states or non-final states for all strings x of length atmost k.

ie $|x| \le k$

Equivalent relation (Reflexive, symmetric, Transitive) say we have n-states

$Q = \langle Q_1, Q_2, Q_3, \ldots Q_N \rangle$

we can make partition of states into final & non-final groups, etc.

$\pi_0 \Rightarrow$ 0 character

$\pi_1 \Rightarrow$ 1 character

$\pi_2 \Rightarrow$ 2 character

$\vdots$

$\pi_k \Rightarrow$ k character

$\downarrow$

$\pi_{k+1} \Rightarrow (k+1)$ character $= \pi_k$

ie keep continuing the group until

$\boxed{\pi_{k+1} = \pi_k}$

k is the reduced number of states

Let see DFA of 8 states as example.

Q \ Σ	a	b
q_0	q_1	q_5
q_1	q_6	q_2
q_2	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

$\Pi_0 = \langle Q_1^0, Q_2^0 \rangle$

$Q_1^0 = \{q_2\}$, $Q_2^0 = \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}$

$\Pi_1 = \langle Q_1^1, Q_2^1, Q_3^1, Q_4^1$

$Q_1^1 = \{q_2\}$

$Q_1^1 = \{q_2\}$

$Q_2^1 = \{q_0, q_4, q_6\}$

$Q_3^1 = \{q_1, q_7\}$

$Q_4^1 = \{q_3, q_5\}$.

