A finite automaton N \( \langle Q, \Sigma, \Gamma, \delta, \delta_0, \mathcal{F} \rangle \) is a Turing machine \( M \) in which every instruction \( (p, c, d, D, q) \) has \( D = R \) (and \( d = c \)). If we use the "liberal" final-state notation \( M = (Q, \Sigma, \Gamma, \delta, \delta_0, \mathcal{F}, \mathcal{H}) \) then we simply don't provide any qrs that read \( B \). \( \mathcal{F} \cap B = \emptyset \).

If we want \( \mathcal{F} \subseteq \{ \text{acc} \} \) and only one other halting state \( \text{ rej (text) } \) then we can do the following "good housekeeping" (like \( \text{ P3.2#3) } \):

\[ L(M) = \{ x: x \text{ ends in } a \text{ or in } b \text{ and has no } bb \text{ substring} \}. \]

If we want to insist that the TM reads all of its input \( x \) even in a deadlock we could replace \( \text{ rej } \) by a loop.

If \( N \) is a DFA then so is \( M \) a DTM. Recall the text has NFAs defined by \( \delta: Q \times \Sigma \to \mathcal{P}(Q) \) but I prefer saying \( \delta \subseteq Q \times \Sigma \times Q \) for NTMs, rather than \( \delta: (Q \times \Gamma) \to \mathcal{P}(\Gamma \times Q, \text{ L, R, T, X, A}) \). Simpler to use:

\[ \delta \subseteq (Q \times \Gamma) \times (\Gamma \times Q, \text{ L, R, T, X, A}) \]

View \( \delta \) as a set of instructions (aka "tuples" or "5-tuples"). A TM is deterministic if there is no \( q \in Q, c \in \Sigma \) st. \( \delta \) has two or more tuples beginning with \( (q, c, \ldots) \).
Definition (53.2): A multi-tape TM has some number $k \geq 1$ of tapes and

$$s = (\Delta \times \Gamma^n)^k \times (\Gamma^n \times \{L,R,S,F\} \times \Delta)^k$$

Typical structure

As an arc in a diagram:

```
    p, (c_1, c_2, ..., c_k) / (d_1, d_2, ..., d_k), (D_1, D_2, ..., D_k), y
```

"Stack up = tape chars for better reading. Could be $\$'.

```
    p
    q
```

```
Input type I
```

Options: all tapes may be "two-way infinite" or have a hard left boundary.

Other options: Initial $\Lambda$ (empty) in column 0.

1 B after $x$ replaced by $\$. $\$

"Good Housekeeping" is extended to mean:

1. $F = 3 \emptyset = \emptyset$, $\emptyset$ is the only other halting state.
2. M always erases its worktapes, and
3. M always ends with its tape head on $B$ (or $\$') to the right of $x$.

ALT 3: If M is computing a function $f(x) = y$, then type I ends with $x$ on it
and M scanning the first bit of $y$ (or all blanks if $y = \emptyset'$).

Given these understandings, we write $M = (Q, \Sigma, \Gamma, \delta, B, S, F)$ as before, but specify $F = 3 \emptyset = \emptyset$ and that

$$s \in (\Delta \times \Sigma \times \emptyset \times \Gamma^n)^k \times (\Gamma^n \times \{L,R,S,F\} \times \Delta)^k$$

Text for 1-type DTM

$$s = (Q \times \Sigma \times \emptyset \times \Gamma^n)^k \times (\Gamma^n \times \{L,R,S,F\} \times \Delta)^k$$
Defn: A pushdown automaton (PDA) is a 2-tape TM $M = (Q, \Sigma, \Gamma, \delta, B, s, F)$ such that every tuple $(P, C_1, d_1, D_1, q) \in S$ satisfies these two restrictions:

- $d_1 = C_1$ and $D_1 \neq L$ (input tape is read-only and one-way)
- If $D_2 = L$, then $d_2 = B$ (second tape behaves as a stack)

The PDA is deterministic if for all $p \in Q$ and pairs $C_1, C_2 \in \Gamma$, there is at most one instruction beginning $(P, C_1, \ldots)$. Then $M$ is a DPDA.

Example: A DPDA $M$ s.t. $L(M) = \{ a^n b^n : n \geq 0 \}$.

---

Tuesday: Examples of NPDA's and TMs.

- There is a "bug" here: inputs $a^n$ when $n \neq 1$ get accepted.
- Exercise: Show how using a bottom-of-stack marker $A$ helps avoid this bug neatly.