

Week 3

Announcements:

① HW1 posted Wed Sep 23

DUE : Oct 1 11:59 PM

② Practice problems for Test 1 will be posted next week

③ Test 1 : 1st 50 mins of class OCT 9

Today

1. Polytime Hierarchy
2. Space Complexity : [See Ch 3 Cai lectures in Comp. Complexity]

Define classes L , NL , $Pspace$

Relationships between classes

Savich's Thm

Complete problems for NL , $Pspace$

Space Hierarchy Thm

Next time : $\begin{cases} \text{Williams surprising Thm} \\ NL = coNL \end{cases}$

Poly time Hierarchy

Recall Verifier Defns of NP & coNP:

NP: $L \in \text{NP}$ iff $\exists$ verifier $V(x, y)$ such that:

2 inputs x : input
 y : extra input

① $x \in L \Leftrightarrow \exists y \ V(x, y) = 1$

② $V(x, y)$ runs in time $\text{poly}(|x|)$

coNP: $L \in \text{coNP}$ iff $\exists$ verifier $V(x, y)$ such that

① $x \in L \Leftrightarrow \forall y \ V(x, y) = 1$

② $V(x, y)$ runs in time $\text{poly}(|x|)$

could equivalently define as :

[NP:] $L \in NP$ iff $\exists$ **polytime** verifier $V(x, y)$ such that :

$$x \in L \Leftrightarrow \exists y \in \{0, 1\}^{\text{poly}(|x|)} [V(x, y) = 1]$$

[coNP:] $L \in \text{coNP}$ iff $\exists$ **polytime** verifier $V(x, y)$ such that

$$x \in L \Leftrightarrow \forall y \in \{0, 1\}^{\text{poly}(|x|)} [V(x, y) = 1]$$

could equivalently define as :

$\boxed{NP :}$ $L \in NP$ iff $\exists$ **polytime** verifier $V(x, y)$ such that :

↑
also called
 Σ_1^P

$$x \in L \Leftrightarrow \exists y \in \{0, 1\}^{\text{poly}(|x|)} [V(x, y) = 1]$$

$\boxed{coNP :}$ $L \in coNP$ iff $\exists$ **polytime** verifier $V(x, y)$ such that

↑
also
called
 Π_1^P

$$x \in L \Leftrightarrow \forall y \in \{0, 1\}^{\text{poly}(|x|)} [V(x, y) = 1]$$

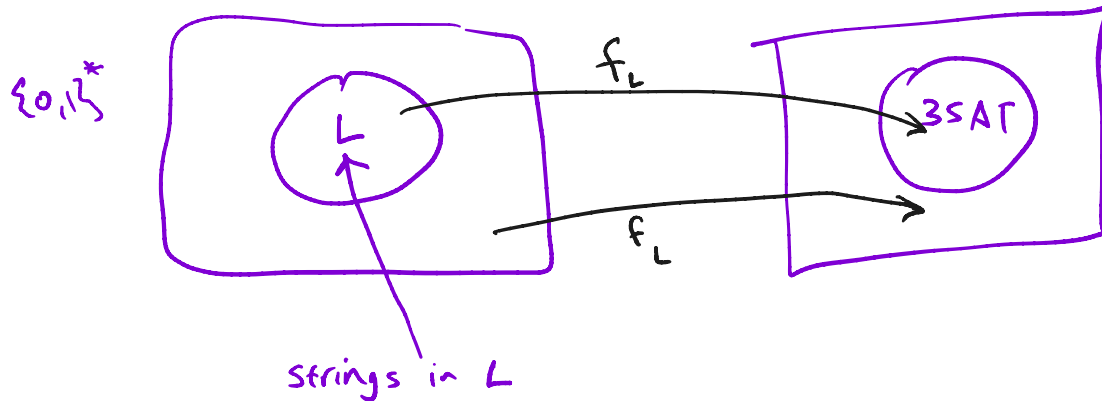
Example

$3SAT = \{ F \mid F \text{ is a satisfiable 3CNF formula} \}$

$3UNSAT \approx \overline{3SAT} = \{ F \mid F \text{ is UNSATISFIABLE 3CNF formula} \}$

