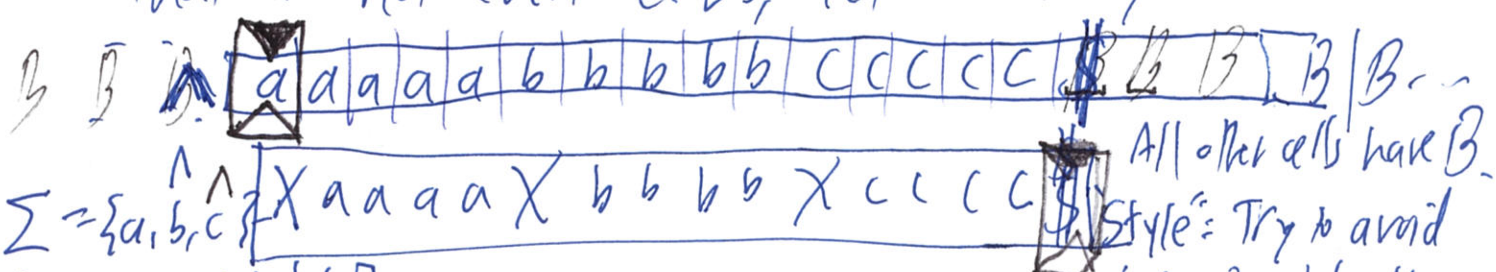
A <u>Turing Machine</u> (TM, defaults to Det^c TM, DTM but there is also an nondet^c, NTM) liberalizes a DFA (or NFA) by:

- allowing to <u>change</u> chars on one or more <u>tapes</u>
- allowing tape heads to move Left (L) or Stay

not in text → Stationary (S) besides moving right (R).

<u>Upshot</u>: TMs can decide languages like $\{a^n b^n c^n : n \geq 1\}$ that are not even CFLs, let alone regular.

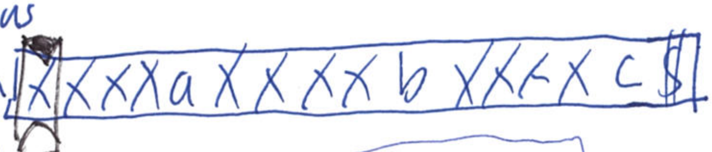


$\Sigma = \{a, b, c\}$

All other cells have B.

Style: Try to avoid interior blanks.

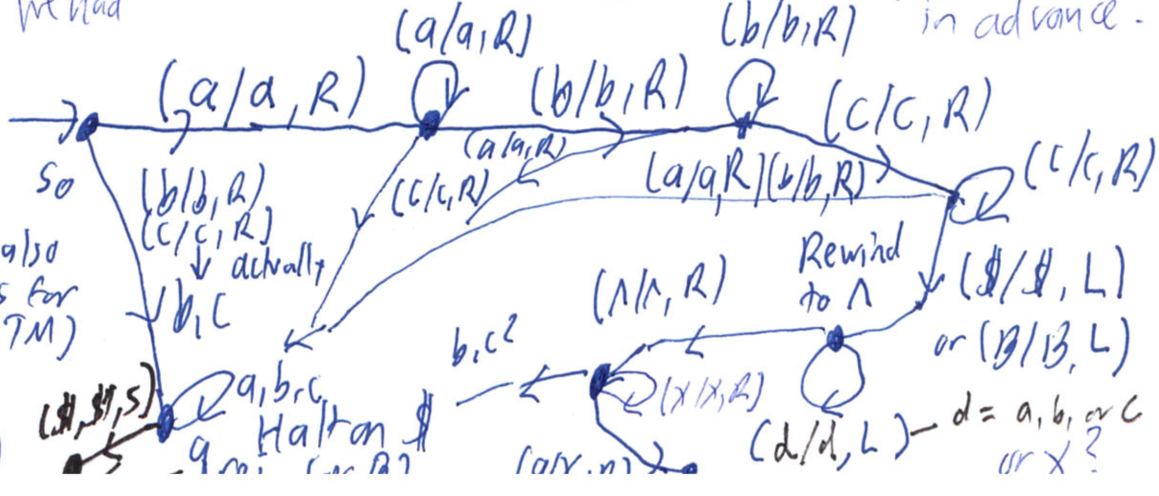
The <u>work</u> alphabet Γ always includes Σ plus the blank B, <u>can</u> include other chars ∧, $, #, X ...



