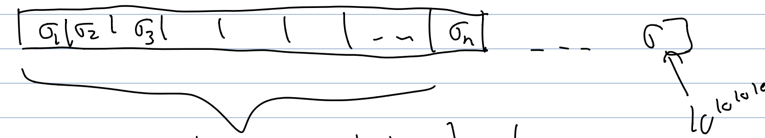
October 4! Begin Turing Machines (Ch. 3 [Sip])

Remark: DFA's:

input




$\sigma_1, \ldots, \sigma_n \in \Sigma$

length arbitrarily long

---

Make our algorithms just a bit more powerful...

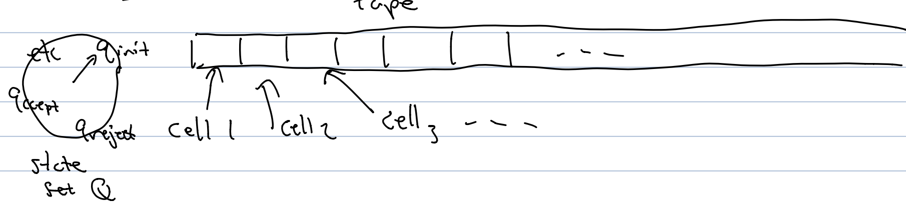
(1) input is on a read/write tape that is infinitely long

(2) can move L = left as well as R = right,

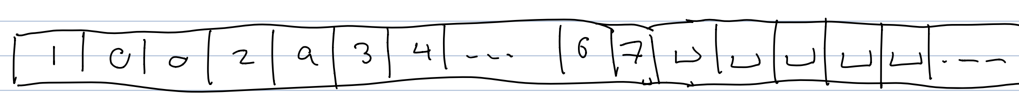
(3) we can write over any tape cell

For now that's it...

Turing Machine:




input is a word in $\Sigma$, $\Sigma$ = some alphabet. We have a "blank" symbol $\sqcup$



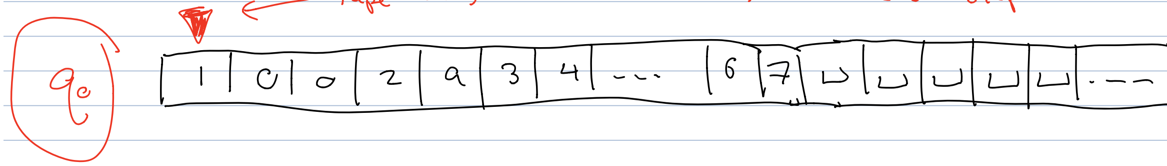

Turing Machine has $\Gamma$ = tape symbols, s.t. $\sqcup \in \Gamma$, $\Sigma \subset \Gamma$

The transition function $\delta : Q \times \Sigma \longrightarrow Q$ (DFA)

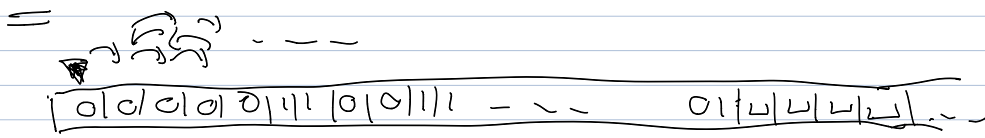
Turing machine $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$

Roughly speaking:

tape head, allowed to move L, R at each step




Example: $L = \{0^n 1^n \mid n \in \mathbb{N}\} = \{01, 0011, 0^3 1^3, \ldots\}$

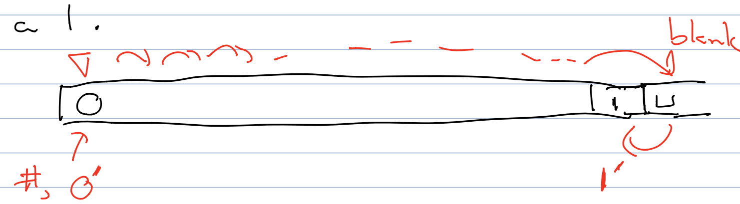



first blank symbol tells you you've seen all the input

You can choose $\Gamma$ ($\sqcup \notin \Sigma$)

$Q$ = set of states must be finite

Idea Algorithm: First you must see a 0, then go to the end, look for a 1.



