Algoritmi e Principi dell'Informatica

Appello d' esame

21 Febbraio 2013

Nome:_____

Matricola:_____

Firma:_____

Tempo a disposizione per la parte di informatica teorica: 1h30'

Esercizio 1 (6 punti)

1. Scrivere un'espressione regolare che descriva il linguaggio L di tutte e sole le stringhe costruibili con l'alfabeto $\{a, b\}$ che hanno un numero pari di a oppure un numero dispari di b (o entrambe le cose).

2. Costruire poi un automa deterministico a potenza minima che riconosca il linguaggio L.

3. Cambierebbe il tipo di automa (a potenza minima) se il linguaggio da riconoscere fosse quel sottoinsieme L' di L nelle cui stringhe il numero di b è pari al numero di a più 1?

Esercizio 2 (5 punti)

Una *lista* è una struttura sintattica definita induttivamente come segue:

- ε è una lista (comunemente chiamata *lista vuota*);
- se h è un carattere e T è una lista, allora $h \circ T$ è una lista (h è comunemente chiamato *testa* della lista ("head"), mentre T è chiamato *coda* della lista ("tail");
- nient'altro è una lista.

Esempio: la lista $c \circ (i \circ (a \circ (o \circ \varepsilon)))$ rappresenta la sequenza di caratteri *c i a o*.

Un *numero con notazione di Peano* è definito induttivamente come segue:

- 0 è un numero (comunemente chiamato *zero*);
- se n è un numero, allora anche $s(n)$ è un numero (comunemente detto *successore* di n);
- nient'altro è un numero.

Esempio: $s(s(0))$ rappresenta il successore del successore di 0 (cioè 2).

Si specifichi in logica del prim'ordine una funzione binaria f che riceve in ingresso una lista e un numero i espresso con notazione di Peano e restituisce il carattere in posizione (i+1)-esima nella lista (cioè il carattere con indice i nella lista, dove 0 è l'indice del primo carattere, 1 del secondo, e così via).

Esempio:

$$f(\, c \circ (i \circ (a \circ (o \circ \varepsilon))), s(s(0))\,) = \mathrm{a}$$

Esercizio 3 (5 punti)

E' noto che il problema di stabilire se una generica macchina di Turing accetti la stringa vuota è indecidibile.
Utilizzare la tecnica della *riduzione* di problemi per stabilire che il seguente problema P è indecidibile:

"Per una generica macchina di Turing M e un generico stato q di M, stabilire se M raggiunge prima o poi lo stato q nel caso in cui inizi la sua esecuzione su un nastro di ingresso vuoto."

Soluzione esercizio 1

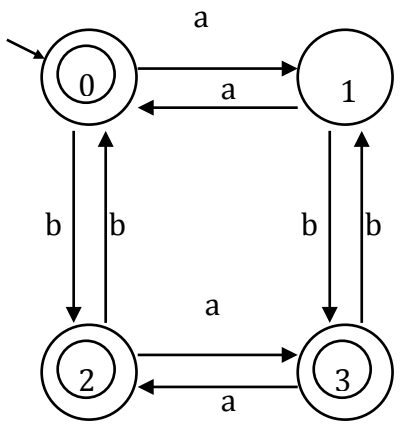
Una possibile espressione regolare è la seguente:

$b^* (b^* a b^* a b^*)^* \mid a^* b a^* (a^* b a^* b a^*)^*$

Anche la seguente espressione (leggermente più semplice) descrive L:

$b^* (a b^* a b^*)^* \mid a^* b a^* (b a^* b a^*)^*$

Per riconoscere L basta un automa a stati finiti, come mostrato di seguito.



Per il linguaggio L' l'automa a potenza minima cambierebbe necessariamente, in quanto un automa a stati finiti non sarebbe in grado di contare un numero n di a arbitrariamente grande per poi confrontarlo con il numero desiderato (n+1) di b.

Soluzione esercizio 2

$\forall h\ \forall T\ \forall n\ (\ f(h \circ T, 0) = h \wedge f(h \circ T, s(n)) = f(T, n)\ )$

Soluzione esercizio 3

Per mostrare l'indecidibilità di P, utilizziamo una riduzione dal problema di stabilire se una generica macchina di Turing accetti la stringa vuota, che, come anticipato, è indecidibile. Data una generica macchina T, basta costruire una macchina T' con gli stessi stati di T, più uno stato aggiuntivo q. T' si comporta esattamente come T, con la sola differenza che, qualora T termini, allora T' si sposta nello stato q e quindi termina. Chiaramente T accetta la stringa vuota se e solo se T' raggiunge lo stato q. Pertanto, se fossimo in grado di decidere il problema P, saremmo anche in grado di decidere il problema di accettazione della stringa vuota, il che è assurdo. Ergo P è indecidibile.