Notation

Σ / Γ : a finite set of symbols called the alphabet

A string | word over Σ is a finite seq^n of elements of Σ & is written as follows:

$W = a_1 a_2 \ldots a_n$. The a_i's $\in \Sigma$ & are called letters

Σ^* : set of all strings over Σ. x, y, z, u, v, w denote strings

λ / ε : null string

Notation

Σ / Γ : a finite set of symbols called the alphabet

A string/word over Σ is a finite seq^n of elements of Σ & is written as follows:

$$w = a_1 a_2 \ldots a_n$$

. The a_i's $\in \Sigma$ & are called letters

Σ^* : set of all strings over Σ. x, y, z, u, v, w denote strings

λ / ε : null string. Note that $\lambda \in \Sigma^*$

Let $w_1 = a_1 a_2 \cdots a_m$ (Note m is called the length of w & is denoted by $|w|$)

$w_2 = b_1 b_2 \cdots b_n$.

$$w_1 * w_2 = a_1 \cdots a_m b_1 \cdots b_n.$$

Clearly $\lambda * w = w * \lambda = w$.

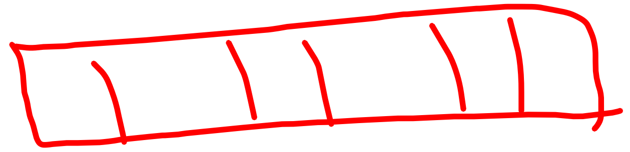
Any subset $\mathcal{L} \subseteq \Sigma^*$ is called a language.

Let $\mathcal{L}_1, \mathcal{L}_2 \subseteq \Sigma^*$ be two languages.

Define $\mathcal{L}_1 * \mathcal{L}_2 = \{ w_1 * w_2 : w_1 \in \mathcal{L}_1 \ \& \ w_2 \in \mathcal{L}_2 \}$

Given $\mathcal{L} \subseteq \Sigma^*$, we define the Kleene closure as follows:

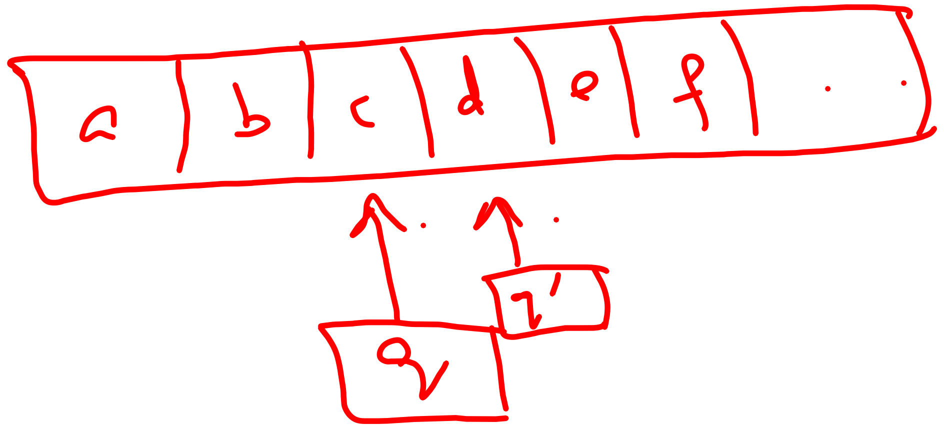
$$\mathcal{L}^* = \{ w_1 * w_2 * \cdots * w_k : k \geq 0 \;\&\; w_i \in \mathcal{L} \}.$$



