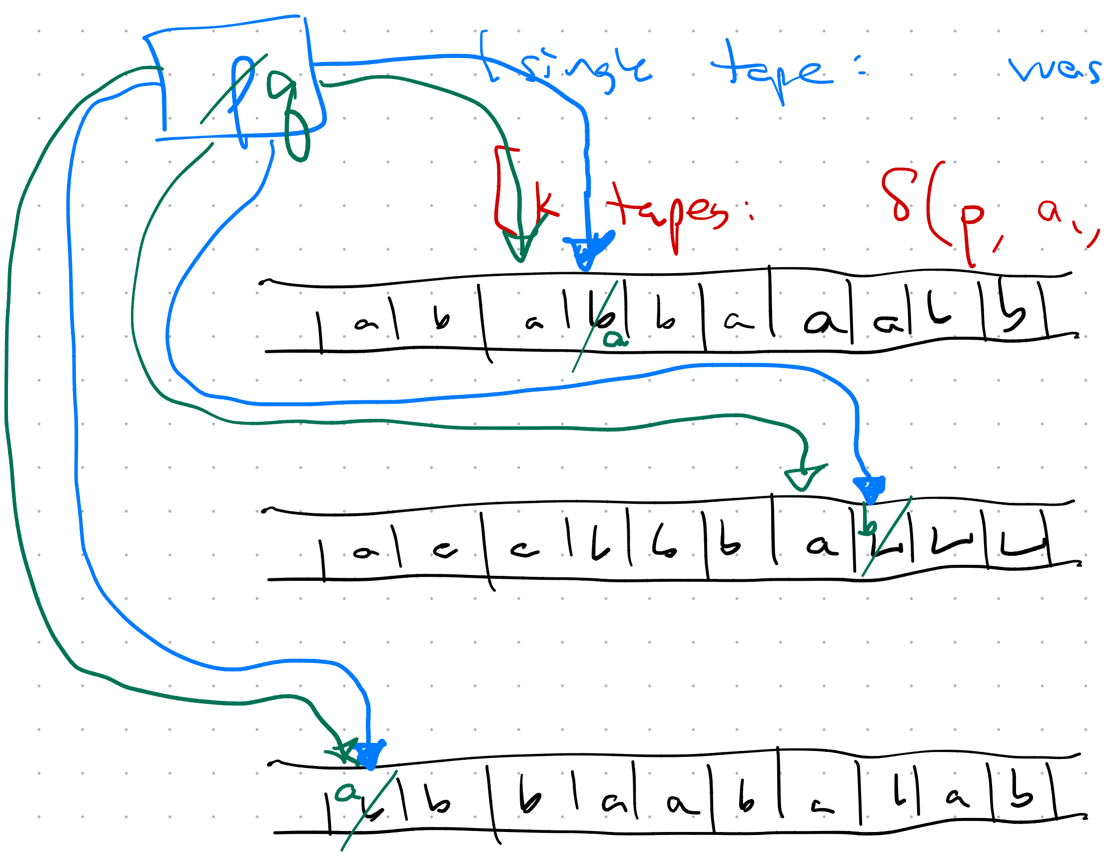
15 Apr 2024   Universal Turing Machine

A multi-tape TM has:

- finite state set
- finite (possibly > 1) number of infinite tapes
  (Input is always on first tape.)
- Read/write head for each tape, capable of moving independently.
- Transition function



[single tape: was $\delta(p, a) = (q, b, L/R)$]

