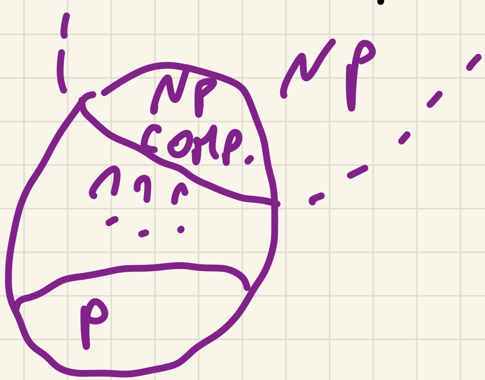
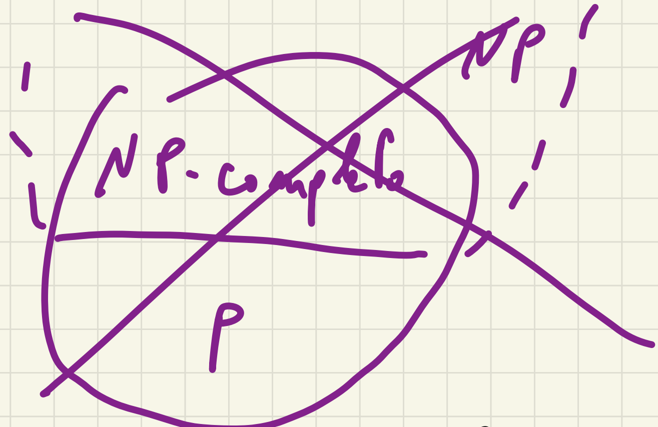
20/12

coNP

Recap: Abbiamo definito complessità di tempo, ovvero P, NP.
NP-completezza: SAT è NP completo.
Altri problemi NP-completi: HAMPATH, CLIQUE, 3COL, ...





TEO Se $P \neq NP$, $\exists L \in NP$ t.c. $L \notin P$ e L non è NP-comp.

Idee:

- NP certifica $x \in L$
- coNP certifica $x \notin L$.

Q: $UNSAT = \overline{SAT} \in NP$?

DEF $coNP = \{ L : \overline{L} \in NP \}$.

$\Rightarrow UNSAT \in coNP$.

Importante: $coNP \neq \overline{NP}$.

TEO $SAT \in P$ sse $UNSAT \in P$.

DIM. Se $SAT \in P$, esiste decisore per SAT.
Decisore per UNSAT: invertire la risposta del decisore per SAT.
Se $UNSAT \in P$, la stessa cosa. $\blacksquare$

Nota: questo non vuol dire: $UNSAT \leq_m^P SAT$.
Non sappiamo dimostrarlo.

Es.: Prendo $\langle \phi \rangle$ e lo mappo in $\langle \neg \phi \rangle$.
$\phi = x \vee y \in SAT$; ma anche $\neg \phi \in SAT$.

TEO. P è chiuso rispetto al complemento: $L \in P$ sse $\overline{L} \in P$.

Dim. La stessa di prima.

COR. $coP = P$. $coEXP = EXP$.

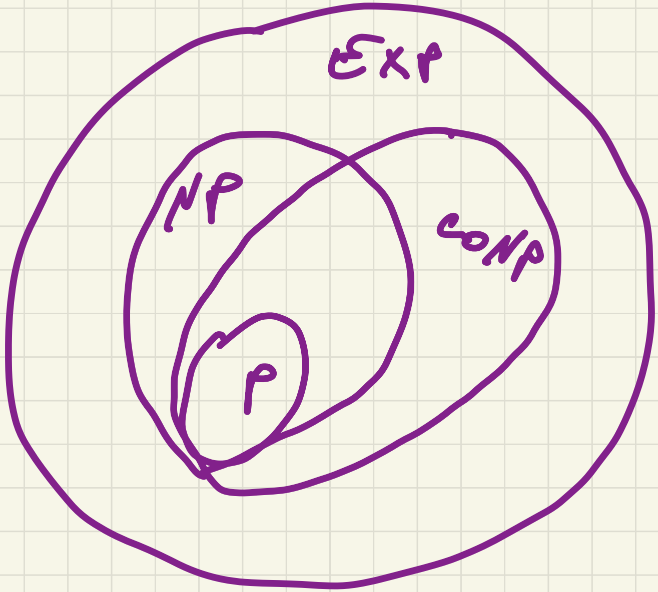
COR. $coNP \subseteq EXP$.

