Defn \( M = (Q, \Sigma, \Gamma, S, q_0, q_{acc}, q_{rej}) \)

- \( Q \): finite state set
- \( \Sigma \): finite input alphabet set; \( \Sigma \subseteq \Sigma \)
- \( \Gamma \): finite tape alphabet; \( \Sigma \subseteq \Gamma \)
- \( S : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\} \): transition function
- \( q_0 \in Q \): start state
- \( q_{acc} \in Q \): accept state
- \( q_{rej} \in Q \): reject state
By definition, no transitions out of $q_{\text{acc}}$, $q_{\text{rej}}$;

$M$ halts if (and only if) it reaches either

$M$ loops if it never halts (“loop” might suggest “simple”, but non-halting computations may of course be arbitrarily complex)

$M$ accepts if it reaches $q_{\text{acc}},$

$M$ rejects by halting in $q_{\text{rej}}$ or by looping

The language recognized by $M$:

$L(M) = \{ w \in \Sigma^* \mid M \text{ accepts } w \}$
L is *Turing recognizable* if $\exists$TM M s.t. $L=L(M)$

L is *Turing decidable* if, furthermore, that M halts on all inputs

*A key distinction!*
Example

\[ L = \{ w \# w \mid w \leq 30,133 \} \]

1. check that there's a single #
2. read, remember & cross off
   left most letter
3. scan to # & compare next letter
4. if ok, cross it off
5. repeat
\[ \delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k \]
Nondeterministic Turing machines:

\[ \delta : Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L,R\}) \]
Non-deterministic Turing machines:
\[ \delta: Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L,R\}) \]

Accept if any path leads to \( q_{\text{accept}} \)
Reject if all (halting) paths lead to \( q_{\text{reject}} \)