Algoritmi e Principi dell'Informatica

Prima prova in itinere - 15 Novembre 2010

Tempo a disposizione: 1h30'

Esercizio 1 (8 punti)

Si scriva un automa (a stati finiti, a pila, macchina di Turing oppure rete di Petri) che riconosce il linguaggio L fatto di tutte e sole le stringhe della forma $a^{n1}b^{n2}c^{n3}$ in cui $0 < n1 < n2 < n3$.

NB: verranno dati 2 ulteriori punti di bonus in caso di modello a *potenza minima*.

Esercizio 2 (11 punti)

Formalizzare mediante formule di logica del prim'ordine le seguenti affermazioni:

2.1. L'insieme dei numeri naturali primi è illimitato.
2.2. C'è una infinità di macchine di Turing che calcolano funzioni totali.
2.3. Esistono macchine di Turing che calcolano funzioni con dominio finito.

Si indichi come al solito con la funzione a due argomenti $f_y(x)$ il valore calcolato dalla y-esima TM con ingresso x.

Esercizio 3 (11 punti)

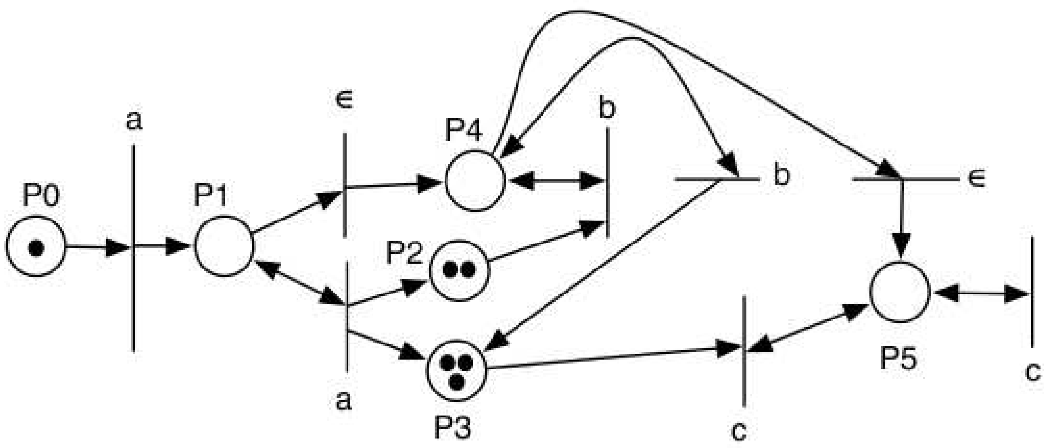
Il professor Rice dà ai suoi studenti il seguente esercizio:

> "Scrivere un automa (a stati finiti, a pila, o macchina di Turing) che riconosca il linguaggio L(G) generato dalla seguente grammatica G:
> $S \rightarrow ABCS \mid cABCABC$
> $ABC \rightarrow a \mid b \mid c$
> L'automa scritto deve essere a potenza riconoscitiva minima tra quelli che riconoscono L(G), o la soluzione proposta sarà considerata sbagliata, e riceverà 0 punti."

3.1. E' decidibile il problema di stabilire se uno studente, in risposta all'esercizio, scrive una macchina di Turing, un automa a pila, o un automa a stati finiti?
3.2. E' possibile scrivere un programma che faccia in automatico la correzione delle soluzioni proposte dagli studenti, e cioè che, data una qualunque soluzione proposta, dica se essa è corretta oppure no?
Suggerimento: si individui il tipo di automa a potenza riconoscitiva minima che risolve l'esercizio del professor Rice.

Soluzioni

Esercizio 1
L'automa a potenza minima tra quelli in grado di riconoscere il linguaggio desiderato è una rete di Petri senza archi inibitori (il linguaggio non è invece riconosciuto da nessun automa a pila, neanche nondeterministico).
Una rete di Petri che riconosce il linguaggio L desiderato è la seguente:



L'insieme F delle marcature finali contiene la sola M_F, che è tale che $M_F(P5) = 1$, e $M_F(P_i) = 0$ per tutti gli altri posti diversi da P5.

Esercizio 2
1. $\forall x \, \exists y \, (y>x \wedge \neg\exists z,q \, (1<z<y \wedge y=zq))$
2. $\forall x \, (\exists z \, (z>x \wedge \forall y \, (f_z(y) \neq \perp)))$
3. $\exists x \, (\exists z \, (\forall y(y > z \rightarrow f_x(z) = \perp)))$

Esercizio 3
La grammatica G, nonostante sia non ristretta, genera un linguaggio regolare; più precisamente, essa genera il linguaggio L(G) fatto di stringhe costruite sull'alfabeto {a,b,c} in cui il terz'ultimo carattere è una c.

1. Il problema è decidibile, in quanto si tratta di fare una semplice analisi sintattica dell'automa scritto dall'utente (non è difficile immaginare di codificare mediante opportune stringhe le funzioni di transizioni corrispondenti agli automi scritti dagli studenti): se sugli archi compaiono solo simboli dell'alfabeto di input, l'automa è a stati finiti; se le etichette sugli archi sono del tipo a,A/β, l'automa è a pila (e si può capire se è deterministico o no analizzando, per ogni stato, le transizioni che escono da esso); se le etichette sono del tipo $a,A/B,\langle M_0,M_1\rangle$ (con $M_i \in \{R,L,S\}$), l'automa è una macchina di Turing (si può capire quanti nastri ha la macchina di Turing da quanti simboli compaiono prima della "/").

2. Il problema è decidibile. Infatti, poiché L(G) è un linguaggio regolare, si danno 2 casi: o lo studente scrive un automa più potente di un FSA, nel qual caso il programma se ne può accorgere (si veda la soluzione del punto 1) e segnalare che l'esercizio è sbagliato; oppure lo studente scrive un FSA, ed in questo caso il problema da risolvere è una semplice equivalenza di FSA (quello scritto dallo studente e quello che riconosce L(G), che è decidibile.