Computation by a DFA: $M = (Q, \Sigma, \delta, q_0, F)$

$w \in \Sigma^* \longrightarrow$ input : $w = \sigma_0 \sigma_1 \sigma_2 \dots \sigma_n$

- $q_0 \to q_1 = \delta(q_0, \sigma_0) \to q_2 = \delta(q_1, \sigma_1) \to q_3 = \delta(q_2, \sigma_2)$
$\dots \to q_{n+1} = \delta(q_n, \sigma_n)$

$f(w) = 1$ iff $q_{n+1} \in F$

$\equiv$ $w \in L(M)$ iff $q_{n+1} \in F$

- $q_{n+1} = \delta(\delta(\dots \delta(\delta(q_0, \sigma_0), \sigma_1) \dots , \sigma_n)$

extended transition function: $\hat{\delta} : Q \times \Sigma^* \to Q$

Base case: $\hat{\delta}(q, \sigma) = \delta(q, \sigma)$ when $|\sigma| = 1$

Recursive step: $\hat{\delta}(q, w\sigma) = \delta(\hat{\delta}(q, w), \sigma)$

Language of a DFA M: For a DFA $M = (Q, \Sigma, \delta, q_0, F)$

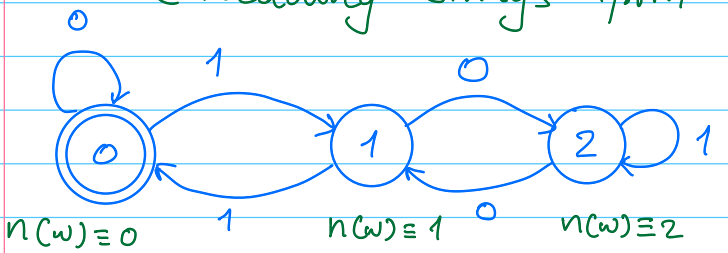
$$L(M) = \{ w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F \}$$

A language $L \subseteq \Sigma^*$ is said to be regular if $\exists$ DFA M s.t $L(M) = L$.

Set of binary strings that are the binary representations of numbers divisible by 3

$000 \in L \qquad 001 \notin L \qquad 00011 \in L \quad \ldots$

(Reading strings from MSB)



$n(w) = $ # corr. to w

$n(w0) = 2.n(w)$

$n(w1) = 2.n(w)+1$

$\delta(q,\sigma) = 2q+\sigma \pmod 3$ → stronger induction hypothesis

$$\hat{\delta}(0,w) \in L \iff \delta(\hat{\delta}(0,w'),\sigma) \in L$$

$$\hat{\delta}(0,w) = \delta(\hat{\delta}(0,w'),\sigma) \quad \text{where } w = w'\sigma$$

$$= \delta(n(w') \bmod 3, \sigma)$$

$$= 2(n(w') \bmod 3) + \sigma \pmod 3$$

$$= 2\,n(w') + \sigma \pmod 3$$

$$= n(w) \pmod 3$$