SEP. 10. COMPLEXITY THEORY

Today:

- Turing Machines
  - Time + space complexity
  - Hierarchy Thms
  - Complexity classes
- Nondeterminism + NP
  - Cook-Levin Thm

(Multitape) Turing Machines

M is a TM w/ input x — say M halts on all inputs.

$t_M(x)$ = #steps M runs on x

$s_M(x) = \sum_i$ (max tape cell used on tape i)

for $n \in \mathbb{N}$,

$t_M(n) = \max_{x \in \Sigma^n} t_M(x)$ = most time needed on input of length n

$s_M(n) = \max_{x \in \Sigma^n} s_M(x)$ = most space needed on input of length n

Asymptotics

- We can always build a lookup table for small inputs, so we are interested in how t_M, s_M scale as $n \to \infty$
- Since we're being agnostic about alphabet size, we can always fudge space/time by constant factors. So we won't care about constant factors.

DEF. For $t: \mathbb{N} \to \mathbb{N}$, $s: \mathbb{N} \to \mathbb{N}$,

$$\mathrm{TIME}(t(n)) = \left\{ L \subseteq \Sigma^* : \begin{array}{c} L \text{ is decided by a TM w/} \\ t_M(n) = O(t(n)) \end{array} \right\}$$

$$\mathrm{SPACE}(s(n)) = \left\{ L \subseteq \Sigma^* : \begin{array}{c} L \text{ is decided by a TM } M \text{ w/} \\ s_M(n) = O(s(n)) \end{array} \right\}$$

SEP 10. p.2

Model + Input dependance

Model could be single tape / multitape / multitape random access

This does matter a bit: eg, PALINDROME = $\{x \in \Sigma^* : x = \mathrm{reverse}(x)\}$.

- single tape needs time $\Theta(n^2)$
- multitape can have $O(n)$ time, but need time · space = $\Omega(n^2)$
- random access can have $O(n)$ time $O(\log(n))$ space

But we won't care too much about the distinction between n and n^2, so this won't bother us too much. Same for reasonable changes in input.

Complexity Classes (Deterministic)

$$P = \bigcup_{c \geq 1} \mathrm{TIME}(n^c) \qquad \mathrm{PSPACE} = \bigcup_{c \geq 1} \mathrm{SPACE}(n^c)$$

$$\mathrm{EXP} = \bigcup_{c \geq 1} \mathrm{TIME}(2^{n^c}) \qquad L = \mathrm{LOGSPACE} = \mathrm{SPACE}(\log(n))$$

$$E = \bigcup_{c \geq 1} \mathrm{TIME}(2^{c \cdot n})$$

— Examples. —

$\{\langle G \rangle \mid G \text{ is bipartite}\} \in P$

$\{\langle G \rangle \mid G \text{ is Hamiltonian}\} \in \mathrm{EXP}$

Universality

Universal Turing Machines:

A TM M can be encoded as a string $\langle M \rangle \in \Sigma_\Gamma^*$ and let's assume every string z encodes M for some TM M. {eg. bogus string $\mapsto$ "hello world"}

Def A UNIVERSAL TM (UTM) takes $\langle M \rangle, x$ as input, and runs M on x.

Thm $\exists$ a UTM, ~~UTM~~ (also called UTM) s.t. $\forall$ TMs M and all inputs x,

$$t_{UTM}(\langle M, x \rangle) = c_M \cdot t_M(x) \log[t_M(x)]$$

$$s_{UTM}(\langle M, x \rangle) = c'_M \cdot s_M(x)$$

[for details, see Arora-Barack ch. 1] basic idea

Why is this interesting? M might have many more tapes/states/alphabet than UTM. To deal with this:

- concatenate M's tapes into one big tape
- Encode M's alphabet in binary

This gives $t_M(x)^2$, but you can be more clever about tape org. to get $t_M(x) \log(t_M(x))$.

HIERARCHY THMS.

"space constructible"

Thm. Let $s : \mathbb{N} \to \mathbb{N}$ be a "nice" function, s.t. $s(n) \geq \log(n)$.

Then $\mathrm{SPACE}(s(n)) \subsetneq \mathrm{SPACE}(s'(n))$

whenever $s(n) = o(s'(n))$.

Thm. Let $t : \mathbb{N} \to \mathbb{N}$ be "nice" w/ $t(n) \geq n$.

Then $\mathrm{TIME}(t(n)) \subsetneq \mathrm{TIME}(T(n))$

whenever $T(n) = \omega(t(n) \log(t(n)))$

eg. $\mathrm{SPACE}(\log(n)) \subsetneq \mathrm{SPACE}(\log^2(n))$

$\mathrm{TIME}(n) \subsetneq \mathrm{TIME}(n^2)$

Proofs go via DIAGONALIZATION + UTM.

Let's prove $\mathrm{TIME}(n) \subsetneq \mathrm{TIME}(n^2)$, and this will generalize.

We need to find

$\mathcal{L} \in \mathrm{TIME}(n^2)$, s.t. $\underbrace{\mathcal{L} \notin \mathrm{TIME}(n)}$

Aka, $\forall M$, $t_M(n) = O(n)$,

$\exists x$ s.t. $\mathcal{L}$, $\mathcal{L}(M)$ differ on x.

Choose the "counterexample" x for M to be $\langle M \rangle$ itself:

Consider the new machine (which uses a UTM) which does the following:

- on input $\langle M \rangle$:
  - run M on $\langle M \rangle$ and output the opposite.
  - for ~~s.t.~~ ~~steps~~ n^2 time
  - ~~supposed~~ ~~definitely~~ ~~halts~~

Then this definitely runs in time $O(n^2)$,

and we will reject every linear time language.

