Turing Machines

- more powerful than PDA’s
- what could it have?

we have a tape - infinite length to the right

the input string

we can - rewrite an entry of the tape (one entry per step of the TM)

from a state on a given symbol:

- we go to a new state
- write a new symbol on the tape
- need to know the direction (left/right/stay) in which we are moving after writing the new symbol
Example: \[ A = \{ a^i b^i c^i \mid i \geq 0 \} \]
Example: \( B = \{ w \# w \mid w \in \{0,1\}^* \} \)

What about: \( \{ ww \mid w \in \{0,1\}^* \} \) ?
Turing Machines

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 □
- \( \Gamma \) is the (finite) tape alphabet, \( \square \cup \Sigma \subseteq \Gamma \)
- \( \delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R,S\} \) 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

We haven't drawn all undrawn options for transitions, go to the reject state.
**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 \( v \).

- Suppose configuration is \[ uaqbv \] and \( \delta(q,b) = (p,c,R) \).

We say that \( uaqbv \) **yields** \( uacpv \): what is the next configuration?

- Next state is \( p \).
- \( b \) gets rewritten to \( c \).
- Move head \( R \).
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: \( q_0, w \) where \( w \) is the input

- accepting configuration: \( x_1, q_{\text{accept}}, x_2 \) where \( x_1, x_2 \in \Gamma^{*} \)

- rejecting configuration: \( x_1, q_{\text{reject}}, x_2 \) where \( x_1, x_2 \in \Gamma^{*} \)

Note: accepting/rejecting configurations are halting

There might be inputs that never reach a halting config. \( \rightarrow \) because they get to an infinite loop.
**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

1. it starts in the initial config.
2. Then goes through a sequence of config., the new config. is yielded by the last conf. and it gets to 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** if there is some TM that recognizes it.

Def 3.6: A language is **Turing-decidable** if there is some TM that decides it.

For every input, the TM gets to either *accept* or *reject* (no infinite loops).
Example: \( A = \{ 0^n \mid n=2^k \text{ for some } k \geq 0 \} \)