DEFN

("M finishes state \( q \)")

\( M \) ends in state \( q \) after reading \( w \in \Sigma^* \) if

1. \( w = w_1 w_2 \ldots w_n \)
   where \( w_i \in \Sigma \)

2. \( \exists \) state \( y_0, y_1, y_2 \ldots y_n \in \Phi \)

\[ r_0 = q_0 \]

\[ y_i \leq i \leq n \]

\[ s(y_{i-1}, w_i) = y_i \]

\[ y_n = q \]

Fact: \( q \) is unique

because \( s \) is a function, basically

\( M \) accepts \( w \in \Sigma^* \) if \( q \) reached by \( M \) after reading

\( w \) is \( \in \mathcal{L}(M) \)

\[ h(M) = \sum_{w \in \mathcal{L}(M)} m \]
Defn:

\( M \text{ accepts } w \iff \text{ the state, } q, \text{ reached by } M \text{ after reading } w \text{ is an accepting state, } \text{i.e. } q \in F \)

Defn:

The language recognized by

\[ L(M) = \{ w \mid M \text{ accepts } w \} \]

Note

Every \( M \) recognizes exactly one language. Implicitly, it "recognizes" only those strings it must accept and those it must reject.
Let $L \subseteq \Sigma^*$.

$L$ is regular if $L = L(M)$ for some Finite Automaton $M$.

$\{0,1\}^*$

$L(M) = \Sigma^*$

Note: $M$ accepts every palindrome, e.g., $a a a b a a a$, but also some non-palindromes, like $a b$.

$L$ recognizes $\Sigma^+$; $L \subseteq \{0,1\}^*$ but $\neq L(M)$.

Regular languages. Example:
- "Even parity" is regular.
- "3rd from right is a" odd length.

Are there general ways to prove languages regular, other than making more & more example $M$'s? Yes 3-2.
Theorem

If \( M = (Q, \Sigma, \delta, q_0, F) \) accepts \( L \) (i.e., \( L = L(M) \))

Then \( M' = (Q, \Sigma, \delta, q_0, Q - F) \) accepts \( \Sigma^* - L \)

Proof

\( M \) accepts \( w \) if it is in a state \( z \in F \) after reading \( w \)

\( \Rightarrow M' \) is in state \( Q \) after reading \( w \).

But \( z \in F \) so \( 8 \in Q - F \)

\( \therefore M' \) rejects \( w \)
Regular languages are closed under complementation.