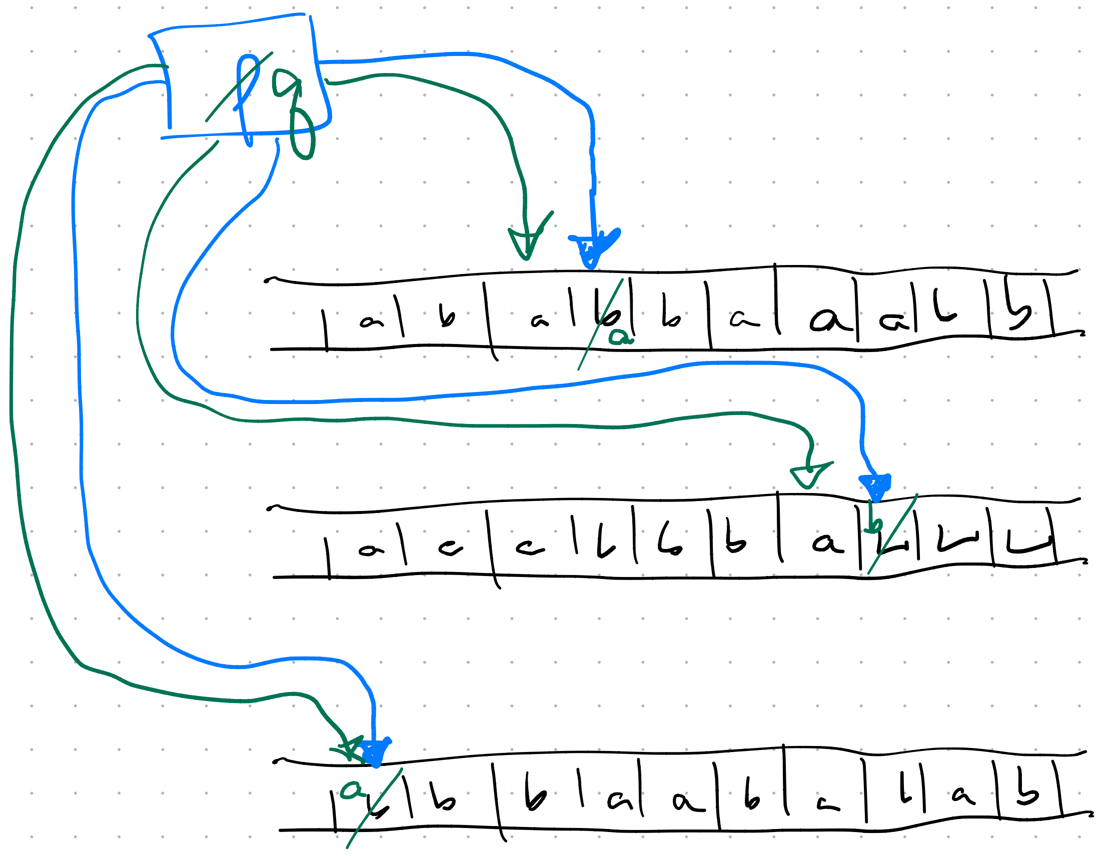
[k tapes: $\delta(p, a_1, \dots, a_k) = (q, \underbrace{b_1, \dots, b_k}_{\text{new symbol to write on each tape}}, \underbrace{d_1, \dots, d_k}_{\text{directions to move each R/W head}})$]

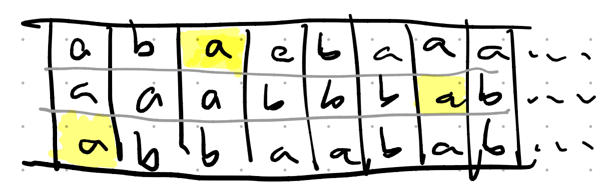
Can be L, R, or stay still

$\delta(p, b, \sqcup, b) =$

$(q, a, b, a, \leftarrow, \leftarrow, -)$

A single tape TM can simulate a multi-tape one.





Simulating M with k tapes and tape alphabet Γ, we will use single-tape machine S with tape alphabet $\Gamma^k \times \{0,1\}^k$

<u>Configuration</u>: The state of a TM, the position of (each) read-write head, and the contents of (each) tape, omitting the infinite sequence of blanks at the end of the tape.

A finite amount of data, that completely describes the state of a computation.

S makes a left-to-right pass, memorizing (in its internal state) each symbol M is reading, then it goes right-to-left, implementing one step of M's transition rule by overwriting symbols and repositioning simulated read-write heads.

Multi-tape TM is the accepted way to quantify space / time complexity of algorithms.

<u>Universal Turing Machine:</u> A machine $\mathcal{U}$, that gets an input $M\#x$ where M describes another TM and x describes the input to M, and $\mathcal{U}$ simulates what happens when M processes input x.

Description of a Turing machine M:

(1) a sequence of 0's and 1's starting with $0^n 1 0^m 1 0^k 1 0^s 1 0^t 1 0^r 1 0^u 1 0^v 1$

n = # states, s = start state, u = endmarker

m = # tape symbols, t = accept, v = blank

k = # inp symbols, r = reject

(2) the description of the transition function. A sequence of $\{0,1\}$-strings, in a standardized format

would be 1 for R

$\delta(p,a) = (q,b,L)$ encoded as $0^p 1 0^a 1 0^q 1 0^b 1 0$

Description of input x: sequence of $\{0,1\}$-strings where symbol $x_i \in \Gamma = [m]$ is encoded as $0^{x_i} 1$.

The string $x = (x_1, x_2, x_3, \ldots, x_a)$
is encoded as $0^{x_1} 1 \, 0^{x_2} 1 \cdots 0^{x_a} 1$

Input to U is $M \# x$ in input alphabet $\{0, 1, \#\}$.

To say U is a univ TM means:

- if M accepts x, U must accept $M \# x$
- if M rejects x, U must reject $M \# x$
- if M loops on x, U must loop on $M \# x$.

How does U work?

3 tapes: input tape (read only: stores $M \# x$)
working tape (stores configuration of M in the simulation)
state tape (stores description of M's state in the simulation)