Could maintain $a^+b^+c^+$ order on the fly, or check it in advance.

what if we had ∧ X X a X X c X X b $? (n = 3)

The initial code of my TM can emulate a DFA M_0 such that $L(M_0) = a^+b^+c^+$ (not yet counting)



Defⁿ: A Turing Machine is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, B, s, F)$

where:
- Q is a finite set of states
- Σ is the finite input alphabet
- B is the blank (␣ in text, or \0, or " " etc.) text. q_{acc}, q_{rej}
- Γ, which always includes $\Sigma \cup \{B\}$, is the work alphabet.
- s is the start state (q_0 in text)
- F is the set of desired final states

{ Text $F = \{q_{acc}\}$ where also without loss of generality there is a unique rejecting state q_{rej}.

- $\delta \subseteq Q \times \Gamma \times \Gamma \times \{L, R, S\} \times Q$

↑ not in text, text puts $\{L, R\}$ last.

Typical tuple or instruction: (p, c, d, D, q) — $p \in Q$, $c \in \Gamma$, $d \in \Gamma$, direction

$d = c$ allowed
$q = p$ allowed.

Diagram

Furthermore:

- ✱ M is deterministic if for all $p \in Q$ and $c \in \Gamma$ there is at most one tuple in δ that begins $(p, c / \ldots)$.
- ✱ M is "completed" if for all $p \neq q_{acc}, q_{rej}$ and $c \in \Gamma$ there is a tuple beginning $(p, c / \ldots)$.

→ Together ⟹ δ is a function from $(Q \setminus \{q_{acc}, q_{rej}\} \times \Gamma)$ to $[\Gamma \times \{L, R, S\} \times Q]$ ≈ text defⁿ of a DTM.

(The Halting states: $\{q_{acc}, q_{rej}\}$)

↑ destination can be q_{rej} or q_{acc}

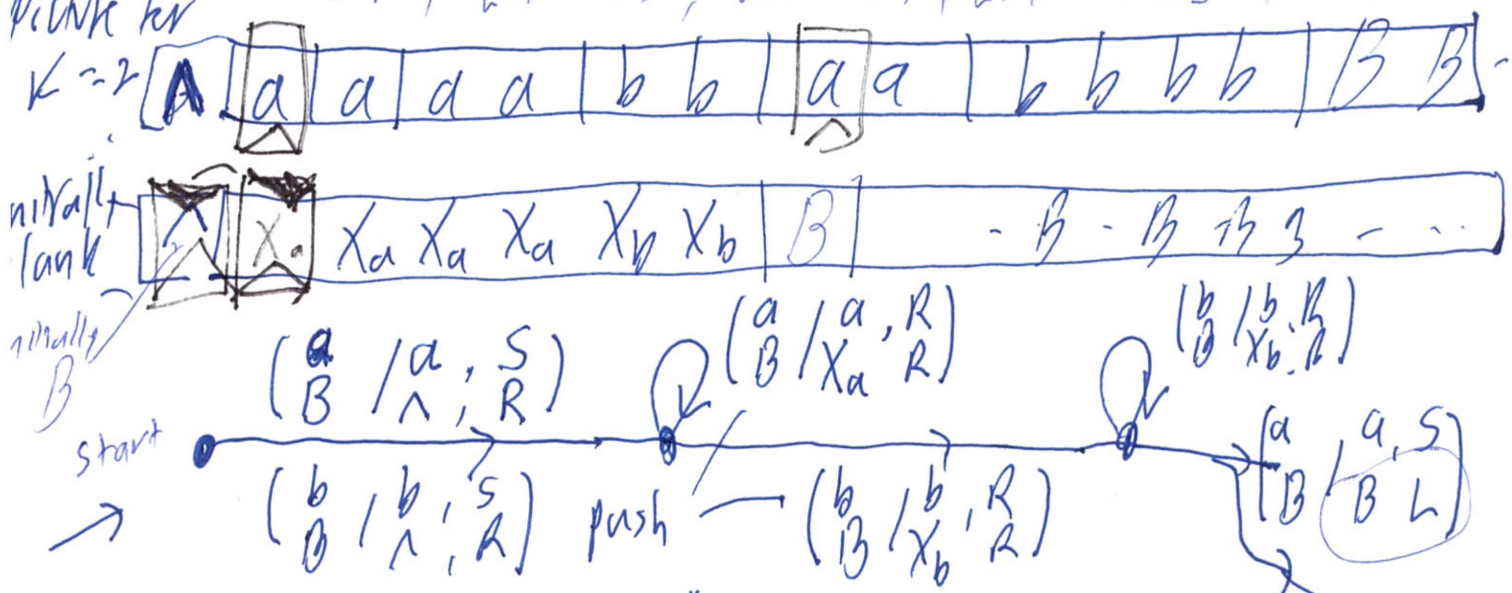
Otherwise, if there is any pair (q, c) with two or more tuples beginning $(q, c / \ldots)$ then M is properly an NTM.

