Vediamo alcune altre folle sulle complessità di tempo.

TEM Se S è NP-completo, allora $S \in P$ sse $P = NP$.

Dim. ($\Leftarrow$) $\forall L \in NP$, $L \leq_m^P S$. Se $S \in P$, anche $L \in P$ perché L è riducibile ad S in tempo polinomiale. Quindi $P = NP$.

($\Rightarrow$) Se $P = NP$ ed S è NP-completo, in particolare $S \in NP$. Ma allora, $S \in P$. ☐

Tuttavia, abbiamo visto che dimostrare SAT sembra difficile. Che possiamo dire sulla rete:

ca di una soluzione. È più difficile?

SAT È SELF-REDUCIBLE: Assumiamo $P = NP$.

Allora $\exists$ TM M con tempo polinomiale t.c.

$$M(\phi) = \begin{cases} \text{ACC} & \text{se } \phi \text{ soddisfacibile} \\ \text{REJ} & \text{se } \phi \text{ non soddisf.} \end{cases}$$

Come posso trovare esplicitamente? Pongo $x_1 = 0$ e considero la formula:

$$\phi_1(x) = \phi(0, x_2, \dots, x_m)$$

Ora se $M(\phi_1) = \text{ACC}$ posso concludere $x_1 = 0$. Altrimenti $x_1 = 1$ e procedo con

x_2. Dopo n invocazioni di M posso trovare x in tempo polinomiale.

Inoltre questo è vero per ogni $L \in NP$, perché sono tutti riducibili a SAT.

Altre osservazioni:

Thm. Se $P = NP$, anche $EXP = NEXP$.

Dim. Uso una tecnica nuova: PADDING.

$$\left(NP = \bigcup NTIME(n^k) \right)$$

$$\left(NEXP = \bigcup NTIME(2^{n^k}) \right)$$

Assumiamo $P = NP$. Devo mostrare $NEXP \subseteq EXP$.
perché è sempre vero che $EXP \subseteq NEXP$.
Ovvero, sia $L \in NEXP$ devo far vedere $L \in EXP$.
Sia N una NTM che decide L in tempo 2^{n^k}.
Considero un linguaggio derivato:

$$L' = \{ \langle x, 1^{2^{|x|^k}} \rangle : x \in L \}$$

dove $1 \notin \Sigma$

Si può vedere che $L' \in NP$. E quindi, siccome $P = NP$, anche $L' \in P$. Ma allora esiste TM M' con tempo polinomiale t.c. $L(M') = L'$.

Ma allora posso decidere L in Tempo esponenziale:

- Dato x, creo $x' = \langle x, 1^{2^{|x|^k}} \rangle$ in tempo $O(2^{n^k})$
- Poi lancio $M'(x')$ che richiede Tempo polinomiale e Acc sse $M'(x') \in Acc$
  - ↳ in $|x'|$ (esp. in $|x|$)

Rimane solo da mostrare: $L' \in NP$. Ecco la NTM N' che decide L':

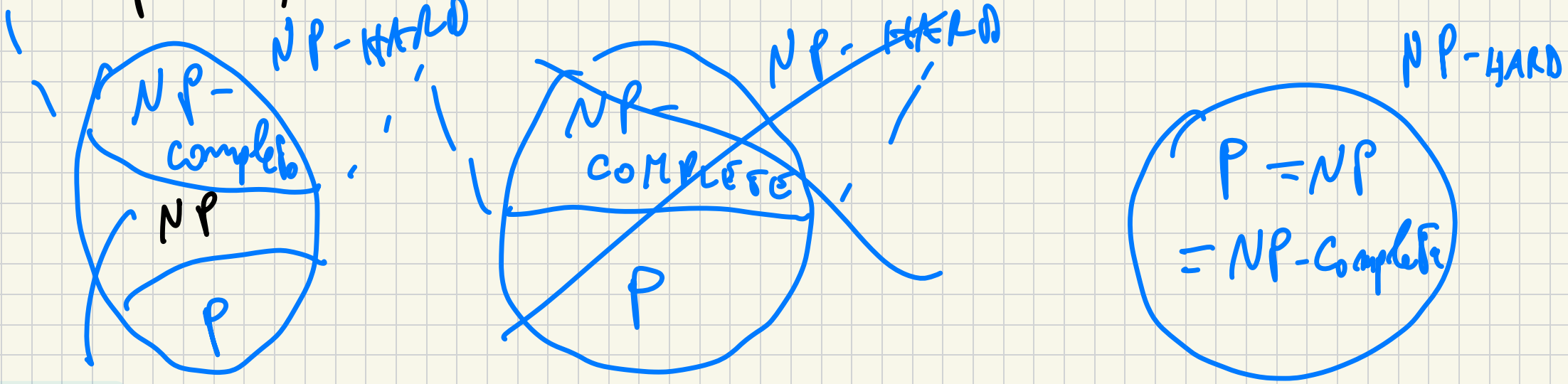
- Controllo che $x' = \langle x, 1^{2^{|x|^k}} \rangle$. Questo è fattibile deterministicamente in $O(|x'|)$

passo. Se non è così rifiuto.