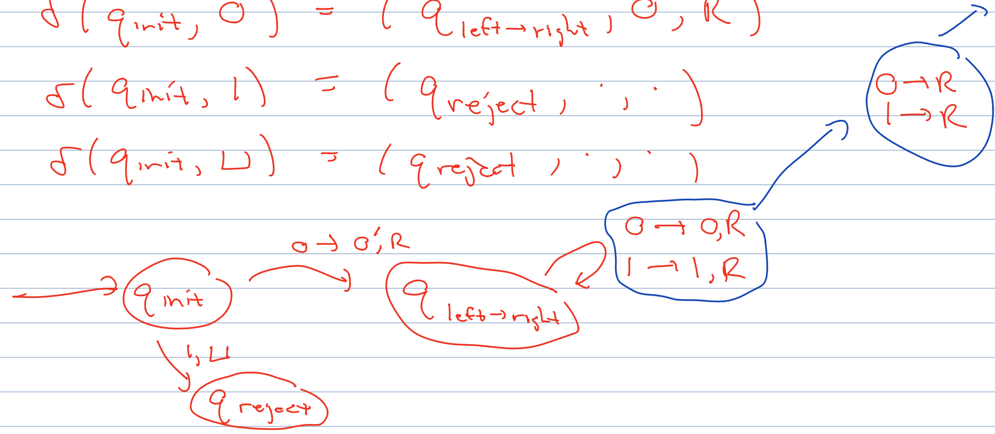

Let's make table of $\delta$ function values

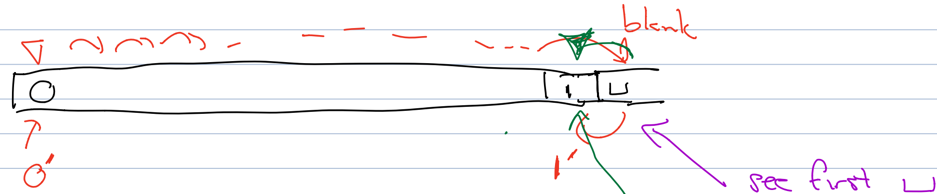
$\delta(q_{init}, 0) = (q_{left \to right}, 0', R)$

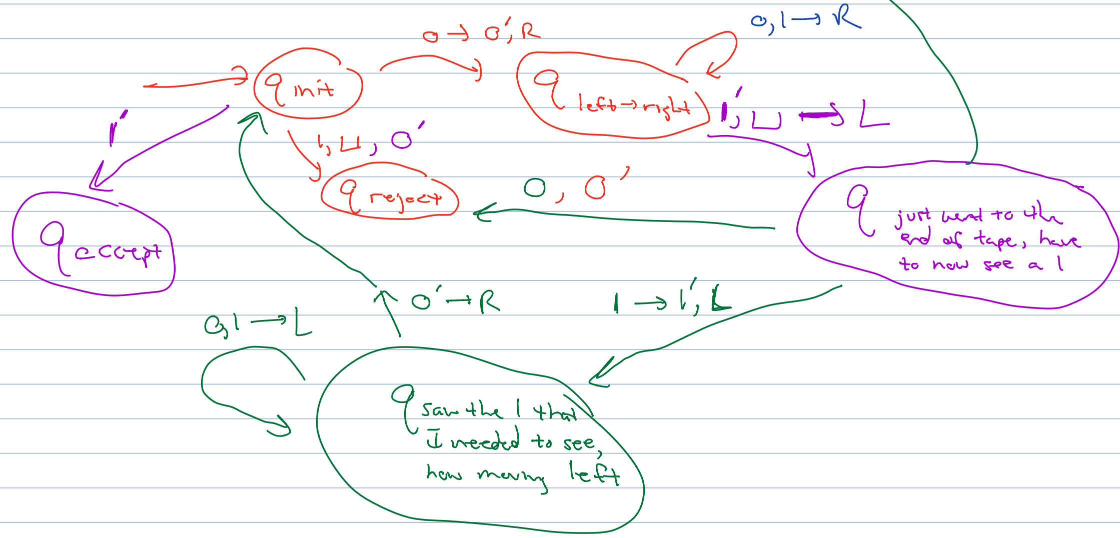
$\delta(q_{init}, 1) = (q_{reject}, \cdot, \cdot)$

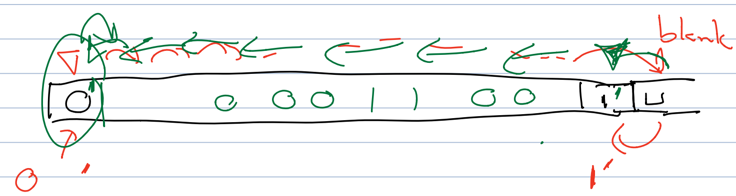
$\delta(q_{init}, \sqcup) = (q_{reject}, \cdot, \cdot)$

see 0, 1, move R
don't change the tape cell
0, 1 → R






Possible bad cases: 0 0 1 1 1 ⊔ ⊔

alg

0' 0 1 1 1'

0' 0' 1 1'

0' 0' 1 1' 1'

OK

0 ⊔ ⊔

0' ⊔ ⊔

0' ⊔ ⊔

0 0 1

0' 0 1'

0' 0' 1'