Extension for any number K of tapes. $M = (Q, \Sigma, \Gamma, \delta, B, s, F)$ ③

with $\delta \subseteq Q \times \Gamma^K \times \Gamma^K \times \{L, R, S\}^K \times Q$,

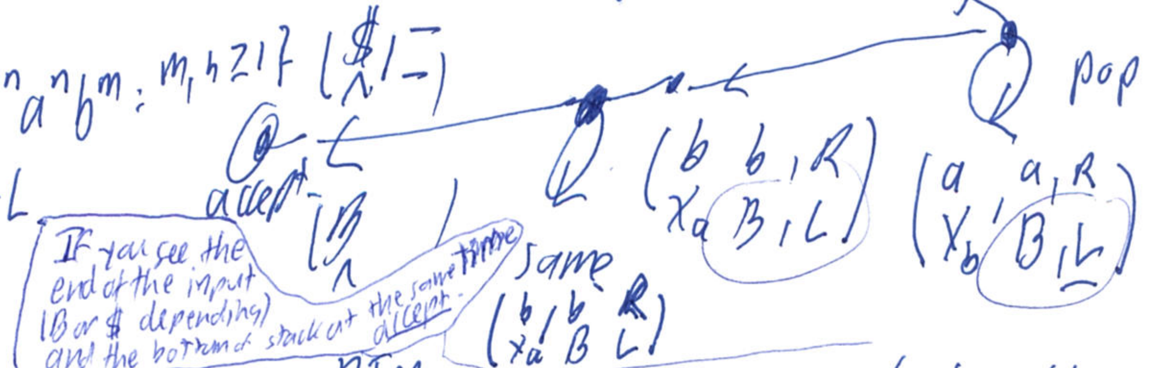
$(p, [c_1 \cdots c_K], [d_1 \cdots d_K], [D_1 \cdots D_K], q)$

Picture for $K = 2$



$L(M) = \{a^m b^n a^n b^m : m, n \geq 1\}$

which is a CFL



A DFA / NFA is a 1-tape DTM / NTM in which every tuple $(p, c/d, D, q)$ has $d = c$ and $D = R$.

A DPDA / NPDA is a 2-tape DTM / NTM in which every instruction

PDA = "Pushdown Automaton"

$\left(p, \begin{matrix} c_1 \\ c_2 \end{matrix} \middle/ \begin{matrix} d_1 \\ d_2 \end{matrix}, \begin{matrix} D_1 \\ D_2 \end{matrix}, q\right)$ has $d_1 = c_1$, $D_1 \neq L$, $D_2 = L \Rightarrow d_2 = B$ ← Tape 2 is a Stack.

Added: The definition of deterministic/nondeterministic is similar for all these forms. A K-tape TM is (properly) nondeterministic if two different instructions begin with the same $(q, [c_1, \ldots, c_K] / \ldots)$, else it is deterministic — and "completed" if there is one such tuple for any q in $Q \setminus \{q_{acc}, q_{rej}\}$ and chars $c_1, \ldots, c_K$ all in Γ, which makes δ into a function.