Dim. Sia $L \in coNP$. $\overline{L} \in NP \subseteq EXP$. Quindi $\overline{L} \subseteq EXP$, ovvero $L \subseteq coEXP = EXP$. ■

TEO. $P \subseteq coNP$.

Dim. $L \in P$; $\overline{L} \in P \subseteq NP$. $\overline{L} \in NP$, $L \in coNP$ ■

Quindi P è nascosto in entrambe (vedi figura).



TEO $P = NP \Rightarrow P = coNP$.

DIM. Basta mostrare $coNP \subseteq P$. Sia $L \in coNP$ ovvero $\overline{L} \in NP = P$. Ma P chiuso per complemento: $L \in P$. ■

COR $P = NP \Longrightarrow coNP = NP$

COR $coNP \neq NP \Longrightarrow P \neq NP$.

TEO $NP = coNP$ sse $UNSAT \in NP$.

DIM. $\Rightarrow$ Sia $NP = coNP$. Allora $UNSAT \in coNP = NP$. Ovvero $UNSAT \in NP$.

($\Leftarrow$) Sia UNSAT $\in NP$. Basta mostrare $coNP \subseteq NP$ e $NP \subseteq coNP$. Dimostro la prima affermazione, l'altra è simile.

Sia $L \in coNP$, $\bar{L} \in NP$. Per Cook-Levin, $\bar{L} \leq_m^p$ SAT. Ma allora $\bar{L} \leq_m^p$ SAT sse $L \leq_m^p$ UNSAT. Ora UNSAT $\in NP$ e quindi $L \in NP$.

In pratica UNSAT è coNP-completo: L è coNP completo se (i) $L \in coNP$ e (ii) $\forall A \in coNP$, $A \leq_m^p L$.

Cor. UNSAT è coNP-completo.

Dim. Chiaramente UNSAT $\in coNP$. Devo far vedere che $\forall A \in coNP$, $A \leq_m^p$ UNSAT. Ma $A \leq_m^p$ UNSAT sse $\bar{A} \leq_m^p$ SAT e $\bar{A} \in NP$ che implica l'affermazione ∎

Interpretazione: $L \in coNP$ sse esiste poly-time $V(x,w)$ t.c. $\forall x$, $x \in \bar{L}$ sse $\exists w$ t.c. $V(x,w) = 1$. E notiamo $x \in \bar{L}$ è equivalente a $x \notin L$!

Se $L \in NP \cap coNP$: Posso certificare in maniera efficiente $x \in L$ e $x \notin L$.

Fatto: Molti problemi in $NP \cap coNP$ si sono poi mostrati essere in P. Es.:

- Stabilire se sistema di disuguaglianze ha soluzione (programmazione lineare).
- PRIMES.
- PERFECT MATCHING nei grafi.

Ma ci sono anche problemi in $NP \cap coNP$ che

Non sappiamo essere in P:

$$\text{FACTOR} = \{ \langle X, A, B \rangle : X \text{ ha un fattore primo tra } A \text{ e } B \}$$

COMPLESSITÀ DI SPAZIO

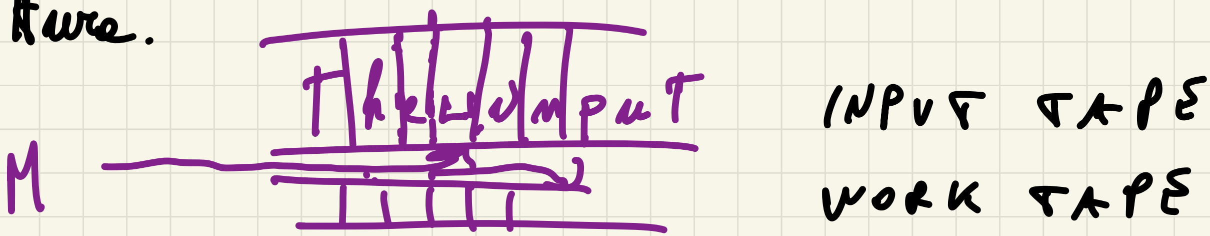
Vogliamo misurare l'efficienza delle TM in termini di spazio. Differenza fondamentale: lo spazio può essere riutilizzato (il tempo no).