Finite State Automaton

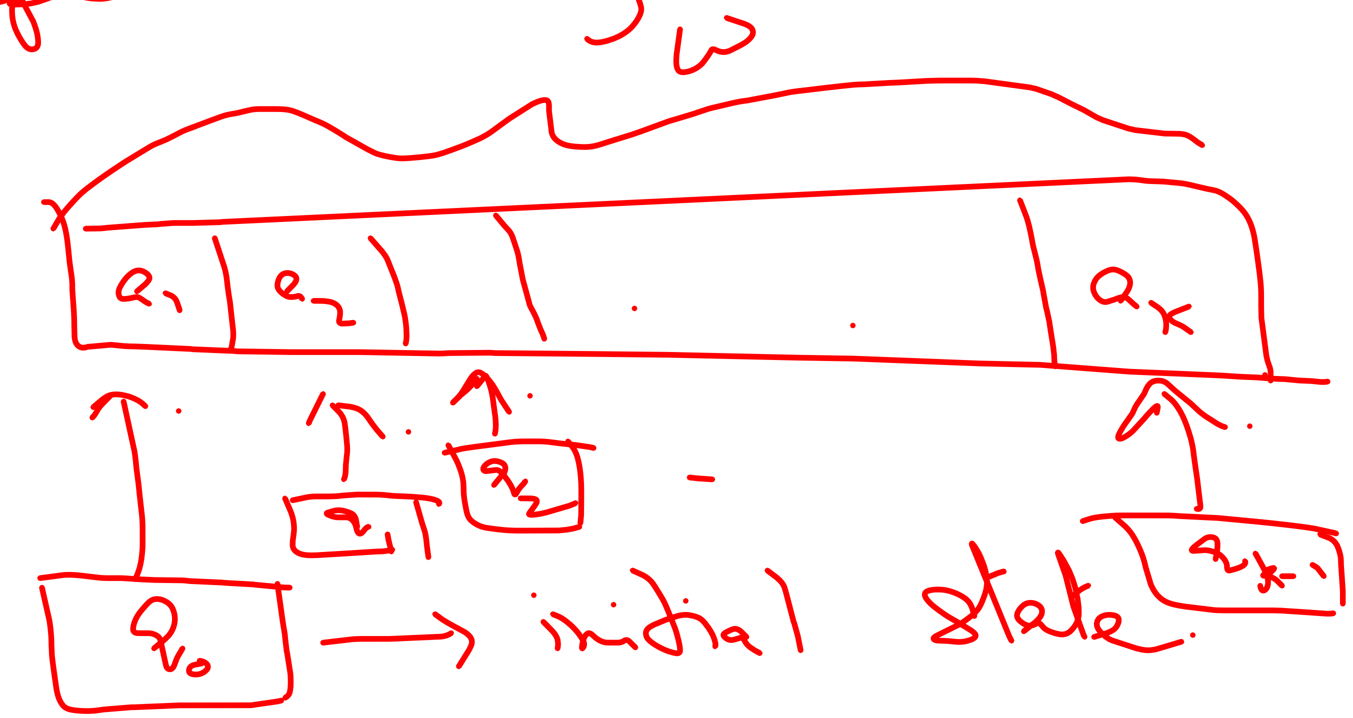
A deterministic finite automata (DFA) consists of a finite tape with cells & a tapehead with a finite control. Each cell of the tape contains a letter from some alphabet Σ

At each instant, the tapehead reads the content of a cell & the finite is in some particular state



At each instant, depending on the letter being scanned by the tapehead & the state of the finite control, the DFA moves the tapehead one cell to the right & perhaps enter a new state q'

Initially the DFA M starts with the tape holding some string w.



Formally a DFA M is a 5-tuple

$(\Sigma, Q, q_0, \delta, F)$, where

Σ : is a finite alphabet

Q : a finite set of <u>states</u> of M.

$q_0 \in Q$ is the initial state.

$F \subseteq Q$ is the set of <u>accepting</u> or <u>final</u> states

& $\delta : Q \times \Sigma \longrightarrow Q$ is the transition fn.

$$\delta(q, a) = q'$$

We extend δ to Σ^* as follows:

$$\delta^* : Q \times \Sigma^* \longrightarrow Q.$$

(1) $\delta^*(q, \lambda) = q$.

& (2) $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$

Ex $\delta^*(q, w_1 w_2)$

$= \delta^*(\delta^*(q, w_1), w_2)$.

Defⁿ A string w is accepted by M if $\delta^*(q_0, w) \in F$.

The lang accepted by M is

$$L(M) = \{ w \in \Sigma^* : \delta^*(q_0, w) \in F \}$$

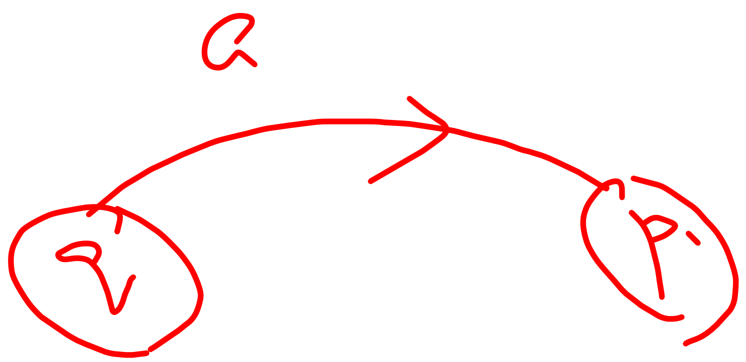
A lang L is said to be regular if it is accepted by some DFA M i.e.

