COMS 4236 Comp. Cxity

Last time: intro
computational problems
computational model (TMs)

Today: → resource bounds: time + space complexity
P, L, PSPACE, etc.
nondeterminism, NP, NP completeness

Readings: (same as last time) Sipser Ch. 3, 7
Arora/Barak: ch. 0, 1.1–1.6
Papad.: 2.1–2.5

Questions?

TIME COMPLEXITY

Def: • A (multitape) TM has __time cxity $T(n)$__
if $\forall |x| = n$ ("all inputs x of length n"),
M makes $\leq T(n)$ steps before haltings.

• Lang L is in TIME(T(n)) if $\exists$ TM M
deciding L in time $T(n)$.

a __cxity class__

Thm ("linear speedup") Let $T(n) = \omega(n)$.
If $L \in TIME(T(n))$, then $\forall\, \varepsilon > 0$,
also $L \in TIME(\varepsilon T(n))$.

(a const; 0.01)

M

Pf sketch: better hardware.

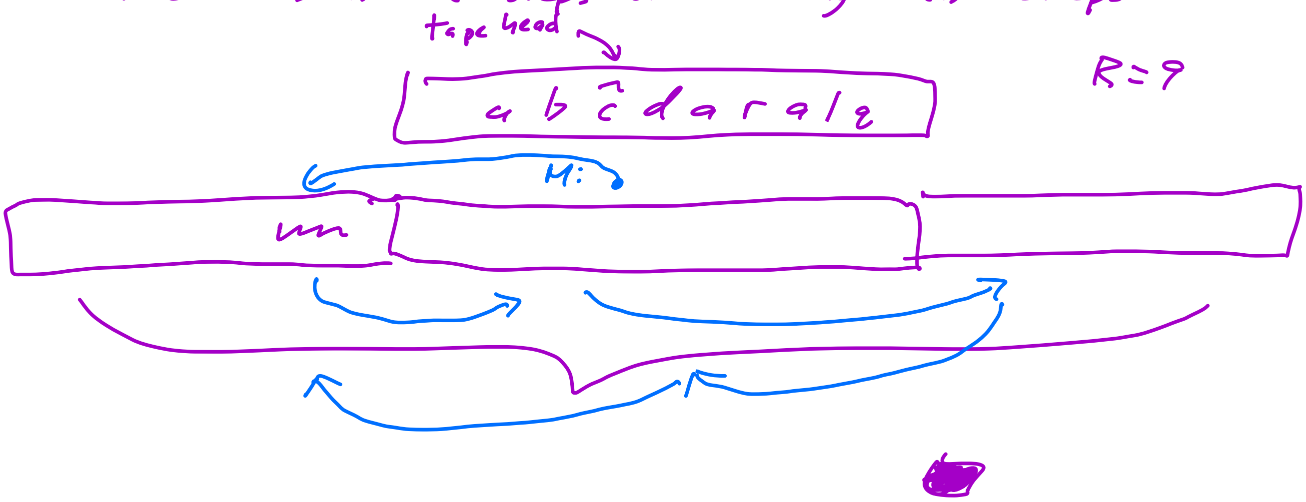
Let M' be TM simulating M but w/ bigger finite control, alphabet:

Σ = M's alph.

Σ^R = M''s alph. (factor R compression).

First n steps: M' copies + condenses x into $|x|/R$ tape cells.

Then sim. R steps of M using $O(1) \to 6$ steps:



So, $TIME(n^2)$ not $TIME(5n^2+6n+11)$

Mult. constants don't matter for our time bounds.

Even in exponent, we barely care about constants:

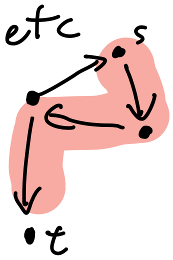
Key c.c.: P

Def $P = \bigcup_{k \geq 1} TIME(n^k)$.

"poly-time"

Ex: REACHABILITY (REACH):

REACH = $\{(G, s, t)$: G a digraph, s, t nodes, $\exists\ s \rightsquigarrow t$ dir. path in $G\}$



YES

Poly-time alg: DFS
BFS etc.

P interesting b/c:

- "efficient computation"
- lots of nat. algs are poly(n) time
- poly's don't grow too fast
- " are robust:
  - closed under $+$ $\quad p(n) + q(n)$
  - " " $\cdot$ $\quad p(n) \cdot q(n)$
  - " " compos $\quad p(q(n))$

all poly if p, q poly.

Holy grail of c.c.: For some nat. problem we're interested in, show <u>not</u> in P.

PARTITION, CLIQUE, CNF-SAT:

in P? Prob. not...

