Variants of Turing Machines

- what if a TM has more tapes? several heads?

This section: we'll give detailed descriptions of our machines but not give detailed $\delta$-functions.

Simulate two-tape TM with one tape:

- idea: have the contents of the second tape after the contents of the first (and now only tape), separated by $\#$
- keep track of the positions of the heads
e.g. $\Gamma_{new} = \Gamma_0 \cup \{ \# \} \cup \{ \#, \}$
e.g. $\Gamma_{new} = \{ a, b, c, \#, \}$

1) walk through the tape (remember the current 2-tape state in your state) remember in the state
2) read the two dotted symbols $\Rightarrow$ know which 2-tape transition to take
3) go back, change the dotted symbols along the way
4) change state

Note: if $\#$ then shift right everything from $\#$ onward, create $\#$
Multitape Turing Machines

- have to redefine $\delta$-function: for 2 tapes

$$\delta: (Q - \{q_{accept}, q_{reject}\}) \times \Gamma \times \Gamma \rightarrow Q \times \Gamma \times \Gamma \times \{L,R\} \times \{L,R\}$$

Thm 3.13: Every multitape TM has an equiv. single-tape TM.

See the previous slide

What about a PDA with 2 stack?

one stack: content of the TM's tape before its head
second stack: after the head
slinky-type push/pop to simulate the tape

also: the Post machine - has a queue instead of a stack
\rightarrow also equivalent to a TM
- have to redefine $\delta$-function:

$$\delta: (Q - \{q_{\text{accept}}\}) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L,R\})$$

$$\delta(q,a) = \{(p,b,R),(r,c,L)\}$$

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

- Simulating det. TM by nondet. TM → immediate (don't use nondeterminism)

  Idea: do a BFS through all possible_configs, until find an accepting config.

  Tape will look like:

  - $w_{aca}$: tape content at the beg. for input $aca$
  - $q_{w_{aca}}$: change it into by shifting right & add state
  - $q_{w_{aca}}$: keep processing the next unprocessed config.
    - generate all subseq. configs.
    - at the end of the tape
    - cross out the one that are already on the tape
    - mark current config as done
An alternative name for Turing-recognizable languages is recursively enumerable languages.

An enumerator is a TM-like “printer” with no input and an extra output tape that prints all strings in a given language.
Thm 3.21: A language is Turing-recognizable iff there is an enumerator for it.