Esercitazione Complessità

20/12/2024

1) Mostrare che se $P = NP$, allora ogni linguaggio NP è anche NP-Completo, eccetto $\emptyset, \Sigma^*$.

Sia $A \in NP$, devo mostrare che A è NP-difficile

$\forall L \in NP, \quad L \leq^P_m A$. { VOGLIAMO MOSTRARE

$\exists$ due stringhe x_{yes}, x_{no}

$x_{yes} \in A$

$x_{no} \notin A$

$\Rightarrow$ devo fare la riduzione, costruisco $R: \Sigma^* \to \Sigma^*$

- dato in input x, uso una TM deterministica che decide $x \in L$, tale TM esiste perché $L \in NP = P$ (per ipotesi)
- Se la TM accetta, $R(x) = x_{yes}$, altrimenti $R(x) = x_{no}$

$\Rightarrow x \in L \iff R(x) \in A$, ed R funziona in tempo polinomiale $\Rightarrow L \leq^P_m A$.

2) Mostrare che $L = \{0^k 1^k ; k \in \mathbb{N}\}$ appartiene a $DTime(n \cdot \log(n))$.

Abbiamo visto che

- $L \in DTime(n^2)$
- L è decidibile in tempo $O(n)$ usando

12/2024

TM deterministiche a 2 nastri. Si vuole mostrare che è DTime($n \cdot \log(n)$) con una TM a singolo nastro.

L'idea è cancellare più 0 e 1 allo stesso tempo, dimezzandoli ogni volta. Nel dettaglio:

- Controllo che l'input abbia la forma $0^x 1^y \Rightarrow$ gli 1 vengono dopo gli 0. Ha costo $O(n)$. Poi si controlla se $x = y$
- ripete la seguente procedura finché la stringa rimanente è 0 1

CICLO

- Scorriamo il nastro e ~~conto il numero~~ controllo che la somma degli 0 più quello degli 1 (#0 + #1) sia pari o dispari, ciò si può fare usando 2 stati q_1, q_2, e passando da uno all'altro durante il conto. ha costo $O(n)$
- Scorriamo nuovamente il nastro, cancellando metà degli ~~zeri~~ e metà degli 1. Cancellando uno sì ed uno no.

  X 0 X 0 X 1 X 1

- cancelliamo alternando gli zeri, e poi gli 1

  X 0 X X 1 X

  ↳ si ricomincia

L'algoritmo è in $O(n \log(n))$.

3) Dimostrare che $A_{DFA} \in L$ dove

$A_{DFA} = \{ \langle M, w \rangle \mid M \in DFA \wedge M(w) \text{ accetta} \}$

La codifica di un DFA ed un input per il DFA. Posso ~~usare~~ rappresentare il numero di stati $|Q|$ del DFA in $O(\log(n))$ spazio, e anche la posizione della testina sull'input del DFA.

basta salvare 1 stato alla volta perché il DFA non ha memoria.

4) Mostrare che $DTime(2^n) = DTime(2^{n+1})$ ma $DTime(2^n) \subset DTime(2^{2n})$.

Ovviamente $2^{n+1} = O(2^n)$. Inoltre, si può usare il teo. di gerarchia dei tempi!

$$2^n \log(n) = o(2^{2n})$$

dato che $\lim_{n \to \infty} \frac{2^n \log(n)}{2^{2n}} = 0$

5) Mostrare che $NTime(n) \subseteq PSpace$

Sappiamo che il tempo limita lo spazio

$NTime(n) \subseteq NPSpace(n)$

$NPSpace(n) \subseteq Space(n^2)$ Savitch

$Space(n^2) \subset Space(n^3)$ gerarchia

$Space(n^3) \subset PSPACE$

6) Sia

$pad(S, l) =$

mostrare el

$A \in P \Leftrightarrow$ pa

$pad(A, n^k) =$

Σ padding

Se $A \in P$

decidere w

$w = S \# j$

d'altro pa

decidere s

usando

7) $P \neq SP$

Assumiamo

gerarchia

~~sia~~ in

possiamo

ho $w \in p$

$\Rightarrow |S| = O$

pad (A, w

ma que

6) Sia $pad : \Sigma^* \times \mathbb{N} \to \Sigma^* \#^*$ tale che $pad(s, \ell) = s\#^j$ dove $j = \max(0, \ell - |s|)$ mostrare che $\forall A \subseteq \Sigma^* \forall k$ si ha che $A \in P \Leftrightarrow pad(A, n^k) \in P$ dove

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

↑ padding del linguaggio.

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

d'altra parte se $pad(A, n^k) \in P$ allora posso decidere se $x \in A$ facendo $pad(x, n^k)$ ed usando poi il decisore per $pad(A, n^k)$.

7) $P \neq Space(n)$

Assumiamo $P = Space(n)$, per il Teo. di gerarchia, $\exists A$ t.c. $A \in Space(n^2)$ ma non in $Space(n)$. Adesso prendo $pad(A, n^2)$, possiamo vedere che $pad(A, n^2) \in Space(n)$, se ho $w \in pad(A, n^2) \Rightarrow w = s\#^j$ con $|w| = n^2$ $\Rightarrow |s| = O(n)$. Siccome $Space(n) = P \Rightarrow$ $pad(A, n^2) \in P \Rightarrow A \in P = Space(n) \Rightarrow P = Space(n)$ ma questa è una contraddizione. ↯

8) mostrare che:

1) Se $L_1, L_2 \in coNP$, allora $L_1 \cap L_2 \in coNP$ e

2) Se $L \in NP \wedge L_1 \subset L \wedge L_1 \in coNP$, allora $L \setminus L_1 \in NP$.

1) $L_1, L_2 \in coNP \Rightarrow L_1, L_2 \in NP \Rightarrow L_1 \cup L_2 \in NP$

$\Rightarrow \overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2 \in coNP$

2) Se $L_1 \in coNP$ allora $\overline{L_1} \in NP$, ma NP è chiuso per intersezione $\Rightarrow L \cap \overline{L_1} = L \setminus L_1 \in NP$.

9) Mostrare che ogni linguaggio PSPACE difficile è anche NP difficile.

$NP \subseteq NPSPACE$ tempo limitato ~~fg~~ spazio

$NP \subseteq PSPACE$ sarebbe

Se A è Pspace difficile è anche NP difficile perché $\forall L \in PSpace$ si riduce col A ma poco quei linguaggi ~~e i~~ ci sono tutti i linguaggi in NP.