$$L(M) = L.$$

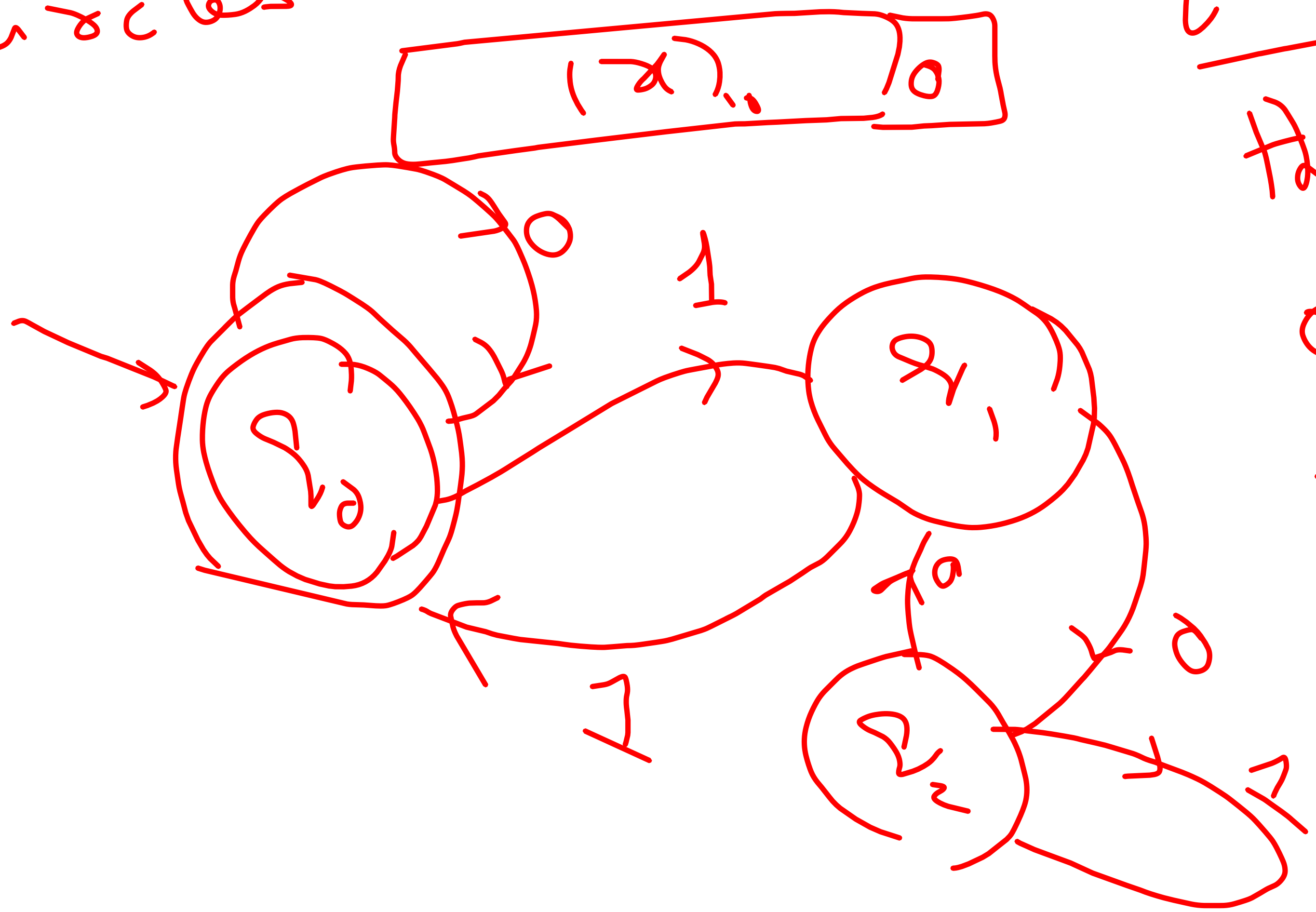
State transition diagram of M is

is a directed graph whose nodes are labeled by the states of M & there is a directed edge, from a node labeled q to another node labeled p, labeled a if

