Defn: A \( K \)-tape TM (\( K \geq 1 \)) has the same components \( M = (Q, \Sigma, \Gamma, \delta, \omega, s, F) \) but \( \delta \subseteq (Q \times \Gamma^K) \times (\Gamma^K \times \Sigma^L \times \Sigma^R \times Q) \).

Instruction: \((p, c_1, c_2, \ldots, c_K, d_1, d_2, \ldots, d_K, q)\) can draw arcs similarly.

Picture:

The initial ID \( I_0(x) \) has all heads in column 1 with the first tape head scanning the first bit \( x_1 \) of \( x \) (or scanning a blank if \( x = \epsilon \)) and all other tapes are blank.

Defn of \( I \vdash^m J, I \vdash^m K \), accepting ID, and \( L(M) \): same.

Alternate convention: \( I \) includes a dedicated \( \ast \) char. Each tape has \( \ast \) in cell 0. Code \( \ast \) always moves \( R \) off a blank. \( \ast \) never moves \( L \) off \( \ast \).
With multi-tape TMs, often the input tape is read-only.

Further, it is often one-way: in $\delta$, always $d_i = C_i$.

(in $\delta$, always $D_i \neq L$, which essentially subsumes $RO$.)

A tape $j$ behaves like a stack if $D_j = L \Rightarrow d_j = \_\_\_\_$ in all cases.

**Def.** A pushdown automaton (PDA: NPDA model) is equivalent to a 2-tape TM whose input tape is one-way and whose sole worktape behaves as a stack.

The one operational quirk is that shifting between push and pop steps needs an extra machine step.

**Theorem:** Given any CFG $G$, we can build an NPDA $M$ s.t. $L(M) = L(G)$, and vice versa.

But, PAL and EVENPAL (without # markers) are examples of CFLs that have an NPDA but not a DPDA.

**Def.** A language $A$ is a DCFL if there is a DPDA $M$ such that $L(M) = A$.

$REG \supsetneq DEFL \supsetneq CFL \supsetneq$ decidable languages.

Section 2.4 talks about "deterministic CFGs" but the ones it gives are not equivalent.