• 3SAT IS NP-COMPLETE

• 3UNSAT is coNP-complete why? $\forall NP \text{ } L \exists \text{ptime } f_L$



Polytime Hierarchy (see LC section 2.3)

$\boxed{\Sigma_2^P :}$ $L \subseteq \{0,1\}^* \in \Sigma_2^P$ iff $\exists$ polytime verifier $V(x, y_1, y_2)$ such that

$$\forall x \ x \in L \Leftrightarrow \exists y_1 \in \{0,1\}^{\text{poly}(|x|)} \forall y_2 \in \{0,1\}^{\text{poly}(|x|)} V(x, y_1, y_2) = 1$$

$\boxed{\Pi_2^P :}$ $L \subseteq \{0,1\}^* \in \Pi_2^P$ iff $\exists$ polytime verifier $V(x, y_1, y_2)$ such that

$$\forall x \ x \in L \Leftrightarrow \forall y_1 \in \{0,1\}^{\text{poly}(|x|)} \exists y_2 \in \{0,1\}^{\text{poly}(|x|)} V(x, y_1, y_2) = 1$$

Polytime Hierarchy (see LC section 2.3)

$$\boxed{\Sigma_k^P :}$$

$L \in \{0,1\}^* \in \Sigma_k^P$ iff $\exists$ polytime verifier $V(x, y_1, y_2, \dots, y_k)$ such that:

$$\forall x \quad x \in L \Leftrightarrow \underbrace{\exists y_1, \forall y_2 \exists y_3 \dots}_{\substack{k \text{ alternations} \\ \text{starting with } \exists}} [V(x, y_1, y_2, \dots, y_k) = 1]$$

where all $|y_i|$ are poly bounded in $|x|$

$$\boxed{\Pi_k^P :}$$

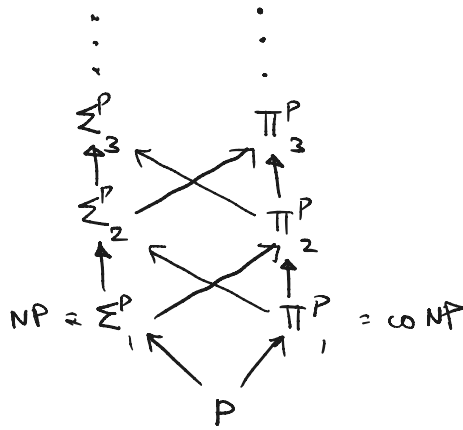
$L \in \{0,1\}^* \in \Pi_k^P$ iff $\exists$ polytime verifier $V(x, y_1, y_2, \dots, y_k)$ such that:

$$\forall x \quad x \in L \Leftrightarrow \underbrace{\forall y_1 \exists y_2 \forall y_3 \dots}_{\substack{k \text{ alternations} \\ \text{starting with } \forall}} [V(x, y_1, y_2, \dots, y_k) = 1]$$

where all $|y_i|$ are poly bounded in $|x|$

Defn (PH) $PH := \bigcup_k \Sigma_k^P = \bigcup_k \Pi_k^P$

Pictorially:



Example of complete Problem for Σ_2^P : Σ_2^b -QBF-SAT

Input : Quantified ^{Boolean} formula G over vars $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_k\}$, ← vars $\vec{x}$ are partitioned into 2 blocks
 $G : \underbrace{\exists \vec{x}_1, \forall \vec{x}_2}_{2 \text{ alternations}} F(\vec{x}_1, \vec{x}_2)$ ← F CNF

$$G \in \Sigma_2^b\text{-QBF SAT} \Leftrightarrow \exists \alpha_1 \in \{0,1\}^{|\vec{x}_1|} \forall \alpha_2 \in \{0,1\}^{|\vec{x}_2|} F(\vec{\alpha}_1, \vec{\alpha}_2)$$

