Turing Machines

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

\underline{Idea:}
1) replace the first symbol of \( w \) with a (\#) and remember this symbol in the state
2) jump over the rest of the first \( w \) until (\#)
3) compare the first symbol of the second \( w \) with the state info and replace with (\#)
4) go back to the beg. and repeat, jumping over (\#)
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, \(\square \cup \Sigma \subseteq \Gamma\)
- \(\delta: (Q \times \Gamma) \rightarrow Q \times \Gamma \times \{L, R\}\) 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 uaqpv
**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:

  \[ \sim q_0 x \]

  accepting string \( x \)

- accepting configuration:

  \[ uq_{accept}v \] for some \( uv \in \{0,1\}^* \)

- rejecting configuration:

  \[ uq_{reject}v \] for some \( uv \in \{0,1\}^* \)

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

- first we define a configuration:

  \( uvq \) - means the tape contains \( uv \), the state is \( q \), and the machine reads the first symbol of \( v \)

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

\[
\text{there is a sequence of config. } \quad \text{s.t.} \quad \begin{align*}
\text{the first is the starting config.} \\
\text{the } i\text{-th config. yields the } (i+1)\text{-st config.} \\
\text{the last config. is accepting}
\end{align*}
\]

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

- every string in L is accepted
- every string not in L is rejected or goes into an infinite loop

Def 3.6: A language is **Turing-decidable** if there is some TM that decides it. (also known as recursive)

- a string in L is accepted
- a string not in L is rejected

(no infinite loops)
Example: \[ A = \{ 0^n \mid n=2^k \text{ for some } k \geq 0 \} \]