– Altrimenti, scrivo 1 e lancio $N(x)$. Questo è fattibile non-det. in $O(z^{|x|^k})$ passi.

Ovviamente N' decide L'. Inoltre il numero di passi di N' è polinomiale in $|x'|$. ∎

Infine, menzioniamo il Teorema di dicotomia:



↳ **THM** Se $P \neq NP$, allora $\exists\, L \in NP$ t.c. L non è NP-completo.

↳ LA DNSR

coNP

Una nuova classe di complessità:

- NP serve a certificare $x \in L$
- coNP serve a certificare $x \notin L$

Q: UNSAT = $\overline{SAT}$. UNSAT $\in$ NP?
Qual è il certificato?

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

Osserviamo: $coNP \neq \overline{NP}$. Infatti UNSAT $\in coNP$.

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

DIM. Questo perché dato un decisore per SAT posso costruirne uno per UNSAT invertendo la risposta ■

THM $P \equiv coP$ ovvero P è chiusa per complemento.

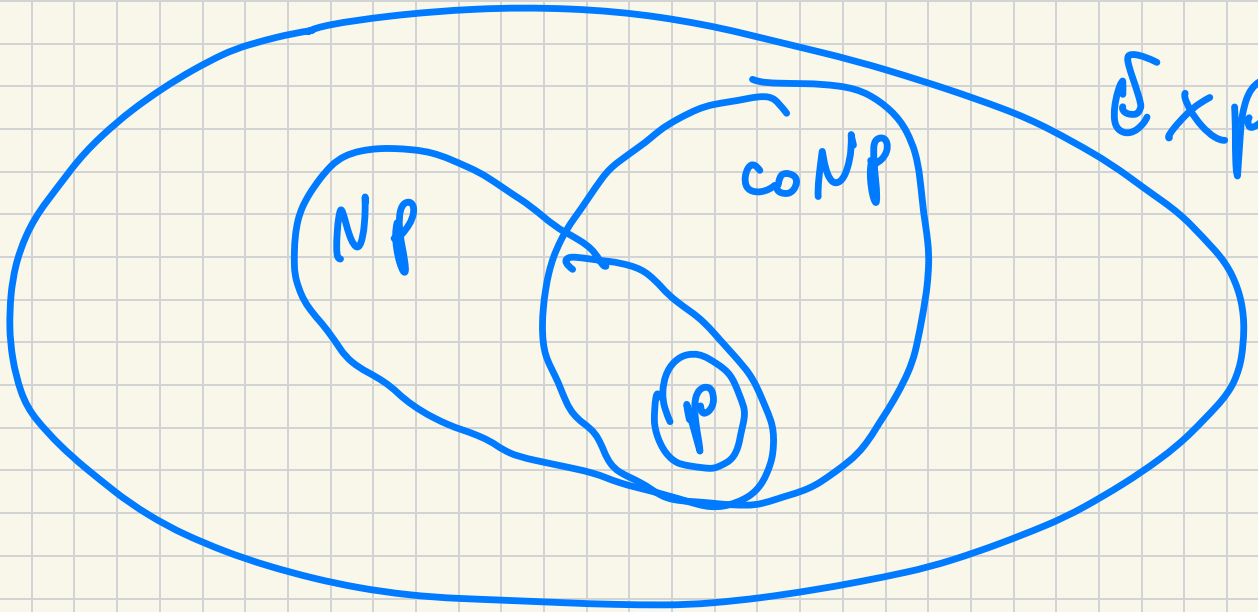
DIM. Immediato, basta invertire ACC e REJ ■

La stessa cosa non è vera per NP, coNP.

Ovvero non sappiamo $NP \overset{?}{=} coNP$. Infatti vedremo anche che $NP \neq coNP \Rightarrow P \neq NP$.
Vediamo perché.

TEOREMA $coNP \subseteq EXP$.

DIM. Sia $L \in coNP$, allora $\overline{L} \in NP \subseteq EXP$.
Quindi $\overline{L} \in EXP$ ed $L \in coEXP = EXP$ ∎



$(P \neq NP)$

Teo $P \subseteq coNP$.

Dim. Se $L \in P \Rightarrow \overline{L} \in P \subseteq NP$. Quindi $\overline{L} \in NP$ e quindi $L \in coNP$ ∎

Teo $P = NP \Rightarrow P = coNP (= NP)$.

Cor. $coNP \neq NP \Rightarrow P \neq NP$.

↳ Dim. Se $L \in coNP$, $\overline{L} \in NP = P$. Quindi $\overline{L} \in P$ ed $L \in P$ ∎

DEF L è coNP completo se:

- $L \in coNP$
- $\forall A \in coNP$, $A \leq_m^P L$. (coNP-HARD).

TEO UNSAT è coNP-completo.

Dim. Devo mostrare $\forall A \in coNP$, $A \leq_m^P UNSAT$.

Ma $A \leq_m^P UNSAT$ sse $\overline{A} \leq_m^P SAT$.