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

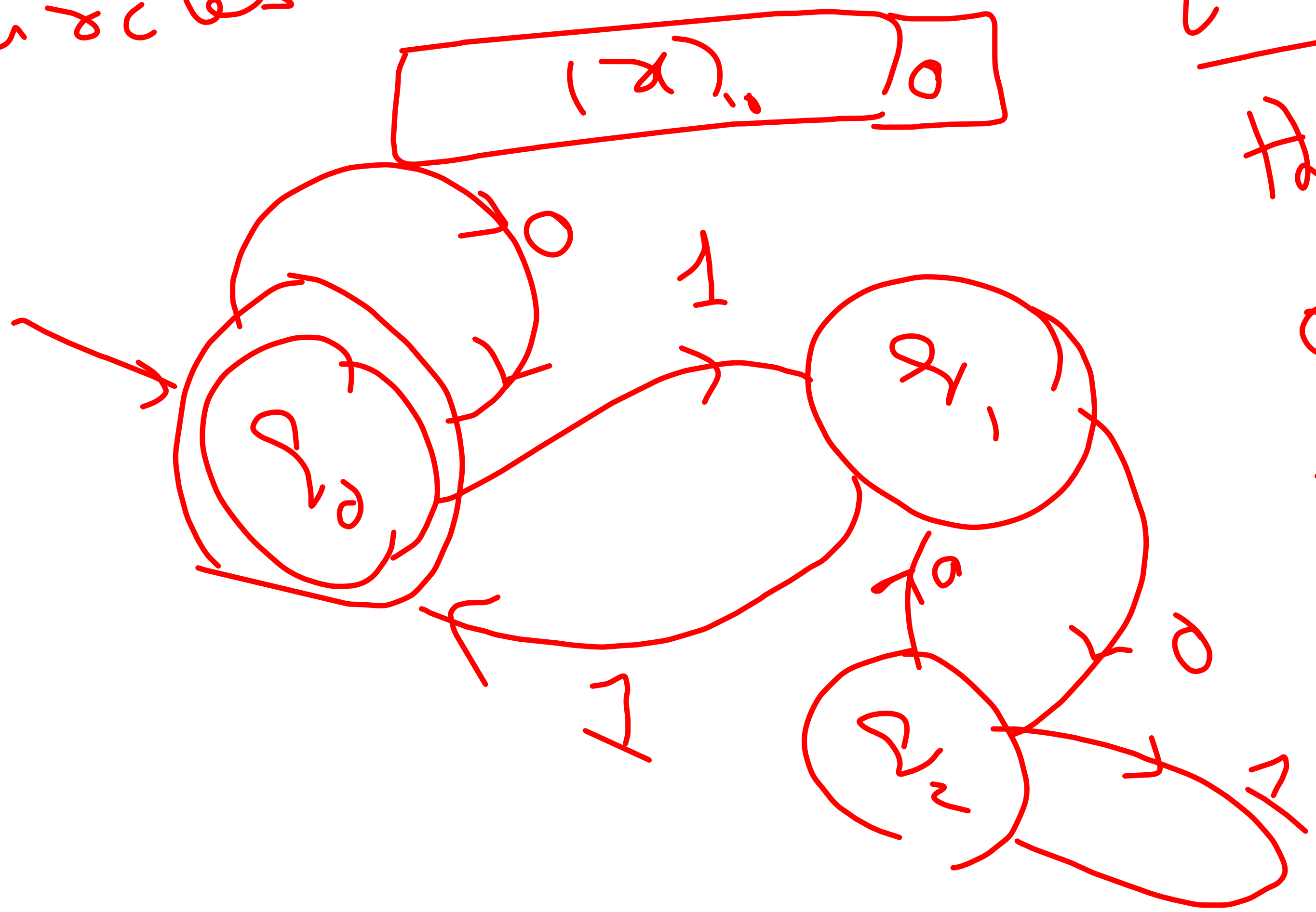


The initial state has an arrow → pointing to it & the accepting states are encircled with 2 circles