$\Pi_2 = \langle Q_1^2, Q_2^2, Q_3^2, Q_4^2$

$Q_1^2 = \{q_2\}$

$Q_2^2 = \{q_0, q_4\}$

$Q_3^2 = \{q_6\}$

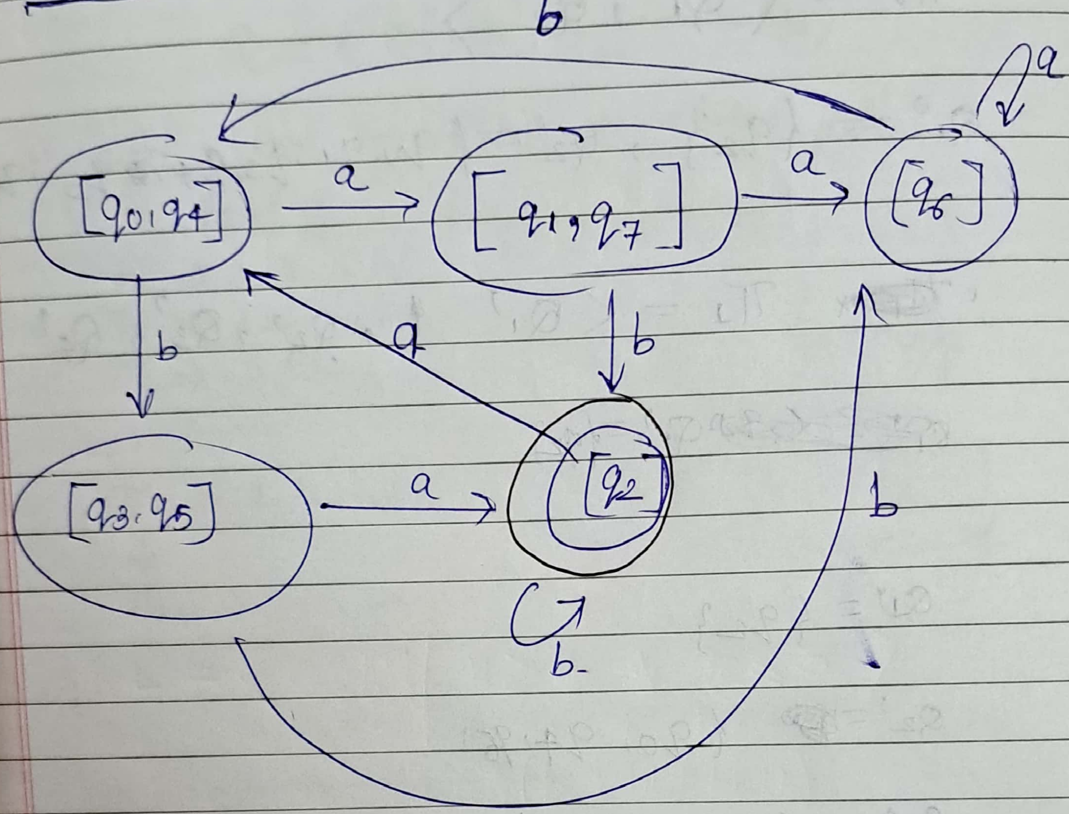
$Q_4^2 = \{q_1, q_7\}$

$Q_5^2 = \{q_3, q_5\}$

Now, Π_3 will also have same parts

Hence

Now, use this groups as minimized states



Regular Expressions

1) Σ

2) ϵ

3) (,)

4) $*$

5) $+$

$r_1 = a(a+b)^*$

$r_2 = (a+b)^* aba$

$r_3 = (a+b)^* aba (a+b)^*$

$r_4 = (a+b)^* (aa+bb) (a+b)^*$

$r_5 = a(a+b)^* b + b(a+b)^* a$

$r_6 = (a+b)^* a (a+b) (a+b)$

$r_7 =$ (even no. of a & even no. of b) -- skip

Only aba selected

$r_8 = aba$

$r_9 = ((a+b)(a+b)(a+b))^*$

Q] Language which should accept 1 'a' anywhere?

ans:- 1) $(a+b)^* a (a+b)^*$

ans 2) $b^* a b^* (a^* b^*)^*$

Q] Exactly 2 b's?

ans:- $a^* b a^* b a^*$

Q] $L = \{ a^n b^m \mid n \geq 4 \ \& \ m \leq 3 \}$

ans:- $aaaa a^* (b + bb + bbb + \epsilon)$

(or)

$aaaa a^* (b+\epsilon)(b+\epsilon)(b+\epsilon)$

Q] Strings composed of a and b; and at most 3 a.

ans:- $b^* (\epsilon + a) b^* (\epsilon + a) b^* (\epsilon + a) b^*$

Q] $L = \{a^n b^m \mid n = \text{even},\ m \geq 0\}$

ans:- $(aa)^* b^*$

Q] $L = \{a^n b^m \mid n+m = \text{even}\}$

ans:- $\boxed{a(aa)^* b(bb)^* + (aa)^*(bb)^*}$ ~~Research~~

(or)

$\boxed{(aa)^* (ab+\epsilon)(bb)^*}$

Q] String composed of a, b ~~and c~~.
Atleast 1 occurence of each symbol should be there

ans:- $(a+b)^* a (a+b)^* b (a+b)^* + (a+b)^* b (a+b)^* a (a+b)^*$

or

$(a+b)^* (ab+ba) (a+b)^*$

Q] string composed of a, b & c, atleast 1 each occurs.