Siccome $\overline{A} \in NP$ allora $\overline{A} \leq_m^P SAT$ e quindi $A \leq_m^P UNSAT$ ☐

Un altro modo:

- $L \in coNP$ sse $\overline{L} \in NP$ ovvero $\exists$ polytime

$V(x,y)$ t.c. $\forall x$, $x \in \bar{L}$ sse $\exists y$ t.c. $V(x,y) = Acc$. Ma $x \in \bar{L}$ è lo stesso che $x \notin L$, quindi coNP è la stessa cosa di certificare $x \notin L$.

Un problema, almeno problemi in NP ∩ coNP sono poi risultati essere in P:

- PRIMES è coNP è banale. Nel '75 è stato dimostrato PRIMES ∈ NP. Successivamente, nel 2001 PRIMES ∈ P.

- MATCHING, stabilire se ∃ PM in un grafo.

- Ci sono problemi che sono in NP ∩ coNP ma non sappiamo essere in P.

SPAZIO

La complessità di spazio di una TM V è $s : \mathbb{N} \to \mathbb{N}$ t.c.

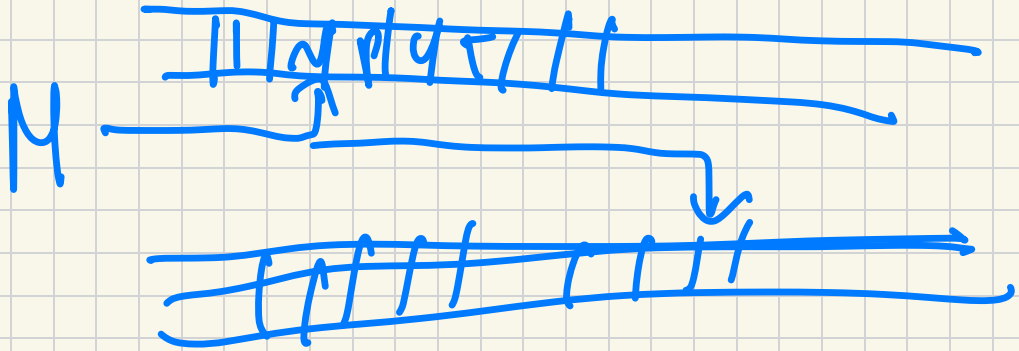
$$s(n) = \max_{|x|=n} \#\text{ celle di memoria usate da } M(x)$$

M deciderà quella funzione

Differenza fondamentale tra spazio e Tempo: lo spazio si può riusare. Per questo motivo con spazio piccolo ($\log n$) non vogliamo

considerare lo spazio necessario a memorizzare input.

Quando si parla di complessità di spazio non si considera l'input:



DEF $SPACE(s(n)) = \{L : \exists \text{ TM } M$ t.c. $L = L(M)$ e M ha comp. di spazio $O(s(n))\}$

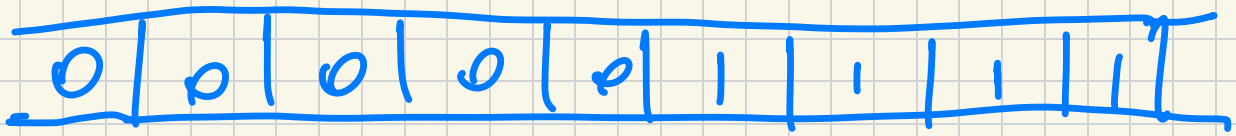
Alcune classi importanti:

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

- $EXPSPACE = \bigcup_k SPACE(2^{n^k})$

- $L = SPACE(\log n)$.

Esempio: $A = \{0^n 1^n : n \in \mathbb{N}\}$. Vedevamo $A \in L$. Avevamo mostrato una TM per questo linguaggio con spazio $O(n)$.



INPUT

L'idea è questa: Sul nastro di lavoro memorizzo solo un contatore, i.e. $\log n$ celle. Fino a che leggo 0, incremento il contatore, quando smetto di leggere 1 decremento. (Ogni volta che c'è o che segue 1 ripeto.)

FATTO. La TM multinastro è equivalente a quella semplice nastro con stesse complessità di spazio. (Esercizio.)

Altro esempio: PALINDROMES $\in L$. Ad alto livello:

- Su input x, determino $n = |x|$.
- For $i = 1, \dots, n$
  - REJ se $x_i \neq x_{n+1-i}$
- ACC.

Come lo realizzo su una TM con spazio $O(\log n)$. Posso usare $K = O(1)$ nastri.
Ad es. sul #1 posso realizzare il passo 1 con spazio $O(\log n)$.
Il passo 2: Memorizzo n su #2 e lo incremento ad ogni passo con spazio $O(\log n)$.

Poi devo calcolare $n+1-i$; lo faccio su #3: calcolo $n+1$; poi copio i su #4 dal #2 poi decremento. Il confronto su #3 e #4 fino a che #4 contiene 0. Resta da confrontare $x_i \neq x_{n+1-i}$. Copio il su #5 e lo decremento muovendolo e sfrutto sul nastro; quando mi fermo leggo x_i. Stessa cosa per x_{n+1-i}. Nella prossima iterazione ripeto lo spostamento.