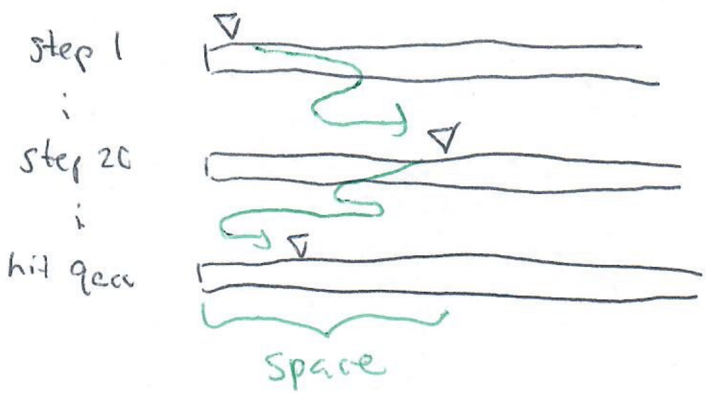
- Multitape Turing machines
- "Time" & "Space" that TM takes:

For a machine, M, on input $w \in \Sigma^*$, the

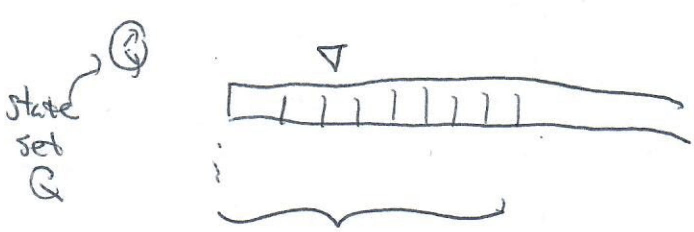
- time M takes on w is number steps in the computation until M reaches q_{acc} or q_{rej}
- space M takes on w: farthest cell that the tape head moves to the right



- Fact $\text{Space}(M,w) \le \text{Time}(M,w)$ since each step of computation move $\{L, R\}$
- Fact $\text{Time}(M,w) \le \text{function}(\text{Space}, |Q|, |\Gamma|) \approx \text{Exponential}_{|Q|,|\Gamma|}(\text{Space})$

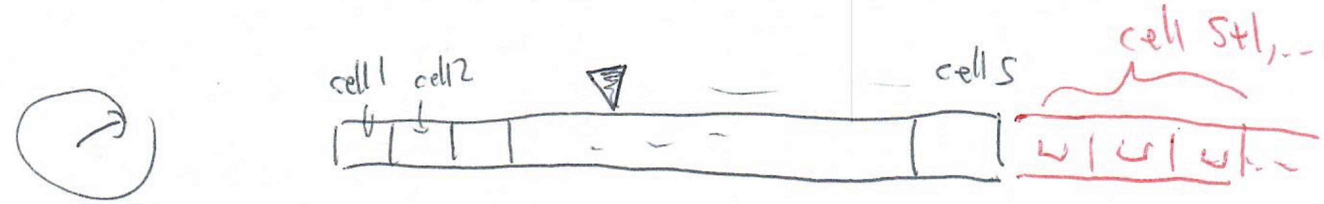
more precisely $\boxed{}^{\text{Space}(M,w)\ \cancel{}}$

function $|Q|, |\Gamma|$



move no more that 20 cells over

Total # of configurations gives space $\leq S$ ②



some state

of $|Q|$

$$\left\{\begin{array}{l} \#\text{ states} \leq |Q| \\ \text{locations} \leq S \\ \text{cell } 1 \leq |\Gamma|,\ \text{cell } 2 \leq |\Gamma|, \ldots \end{array}\right.$$

Total # poss configurations

$$\leq |Q| \cdot S \ |\Gamma|^S \leq |Q| S |\Gamma|^S$$

$$\leq \text{const} \cdot S \cdot (\text{const}')^S$$

If computation terminates... can't visit the same config twice (since computation is deterministic).

Time $\leq$ # possible configurations...

