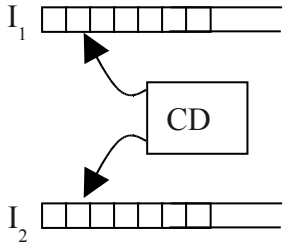
Informatica Teorica

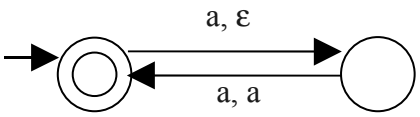
Prima prova in itinere - 8 Maggio 2008
NB: il punteggio è espresso in 13-esimi e riflette il peso relativo della prova rispetto all'intero esame il cui punteggio è in 30-esimi. Come già in altre occasioni è tuttavia possibile ottenere un punteggio complessivo superiore a 13/13.

Esercizio 1 (10/13 punti)
Parte 1
Si formalizzi la nozione di automa trasduttore finito deterministico FTε 1 che, al contrario del modello di trasduttore finito presentato a lezione (chiamiamolo FT), può anche effettuare ε -mosse ma può emettere, a ogni transizione, al più un solo simbolo sul nastro di uscita.

Dimostrare che il modello FTε 1 è almeno altrettanto potente del modello FT deterministico (ossia che qualsiasi FT può essere simulato da un opportuno FTε 1).

Parte 2
Formalizzare poi la nozione di automa finito deterministico a due ingressi, 2FA, tratteggiato in figura a sinistra.





2FA può fare ε -mosse, non facendo avanzare la testina di ingresso su uno o entrambi i nastri di ingresso I_1 e I_2. L'automa 2FA può essere visto come traduttore nel modo seguente: la stringa di ingresso x viene accettata e tradotta in τ (x) se e solo se la coppia di stringhe <x, τ (x)> sui due nastri I_1 e I_2 è accettata. Per esempio, l'automa 2FA nella figura a destra definisce la traduzione τ $(a^n)=a^{n/2}$ per $n \geq 0$ e pari. Mostrare che il modello 2FA è almeno altrettanto potente di FT deterministico, nel senso che se un FT deterministico accetta una stringa x e la traduce nella stringa τ (x), allora esiste un opportuno 2FA che accetta la coppia di stringhe <x, τ (x)> sui due nastri I_1 e I_2.

Valutare se il modello 2FA ha *esattamente* la stessa potenza espressiva di FT, cioè se qualsiasi traduzione definita da un 2FA nel modo sopra indicato può essere calcolata da un opportuno FT.

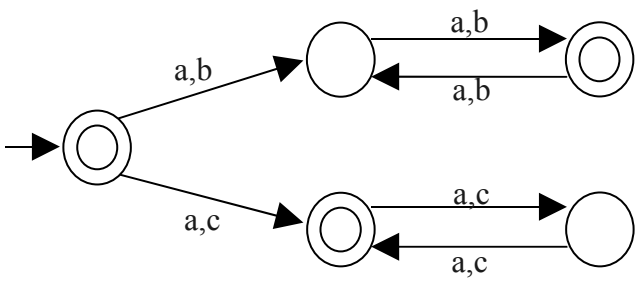
Esercizio 2 (4/13 punti)
Scrivere una grammatica G che generi il linguaggio {a^nb^n se n è pari >= 0, a^nc^n se n è dispari}