$x^* a\ x^* c\ x^* b\ x^*$

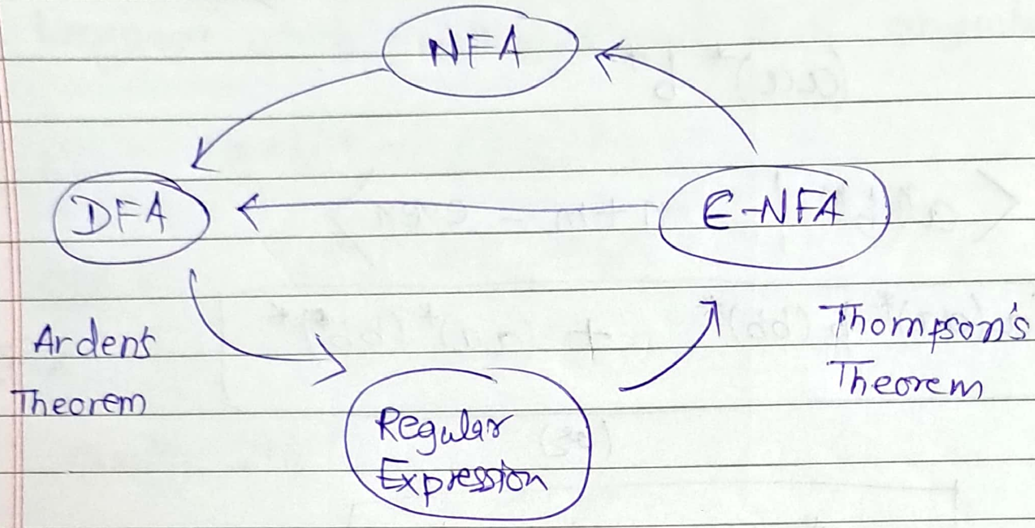
$x^* a\ x^* b\ x^* c\ x^*$

$x^* b\ x^* a\ x^* c\ x^*$

$x^* b\ x^* c\ x^* a\ x^*$

$x^* c\ x^* a\ x^* b\ x^*$

$x^* c\ x^* b\ x^* a\ x^*$



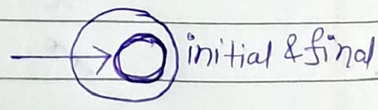
Let r be a regular expression
Then there exist a ϵ-NFA
which accepts $L(r)$

<u>Proof</u> :- Mathematical
Induction on the no. of operators

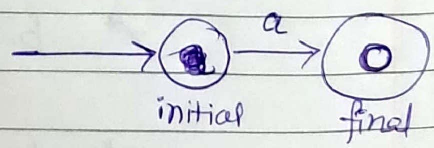
in Regular expressions are:-

1) There is ϵ-NFA having 1 final state and
no transition goes out of this final state

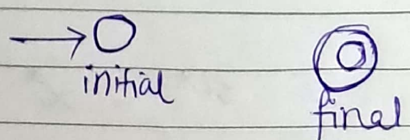
<u>Basis</u> :- $r_1 = \epsilon$



$r_2 = a$



$r_3 = \phi$



classmate

Date

Page

2) Now, we will increase operators 1 at a time.

Assume that the theorem is true for the regular expression with fewer than i operators $(i \geq 1)$

Let r have "i" operator, then there exist 3 case.

