- 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.
Multitape Turing Machines

- have to redefine $\delta$-function:
  \[ \delta_{\text{multi}} : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L,R\}^k \]

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

Init:

To simulate $\delta(q, x_1, x_2, \ldots, x_k) = (p, y_1, y_2, \ldots, y_k, L, R, L, \ldots)$

Idea: 1) "fold" the $k$ tapes into a single tape by concatenating the contents of the individual tapes, use a delimiter symbol
2) use dotted symbols to keep track of the head positions

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

single tape

a+b,b,l
Thm 3.13: Every multitape TM has an equiv. single-tape TM.

1) Scan the tape to the right, remembering all the dotted symbols in the stack, i.e., we will have a state.
2) Go back to the front of the tape.
3) Scan the tape to the right, changing the dotted symbols as you go: if \( \rightarrow \), put the dot above the symbol on the R.
   also update the state.
4) Go back to the front of the tape and change the state to p.
- have to redefine \( \delta \)-function:
\[
\delta : Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L,R\})
\]

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

1) Start with the initial config:
   \[
   \text{go aabab}
   \]
   input string

2) BFS (breadth-first search) through the config.
   we get:
   \[
   \text{go aabab \# bp.abab \# cpz.abab \# p3.babab \# ...}
   \]
   (level by level)

Note:
BFS does not work due to infinite branches

\( \Rightarrow \) Thus, if \( \exists \) accepting config, we will find it since #config per level is finite

... accept...
accepted
if \( \exists \) path to accept
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.