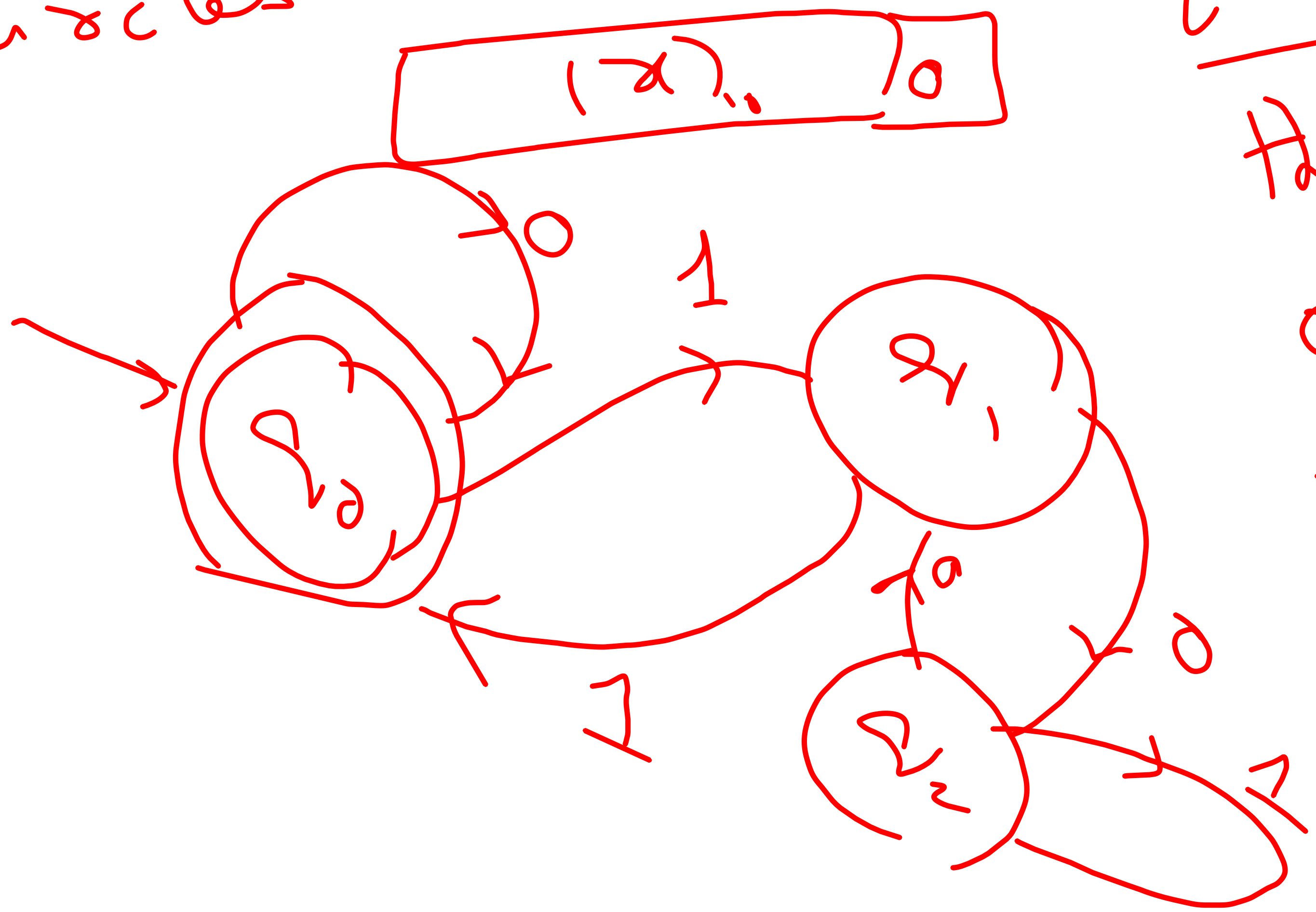
Ex 1 Construct a DFA M that accepts all strings over $\{0,1\}$ whose decimal representation is divisible by 3

The initial state has an arrow → pointing to it & the