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:**

\[ S(p, a) = (q, b, L/R) \]

**k tapes:**

\[ S(p, a_1, \ldots, a_k) = (q, b_1, \ldots, b_k, d_1, \ldots, d_k) \]

- new symbol to write
- directions to move
- on each tape
- R/W head
- Can be L, R, or stay still

\[ S(p, b, u, b) = (q, a, b, a, <, <, \downarrow) \]
A single tape TM can simulate a multi-tape one.

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

**Configuration:** 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.

**Universal Turing Machine**: A machine $U$, that gets an input $M \# x$ where $M$ describes another TM and $x$ describes the input to $M$, and $U$ 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 $0^1 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0$

   - $n = \# \text{states}$
   - $s = \text{start state}$
   - $m = \# \text{tape symbols}$
   - $t = \text{accept}$
   - $v = \text{blank}$
   - $r = \text{reject}$

2. The description of the transition function:
   A sequence of $\{0,1\}$-strings, in a standardized format:
   $\delta(p,c) = (q,b,L)$ encoded as $0^1 1^0 10^1 0^1 0^1 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0 1^0$

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

\[\text{End of Description}\]
The string $X = (x_1, x_2, x_3, ..., x_a)$ is encoded as $0^{x_1}10^{x_a}1 ... 0^{x_1}1$.

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

To say $U$ is a universal 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).