Verifier : $V(q, \vec{x}_1, \vec{x}_2)$: check if $q(\vec{x}_1, \vec{x}_2)$ is satisfiable

Similarly Σ_k^b -QBF SAT is complete for Σ_k^P

Space Complexity

Recall A TM for deciding a Language $L \subseteq \{0,1\}^*$ has

- a read-only input tape (with input-tape head)
- k read/write work tapes

Defn A TM M has space complexity $S(n)$ if

$\forall x, |x|=n$, M scans $\leq S(n)$ cells on every work tape
on input x (need not halt)

Space Complexity

Defn A TM M has space complexity $S(n)$ if

$\forall x, |x|=n$, M scans $\leq S(n)$ cells on every work tape on input x (need not halt)

Defn $L \in \text{SPACE}(S(n))$ if $\exists$ a deterministic TM M that decides L and has $\text{SPACE } O(S(n))$ always halts

Note space can be $\ll$ length of input

Defn $\text{LogSPACE} = \text{SPACE}(\log n)$

$\text{PSPACE} = \bigcup_{k \geq 0} \text{SPACE}(n^k)$

Space Complexity

Defn A non-det TM M has non-det space complexity $S(n)$ if

$\forall x, |x|=n$, M scans $\leq S(n)$ cells on every work tape on input x (need not halt)

Defn $L \in \text{NSPACE}(S(n))$ if $\exists$ a non-det deterministic TM M that decides L and has $\text{NSPACE } O(S(n))$

Note space can be $\ll$ length of input

Defn " L " = $\text{LogSPACE} = \text{SPACE}(\log n)$

$$\text{PSPACE} = \bigcup_{k \geq 0} \text{SPACE}(n^k)$$

"NL" = $\text{NSPACE}(\log n)$

$$\text{NPSPACE} = \bigcup_{k \geq 0} \text{NSPACE}(n^k)$$

The Reachability Problem for Space Bounded TMs

- Let M be a $NSPACE(s(n))$ TM, deciding $L \subseteq \{0,1\}^*$

Then we can formulate whether $x \in L$

by a directed graph reachability problem,
on a graph $G_{M,x}$

$G_{M,x}$: directed graph

vertices :

- $\log n$ bits to specify input head location
- $O(1)$ bits to specify state
- $O(s(n))$ bits to specify contents of work tape(s) and head locations

V_0 = start config : V_0 : input head at posn 1, work head posn 1,
state = q_0
work tape all blank

V_{acc} = accept config : merge all vertices with state = q_{accept}

$\therefore$ # vertices = $2^{O(s(n))}$ (since $s(n) \geq \log n$)

* Note the input tape always contains x (since input tape is read only)
so x is "hard-coded" and doesn't need to be
part of description of configurations/vertices of $G_{M,x}$

$G_{M,x} :$

Edges : edge from $v \rightarrow v'$

iff $M(x)$ can transition from config v
to config v' in one transition

since M is nondeterministic, can have more than
one edge out of any vertex

Thus we have:

Proposition Let M be an $NSPACE(s(n))$ TM accepting L .

Let x be an input. M accepts x iff

there is a path from v_0 to v_{acc} in $G_{M,x}$

Reachability

Input: Directed graph g , vertices s, t

Accept iff there is a (directed) path from s to t in g

If $g = (V, E)$ $|V| = n$ then $|g| = n^2$ (adjacency matrix)

so input $\langle g, s, t \rangle$ has length $n^2 + 2 \log n = O(n^2)$

Lemma 0 Reachability $\in$ TIME ($\text{poly}(n)$)

Pf: By depth-first search

Theorem For any $s(n) \geq \log(n)$
 $NSPACE(s(n)) \leq DTIME(2^{O(s(n))})$

Pf Let M decide L in $NSPACE(s(n))$.

on input x , create directed graph $G_{M,x}$ on $2^{O(s(n))}$ vertices

Run time $\text{poly}(2^{O(s(n))}) = 2^{O(s(n))}$ reachability alg [Lemma 0]

on $(G_{M,x}, v_0, v_{acc})$. Accept iff v_{acc} reachable from v_0

Correctness by proposition.

