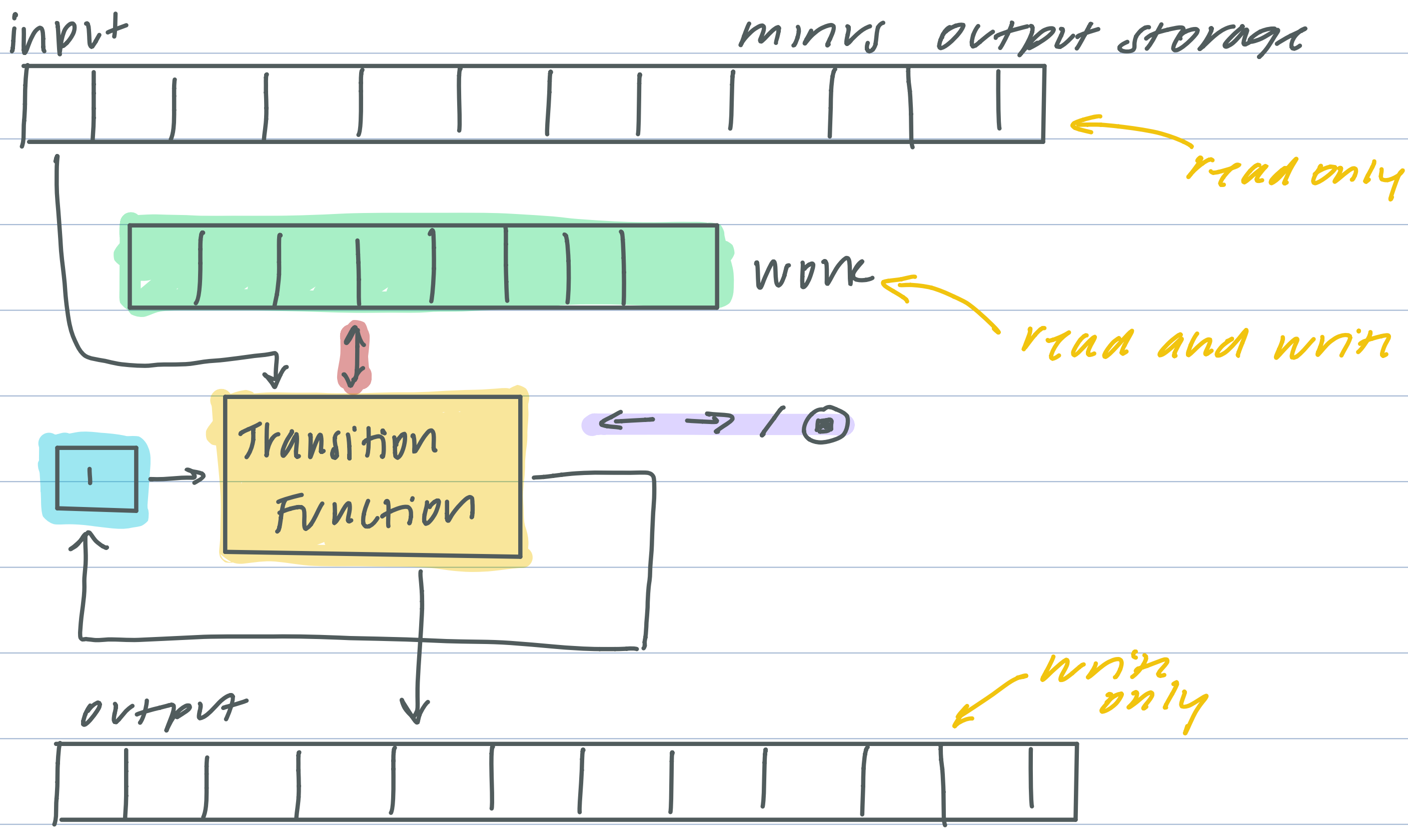
CS121 Lecture 23: Space Complexity Nov 19

Space Bounded Computation

- count memory used, minus input storage

minus output storage



Example Space Complexity

- adding two n-bit integers: $O(\log n)$
- Multiplication of two n-bit integers:
  - long mult: $O(n^2)$
  - better: $O(\log n)$
- Multiplication of nxn matrices w/ $O(\log n)$ bit entries: $O(\log n)$
- Sorting n elements w/ $O(\log n)$
  - Bubble: $O(\log n)$
  - merge: $O(n)$

