A finite automaton N "Is-A" Turing machine M in which every instruction (p, c, d, D, q) has $D = R$ (and $d = c$).
(or $c = B$ and $D = S$ is OK)

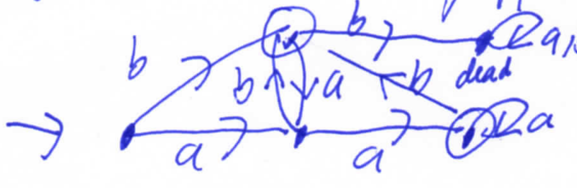
Γ = Gamma

FA: $p \xrightarrow{c} q$ $\Rightarrow$ TM: $p \xrightarrow{(c/c,R)} q$.

text ⌴ any $F \subseteq Q$

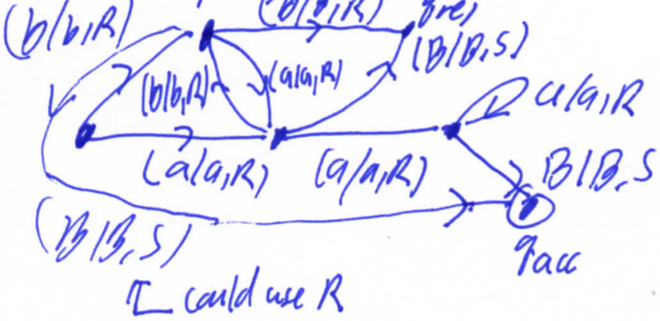
If we use the "liberal" final-state notation $M = (Q, \Sigma, \Gamma, \delta, B, s, F)$ then we simply don't provide any arcs that read B. $(\Gamma = \Sigma \cup \{B\})$

If we want $F = \{q_{acc}\}$ and only one other halting state q_{rej} (text) then we can do the following "good housekeeping" (like PS2 #2):



$L(M) = \{x : x$ ends in aa or in b, and has no bb substrings$\}$.

$\Longrightarrow$ TM



If we want to insist that the TM reads all of its input x even in a dead case, we could replace q_{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 $\delta : (Q \times \Gamma) \to \mathcal{P}(\Gamma \times \{L, R\} \times Q)$, simpler to use
$$\delta \subseteq (Q \times \Gamma) \times (\Gamma \times \{L, R, S\} \times Q).$$
View δ as a set of <u>instructions</u> (aka "tuples" or "5-tuples"). A TM is <u>deterministic</u> if there is no $q \in Q$, $c \in \Gamma$ s.t. δ has two or more tuples beginning with $(q, c, \ldots)$.

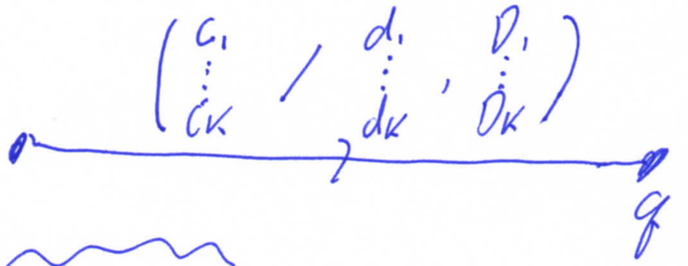
Defn (§3.2): A multitape TM has some number $k \geq 1$ of tapes and

$$\delta \subseteq (Q \times \Gamma^k) \times (\Gamma^k \times \{L,R,S\}^k \times Q)$$

Typical instruction

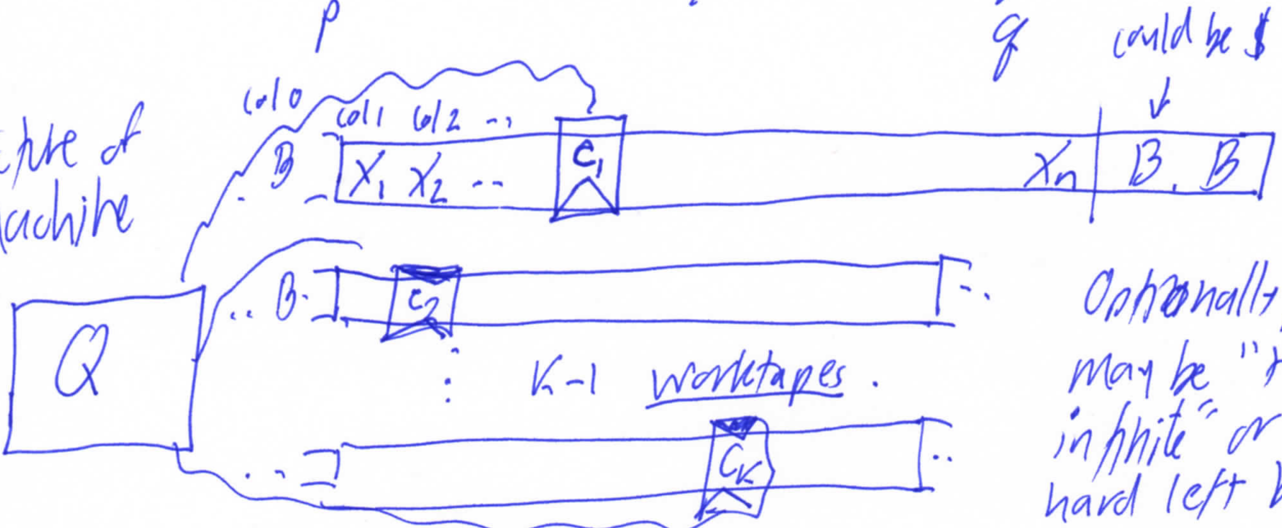
$$(p, (c_1, c_2, \dots, c_k) / (d_1, d_2, \dots, d_k), (D_1, D_2, \dots, D_k), q)$$

As an arc in a diagram:



"Stack up" tape chars for better reading.