DEF La complessità di spazio di una TM M decisore è una funzione $f: \mathbb{N} \to \mathbb{N}$ t.c.

$$f(n) = \max_{\substack{\text{input } x \\ |x| = n}} \{ \# \text{ di celle di nastro usate da } M(x) \}$$

È necessario cambiare il modello: Non voglio penalizzare la TM per dover memorizzare l'input.

Nuovo modello: TM con 1 input tape a sola lettura.



TM multinastro: Ci ricordiamo che una TM multinastro di tempo $f(n)$ si può simulare con singolo nastro in tempo $O(f^2(n))$.
Per lo spazio: $O(f(n))$ spazio. (Esercizio).

DEF. $SPACE(f(n)) = \{ A : \exists$ TM M che decide A con complessità di spazio $O(f(n)) \}$.

Pertanto:

- $L = SPACE(\log n)$; $PSPACE = \bigcup_k SPACE(n^k)$

  $\bigcup_k SPACE(2^{n^k}) = EXPSPACE$ = spazio esponenziale.

- NL, NPSPACE, NEXPSPACE ovvero TM non deterministica.

Esempio: $A = \{0^n 1^n : n \in \mathbb{N}\} \in L$. Ecco la TM:

- Controlla input forma corretta, ovvero non 1 seguito da 0.
- Sul nastro di lavoro: incrementa contatore fino a che legge 0. Decrementa fino a che legge 1. Alla fine accetta sse $ctr = 0$.

PALINDROMES È L. La TM:

- Su input x, determina $n = |x|$.
- Per $i = 1, \dots, n$
  - Rifiuta se $x_i \neq x_{n+1-i}$
- Accetta.

Chiaramente funziona. Come lo faccio in spazio logaritmico: Supponiamo $O(1)$ nastri.

1) Determinare n costa $O(\log n)$ spazio.
2) Memorizzo i sul nastro #2 e lo incremento; calcolo $n+1-i$: su nastro #3 copio n, lo incremento $+1$, copio i sul nastro #4 e decremento nastro #3 e #4 fino a che #4 arriva a 0.

3) Controllo $x_i \neq x_{n+1-i}$. Prelevo x_i decrementando nastro #2 e muovendo testina di input a DX.

Stessa cosa per x_{n+1-i}

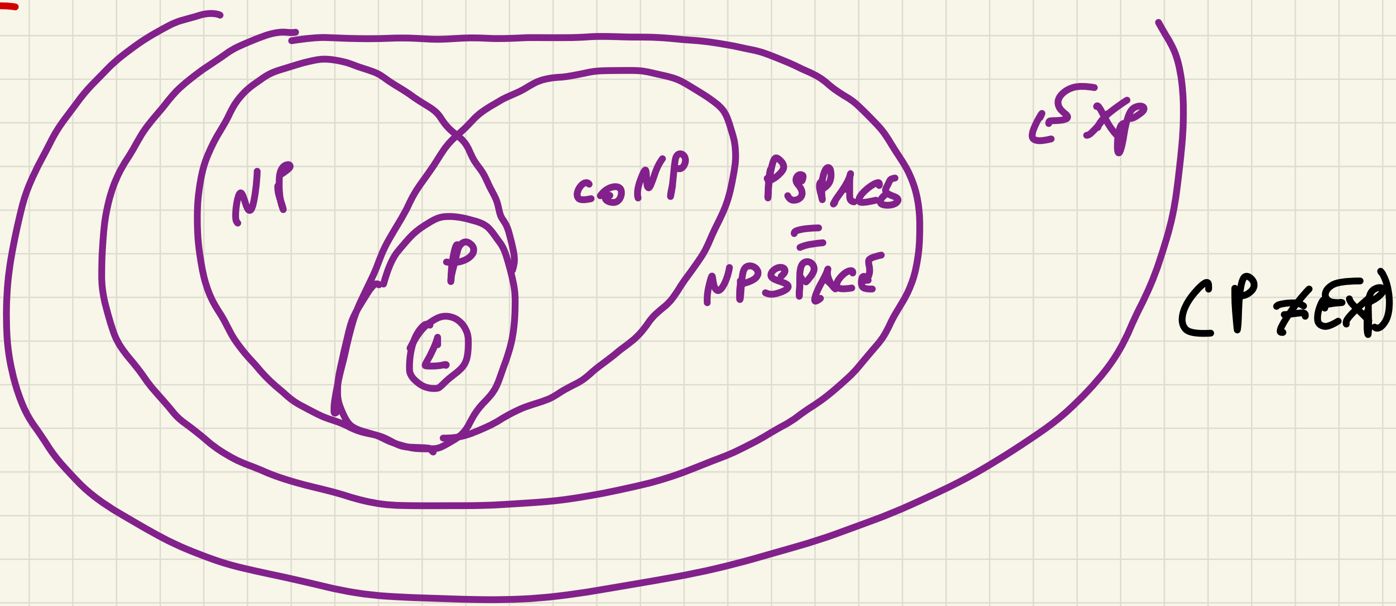
Riuso lo spazio, per ogni i. Totale $O(\log n)$.