Basic Problems

- CIRC-EVAL $(C, x) = C(x)$ for NAND-CIRC C and $x \in \{0,1\}^n$
  - Space $O(|C|)$
- PATH $(G, s, t) = 1$ if $\exists$ path from vertex s to t in directed graph G
- SAT$(\phi) = 1$ if $\exists x$ s.t. $\phi(x) = 1$
  - Space $O(n)$
- QSAT$(\phi) = 1$ if $\exists x_1 \forall x_2 \exists x_3 \cdots \forall x_{2n}\ \phi(x_1, \ldots x_{2n})$
  - Space $O(n)$
- Games (Chess, Go, Mahjong)
  - Space = O(board size)

Space Complexity Classes

SPACE$(s(n))$: class of all function $F: \{0,1\}^* \to \{0,1\}$ that can be computed by a TM in space $s(n)$

- Space $(O(1))$ = REGEXPs
- $L \stackrel{def}{=} \bigcup_c$ SPACE$(c \log n)$ - space feasible class
- PSPACE $= \bigcup_c$ SPACE(n^c) - infeasible w/ interesting problems
- * $L^2 = \bigcup_c$ SPACE$(c \log^2 n)$

Space Hierarchy Thm

- Universal Algorithm : $\exists$ universal algo U that takes as input a pair (M, x) and outputs $M(x)$. If M takes space $s(n)$ then $V(M, \cdot)$ uses space $O(s(n)+\log(n))$

- Space Hierarchy Thm: For every "nice" $s_1(n), s_2(n) \geq \log n$ w/ $s_1(n) = o(s_2(n))$, we have
  $SPACE(s_1(n)) \subsetneq SPACE(s_2(n))$.
  - Corollary: $L \subsetneq L^2 \subsetneq PSPACE$

Relation to Time Classes

- $TIME(f(n)) \leq \underline{SPACE(f(n)) \leq TIME(2^{f(n)})}$
  ↳ if it didn't halt, then repeats memory
- Thm: $TIME(f(n)) \leq SPACE\left(\frac{f(n)}{\log f(n)}\right)$ ← only for TMs
- Corollary: $L \leq P \leq PSPACE \leq EXP$
  * strictness unknown:
    (1) $L = P$?   (2) $P = PSPACE$?   (3) $PSPACE = EXP$?
    cannot both be true, violates space hierarchy
    cannot both be true, violates Time Hierarchy
  * if $L = P$, implies $PSPACE = EXP$
    true because of padding

Reductions and Completeness

- Def: $F \leq^{space}_{s(n)} G$ if $\exists$ an $s(n)$-space algorithm R that maps inputs of F to inputs of G s.t. $\forall X$, $F(X) = G(R(X))$
  Lemma 1: if (1) $F \leq_{s_1(n)} G$ w/ reduction R s.t. $R_n: \{0,1\}^n \to \{0,1\}^m$
  |

(2) $G \in SPACE(s_2(n))$

then $F \in SPACE(s_1(n) + s_2(m) + \log m)$

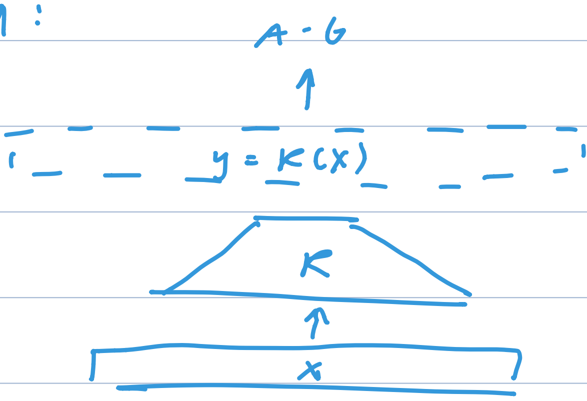
Proof: ~~(1) given input x to F, $y = K(x)$~~

~~(2) return $G(y)$~~

no space to write y.

space complexity $m + \max\{s_1(n), s_2(m)\}$

Idea:



Time increment:

