Informatica Teorica - Sezione Pradella

Appello del 5 Febbraio 2005

Attenzione!!!

I voti proposti verranno pubblicati sul sito seguente:

http://www.elet.polimi.it/upload/pradella/IT.html

Gli studenti avranno tempo due giorni (48 ore, per la precisione) dalla data della pubblicazione per rifiutare il voto proposto, se sufficiente. L'eventuale rifiuto deve essere comunicato via email, facendo uso dell'indirizzo ufficiale del Poliself, non di indirizzo privato! Trascorsi i due giorni, i voti verranno registrati e trasmessi alla segreteria.

Per evitare il ripetersi di recenti disguidi, probabilmente dovuti a malfunzionamenti del servizio di posta elettronica, il docente spedirà a ogni studente "rinunciatario" un esplicito messaggio di "ricevuta". In caso di mancata ricezione di tale ricevuta si consiglia di contattare il docente telefonicamente o di avvisare la segreteria didattica del DEI.

Chi risulterà insufficiente dopo il recupero di febbraio o rinuncerà al voto proposto dovrà ri-iscriversi al corso per l'anno accademico 2004/05.

Esercizio 1 (punti 8)

Si specifichi, mediante una formula del prim'ordine il seguente comportamento di un ipotetico sistema:

Tutte le volte che, in un certo istante, si verifica l'evento A, dopo esattamente k unità di tempo si verifica l'evento B, a meno che, contemporaneamente ad A, non si verifichi anche l'evento C. In tal caso, B non si verifica dopo k unità di tempo, bensì dopo k+1.

Esercizio 2 (punti 8)

Si considerino i seguenti linguaggi:

$L1 = \{a^n b^{2m} | n, m >= 1\}$, $L2 = \{a^n b^{3n} | n >= 0\}$

Si costruisca una grammatica G che generi il linguaggio $L = L1 \cap L2$. E' preferibile una grammatica a potenza minima, ossia regolare se ne esiste una, non contestuale se ne esiste una ma non ne esiste una regolare, ecc. Nel caso, si spieghi brevemente perché non esstono grammatiche appartenenti a classi inferiori a quella proposta.

Esercizio 3 (punti 6)

Si dica, giustificando brevemente la risposta, se è decidibile il problema di stabilire se il linguaggio L di cui all'esercizio 2 è vuoto o no.

Esercizio 4 (punti 8)

Si dica, giustificando brevemente la risposta, qual è l'ordine di grandezza minimo per il riconoscimento del linguaggio L di cui all'esercizio 2 nei due casi in cui si usi

a) Una macchina di Turing.

b) Una RAM, assumendo il criterio di costo logaritmico.

Soluzioni

Esercizio 1

Codificando l'occorrenza di un generico evento E all'istante t mediante il predicato E(t), la formula seguente specifica il comportamento desiderato:

$$\forall t((A(t) \wedge \neg C(t)) \to B(t+k)) \wedge \\ ((A(t) \wedge C(t)) \to (\neg B(t+k) \wedge B(t+k+1)))$$

Si noti che la formula di cui sopra porta ad una contraddizione se non solo all'istante *t*, ma anche all'istante *t-1* accadono sia A che C (cosa che la specifica a parole in principio non vieta). In questo caso, infatti, il secondo congiunto applicato a *t-1* vincola B ad accadere a *t+k*, mentre lo stesso congiunto riferito all'istante *t* proibisce che B accada a *t+k*. Una formula che evita questo fenomeno, reinterpretando la specifica a parola è la seguente:

$$\forall t((A(t) \wedge \neg C(t)) \to B(t+k)) \wedge \\ ((A(t) \wedge C(t)) \to B(t+k+1))$$

Esercizio 2

Il linguaggio L = L1 ∩ L2 è l'insieme $\{a^{2n}b^{6n} | n >= 1\}$. Infatti, L1 è il linguaggio delle stringhe con un numero *pari* di 'b' che seguono un numero qualunque di 'a'. Tra le stringhe di L2, del tipo '$a^n b^{3n}$', quelle che appartengono anche a L1 sono esattamente quelle con un numero pari di 'b', ossia quelle nella forma '$a^{2n}b^{3\cdot(2n)} = a^{2n}b^{6n}$'.

L è generato dalla seguente grammatica non contestuale (e da nessuna grammatica regolare essendo necessaria una pila per il suo riconoscimento):

S → AB | ASB

A → aa

B → bbbbbb

Esercizio 3

L non è vuoto: contiene aabbbbbb, ecc. Quindi il problema L = ∅ è *deciso* e perciò *decidibile*.

Esercizio 4

Una MT che simuli un automa a pila deterministico può facilmente riconoscere L in tempo lineare. Una RAM deve necessariamente aggiornare un contatore (il cui contenuto è un numero proporzionale ad n) per ogni carattere letto. Perciò, a criterio di costo logaritmico, la sua complessità è necessariamente Θ(n.log(n)).