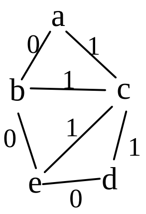
Fondamenti dell'Informatica
A.A. 2005/2006
Corso A

Prova scritta: 13/7/2006 ore 8.30 – 11.00

1. Definire una macchina di Turing non deterministica. Definire il concetto di configurazione, di transizione e di accettazione/rifiuto di una stringa *(3 punti)*. Progettare una macchina di Turing non deterministica[1] che accetti il linguaggio

   $L=\{w\in(0+1)^+ \mid w$ rappresenta una sequenza lecita di attraversamento dei nodi del grafo G$\}$

   dove G è il seguente grafo con etichette 0 e 1 lungo gli archi

   Un esempio di sequenza lecita di attraversamento è 0011 perché, se si parte dal nodo b, essa corrisponde al percorso b → e → d → c → b. La sequenza 010 non è lecita, e quindi non appartiene a L, perché da qualunque nodo si parte, non è possibile seguire una sequenza di 3 archi etichettati con 1. (Suggerimento: oltre allo stato iniziale, associare a ogni nodo di G uno stato della MTN che si suppone di aver attraversato. Tutti gli stati, tranne quello iniziale, sono anche finali). (*5 punti*)

2. Definire un programma RAM, e spiegare brevemente le differenze fra i diversi modelli di costo proposti per i programmi RAM. *(3 punti)*. Scrivere un programma RAM per il calcolo di $x!$, $x\geq 0$ e valutarne la complessità rispetto ai diversi modelli di costo proposti. (*7 punti*)

3. Mostrare come si possono esprimere le funzioni ricorsive mediante i costrutti del linguaggio funzionale SLF (*4 punti*). Mostrare che la funzione $f(x)=x^x$ è ricorsiva primitiva *(3 punti)* e quindi scrivere un programma SLF per il calcolo di 5^5. (*2 punti*)

4. Illustrare la struttura della gerarchia aritmetica (o gerarchia di Kleene) (*5 punti*).

5. Definire il concetto di chiusura di una classe di complessità rispetto a una relazione di riduzione tra problemi (*2 punti*). Dimostrare che la classe LOGSPACE è chiusa rispetto alla riduzione $\leq_m^{logspace}$ o che la classe P è chiusa rispetto alla riduzione $\leq_m^p$ (*4 punti*).[2]

Soluzione Es. 5

Ricordiamo che:

Definizione 8.19 *Una classe di complessità $\mathcal{C}$ è detta chiusa rispetto ad una riducibilità $\leq_r$ se e solo se per ogni coppia di problemi $\mathcal{P}_1$, $\mathcal{P}_2$ tali che $\mathcal{P}_1 \leq_r \mathcal{P}_2$, $\mathcal{P}_2 \in \mathcal{C}$ implica $\mathcal{P}_1 \in \mathcal{C}$.*

e che se esiste una riduzione R da P_1 a P_2 allora per una istanza $x\in I_{P1}$ trasformata in una istanza $y\in I_{P2}$ si ha $t_{A1}(|x|) = t_R(|x|) + t_{A2}(|y|)$, dove A_1 (A_2) è l'algoritmo che risolve P_1 (P_2).
Ora, se $P_1 \leq_m^p P_2$ (cioè è P_1 Karp-riducibile polinomialmente a P_2) allora $t_R(|x|)$ è un polinomio in $|x|$. Inoltre se $P_2\in\mathcal{P}$ (cioè è risolvibile in tempo polinomiale con MT deterministica) allora possiamo prendere come algoritmo risolutivo A_2 proprio uno tale che $t_{A2}(|y|)$ è un polinomio. Ma a questo punto, abbiamo trovato un algoritmo A_1 per P_1 (quello che prima trasforma $x\in I_{P1}$ in $y\in I_{P2}$ e poi applica A_2 a y) di complessità polinomiale (perché somma di due polinomi). Allora anche $P_1\in\mathcal{P}$ e quindi è dimostrata la chiusura di $\mathcal{P}$ rispetto a $\leq_m^p$.

---

[1] Assicurarsi che il grado di non determinismo sia superiore a 1.
[2] La totalizzazione di un punteggio superiore a 30 punti equivale al *30 con lode*.