Algoritmi e Principi dell'Informatica
Modulo di Informatica Teorica
Appello del 6 luglio 2011

Tempo a disposizione: 1h30'

Esercizio 1 (11 punti)
Si consideri il linguaggio $L \subseteq \{a,b,c\}^*$ composto dalle stringhe di lunghezza ≥2 che finiscono con 'bc', e una funzione di traduzione τ che, per ogni stringa $x \in L$, cancella i simboli 'a', raddoppia le 'b', e mantiene inalterate le 'c'. Per esempio

τ(abc)=bbc, τ(cbabc)= cbbbbc, τ(c)=$\perp$, τ(cbab)= $\perp$.

- Si scriva il trasduttore finito che calcoli la traduzione descritta.
- Si scriva l'automa a stati finiti che accetti il linguaggio $\tau(L)$; se possibile si mostri come l'automa che accetta $\tau(L)$ si possa derivare dal trasduttore definito precedentemente.
- Basandosi sulla domanda precedente, si delinei una procedura sistematica per ottenere da un trasduttore a stati finiti, l'automa che accetti il linguaggio di uscita del trasduttore stesso.

Esercizio 2 (10 punti)
Si consideri un sistema composto da una lampada dotata di N>1 bottoni per accenderla e spegnerla. La lampada viene accesa premendo simultaneamente almeno due bottoni, mentre viene spenta premendo esattamente un bottone.
Si descriva il sistema attraverso una rete di Petri oppure formule logiche, usando i seguenti predicati:

L(t): la lampada è accesa al tempo *t*;
P(*b*, *t*): il bottone *b* viene premuto al tempo *t*.

Esercizio 3 (11 punti)
Si consideri l'insieme di macchine di Turing (MT) a nastro singolo con n stati definite su un alfabeto prefissato. Come noto, c'è un numero finito di tali MT a n stati. Iniziando l'esecuzione con il nastro con tutti blank, alcune di queste MT terminano, mentre altre non terminano.

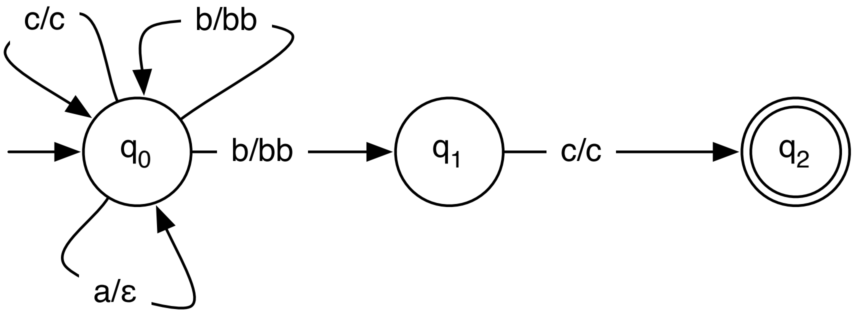
1. Dire se è decidibile il problema di stabilire se, data una generica MT dell'insieme considerato, questa termina partendo con il nastro con tutti blank.

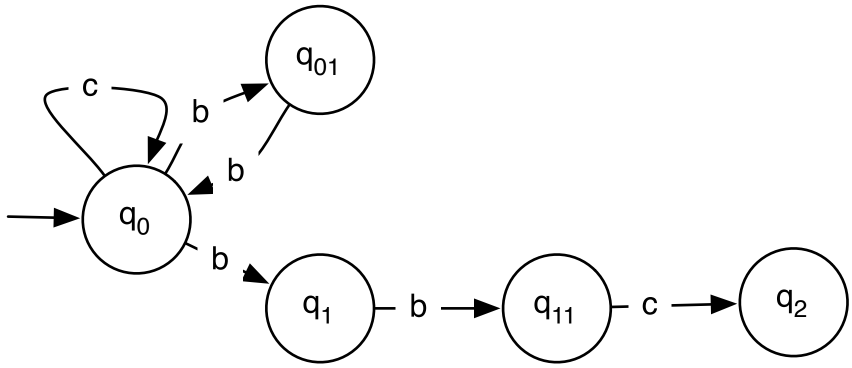
Si definisce S(n) il numero di passi eseguiti dalla MT che termina più tardi, tra quelle a n stati che terminano (iniziando l'esecuzione con il nastro con tutti blank).
Tale numero, noto anche come l'n-esimo numero "alacre castoro", è definito in modo univoco e, intuitivamente, rappresenta la durata dell'esecuzione della MT più "indaffarata" tra quelle che terminano. La funzione S(n) cresce molto velocemente al crescere di n.

2. Dimostrare che la funzione S(n) non è computabile.

Soluzioni

Es 1





Costruzione: in pratica basta "buttar via" la parte di input, mentre si trasformano in transizioni le stringhe di output: se una stringa è vuota, si perde la transizione; se è fatta da più di un carattere si mette una catena di stati con una "maglia" per ogni carattere.

Es 2

Versione a tempo discreto

$\exists b_1\ \exists b_2\ (b_1 \neq b2\ \land\ P(b_1, t)\ \land\ P(b_2, t)) \Rightarrow L(t+1)$

$\exists b_1\ (P(b_1,t)\ \land\ \neg\exists b_2\ (b_1 \neq b_2\ \land\ P(b_2, t)) \Rightarrow \neg L(t+1)$

$\neg\exists b\ P(b, t) \Rightarrow (L(t) \Leftrightarrow L(t+1))$

Es 3

Il problema di stabilire se, data una generica MT, questa termina partendo con il nastro con tutti blank è indecidibile. Questo si può vedere ad esempio usando il teorema di Rice. Infatti, l'insieme delle MT che terminano come descritto non è né l'insieme vuoto, né l'insieme di tutte le funzioni calcolabili.

Per dimostrare che la funzione S(n) non è computabile procediamo per assurdo.

Supponiamo di essere in grado di calcolare S(n) per ogni n. Allora possiamo decidere se una qualunque MT con n stati termina iniziando la sua esecuzione con il nastro d'ingresso con tutti blank. Infatti basta mettere in esecuzione la macchina: se si ferma, sappiamo che termina; se non si ferma entro S(n) passi, allora sappiamo che non terminerà mai, poiché S(n) è il numero massimo di passi eseguibili da una MT che termina.

Ma stabilire se una MT termina iniziando la sua esecuzione con il nastro d'ingresso vuoto è indecidibile, come dimostrato in precedenza. Assurdo.