Theorem (Savitch) Let $S(n) \geq \log n$. Then
 $NSPACE(S(n)) \subseteq SPACE(S^2(n))$

Pf Key idea:
Reachability solvable in space $O((\log n)^2)$, $n = \# \text{ vertices}$

* note DFS alg uses space n

Lemma 1 Reachability $\in \text{SPACE}(\log^2 n)$

← deterministic $\log^2 n$ space

Proof of Savitch's Thm from Lemma 1

Let M be $\text{SPACE}(s(n))$ TM

given M, x

Run Reachability alg from Lemma 2
on $(g_{M,x}, v_{0,x}, v_{acc})$

Note: $g_{M,x}$ is
given implicitly on
input tape by M, x

since $|g_{M,x}| = 2^{O(s(n))}$, alg runs in space

$$O(\log(|g_{M,x}|)^2) = O(s(n)^2)$$

Lemma 2 Reachability $\in \text{SPACE}(\log^2 n)$

← deterministic $\log^2 n$ space

Idea: Divide and conquer / recursive algorithm

$\exists$ path from $s \rightsquigarrow t$ of length $\leq n$ iff

$\exists$ vertex x such that: (i) $\exists$ path $s \rightsquigarrow x$ of length $\leq \frac{n}{2}$ and
(ii) $\exists$ path $x \rightsquigarrow t$ " " $\leq \frac{n}{2}$

↑
can be implemented using $O(\log^2 n)$ space

Main tape 1: stack that stores recursive history of calls

Reach (u, v, t) :

IF $t = 1$ then

IF $(u = v)$ or $(u, v) \in E \Rightarrow \text{accept}$

Else $\Rightarrow \text{reject}$

IF $t > 1$ then

For each vertex x in g :

IF $[\text{Reach}(u, x, t/2) \wedge \text{Reach}(x, v, t/2)] \Rightarrow \text{accept}$

IF haven't yet accepted $\Rightarrow \text{reject}$

Initially call $\text{Path}(s, t, n)$

Reach (u, v, t) :

If $t = 1$ then

If $(u = v)$ or $(u, v) \in E \Rightarrow \text{accept}$

Else $\Rightarrow \text{reject}$

If $t > 1$ then

For each vertex x in g :

If $[\text{Reach}(u, x, t/2) \wedge \text{Reach}(x, v, t/2)] \Rightarrow \text{accept}$

If haven't yet accepted $\Rightarrow \text{reject}$

Space complexity of Reach (s, t, n) :

Recursion depth $\log n$

Stack needs to store info about $\log n$ recursive calls

Each call : store u, v, t, x used in that call $] O(\log n)$

$\therefore O((\log n)^2)$ space

Space Complexity - Relationships

* Recall space(Time $f(n)$)
means $O(f(n))$

$$(1) \text{SPACE}(f(n)) \subseteq \text{NSPACE}(f(n))$$

$$(2) \text{TIME}(f(n)) \subseteq \text{NTIME}(f(n))$$

$$(3) \text{NTIME}(f(n)) \subseteq \text{TIME}(2^{O(f(n))})$$

$$(4) \text{TIME}(f(n)) \subseteq \text{SPACE}(f(n)) \text{ easy}$$

stronger: $\text{NTIME}(f(n)) \subseteq \text{SPACE}(f(n))$?

