Informatica Teorica (Sez. Pradella)

Appello del 13 Settembre 2005

NOTE:

- **DURATA 1H 15'**
- **SCRIVERE CHIARAMENTE "*LAUREANDO*" SUL COMPITO, SE SI È TALI: I RISULTATI DEI LAUREANDI SARANNO PUBBLICATI SUL SITO DEL CORSO ENTRO DOMANI MATTINA.**

Esercizio 1 (punti 10/30-esimi)

Si scrivano un automa e una grammatica che, rispettivamente, riconosca e generi il linguaggio $L = \{a^n a^m a^m a^n \mid n, m >= 1, n \text{ pari}, m \text{ dispari}\}$.

NB: il punteggio massimo verrà assegnato solo se l'automa ideato sarà a potenza riconoscitiva minima tra quelli che riconoscono il linguaggio desiderato e la grammatica avrà il minimo numero di produzioni.

Esercizio 2 (punti 10/30-esimi)

Si scriva una formula logica che descrive la seguente regola per controllare atterraggi e decolli in un aeroporto:

- Se c'è un aereo in fase di decollo nessun altro aereo può decollare entro 3 minuti dall'inizio del decollo e nessun altro aereo può atterrare entro 5 minuti dall'inizio del decollo dell'aereo precedente;
- Se c'è un aereo in fase di atterraggio nessun altro aereo può atterrare entro 3 minuti dall'inizio dell'atterraggio e nessun altro aereo può decollare entro 5 minuti dall'inizio dell'atterraggio dell'aereo precedente.

Esercizio 3 (punti 10/30-esimi)

Dire, giustificando brevemente la risposta, se la seguente funzione f: N → N è:

a) Computabile

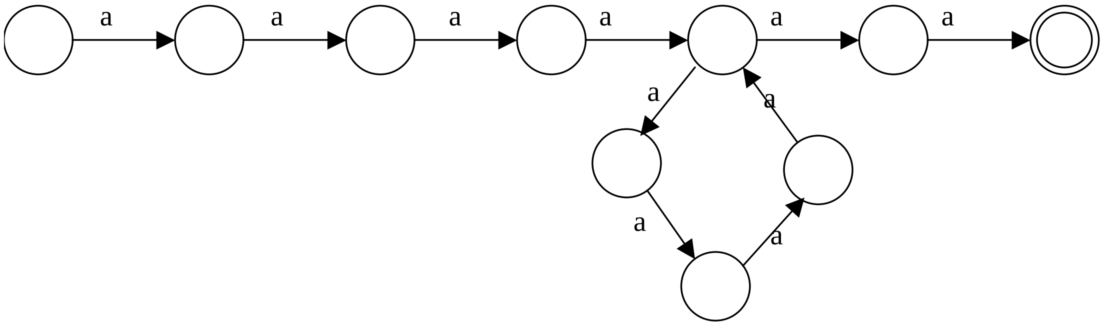
b) Totale.

f(n) = n-esimo numero primo. Per convenzione, si assuma f(0) = 1, e, di conseguenza, f(1) = 2, f(2) = 3, f(3) = 5, …

Soluzioni

Esercizio 1

La definizione di L equivale a L = {a^{2n}| n dispari e > 2}, ossia L = {a^6 , a^{10}, a^{14}, a^{18}, …}, ossia, detto in altra maniera L = { $a^{2(2k+1)}$ | k ≥ 1 } = { a^{4k+2} | k ≥ 1 }. Quindi L può essere riconosciuto mediante il seguente semplice automa a stati finiti:



Da esso si può ricavare immediatamente una grammatica regolare equivalente, che può essere "compattata" nella seguente grammatica (non contestuale!) per minimizzare il numero di produzioni:

S → a^4A

A → a^4A| a^2

Esercizio 2

Si definscano i predicati StartTakeOff (x, t) e StartLand(x, t) per indicare, rispettivamente, che l'aereo x ha iniziato il decollo o l'atterraggio all'istante t.

Il requisito richiesto è allora formalizzato dalla formula seguente:

$$\begin{aligned}
&\forall t, x(\ StartTakeOff\,(x,t) \rightarrow \\
&\qquad \neg\exists y, t_1 (y \neq x \wedge (t \leq t_1 \leq t+3) \wedge StartTakeOff\,(y,t_1)) \wedge \\
&\qquad \neg\exists y, t_1 (y \neq x \wedge (t \leq t_1 \leq t+5) \wedge StartLand(y,t_1))\ ) \\
&\wedge \\
&\forall t, x(\ StartLand(x,t) \rightarrow \\
&\qquad \neg\exists y, t_1 (y \neq x \wedge (t \leq t_1 \leq t+3) \wedge StartLand(y,t_1)) \wedge \\
&\qquad \neg\exists y, t_1 (y \neq x \wedge (t \leq t_1 \leq t+5) \wedge StartTakeOff\,(y,t_1))\ )
\end{aligned}$$

Altri predicati e formule, non strettamente necessari ai fini della formalizzazione del requisito ma utili a chiarire il significato dei termini potrebbero definire la fase di atterraggio/decollo dell'aereo x come quella compresa tra i relativi inizio (Start) e fine (End).

Esercizio 3

La funzione f è computabile: essendo infatti decidibile se un numero sia primo o no, dopo aver costruito l'n-esimo numero primo si può ottenere l'n+1-esimo enumerando tutti i numeri ad esso successivi fino ad ottenere il prossimo numero primo.

f è anche totale perché i numeri primi sono infiniti. Quindi, per ogni n, esiste l'n+1-esimo numero primo che verrà individuato dall'algoritmo suddetto.