case 1) $r = r_1 + r_2$

case 2) $r = r_1 r_2$

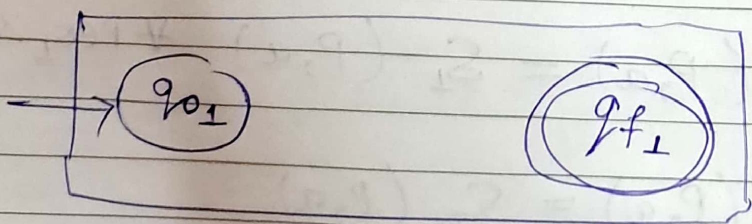
case 3) $r = r_1^*$

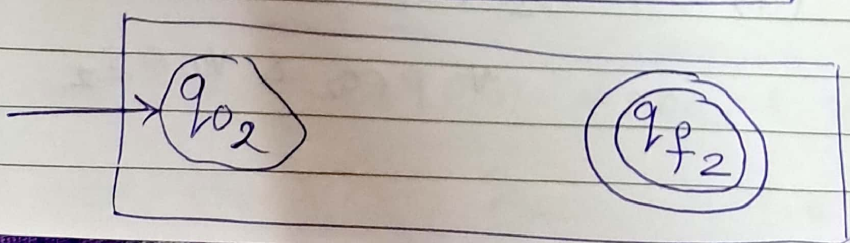
r_1 $L(r_1)$ M_1

r_2 $L(r_2)$ M_2

$M_1 = \langle Q_1, \Sigma_1, \delta_1, q_{0_1}, F_1 \rangle$

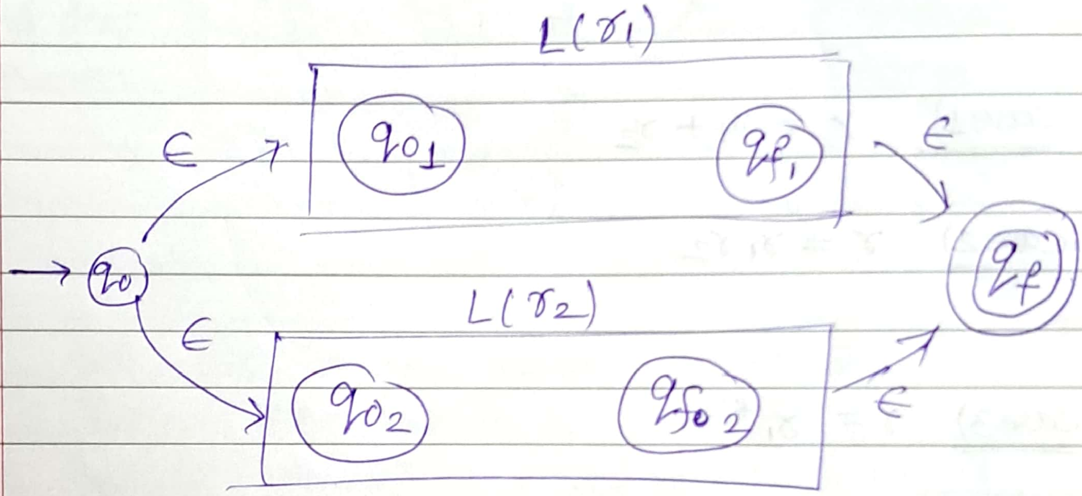
$M_2 = \langle Q_2, \Sigma_2, \delta_2, q_{0_2}, F_2 \rangle$





case 1) $r_1 + r_2$

$M_{union} = \langle Q_{union}, \Sigma_{union}, \delta_{union}, q_{0\,union}, F_{union} \rangle$



$Q_{union} = Q_1 \cup Q_2 \cup \{q_0, q_f\}$

$\Sigma_{union} = \Sigma_1 \cup \Sigma_2 \cup \{\epsilon\}$

$F_{union} = \{q_f\}$

1) $\delta_{union}(q_0, \epsilon) = \{q_{0_1}, q_{0_2}\}$

2) $\delta_{union}(P, a) = \delta_1(P, a) \quad \forall P \in Q_1 \,\&\, \forall a \in \Sigma_1$

3) $\delta_{union}(P, a) = \delta_2(P, a)$

$\forall P \in Q_2 \,\&\, \forall a \in \Sigma_2$

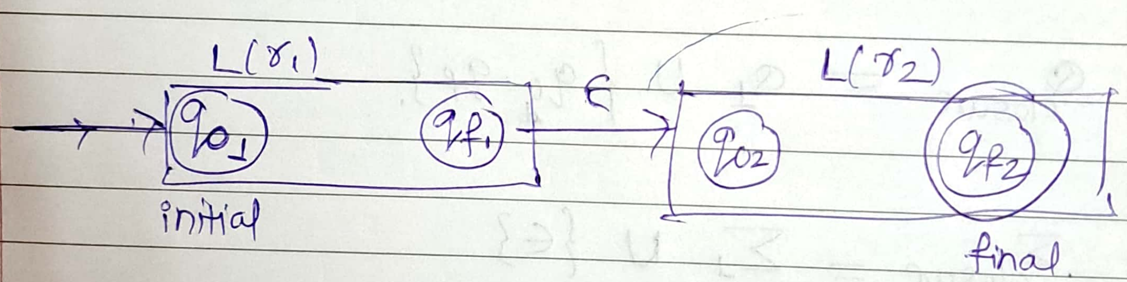
4) $\delta_{union}(q_{f_1}, \epsilon) = q_f \quad \forall q_{f_1} \in F_1$

