Turing Machines

- more powerful than PDA's
- what could it have?
Example: \[ A = \{ a^i b^i c^i \mid i \geq 0 \} \]
Example: \[ B = \{ w\#w \mid w \in \{0,1\}^* \} \]
Def 3.3:

A **Turing machine** (TM) is a 7-tuple \((Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})\), where

- \(Q\) is the (finite) set of states
- \(\Sigma\) is the (finite) input alphabet, not containing \(\square\)
- \(\Gamma\) is the (finite) tape alphabet, \(\square \cup \Sigma \subseteq \Gamma\)
- \(\delta\): \(Q - \{q_{\text{reject}}\} \times \Gamma \to Q \times \Gamma \times \{L,R\}\) is the transition function
- \(q_0 \in Q\)
- \(q_{\text{accept}} \in Q\) is the accept state
- \(q_{\text{reject}} \in Q - \{q_{\text{accept}}\}\) is the reject state
**Computation** of Turing machines

- first we define a **configuration**:
  
  $uqv$ - means the tape contains $uv$, the state is $q$, and the machine reads the first symbol of $v$

- suppose configuration is $uaqbv$ and $\delta(q,b) = (p,c,R)$

We say that $uaqbv$ **yields** $uacpv$

Note: other possibilities for yielding a new config., e.g. moving left, being at the end of the tape (reading $\lambda$), etc.
Turing Machines

Computation of Turing machines

- first we define a configuration:
  \( uqv \) - means the tape contains \( uv \), the state is \( q \), and the machine reads the first symbol of \( v \)

- start configuration:
  \( q_0 \) - initial state

- accepting configuration:
  \( uqv_{\text{accept}} \) - \( u, v \in \Sigma^* \)

- rejecting configuration:
  \( uqv_{\text{reject}} \) - \( u, v \in \Sigma^* \)

Note: accepting/rejecting configurations are halting
Computation of Turing machines

- first we define a configuration:

  \[ uqv \] means the tape contains \( uv \), the state is \( q \), and the machine reads the first symbol of \( v \)

A TM \( M \) accepts \( w \) if

there exists a sequence of configurations \( c_0, c_1, \ldots, c_k \)

where
1) \( c_0 \) is the starting conf.
2) \( c_i \) yields \( c_{i+1} \), \( i \in \{0, \ldots, k-1\} \)
3) \( c_k \) is an accepting conf.

The language of a TM \( M \) is the set of strings that \( M \) accepts/recognizes.
Def 3.5: A language is **Turing-recognizable** (or recursively enumerable) if there is some TM that recognizes it.

Def 3.6: A language is **Turing-decidable** (or recursive) if there is some TM that decides it.
Example: $A = \{ 0^n \mid n = 2^k \text{ for some } k \geq 0 \}$

We keep dividing by 2, by crossing out every other 0.

When to accept? (Add a couple of states to the TM)