Catch: it might be that for small n, before the $O(\cdot)$ kicks in, we might ~~miss~~ fail. Hack to fix this: fix an enumeration of TMs so that each TM occurs infinitely many times.

← See Arora+Barak for details.

COROLLARIES.

(1) $P \subsetneq \mathrm{EXP}$

(2) $\mathrm{LOGSPACE} \subsetneq \mathrm{PSPACE}$

FACTS.

- $\mathrm{PSPACE} \subseteq \mathrm{EXP}$
- $L \subseteq P$

More generally, if $s(n) \geq \log(n)$, then

$$\mathrm{SPACE}(s(n)) \subseteq \bigcup_{c \geq 1} \mathrm{TIME}\left(2^{c \cdot s(n)}\right)$$

Idea: Configuration graph.

#configurations in a TM w/ space $s(n)$

is $\underbrace{O(1)}_{\text{state}} \cdot \underbrace{2^{O(s(n))}}_{\text{tape contents}} \cdot \underbrace{s(n)^{O(1)}}_{\text{heads}} \cdot \underbrace{n}_{\text{head on input tape}}$

$= 2^{O(s(n))}$ if $s(n) \geq \log(n)$

So together this gives

$L \subseteq P \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXP}$

strict strict

but we don't know if any of the others are strict

NONDETERMINISM

Recall that a (deterministic) TM has a transition fn $\delta : Q \times \Sigma^k \to Q \times \Sigma^k \times \{L, R\}^k$

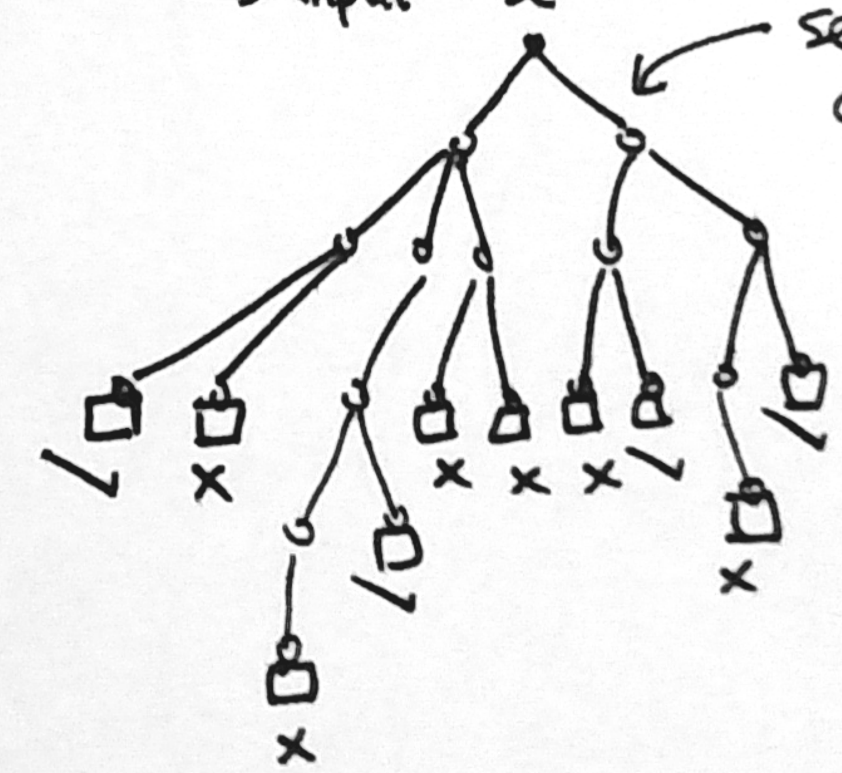
For a NON-DETERMINISTIC TM (NDTM), ~~the trans~~

we have a transition RELATION $\Delta \subseteq (Q \times \Sigma^k) \times (Q \times \Sigma^k \times \{L, R\}^k)$.

Visualize a computation as a tree:

→ input x

several possible computation paths



every leaf halts and either accepts or rejects

then define

$\mathcal{L}(M) = \{x : \exists$ a branch that accepts $\}$

Def

$$\mathrm{NTIME}(t(n)) = \{\mathcal{L} \mid \mathcal{L} = \mathcal{L}(M) \text{ for some NDTM that runs in time } O(t(n)), \forall \text{ inputs } x \text{ of length } n, \text{ on every path}\}$$

$$NP = \bigcup_{c \geq 1} \mathrm{NTIME}(n^c).$$

SEP 10. p.5.

Alternative characterization of NP :

Thm $\mathcal{L} \in NP \iff \exists$ $c<\infty$ and polynomial-time decidable relation A

s.t. $\mathcal{L} = \{ x \mid \exists y,\ |y| \leq |x|^c,\ (x,y) \in A \}$

aka, $\mathcal{L}$ is "efficiently verifiable."

Pf sketch $\Leftarrow$ Given such a relation A, build a NDTM as follows:

- branch by guessing y
- run the computation (that is guaranteed by A) to check if $(x,y) \in A$.

$\Rightarrow$ Given an NDTM, come up w/ A as follows:

- the y encodes all the nondeterministic choices that lead to the accepting path.

FACT. $P \subseteq NP$. Whether this is strict is the big open problem.

Next time: NP completeness + Cook-Levin Thm.