Algoritmi e Principi dell'Informatica

Appello del 25 Febbraio 2011 – soluzioni

Esercizio 1

1. Il linguaggio riconosciuto dalla rete di Petri è: $L = (a(b|c)(b|c)c)^*$.

   Una grammatica G con il numero minimo di nonterminali (1) che generi L è caratterizzata dalle seguenti produzioni:

$$S \rightarrow \epsilon \mid abbcS \mid abccS \mid acbcS \mid acccS$$

2. La grammatica G è di tipo non contestuale. Il linguaggio L è regolare, pertanto G non è a potenza minima in quanto basterebbe una grammatica regolare per generare L.

3. La rete di Petri del testo fa uso di archi inibitori, pertanto non è a potenza minima, in quanto per riconoscere L, che è un linguaggio regolare, sarebbe senz'altro sufficiente una rete di Petri senza archi inibitori.

Esercizio 2

1. Si tratta di una domanda con risposta chiusa sì/no che non dipende da alcun input, pertanto il problema è decidibile.

2. Non è decidibile per il teorema di Rice, in quanto l'insieme di funzioni descritto non è l'insieme vuoto, né l'insieme universo.

3. È semidecidibile, in quanto basta provare le computazioni da 0 a 10 di f_y, alternando l'esecuzione di $f_y(0)$, $f_y(1)$, $\ldots$, $f_y(10)$ eseguendole un passo per ciascuna. Se tutte e 11 le esecuzioni terminano, prima o poi lo scopriamo, da cui la semidecidibilità.

4. No, per Rice, per le stesse ragioni del punto 2.

5. Non è nemmeno semidecidibile. Se fosse semidecidibile anche questo problema, unitamente al fatto che è semidecidibile stabilire se f_y sia definita per $x \leq 10$ (vedi punto 3), allora sarebbe semidecidibile anche il problema di stabilire se una generica funzione sia definita per ogni x, ovvero sia totale, il che è notoriamente falso.

Esercizio 3

1. Il linguaggio generato da G_2 è il linguaggio di tutte e sole le stringhe in cui il numero di a è pari al numero di b più uno, ossia $L(G_2) = \{x | x \in \{a, b\}^* \wedge \#a(x) = \#b(x) + 1\}$. Come affermato nel testo, le stringhe generate a partire da T sono quelle del linguaggio $L(G_1)$ di tutte e sole le stringhe in cui le a e le b sono in ugual numero. Le stringhe di $L(G_2)$, tramite la produzione $S \rightarrow TaT$, sono generate dalla giustapposizione di una a tra due stringhe di $L(G_1)$.

2. Da quanto scritto sopra, le stringhe generate da G_2 hanno la forma $x \cdot a \cdot y$, dove $x, y \in L(G_1)$. È immediato concludere che G_2 genera *soltanto* stringhe in cui vi sia una a in più rispetto alle b, poichè sia x sia y sono già bilanciate, appartenendo a $L(G_1)$. Per mostrare che G_2 genera *tutte* le stringhe in cui vi sia una a in più rispetto alle b, si consideri una generica stringa w tale per cui $\#a(w) = \#b(w) + 1$. Poiché w contiene più a che b, scandendo w da sinistra a destra, esiste necessariamente un carattere di w che è la prima a in soprannumero rispetto alle b incontrate (incluso il caso in cui non si siano incontrate b e cioè che la a sia il primo carattere di w). Sia x la sottostringa composta da tutti i caratteri a sinistra di quella a, e y quella composta da tutti i caratteri a destra della a. Per quanto scritto, la a che separa x da y è la prima a in soprannumero, quindi in x le a e le b sono in ugual numero. A questo punto segue necessariamente che anche nella y le a e le b sono in ugual numero, altrimenti la stringa $w = x \cdot a \cdot y$ non soddisferebbe più il vincolo $\#a(w) = \#b(w) + 1$.