Other time classes: $TIME(n^{poly(\log n)})$

$$E = TIME(2^{O(n)}) = \bigcup_{k\geq 1} TIME(2^{kn})$$

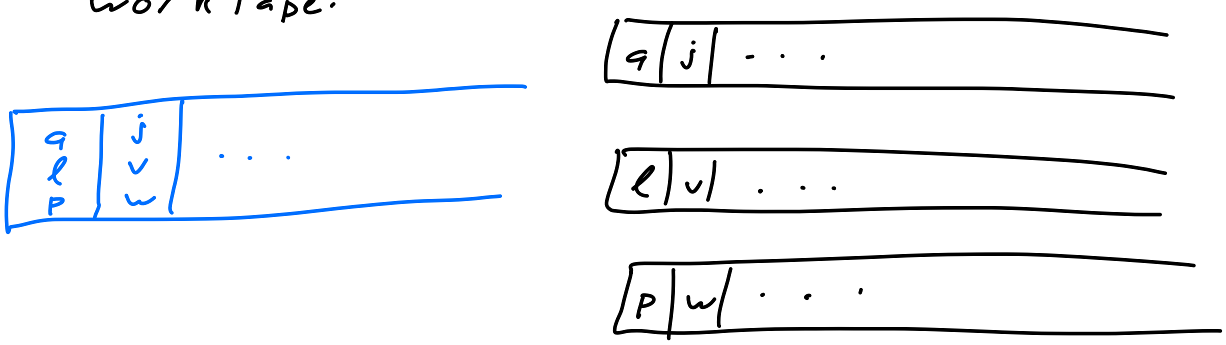
$$EXP = \bigcup_{k\geq 1} TIME(2^{n^k})$$

We'll show $P \neq E$, $P \neq EXP$.

SPACE CXITY

<u>Def:</u> A (multitape) TM M has <u>space cxity $S(n)$</u> if $\forall |x| = n$, M scans $\leq S(n)$ cells on each worktape on input x.

- M need not halt
- Don't count output length
- WLOG, can view as having one worktape:



Def: $L \in SPACE(S(n))$ if $\exists$ M deciding L with space cxity $S(n)$.

- Very reasonable to consider $s(n) = o(n)$.

Ex: $\{ww : w \in \Sigma^*\} \in SPACE(O(\log n))$

$O(\log n)$ space lets you maintain $O(1)$ many counters (to n) / pointers into input.

Thm: (lin. compr. for space): if $L \in SPACE(S(n))$, then $\forall \varepsilon > 0$, $L \in SPACE(\underbrace{\varepsilon \cdot S(n)})$ $\rightarrow$ actually $\max\{1, \lceil \varepsilon \cdot S(n) \rceil\}$

Impt space classes:

- $L = LOGSPACE = SPACE(\log n)$. $\rightarrow$ $L \subseteq P$ (we'll show)
- $PSPACE = \bigcup_{k>0} SPACE(n^k)$ $\qquad$ $P \subseteq PSPACE$.
- $EXPSPACE = \bigcup_{k>0} SPACE(2^{n^k})$, etc.

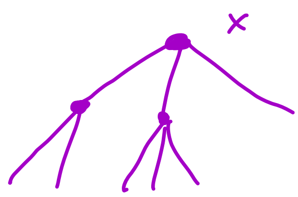
Nondeterminism

(AB chap 2, Sipser 7.3, Pap. 2.7)

↳ det TM: transit. rule δ is a function:
state q, tape cells $a_1, \ldots, a_k$ under heads:
only one next outcome of q, tape cells.

Nondet TM: δ is a relation. More than 1
poss. next outcome.

View as tree of poss. comput's on a fixed input.



- each poss. comput. = path thru tree.
- branching factor: fixed const. Think of binary nondeterm.
  (bin tree)

Def: NTM M decides lang L : $\forall x$,

$x \in L \iff$ there's some comput. of M on x that accepts;

$x \notin L \iff$ every comput. of M on x rejects.

"M can make lucky guesses".

Def: Runtime/time cxity of NTM M on

input x : max runtime (poss) of M on x over all seq. of comput. (acc, rejecting); = depth of tree.

Space cxity : analogous.

(Easy to "add a clock" to NTM to time out within a specified time/space bd.)

Def: $NTIME(T(n)) = \{ L : \exists$ NTM deciding L in $T(n)$ time$\}$.

$NSPACE(S(n))$ likewise.

Def: $NP = \bigcup_{k \geq 1} NTIME(n^k)$

More intuit. way to view NP:
$L \in NP$ iff $\forall x \in L$, $\exists$ eff. verifiable witness w that $x \in L$. (proof/cert.)
(+ if $x \notin L$, no such w exists).

Def: A verifier for lang. L: det alg. V s.t.

$x \in L \Rightarrow \exists$ witness w s.t. V accepts (x,w)
$x \notin L \Rightarrow \neg\exists$ " " " " " ".

A poly-time verifier for L is a verif V running in poly($|x|$) time $\forall (x,w)$.

WLOG $|w| \le$ poly($|x|$) if V is poly-time.

Claim: $NP = \{L : L$ has poly-time verifier$\}$.

Pf: $\Leftarrow$: Sps L has poly-time verif. NTM can guess w, then run det alg for V.

$\Rightarrow$: if $L \in NP$: N nondet TM for L, poly-time det verif V:

on input (x,w): view w as descrip. of seq. of nondet. choices that N should make.

V accepts (x,w) iff N acc. x under w as its seq of nondet. choices.

So, $\exists w$ causing V to acc (x,w)

$\Updownarrow$

N acc. x.