Simulation in 3 only needs space $O(f(n))$

Space Complexity - Relationships

- ① $SPACE(f(n)) \subseteq NSPACE(f(n))$
- ② $TIME(f(n)) \subseteq NTIME(f(n))$
- ③ $NTIME(f(n)) \subseteq TIME(2^{O(f(n))})$
- ④ $TIME(f(n)) \subseteq SPACE(f(n))$
stronger: $NTIME(f(n)) \subseteq SPACE(f(n))$
- ⑤ For $f(n) \geq \log n$: $NSPACE(f(n)) \subseteq TIME(2^{O(f(n))})$
- ⑥ For $f(n) \geq \log n$: $NSPACE(f(n)) \subseteq SPACE(f(n)^2)$

Savich '70

$$\textcircled{1} \quad \text{SPACE}(f(n)) \subseteq \text{NSPACE}(f(n))$$

$$\textcircled{2} \quad \text{TIME}(f(n)) \subseteq \text{NTIME}(f(n))$$

$$\textcircled{3} \quad \text{NTIME}(f(n)) \subseteq \text{TIME}(2^{O(f(n))})$$

$$\textcircled{4} \quad \text{TIME}(f(n)) \subseteq \text{SPACE}(f(n))$$

stronger: $\text{NTIME}(f(n)) \subseteq \text{SPACE}(f(n))$

$$\textcircled{5} \quad \text{For } f(n) \geq \log n : \quad \text{NSPACE}(f(n)) \subseteq \text{TIME}(2^{O(f(n))})$$

$$\textcircled{6} \quad \text{For } f(n) \geq \log n : \quad \text{NSPACE}(f(n)) \subseteq \text{SPACE}(f(n)^2) \quad \text{Savich '70}$$

Consequences of ①-⑥:

$$L \subseteq NL \subseteq P \subseteq NP \subseteq \text{PSPACE} = \text{NPSPACE} \subseteq \text{EXPTIME}$$

①

⑤

②

④

①

⑤

+

⑥

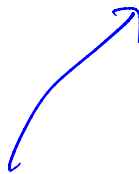
SPACE SURPRISES! (Next class)

⑦ Immerman, Szelepcsenyi (1988)

$$\text{NSPACE}(f(n)) = \text{coNPSpace}(f(n))$$

⑧ Williams (2025)

$$\text{TIME}(f(n)) \subseteq \text{SPACE}(\sqrt{f(n) \log f(n)})$$

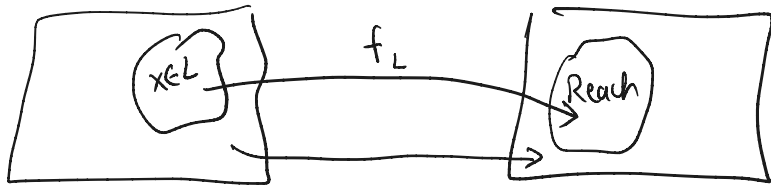
 we'll do this next class

NL-completeness

Thm

(Directed) Graph reachability is complete for NL
(under logspace reductions)

That is: $\forall L \in \text{LOGSPACE} \exists \text{ logspace function } f_L : \{0,1\}^* \rightarrow \{0,1\}^*$
such that $\forall x \quad x \in L \Leftrightarrow \underbrace{f_L(x)}_{\text{input } (g,s,t) \text{ to Reach}} \in \text{Reach}$



PSPACE - COMPLETENESS

① Any $L \in PH$ is in $PSPACE$ (exercise)

② QBF - SAT is PSPACE-complete

Input : QBF : $Q_1 x_1 Q_2 x_2 Q_3 x_3 \dots Q_n x_n \underbrace{F(x_1 \dots x_n)}_{\text{CNF}}$
each Q_i either $\forall$ or $\exists$

generalizes
 Σ_k -QBF SAT
Here number of
alternations can
be as large as n

Ex: $\exists x_1 \exists x_2 \forall x_3 \exists x_4 F(x_1, x_2, x_3, x_4)$

accept iff $\exists \alpha_1 \in \{0,1\} \exists \alpha_2 \in \{0,1\} \forall \alpha_3 \in \{0,1\} \exists \alpha_4 \in \{0,1\} F(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = 1$

space hierarchy Thm: Let $t(n) = o(T(n))$, $t(n) \geq \log n$,
and t, T are "nice". Then $SPACE(T(n)) \not\subseteq SPACE(t(n))$

Proof similar to time-hierarchy Thm

Consequences: $PSPACE \not\subseteq LOGSPACE$

$EXSPACE \not\subseteq PSPACE$