$T_K(n) \cdot T_G(m)$

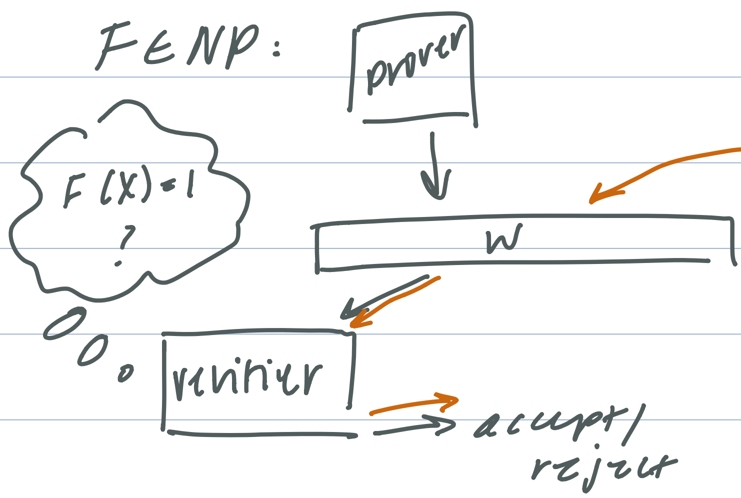
compared to $T_K(n) + T_G(m)$

Some complete problems

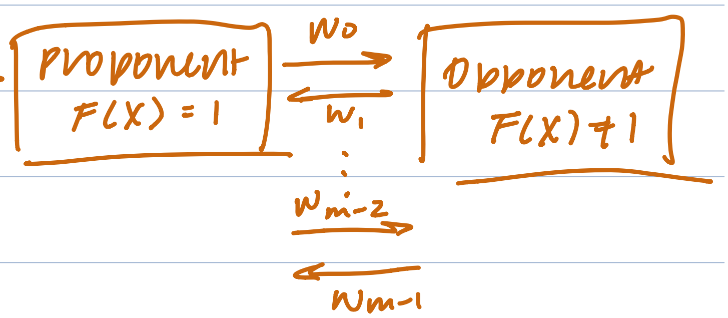
- CIRC-EVAL: P complete under $\leq_L^{space}$ reductions.
  - Implications: CIRC-EVAL $\in L \iff L = P$
- QBF: PSPACE complete under $\leq_p^{time}$ reductions
  - Also versions of Go / Chess / Debates
  - Implication: QBF $\in P \iff$ PSPACE $= P$

PSPACE

$F \in NP$:



$F \in PSPACE$



$F(x)=1 \rightarrow$

$\exists$ proponent $\forall$ opponent
$V(x,w)$ = Accept

$F \in PSPACE \rightarrow$

such a polynomial time V exists

Randomness and Nondeterminism

random	Nondeterminism
· tosses random coins internally	· takes input on input tape and witness on witness tape
· one-way access to random string	· two-way access to input, one-way access to witness tape

P, for example

$F \in NX$: (nondeterminism)

$\exists A \in X$ s.t. $\left(\forall x\ F(x)=1 \longleftrightarrow \Pr_y [A(x,y)=1] > 0\right)$

$F \in BPX$: (random)

$\exists A \in X$ s.t. $\forall x\ \Pr_y [A(x,y) = F(x)] \geq 2/3$

Classes

· random
BPPSPACE, BPL

· nondeterm
NPSPACE, NL

* $BPSPACE(s(n)) \subseteq SPACE(s(n)^2)$
  · corollary: BPPSPACE = PSPACE
  $L \subseteq BPL \subseteq L^2$
* $NSPACE(s(n)) \subseteq SPACE(s(n)^2)$
  · corollary: NPSPACE = PSPACE
  $L \subseteq NL \subseteq L^2$, unknown NL vs BPL

Interesting Problems

PATH(G, s, t)

- $\exists$ path from s to t in directed G
- PATH $\in$ NL , PATH $\in L^2$
- PATH is NL-complete

UPATH(G, s, t)

- same as PATH but undirected
- not known to be BPL-complete
- UPATH $\in$ L
- UPATH $\in$ BPL
  - Algo: random walk for n^3 steps starting at s

PATH $\in L^2$ Proof

$PATH_k(G, s, t) := 1$ iff $\exists$ path from s to t of len $\le k$

Claim 1:

$PATH_k \in SPACE(\log k \cdot \log n)$

Claim 2:

- $\exists$ log-space algo $R: G = (V, E) \to G^2 = (V, E^2)$ s.t.
  $PATH_{2k}(G, s, t) \leftrightarrow PATH_k(G^2, s, t)$ ☒