Back to $\{0^n 1^n\}$

$$\text{PALINDROME}_{\{0,1\}} = \{ s \in \{0,1\}^* \mid s = s^{\text{reverse}} \}$$

e.g. $011101110 \in$ PAL

$0111011 \notin$ PAL

Recognize $0^n 1^n$ with TM: 000011110 1...

→ move n

← move n

⋮

n each direction

$\underbrace{0 \ldots 0}_{n} \underbrace{1 \ldots 1}_{n}$

$O(n^2)$

PALINDROME similar:

w = 01110 . . . 010⊔⊔ is palindrome

|w| size n

Compare

n steps
n-1 steps
n-2 steps
⋮

Order n^2

TM: Use states to remember

q I just saw a 1 on the leftmost cell, and the cell before saw 1
q " — — 0 — — — — — " " — — — — — 0
" " — — — — — — 1
" " — — — — — — 0

You can remember any finite information,
at the cost of getting more states . . . ⇝ "speed-up" theorems

Why 2-tape machine?

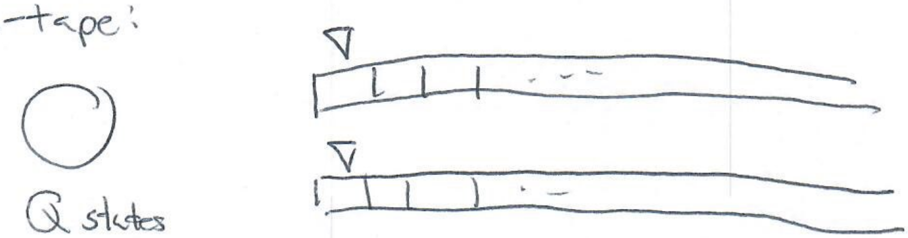
- more convenient
- PALINDROME takes time $\geq$ const n^2 1-tape machine
  $= O(n)$ 2-tape machine

$\underbrace{\text{1-tape machine / 2-tape machine}}_{\text{depends on machine model}}$

PALINDROME algorithm

0011011 0 . . . 00⊔⊔⊔

2-tape:



← input written on first tape

extra 2nd tape

2-tape machine: $(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

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

current state

~~tape~~ 1 tape head is reading

tape 2 tape...

$$\delta : Q \times \Gamma^2 \longrightarrow Q \times \Gamma^2 \to \{L,R,S\}^2$$

Why S = stay in $\{L,R,S\}$?

Palindrome: 1 0 1 0 1 1 0 1 0 ⊔ ⊔ tape 1

⊔ ⊔ ⊔ - - - - tape 2

Step 1: Mark beginning of tape 1

1' 0 1 0 1 - -

1' ⊔ ⊔ ⊔

Step 2 moving R

1' 0 1 ⊔

1' 0 ⊔ ⊔

⋮

1' 0 1 - - - 1 ⊔

1' 0 1 - - - 1 ⊔

Simpler example for $\{0^n 1^n\}$:

input → $\triangledown$ 0 0 0 . . 0 1 1 1 1 1 . . 1 ⊔ ⊔ tape 1

is it in $\{0,1\}^n$? $\triangledown$ ⊔ ⊔ ⊔ . . . tape 2

Stage 1: → R

0′ 0 . . . 0 1 1 1 . . . 1 ⊔ $\triangledown$ tape 1

0′ 0 0 . . 0 1 1 1 . . . 1 $\triangledown$ tape 2

→ R

input size m, m steps

Stage 2: $\triangledown$ ← R

0′ 0 0 . . 0 | 1 . . . 1 ⊔ tape 1

0′ 0 0 . . 0 | 1 . . . 1′ ⊔ tape 2

Staying = S

m steps

stage 3 → R

$\triangledown$ input

input ← $\triangledown$ L

m/2 steps

if get to:

. . 0 1 1 . .

$\triangledown$ 0 1 1 . .

\# 0's is same # 1's accept.

→ linear time, i.e. linear # of steps, if 2-tapes.

Friday: for deciding languages: 1-tape ⟺ 2-tape