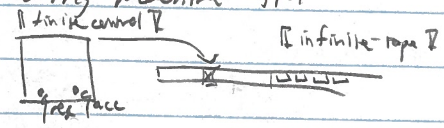
Michael Forbes
Mforbes@illinois.edu
2019-01-15.4 → 2019-01-17.1
2019-01-17.2 ← cs579

cs 579: Computational Complexity: Lecture 2

today: time complexity
- defn
- examples
- robustness

admin:
- ps1 out tonight, due 01-31
- sign up for piazza

def: a Turing machine TM:

M 

each step:
- read tape at head position
- write on tape at head position
- update finite control
- move tape head

M accepts x if reach q_{acc}

$L(M) = \{x : M \text{ acc } x\}$

Prop: exists TM M w/ $L(M) = \{0^n 1^n : n \in \mathbb{N}\}$

pf: [last time] $\emptyset \emptyset \times \times$

on input x:

$O(n)$ steps 1) pass over x, check if $x \in 0^*1^*$, reject if not

$\le n$ rounds 2) repeat until all symbols crossed off:

$O(n)$ steps [pass over input a) cross a single 0 → "$\emptyset$" $\in \Gamma$
b) cross a single 1

reject if 0's finish before 1's or vice versa

accept otherwise "

Q: how many steps does this take? $n = |x|$

$\le O(n) + n \cdot O(n) = O(n^2)$

Q: do better? [not just by constant factors]

Fact: cannot achieve $O(n)$

↳ Sipser #7.47 [won't solve here]

Prop: exists TM M w/ $L(M) = \{0^m 1^m\}$ that runs in $O(n \lg n)$ steps on length n inputs

idea:
- check for 0^*1^*
- given $0^a 1^b$ $\underbrace{\text{check if}}$ $a = b$ ← $\lg n$ bits

↳ $\lg n$ bits each check in $O(n)$ time

M = " on input x:

1) check if $x \in 0^*1^*$, reject if not

2) repeat until all symbols crossed off

a) count parity of #0's
b) count parity of #1's] reject if unequal

Michael Forbes
miforbes@illinois.edu
2019-01-17.2 ← 2019-01-17.2
→ 2019-01-17.3
cs579

c) cross off every other 0   eg $\cancel{0}0$, $\cancel{0}0\cancel{0}$

d) cross off every other 1

accept if we see ... ← $a \bmod 2'' \in \{0,1\}$

correctness — $0^a 1^b$ $a = 2a' + a''$

$b = 2b' + \underbrace{b''}_{b \bmod 2 \in \{0,1\}}$

- $a'' \neq b'' \Rightarrow a \neq b$

- crossing off: $0^a 1^b \to 0^{\lfloor a/2 \rfloor} 0^{\lfloor b/2 \rfloor}$ [ignoring crossed off]

$= 0^{a'} 0^{b'}$

$a \neq b$ but $a'' = b'' \Rightarrow a' \neq b'$ [now recurse]

run time: - each round is $\underbrace{O(1) \text{ passes on input}}_{O(n) \text{ steps}}$

- $O(\lg n)$ rounds as halve # alive symbols each round

$\Rightarrow O(n \lg n)$ ☒

Q: do better?

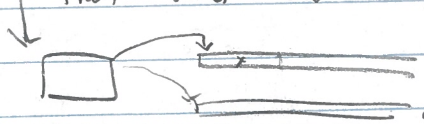
Fact: no $o(n \lg n)$ algo ← Sipser #7.47

[hence this is asymptotically optimal]

[but is it?]

Prop: exists two-tape TM M w/ $L(M) = \{0^m 1^m\}$ that takes $O(n)$ steps on any input

Pf.



← same as before

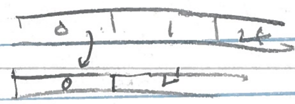
← initialized to blank

new: - both heads get updated each step
- can move independently

M = " on input x:

$O(n)$ 1) check if $x \in 0^* 1^*$, reject if not

$O(n)$ 2) copy all 0's onto 2nd tape



$O(n)$ 3) check # 1's on first tape = # 0's on 2nd tape "

Remarks: - time complexity changed w/ model

- but not by much

want: - theory of computation independent of specific model

- specific model to study = one tape TMs

- universality of specific model.

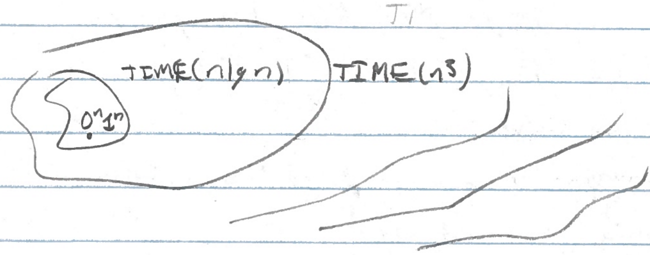
Michael Forbes
mforbes@illinois.edu

def: - TM M runs in time $t(n)$, $t(n): \mathbb{N} \to \mathbb{N}$, if for all inputs $x \in \Sigma^{n}$, M halts within $t(n)$ steps

input size

$\text{TIME}(t(n)) = \{ L \mid L = L(M),\ M \text{ runs in } \underline{O(t(n))} \text{ steps} \}$

ignores constant factors
[appropriate resolution of study]



Prop. $\text{TIME}_{2\text{-tape TM}}(t(n)) \subseteq \text{TIME}(t(n)^2)$ [one-tape] w/ $O(n^2)$ initialization

Pf idea - simulate $\underbrace{\text{each step}}_{\le t(n) \text{ steps}}$ on 2-tape TM by $\le t(n)+n$ steps on 1-tape

$\to \le t \cdot t + n^2$ steps $\approx t^2$ steps

assume $t(n) \ge n$ to read entire input

- encoding two tapes into one

$a_1 a_2 a_3$
$b_1 b_2 b_3$ $\longrightarrow$ $a_1 b_1 a_2 b_2 a_3 b_3 \ldots$

via interleaving

- 2-tape TM M has tape alphabet Γ

$\mapsto$ 1-tape TM has tape alphabet $\Gamma \cup \{ \dot{\gamma} : \gamma \in \Gamma \}$

signifies position of tape head

M' = "on input x:

$O(n^2)$ 1) convert tape into interleaved format

$x \sqcup \sqcup \sqcup \mapsto \dot{x}_1 \dot{\sqcup} x_2 \sqcup x_3 \sqcup x_4 \sqcup \cdots$

head position of first tape

position of second head

$O(t(n))$ 2) simulate each transition

[$\le O(t(n))$ many cells to check]

$O(t(n))$ a) scan tape to find head of first tape, do transition

$O(t(n))$ b) " second "

[heads may be far apart]

time $\Rightarrow O(t^2+n^2)$

correctness: clear

Michael Forbes
miforbes@illinois.edu
2019-01-17.4 ← 2019-01-17-3
→ 2019-01-22.1
CS579

Rmk: says 2 tape TM ≅ 1-tape TM, up to polynomial factors in runtime
↖ true of all known deterministic realistic models of computation

Extended Church Turing thesis ↖ for all realistic models [hence studying 1-tape is robust choice]
↳ might be false for quantum computation

defn $P = \bigcup_k \mathrm{TIME}(n^k) = \mathrm{TIME}(\mathrm{poly}(n))$ = polynomial time

Rmk: P is model invariant
roughly corresponds to "practical" solvability [n^{1000} not practical, $2^{n/1000}$ may be practical!?]
closed under subroutines

Q: what is in P?

today: time complexity
robustness
P

next time: - non determinism
- NP