accepting states are encircled with 2 circles



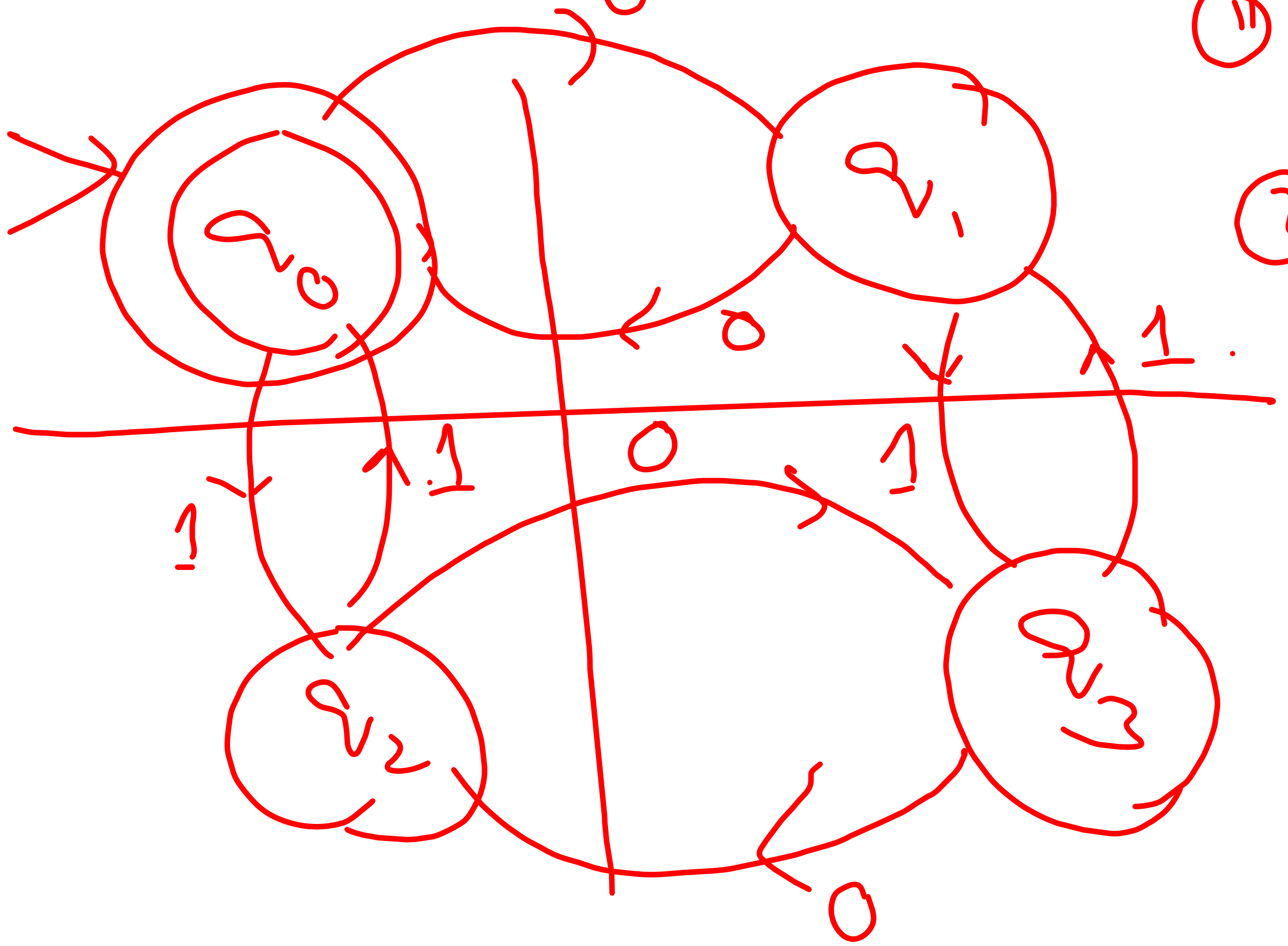
Ex 1 Construct a DFA M that accepts all strings over $\{0,1\}$ whose decimal representation is divisible by 3

