Variants of Turing Machines

- what if the tape is infinite in both directions?

  - if given a standard TM, can we simulate it on a 2WITTM?
    
    yes, just do not use the left part of the tape (exactly the same transition diagram)

  - if given a 2WITTM, can we simulate it with a standard TM?

    idea 1:
    
    - treat odd positions as going right (the right half of the tape)
    - even as the left half of the tape

    idea 2:
    
    - simulate the 2WITTM until it tries to go left of the leftmost position on the tape (mark the position),
    - then go to dedicated states that shift the contents of the tape one position to the right, making a blank at the left, then continue the computation

    idea 3: "fold the tape in half, create double-decker symbols"

    1. shift the input right, create a new start delimiter, and change the input symbols as follows

    2. \( \delta_{\text{new}}(q^T, a) = (p^T, b, R) \)
    
    \( \delta_{\text{new}}(q_i, b) = (p_i, b, L) \)

    where \( T \) means reading the top part
Variants of Turing Machines

- what if a TM has several heads?
  - can we simulate a standard TM using a 2-headed dragon TM?
    YES, in every transition, move the 2nd head w. the first (same direction)
    and the 2nd head does not rewrite the symbol,
    the 1st head simulates original TM
  - can we simulate a 2-headed dragon TM using a standard TM?

This section: we’ll give detailed descriptions of our machines but not give detailed δ-functions.
Variants of Turing Machines

- what about several tapes (and heads)?

Sketch:
create double-decker symbols to capture the contents of both tapes using a single tape, then simulate with 2 heads, 1 tape (similarly to before)
- have to redefine $\delta$-function:

**Thm 3.13**: Every multitape TM has an equiv. single-tape TM.
Nondeterministic Turing Machines

- have to redefine $\delta$-function:

\[ Q \times \{\text{accept}, \text{reject}\} \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R, S\}) \]

Thm 3.16: Every nondeterministic TM has an equivalent deterministic TM.

Note: DFS does not work! (bec. of possible branches of infinite length)

idea: BFS through the config. tree,

if an accept. config. is found, accept

how to do this using a TM (deterministic):

1) go to $abba$ #

2) go to $abba$ #, $xpbb$ #, or $bbab$ #

for the current config., create its children config.
at the end of the tape, then mark current as done
and move to the next config. (becomes current)