Picture of Machine



Input tape 1

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

Other options: • Initial $\wedge$ (caret) in column 0.
• B after x replaced by $.

"Good Housekeeping" is extended to mean:

Text ① $F = \{q_{acc}\}$, q_{rej} is the only other halting state.

② M always erases its worktapes, and

③ M always ends with its Tape 1 head on B (or $) to the right of x.

④ M never writes a B symbol except when erasing rightmost char on a tape.

Alt 3: If M is computing a function $f(x) = y$, then tape 1 ends with y on it and M scanning the first bit of y (or all blank if $y = \epsilon$).

Given these understandings, we write $M = (Q, \Sigma, \Gamma, \delta, B, s, F)$ as before, but specify $F = \{q_{acc}\}$ and that

$$\delta \subseteq (Q \setminus \{q_{acc}, q_{rej}\} \times \Gamma^k) \times (\Gamma^k \times \{L,R,S\}^k \times Q).$$

Text for 1-tape DTM:

$$\delta : (Q \setminus \{q_{acc}, q_{rej}\}) \times \Gamma \to \Gamma \times \{L,R,S\} \times Q)$$

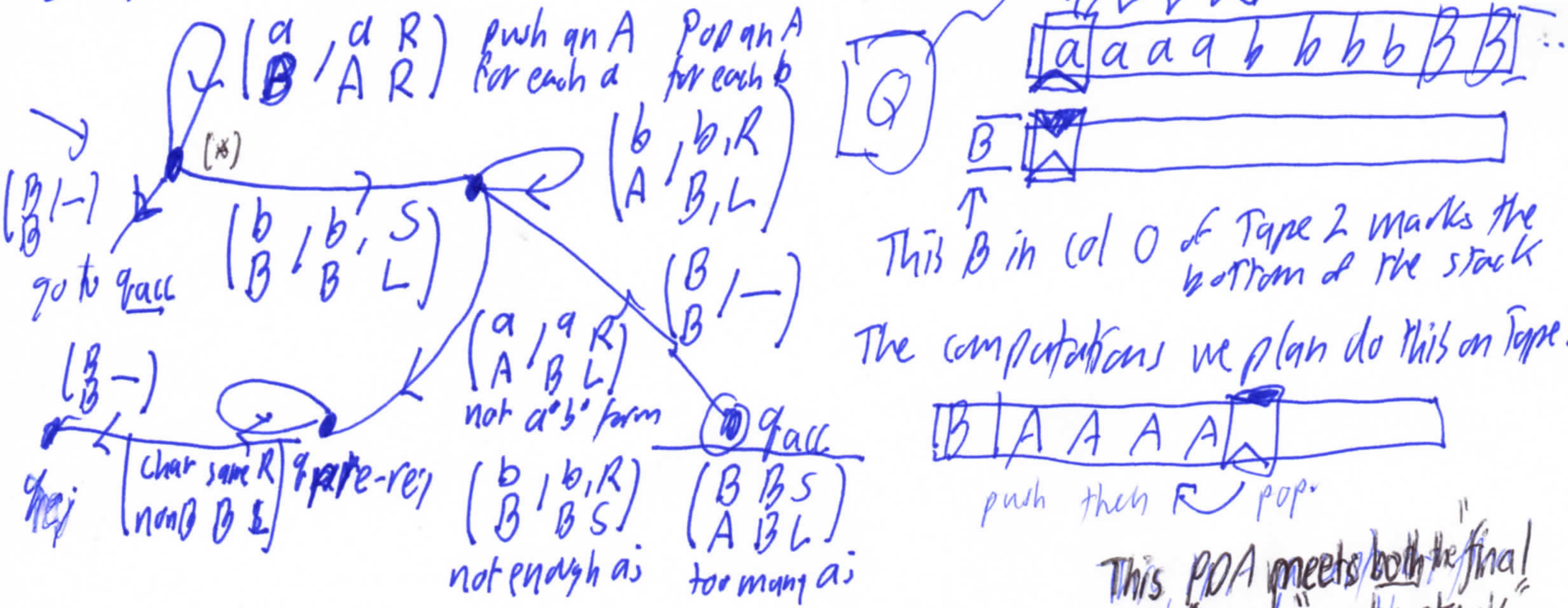
Defining Pushdown Automata as 2-TAPE machines with "good housekeeping" and two other restrictions. (by my defn)

Defn: A pushdown automaton (PDA $\xrightarrow{\text{defaults}}$ NPDA; $\xrightarrow{\text{if det}}$ DPDA) IS A 2-tape TM $M = (Q, \Sigma, \Gamma, \delta, B, s, F)$ such that every tuple $(p, {c_1 \atop c_2} / {d_1 \atop d_2}, {D_1 \atop D_2}, q) \in \delta$ 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 det^c if for all $p \in Q$ and pairs $c_1, c_2 \in \Gamma$, there is at most one instruction beginning $(p, {c_1 \atop c_2} / \cdots)$. Then M is a DPDA.

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



This PDA meets both the "final state" and "empty stack" criteria for acceptance.

Tuesday: examples of NPDAs and TMs.

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