5) $\delta_{union}(q_{f_2}, \epsilon) = q_f \quad \forall q_{f_2} \in F_2$

— x —

<u>case 2)</u> $r_1 r_2$

$$M_{concatenation} = \langle Q_{concat}, \Sigma_{concat}, \delta_{concat}, q_{0_{concat}}, F_{concat} \rangle$$



$Q_{concat} = Q_1 \cup Q_2$

$\Sigma_{concat} = \Sigma_1 \cup \Sigma_2 \cup \{\epsilon\}$

$F_{union} = F_2$

1) $\delta_{concat}(p, a) = \delta_1(p, a) \quad \forall p \in Q_1 \ \& \ \forall a \in \Sigma_1$

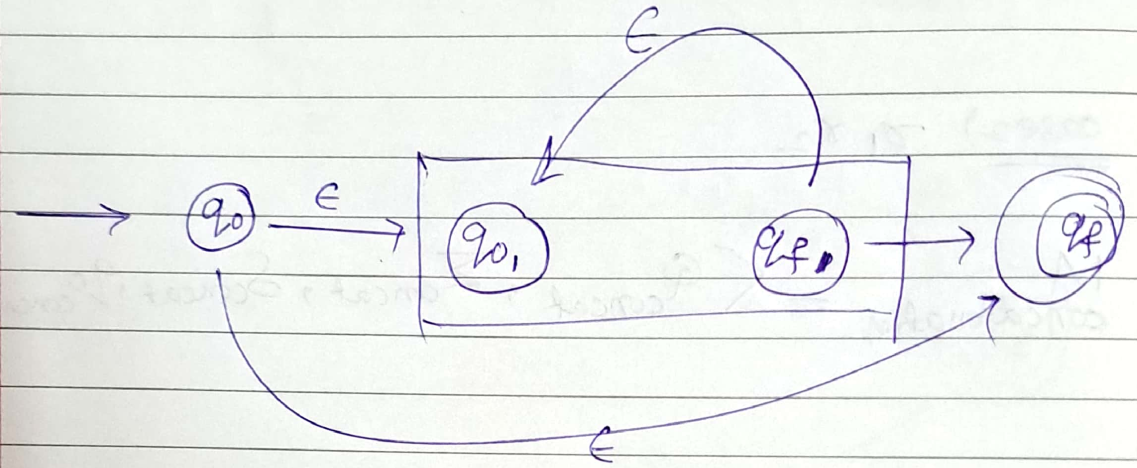
All transitions of both machines are retained.

2) $\delta_{concat}(p, a) = \delta_2(p, a) \quad \forall p \in Q_2 \ \& \ \forall a \in \Sigma_2$

3) $\delta_{concat}(q_{f_1}, \epsilon) = q_{0_2} \quad \forall q_{f_1} \in F_1$

case 3 $\quad r = r_1^*$

$$M_{closure} = \langle Q_{closure}, \Sigma_{closure}, \delta_{closure}, q_{0_{closure}}, F_{closure} \rangle$$



$$Q_{closure} = Q_1 \cup \{q_0, q_f\}.$$

$$\Sigma_{closure} = \Sigma_1 \cup \{\epsilon\}$$

$$F_{closure} = \{q_f\}$$

1) $\delta_{closure}(q_0, \epsilon) = \{q_{0_1}, q_f\}$

2) $\delta_{closure}(P, a) = \delta_1(P, a) \quad \forall p \in Q_1 \;\&$
$\forall a \in \Sigma_1$

3) $\delta_{closure}(q_{f_1}, \epsilon) = \{q_f, q_{01}\}$