Soluzioni

Esercizio 1

- Un FTε 1 è una tupla (Q, I, O, δ, q_0, F) dove Q, I, O, q_0, e F hanno lo stesso significato che negli automi trasduttori tradizionali. La funzione di transizione δ: $Q \times I \cup \{\varepsilon\} \rightarrow Q \times O \cup \{\varepsilon\}$ definisce, per ogni coppia (q, i) di stato corrente e carattere in input (oppure ε) lo stato prossimo q' e il singolo carattere scritto in output (oppure ε), cioè δ(q,i) = (q', o). Affinchè sia deterministico si richiede che se δ(q,ε) è definito, allora δ(q,i) è indefinito per tutti gli $i \in I$.
  Conviene poi fornire la definizione di configurazione, transizione tra configurazioni, e trasduzione seguendo la prassi standard usata per automi a pila e macchine di Turing.
- Dato un FT = (Q, I, O, δ, q_0, F) deterministico, è sempre possibile definite un FTε 1 = (Q', I, O, δ', q_0, F) che definisce la stessa traduzione facendo in modo che, quando FT emette una stringa di lunghezza >1, l'FTε 1 corrispondente emetta in sequenza i caratteri della stringa, uno alla volta, mediante una serie di ε -mosse che lo fanno passare attraverso una serie di stati aggiuntivi, diversi da tutti gli altri, introdotti a questo scopo. Q' e δ' sono quindi definiti in questo modo. Q' contiene tutti gli stati che sono in Q. Inoltre, per ogni δ(q,i) = (q', s) con $s \in O^*$, se $|s| \leq 1$ si ha semplicemente δ'(q,i) = (q', s); se $|s| > 1$si aggiungono |s| stati q^s_1, q^s_2, ..., $q^s_{|s|}$ a Q' e si pone: (1) δ'(q,i) = (q^s_1, s[1]); (2) per ogni $2 \leq k \leq |s|$: δ'(q^s_{k-1},ε) = (q^s_k, s[k]); (3) δ'($q^s_{|s|}$,ε) = (q', ε);.
- Un 2FA è una tupla (Q, I_1, I_2, δ, q_0, F) dove I1 e I2 sono gli alfabeti dei due nastri di ingresso, Q, q_0, e F hanno il significato convenzionale. La funzione di transizione δ: $Q \times I_1 \cup \{\varepsilon\} \times I_2 \cup \{\varepsilon\} \rightarrow Q$ definisce per ogni tupla (q, x, y) di stato corrente q, carattere sul primo nastro x (o ε), e carattere sul secondo nastro y (o ε) lo stato prossimo δ(q, x, y). Si richiede per il determinismo che se δ(q, x, ε) (rispettivamente δ(q, ε , y)) è definito allora δ(q, x, y) è indefinito per ogni $y \in I_2$ (rispettivamente per ogni $x \in I_1$).
  La definizione di configurazione e transizione tra configurazioni sono definite di conseguenza nel solito modo (includendo nella configurazione il contenuto dei due nastri dalla posizione della testina in poi).
- Il modello 2FA è almeno altrettanto potente del modello FTε 1 (e quindi, transitivamente, del modello FT): dato un qualsiasi FTε 1, esiste un 2FA che simula ogni sua transizione, leggendo dal secondo nastro di ingresso lo stesso carattere che l'FTε 1 emette sul nastro di uscita ed effettuando ε -mosse sul primo o sul secondo nastro di ingresso quando, rispettivamente, l'FTε 1 fa una ε -mossa (i.e., non consuma ingresso) o non emette alcun simbolo in uscita. Formalmente, dato un FTε 1 = (Q, I, O, δ, q_0, F) si definisce il 2FA = (Q, I, O, δ', q_0, F) dove δ '(q, x, y) = q' se e solo se δ (q, x)=(q',y), $\forall q,q' \in Q$ e $\forall x \in I \cup \{\varepsilon\}$ e $\forall y \in O \cup \{\varepsilon\}$.
- Il modello 2FA è strettamente più potente del modello FT, perchè permette di definire una traduzione, accettandola come stringa in ingresso sul secondo nastro, anche nel caso in cui la traduzione dipenda da una proprietà della stringa di ingresso (quella sul primo nastro) che può essere decisa solo al termine della scansione, dopo un numero di passi illimitato a priori. Ciò non può essere fatto dal modello FT, a causa del vincolo di memoria finita che lo obbliga a emettere la traduzione "in tempo reale", durante la scansione della stringa in ingresso. Come esempio di ciò si consideri la seguente traduzione: τ (a^n)=b^n se n è pari, τ (a^n)=c^n se n è dispari. Nessun FT può calcolare questa traduzione, perchè il riconoscimento della stringa come avente lunghezza pari o dispari avviene solo al termine della scansione, mentre l'emissione della traduzione deve avvenire durante la scansione. Il 2FA mostrato in figura invece *definisce* la traduzione in questione accettando il linguaggio $L=\{(a^n,b^n) \mid n \geq 0, n \text{ pari}\} \cup \{(a^n,c^n) \mid n \text{ dispari}\}$

.



Esercizio 2

S ▪ X | Y
X ▪ aAb | ε
A ▪ aXb
Y ▪ aBc
B ▪ aYc | ε