In generale: In spazio logaritmico posso:

- $n = |x|$ è noto.
- mantenere $O(1)$ contatori i, j, k da 0 e $\log(n)$ in $O(\log n)$ spazio.
- Posso leggere caratteri input.
- Fare semplici operazioni aritmetiche. AOL es.: $a \cdot b = c$.

In seguito, dimostriamo un risultato importante per la complessità di spazio.

TEO (SAVITCH) NPSPACE = PSPACE.



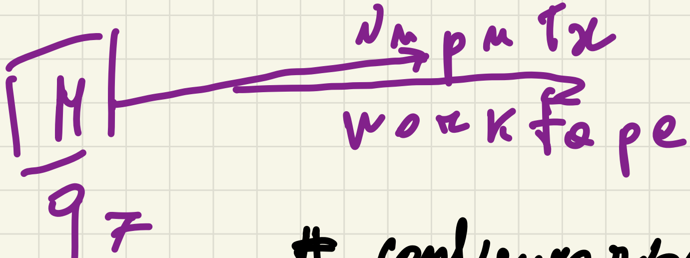
TEO $TIME(f(n)) \subseteq SPACE(f(n))$

Dim. Il tempo limita lo spazio: una TM di tempo $f(n)$ può usare al max $f(n)$ celle ∎

TEO Per $f(n) \geq \log n$, $SPACE(f(n)) \subseteq TIME(2^{O(f(n))})$.

Es.: $L \subseteq P$ e $PSPACE \subseteq EXP$.

Dim. Data una TM di spazio $O(f(n))$, allora $2^{O(f(n))}$ è il numero di possibili configurazioni della TM. Suppongo: 1 nastro input e 1 nastro lavoro



work q_7 tape; 3

$$\# \text{configurazioni} \leq |\Gamma|^{f(n)} \cdot |Q| \cdot n$$

Siccome $n \leq 2^{f(n)}$ allora $\leq 2^{O(f(n))}$.

$|\Gamma|, |Q| = O(1)$

Siccome M è decisore, non ripete le configurazioni. Il tempo è allora $\le 2^{O(f(n))}$ $\blacksquare$

TEO. $PATH \in SPACE(\log^2 n)$.

Dim: Dato $\langle G, s, t \rangle$ devo decidere se esiste $s \rightsquigarrow t$ in G in spazio $\log^2 n$.
Algoritmo ricorsivo: Considero la seguente procedura:
$PATH?(x, y, k)$: Ritorna SI/NO se $\exists$ $x \rightsquigarrow y$ con lunghezza $\le 2^k$.
Osservazione: Il cammino $s \rightsquigarrow t$ in G se esiste ha lunghezza $\le n$.
$\Rightarrow$ $PATH?(s, t, \lceil \log_2 n \rceil)$

PATH?(x, y, k):

- Se $k=0$, accetta sse $(x,y) \in E$ oppure $x=y$. (Altrimenti rifiuta.)
- Se $k>0$, per ogni nodo $w \in V$
  - Se PATH?$(x, w, k-1)$ AND PATH?$(w, y, k-1)$, accetta
  - Altrimenti rifiuta.

Profondità ricorsione: $O(\log n)$.
In memoria: n, s, t, x, y, k. Spazio $O(\log n)$
$\Rightarrow$ Complessità di spazio $O(\log^2 n)$. ∎

Osservazione: Non è di tempo polinomiale!