Algoritmi e Principi dell'Informatica
Prima Prova in Itinere

25 Novembre 2014

Il tempo a disposizione è di **1 ora e 45 minuti.**

Esercizio 1 (punti 6/15 senza la parte facoltativa, 9 con la parte facoltativa)

Si ricordi che la *stringa inversa* di una generica stringa w, qui denotata w^R, è ottenuta da w scrivendo i suoi caratteri in ordine inverso, ad esempio $abaabb^R$ = bbaaba, $aaabbb^R$ = bbbaaa. *L'operazione di inversione* viene estesa dalle stringhe ai linguaggi nel modo naturale:

$L^R = \{w^R \mid w \in L\}$

Si dica, con brevi spiegazioni, quali delle seguenti famiglie di linguaggi sono chiuse rispetto all'operazione di inversione:

- Il linguaggi regolari
- I linguaggi non contestuali
- I linguaggi generati da grammatiche non ristrette
- **(Parte facoltativa)** I linguaggi non contestuali deterministici

NB: la parte facoltativa verrà valutata solo se nelle restanti parti si saranno ottenuti almeno 5 punti.

Esercizio 2 (punti 7/15)

Due macchine di Turing (MT) si dicono *isomorfe* se:

- hanno lo stesso numero di nastri;
- ad ogni mossa spostano le rispettive testine del nastro i-esimo alla stessa maniera (entrambe a destra o entrambe a sinistra o lasciano entrambe ferme);

Si noti che di conseguenza due MT isomorfe usano esattamente le stesse celle di memoria.

E' decidibile il fatto che due qualsiasi MT isomorfe siano equivalenti, ossia calcolino la stessa funzione o riconoscano lo stesso linguaggio? Giustificare brevemente la risposta

Esercizio 3 (punti 5/15)

Specificare in logica del prim'ordine il linguaggio $L_1 = \{ a^n b^n c^n \mid n \geq 0 \}$. Nella specifica si possono usare esclusivamente i seguenti predicati e funzioni:

- il predicato di appartenenza insiemistica $\in$
- il predicato di uguaglianza =
- la funzione di concatenazione $\cdot$
- le costanti dei caratteri alfabetici a, b, c e la stringa vuota ε
- il simbolo L_1 per riferirsi al linguaggio da specificare (più eventuali altri simboli L_2, L_3, … per eventuali linguaggi intermedi)

Non si può quindi utilizzare la funzione di elevamento a potenza.

Tracce di soluzioni

Esercizio 1

- Data una grammatica G non-contestuale che genera il linguaggio L, la grammatica G^R ottenuta trasformando ogni produzione di G, $A \rightarrow \alpha$, nella produzione $A \rightarrow \alpha^R$, genera evidentemente L^R.
- In particolare, se G è anche regolare, ogni produzione del tipo $A \rightarrow Ba$ viene trasformata in $A \rightarrow aB$ ottenendo quindi una grammatica che genera L^R e soddisfa ancora la definizione di grammatica regolare (è noto infatti che le grammatiche lineari a sinistra o lineari a destra sono tra loro equivalenti e generano entrambe tutti e solo i linguaggi regolari).
- Anche i linguaggi generati da grammatiche non ristrette sono chiusi rispetto all'inversione; infatti queste grammatiche sono equivalenti alle macchine di Turing. Data quindi una MT che riconosca L basta costruirne una che inverta la stringa di ingresso in un nastro ausiliario e successivamente simuli il comportamento della MT originaria usando il nastro ausiliario al posto del nastro di ingresso.
- **(Parte facoltativa)**
  I linguaggi non contestuali deterministici non sono invecc chiusi rispetto all'inversione. Ciò si può dedurre ricordando che $L = \{a^nb^n\} \cup \{a^nb^{2n}\}$ è nondeterministico e osservando invece che $L_1 = \{1a^nb^n\} \cup \{0a^nb^{2n}\}$ è deterministico (la lettrura del primo simbolo permette di stabilire subito se si dovranno contare 2 b per ogni a oppure un solo b). L_1^R però non è deterministico perché a inizio stringa non è possibile stabilire quale sarà il rapporto corretto tra il numero di b e quello di a.

Esercizio 2

Il problema non è decidibile: ogni coppia di MT può essere trasformata in una coppia equivalente che soddisfi la proprietà indicata: basta arricchire l'una con la memoria e il controllo dell'altra (che poi non verranno utilizzati per la computazione "vera"): quindi, date due MT con k_1 e k_2 nastri rispettivamente se ne costruscono altre due, entrambe di $k_1 + k_2$ nastri, rispettivamente equivalenti alle originarie. Perciò se fosse possibile decidere l'equivalenza delle due nuove MT sarebbe decidibile anche l'equivalenza tra due MT generiche.

Esercizio 3

$\forall x(x \in L_1 \leftrightarrow x = \varepsilon \vee \exists y \exists z \exists w\ x = y \cdot z \cdot w \wedge y \cdot z \in L_2 \wedge z \cdot w \in L_3 \wedge y \in L_4 \wedge z \in L_5 \wedge w \in L_6)$

$\wedge$

$\forall x(x \in L_2 \leftrightarrow x = \varepsilon \vee \exists y\ x = a \cdot y \cdot b \wedge y \in L_2)$

$\wedge$

$\forall x(x \in L_3 \leftrightarrow x = \varepsilon \vee \exists y\ x = b \cdot y \cdot c \wedge y \in L_3)$

$\wedge$

$\forall x(x \in L_4 \leftrightarrow x = \varepsilon \vee \exists y\ x = a \cdot y \wedge y \in L_4)$

$\wedge$

$\forall x(x \in L_5 \leftrightarrow x = \varepsilon \vee \exists y\ x = b \cdot y \wedge y \in L_5)$

$\wedge$

$\forall x(x \in L_6 \leftrightarrow x = \varepsilon \vee \exists y\ x = c \cdot y \wedge y \in L_6)$