The initial state has an arrow → pointing to it & the accepting states are encircled with 2 circles



Ex 1 Construct a DFA M that accepts all strings over {0,1} whose decimal representation is divisible by 3

Ex 2. Construct a DFA that accepts all strings over $\{0,1\}$ that contains a even no. of 0's & even no. of 1's.

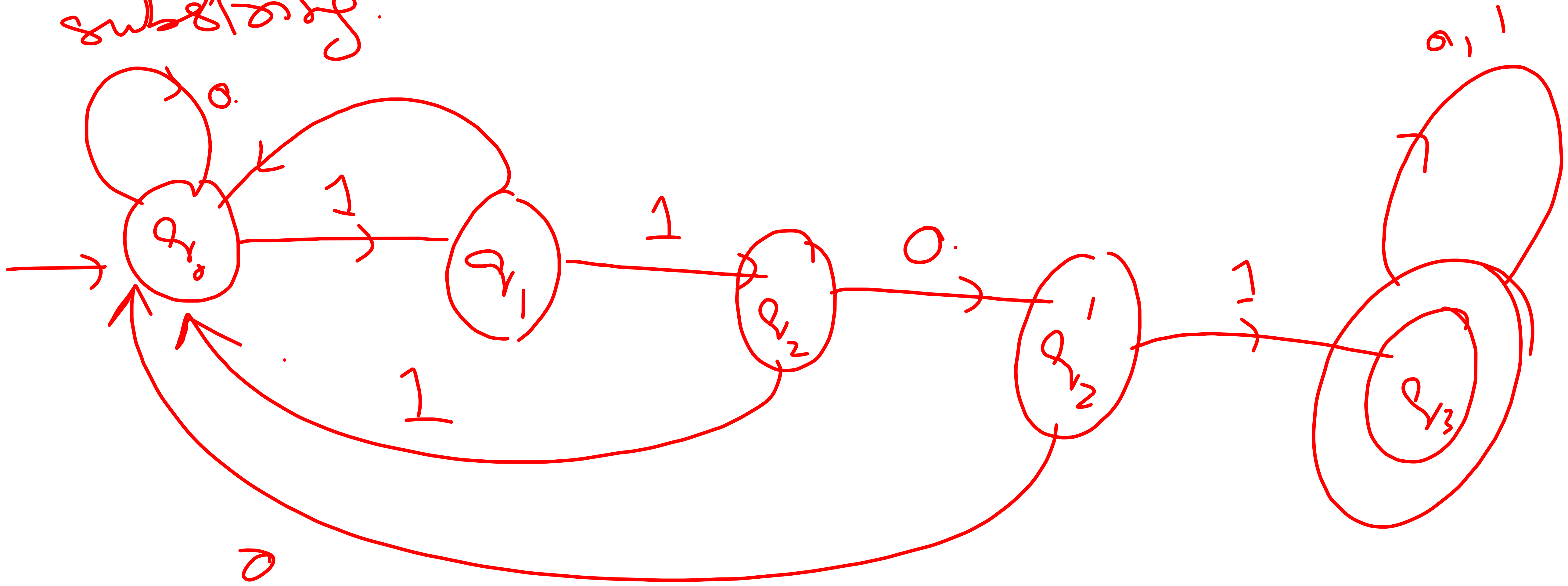


① $\delta_0(q_0, w) = q_0$

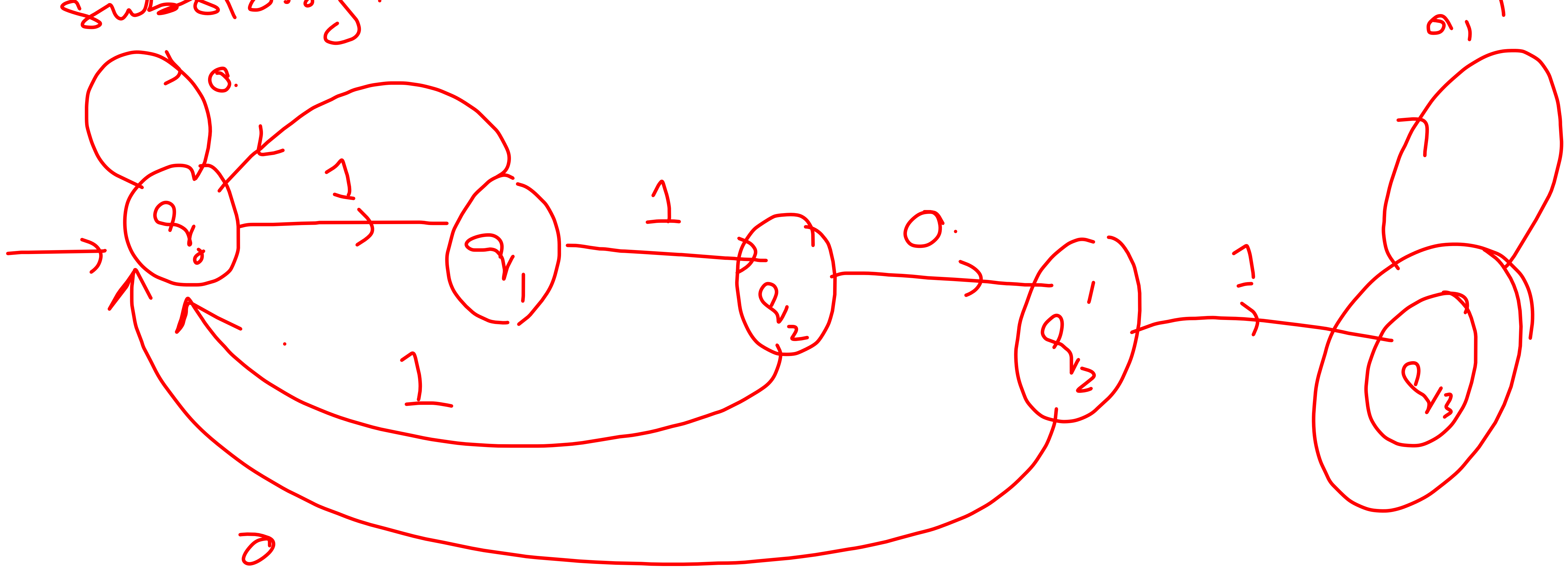
② $\delta_0(q_0, w) = q_1$

iff ...

Ex 3 Construct a DFA that contains 1101 as a substring.



Ex 3 Construct a DFA that contains 1101 as a substring.



Non-deterministic finite automaton (NFA).