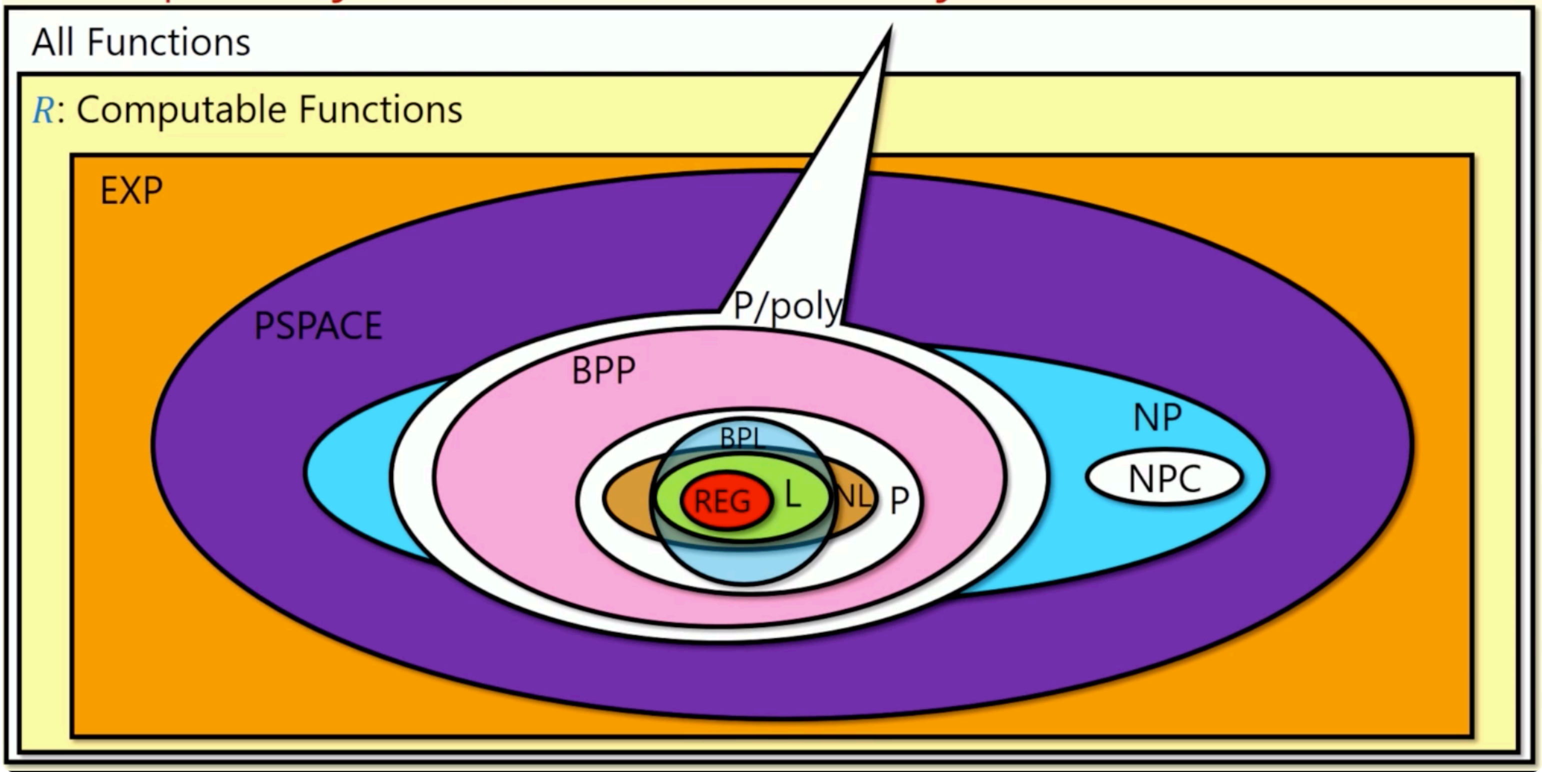
* reduction!

Claim 2 $\longrightarrow$ Claim 1

$PATH_{2k}(G, s, t) \le_{\log n} PATH_k(G^2, s, t) \in SPACE(\log k \cdot \log n)$

$\longrightarrow PATH_{2k}(G, s, t) \in SPACE((1 + \log k) \cdot \log n)$ ☒

Complexity Classes (Pictorially)



Take away lessons

- Space complexity: interesting + relevant measure: <u>Space is reusable</u>
- Classes:
  - NPSPACE = BPSPACE = PSPACE. (Complete problems = 2player games)
  - $L \subseteq BPL \subseteq L^2$; $L \subseteq NL \subseteq L^2$ (NL-Complete problem = PATH)
  - $L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXP$
- Proofs:
  - $F \leq_{s_1} G$ & $G \in \text{SPACE}(s_2) \Rightarrow F \in \text{SPACE}(s_1 + s_2)$
  - $\text{PATH} \in NL \cap L^2$
- References for today's lecture: These slides, Arora-Barak (Chapter 4) + references posted on Ed.