Informatica Teorica

Seconda prova in itinere e appello del corso del Vecchio ordinamento

29 Giugno 2004

Coloro che sostengono la seconda prova in itinere svolgano gli esercizi 1,2,3 in un'ora e un quarto. Chi sostiene l'appello completo (o la parte di Informatica teorica del corso integrato Algebra + Inf. Teorica) svolga gli esercizi 1, 3 e 4 nello stesso tempo.

NB I punteggi assegnati agli esercizi 1,2,3 hanno valore solo con riferimento alla II prova in itinere.

Si suggerisce di affrontare per primo l'esercizio 3.

Esercizio 1 (Punti 6)

Si dica, giustificando brevemente la risposta, se le seguenti affermazioni sono vere o false:

a) Il problema di stabilire se un generico programma P, codificato in C, eseguito su un generico dato di ingresso x, non terminerà mai la sua esecuzione è semidecidibile.

b) Il problema di stabilire se un generico programma P, codificato in C, eseguito su un generico dato di ingresso x, "andrà in loop", nel senso stretto del termine, ossia ripeterà indefinitamente una stessa sequenza di "mosse" ritornando ogni volta nello stesso stato di memoria, è semidecidibile.

Esercizio 2 (Punti 4)

Il signor Furbetti sta realizzando un archivio mediante una tabella hash. I record dell'archivio hanno una chiave costituita da sequenze di al più 30 caratteri e la tabella ha una dimensione di 5 MB.

Furbetti è particolarmente soddisfatto perché ha trovato una funzione di hash h "perfetta" (ossia è riuscito a dimostrare che la sua funzione h è tale che $x \neq y$ implica $h(x) \neq h(y)$). Accingendosi però a implementare un algoritmo per il calcolo di h non riesce a trovarne uno. Egli viene allora assalito dal dubbio che la sua h non sia calcolabile.

Potete aiutare Furbetti a risolvere il suo dubbio?

Esercizio 3 (Punti 8)

a) Si descriva sinteticamente –senza necessariamente codificarlo in modo completo- un algoritmo per la macchina RAM per riconoscere il linguaggio $L = \{a^n b^n c^n \mid n >= 1\}$. Se ne valutino complessità spaziale e temporale –a meno della relazione Θ-mediante il criterio di costo logaritmico.

b) E' possibile ottenere le stesse complessità (sia spaziale che temporale) mediante una macchina di Turing? Giustificare brevemente la risposta.

Esercizio 4

Si scriva una grammatica che generi il linguaggio $L = \{a^n w \mid w \in \{b,c\}^* \wedge \#(b,w) = \#(c,w) = n, n >= 1\}$, dove la funzione #(i,x) denota il numero di occorrenze del carattere i nella stringa x.

La grammatica così ottenuta è a potenza minima, ossia non esiste una grammatica ad essa equivalente appartenente ad una classe di minor potenza generativa? Giustificare brevemente la risposta.

Soluzioni

Esercizio 1

a) Falsa. Infatti il problema complementare è notoriamente non decidibile ma semidecidibile. Quindi non può essere semidecidibile anche il problema della ***non***-terminazione.

b) Vera. E' concettualmente possibile registrare ogni stato dell'esecuzione del programma e verificare se lo stato attuale è identico a uno già attraversato. Se ciò accade, da quel momento in poi l'esecuzione si ripeterà identicamente all'infinito.

Esercizio 2

h è certamente calcolabile perché ha un dominio finito. Ciò però aiuta solo in parte Furbetti perché non fornisce indicazioni su come trovare l'algoritmo di calcolo di h.

Esercizio 3

a) La RAM aggiorna un contatore per ogni a letto (costo $\Theta(i)$, i = 1, …, n). Salva il valore del contatore al termine della lettura degli a in due celle diverse. Decrementa la prima copia per ogni b letto e la seconda copia per ogni c letto successivamente. Al termine entrambi i contatori devono contenere esattamente 0. Ne deriva una complessità spaziale, a criterio logaritmico, $\Theta(\log(n))$ e una temporale $\Theta(n.\log(n))$

b) Sì. Una MdT può simulare il comportamento della RAM esattamente allo stesso modo, usando un nastro per ogni contatore e codificandone il contenuto in numerazione binaria. Per ogni carattere letto l'incremento o il decremento del nastro contatore richiede un numero di mosse al più proporzionale alla lunghezza del nastro, ossia $\Theta(\log(n))$. Perciò complessità spaziale e temporale coincidono con quelle della RAM.
Si noti che un MdT potrebbe riconoscere lo stesso linguaggio con complessità $\Theta(n)$ sia spaziale che temporale.

Esercizio 4

S → aSH | aH

H → BC

BC → CB

CB → BC

B → b

C → c

La grammatica di cui sopra non è non-contestuale (appartiene però alla classe delle grammatiche contestuali, una classe non considerata in questo corso di minor potenza delle grammatiche generali e di maggior potenza di quelle non-contestuali; questa classe genera linguaggi decidibili al contrario delle grammatiche generali). Non è possibile generare lo stesso linguaggio mediante una grammatica non-contestuale perché per riconoscere L occorre tenere aggiornati i conteggi sia delle be che delle c per confrontarli con il numero di a letto inizialmente.