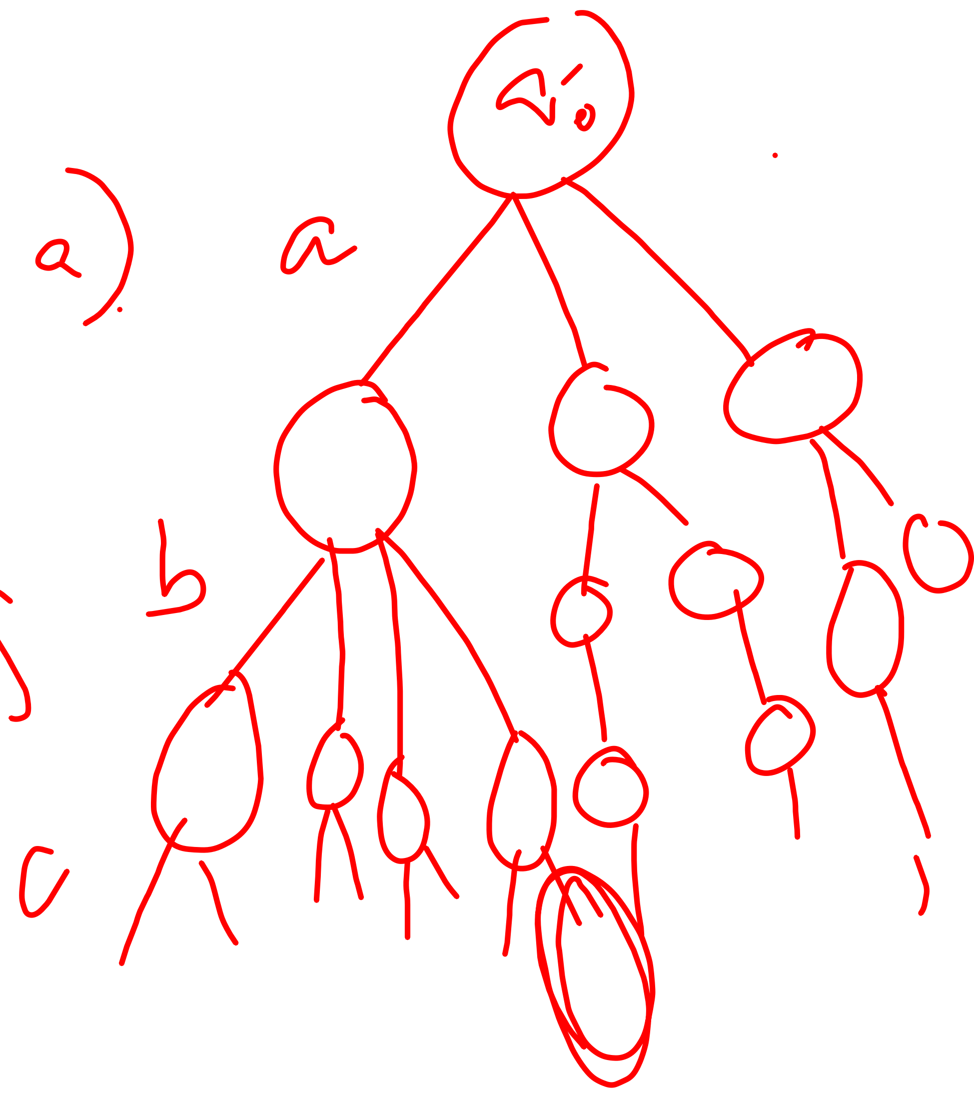
In NFA, the automaton at each step has the option of moving into one of several states & hence $\delta : Q \times \Sigma \longrightarrow \mathcal{P}(Q)$

Hence $\delta(q, a) = \{p_1, \ldots, p_k\}$ means that in state q reading the letter a, M has the option of moving into one of the states $p_1, \ldots, p_k$

We extend δ to Σ^* by induction as follows:

(a) $\delta^*(q, \lambda) = \{q\}$.

(b) $\delta^*(q, wa) = \bigcup\limits_{q' \in \delta^*(q,w)} \delta(q', a)$



Def'n An NFA M accepts w if

$$\delta^*(q_0, w) \cap F \neq \phi$$

The lang accepted by M is

$$L(M) = \{ w \in \Sigma^* : \delta^*(q_0, w) \cap F \neq \phi \}$$

Thm (Rabin) There is an algorithm that transforms a given NFA M into an equivalent DFA $\hat{M}$.

Pf Let $M = (\Sigma, Q, q_0, \delta, F)$. We shall construct

$$\hat{M} = (\Sigma, \hat{Q}, \hat{q}_0, \hat{\delta}, \hat{F}),$$

where

$$\hat{Q} = \mathcal{P}(Q).$$

$$\hat{q}_0 = \{q_0\}.$$

$$\hat{F} = \{Q' \subseteq Q : Q' \cap F \neq \phi\}.$$

$\hat{\delta} : \mathcal{P}(Q) \times \Sigma \longrightarrow \mathcal{P}(Q)$. is defined as follows:

$$\hat{\delta}(P, a) = \bigcup_{p \in P} \delta(p, a)$$

Claim $\quad \hat{\delta}^*(P, w) = \bigcup_{p \in P} \delta^*(p, w)$

$|w| = 0$.

$$\hat{\delta}^*(P, \lambda) = P = \bigcup_{p \in P} \{p\} = \bigcup_{p \in P} \delta^*(p, \lambda).$$

$$\hat{\delta}^*(P, wa) = \hat{\delta}(\hat{\delta}^*(P, w), a)$$

$$= \hat{\delta}\Big(\bigcup_{p \in P} \delta^*(p, w), a\Big)$$

$$= \bigcup_{p \in P} \hat{\delta}(\delta^*(p, w), a)$$

$$= \bigcup_{p \in P} \bigcup_{q \in \delta^*(p, w)} \delta(q, a) = \bigcup_{p \in P} \delta^*(p, wa)$$

$$w \in \mathcal{L}(\hat{M})$$

$$\Longleftrightarrow \hat{\delta}^*(\hat{q}_0, w) \in \hat{F}$$

$$\Longleftrightarrow \delta^*(q_0, w) \in \hat{F}$$

$$\Longleftrightarrow \delta^*(q_0, w) \cap F \neq \phi$$

$$\Longleftrightarrow w \in \mathcal{L}(M)$$