Esercizi

*) Mostrare che se $P = NP$, allora $NP =$ NP-completo (eccetto $A = \emptyset, \Sigma^*$).

Sia $A \in NP$; devo mostrare che A è NP-completo.

Ovvero, per ogni $L \in NP$ abbiamo $L \leq_m^P A$.

Siccome $A \neq \emptyset, \Sigma^*$, esistono x_{YES}, x_{NO} t.c. $x_{YES} \in A$ mentre $x_{NO} \notin A$.

Costruisco riduzione $R : \Sigma^* \to \Sigma^*$:

- Dato x, usa la TM det. che decide $x \in L$.

Tale TM esiste in quanto $L \in NP = P$.

- Se la TM di cui sopra accetta, $R(x) = x_{YES}$

se rifiuta $R(x) = x_{no}$.
Ovviamente: $\forall x \in \Sigma^*$

$$x \in L \quad \text{sse} \quad R(x) \in A.$$

Quindi $L \leq_m^p A$.

*) Mostrare che $L = \{ 0^k 1^k : k \in \mathbb{N} \}$ appartiene a $DTIME(n \log n)$.

Nel corso abbiamo visto:

- $L \in DTIME(n^2)$
- L è decidibile in tempo $O(n)$ usando TM det. a 2 nastri.

L'idea è cancellare più 0/1 allo stesso tempo,

Made with Goodnotes

infatti dimezzarlo ogni volta (# $\log n$ volte).
Nel dettaglio:

- Controllo che l'input abbia la forma $0^* 1^*$. Se no, rifiuto. Costo: $\Theta(n)$.
- Ripete fino a che sopravvive almeno uno 0/1:
  - Scorrimento il nastro e controllo che #0 + #1 è PARI/DISPARI. Questo è possibile usando 2 stati q_0, q_1 e passando da uno all'altro scorrendo l'input. Se DISPARI rifiuta. Costo: $\Theta(n)$.
  - Scorrimento il nastro:
    - Rimpiazza con X la metà degli '0'

e le metà degli '1' (uno su e uno no).
Costo : $O(n)$.

- Controlla che ci siano solo 'x'.

0 1 1 1

X 1 X 1

0 0 0 1 1 1 1

X 0 X X 1 X

X X X X X X ACCETTA

0 0 0 1 1

X 0 X X 1

X 0 X 0 X 0 1 X 1

0 0 0 0 0 0 1 1 1 1

X 0 X 0 X 0 X 1 X 1

*) Dimostrare che $A_{DFA} \in L$

$$A_{DFA} = \{ \langle M, w \rangle : M \in DFA \text{ e } M(w) = ACC \}.$$

Perché? Posso rappresentare # stati del DFA in $O(\log n)$ spazio e anche la posizione della testina sull'input.

*) Mostrare che $DTIME(2^n) = DTIME(2^{n+1})$ ma $DTIME(2^n) \subsetneq DTIME(2^{2n})$.

$2^{n+1} = O(2^n)$ quindi il primo è banale.

Inoltre $2^n \log n = o(2^{2n})$ e quindi il secondo punto segue dal T.H.T.

*) $NTIME(n) \subsetneq PSPACE$.

Abbastanza facile:

$NTIME(n) \subseteq NSPACE(n)$ Tempo limita spazio

$NSPACE(n) \subseteq SPACE(n^2)$ SAVITCH

$SPACE(n^2) \subsetneq SPACE(n^3)$ S.H.T.

$SPACE(n^3) \subseteq PSPACE$.

*) Sia $pad : \Sigma^* \times \mathbb{N} \to \Sigma^* \#^*$ t.c.

$pad(s, l) = s\#^j$ dove $j = \max(0, l - |s|)$.

Mostrare che $\forall A$ e $\forall k \in \mathbb{N}$: $A \in P$ sse $pad(A, n^k) \in P$ dove

$$pad(A, n^k) = \{ pad(x, n^k) : x \in A \}.$$

Se $A \in P$, allora $pad(A, n^k) \in P$ perché posso decidere $w \in pad(A, n^k)$ scrivendo $w = s\#^j$ dove $s \in A$ e $|w| = n^k$.

D'altra parte, se $pad(A, n^k) \in P$, allora posso decidere $x \in A$ facendo $pad(x, n^k)$

e usando il decisore per $\mathrm{pad}(A, n^k)$.

*) $P \neq SPACE(n)$.

Usiamo l'esercizio precedente. Sia $P = SPACE(n)$ per il S.H.T. esiste A t.c. $A \in SPACE(n^2)$ ma $A \notin SPACE(n)$.
Adesso considero $\mathrm{pad}(A, n^2)$. Possiamo vedere che $\mathrm{pad}(A, n^2) \in SPACE(n)$.

$$w \in \mathrm{pad}(A, n^2) : \quad w = s\underbrace{\#^j}_{n^2}$$

$$|s| = O(n)$$

$x \in pad(A, n^2)$

$$x = s\#^{|s|^2}$$

Siccome $SPACE(n) = P$, $pad(A, n^2) \in P$

$\Rightarrow A \in P = PSPACE(n)$.

*) Mostrare che:

- Se $L_1, L_2 \in coNP$, allora $L_1 \cap L_2 \in coNP$.
- Se $L \in NP$, $L_1 \subsetneq L$ e $L_1 \in coNP$ allora $L \setminus L_1 \in NP$.

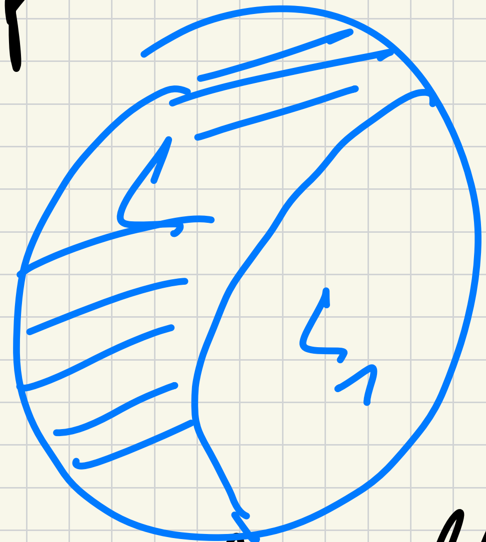
Made with Goodnotes

Primo punto: $L_1, L_2 \in coNP$ equivale a $\overline{L_1}, \overline{L_2} \in NP$. Segue $\overline{L_1} \cup \overline{L_2} \in NP$.

$$\overline{\left(\overline{L_1} \cup \overline{L_2}\right)} \in coNP$$

$$\|$$

$$L_1 \cap L_2$$



Secondo punto: Se $L_1 \in coNP$ allora $\overline{L_1} \in NP$. Ma NP è chiuso per intersezione: $L \cap \overline{L_1} = L \setminus L_1 \in NP$.

*) Mostrare che ogni linguaggio PSPACE-hard è anche NP-hard.

A PSPACE-hard: $\forall L \in PSPACE$

$$L \leq_m^p A$$

A NP-hard: $\forall L \in NP$

$$L \leq_m^p A$$

Ora: $NP \subseteq NPSPACE$ (tempo limita spazio). $NPSPACE \subseteq PSPACE$ per SAVITCH. Quindi se A è PSPACE hard è anche NP-hard perché ogni $L \in PSPACE$ si riduce ad A ma tra questi linguaggi ci sono tutti i linguaggi in NP.