Informatica Teorica

Appello d'esame dell'8 Settembre 2008
Il tempo a disposizione è di 2 ore

Esercizio 1 (Punti 13)

Si considerino le seguenti regole relative al funzionamento di un apparecchio per l'uso del Bancomat:

1. L'apparecchio si trova inizialmente in uno stato di riposo
2. L'introduzione della tessera provoca la transizione in uno stato di immissione codice
3. Il codice consta di 5 cifre
4. L'immissione del codice è ultimata se l'utente ha digitato 5 cifre e successivamente premuto il tasto ENTER, oppure dopo che sono trascorsi 5 secondi dall'immissione della quinta cifra.
5. Una volta ultimata l'immissione del codice l'apparecchio passa in uno stato di verifica.
6. Se, nello stato di immissione, trascorrono più di 5 secondi senza che si digiti una cifra e non è stata ancora digitata la quinta cifra, viene espulsa la tessera e l'apparecchio torna nello stato di riposo.

Si formalizzino mediante formule del prim'ordine solo le regole 4, 5 e 6 (le altre servono da contesto ma non è necessario che vengano formalizzate esplicitamente). Si suggerisce di utilizzare i seguenti predicati che denotano possibili *stati* in cui l'apparecchio si trova in un generico istante t:

- Riposo(t)
- ImissioneDati(t)
- Verifica(t)
- CifreImesse(t, k) indica che all'istante t le cifre immesse sono k; lo stato ha senso solo durante lo stato ImmissioneDati.

e i seguenti predicati che denotano possibili *eventi* che si verificano in un generico istante t:

- PremiCifra(t)
- PremiEnter(t)
- EspulsioneTessera(t)

Esercizio 2 (Punti 11)

Dire se i seguenti problemi sono decidibili, solo semidecidibili, o né uno né l'altro:

1. Stabilire se, data una generica Macchina di Turing M, il linguaggio riconosciuto da M è vuoto.
2. Stabilire se, data una generica Macchina di Turing M che non ha cicli (cioè il cui grafo corrispondente è aciclico), il linguaggio riconosciuto da M è vuoto.
3. Stabilire se, data una generica Macchina di Turing M che non ha autoanelli (cioè il cui grafo corrispondente non ha archi che puntano al nodo da cui originano; un autoanello è un caso particolare di ciclo), il linguaggio riconosciuto da M **non è** vuoto.

Esercizio 3 (Punti 10)

Si consideri la traduzione di una stringa di ingresso w $\in$ $\{a,b,c\}^+$ in una stringa $a^{n/2}b^m$, dove n è il numero di a nella stringa w e m è il numero di b nella stringa w.

1. Si descriva a parole una Macchina di Turing a k nastri che realizza la traduzione indicata, e se ne diano le complessità spaziale e temporale.
2. Si descriva a parole una Macchina RAM che realizza la traduzione indicata, e se ne diano le complessità spaziale e temporale usando il criterio di costo logaritmico.
3. Si descriva a parole una Macchina di Turing a nastro singolo che realizza la traduzione indicata, e se ne diano le complessità spaziale e temporale.

NB: Il punteggio conseguito sarà tanto più alto tanto migliori saranno le complessità delle macchine ideate.

Soluzioni

Esercizio 1

Come già in altre occasioni, per un generico stato S, Up_to_now_S(t) è una notazione abbreviata per la formula seguente

$$\exists\delta(\forall t_1(t-\delta<t_1<t)\rightarrow S(t_1)) \qquad (\text{sottinteso}: \delta>0)$$

Similmente From_now_on_S(t) è una notazione abbreviata per la formula seguente:

$$\exists\delta(\forall t_1(t\le t_1<t+\delta)\rightarrow S(t_1))$$

Ciò premesso le regole 4 e 6 possono essere formalizzate nel modo seguente, sottintendendo per ognuna la quantificazione universale ∀t, k:

Regole 4 e 5

(

Up_to_now_CifreImmesse(t, 5) ∧PremiEnter(t)

→

From_now_on_¬ImmissioneDati(t) ∧From_now_on_Verifica(t)

)

∧

(

Up_to_now_CifreImmesse(t, 5) ∧PremiCifra(t-5) ∧

(∀t1 (t-5 < t1 < t) →(¬PremiEnter(t) ∧¬ PremiCifra(t)))

→

From_now_on_¬ImmissioneDati(t) ∧From_now_on_Verifica(t)

)

Regola 6

Up_to_now_ImmissioneDati(t) ∧Up_to_now_CifreImmesse(t, k) ∧(0 ≤ k <5)

∧

(∀t1 (t-5 < t1 < t) →(¬PremiEnter(t) ∧¬ PremiCifra(t)))

→

From_now_on_¬ImmissioneDati(t) ∧From_now_on_Riposo(t) ∧EspulsioneTessera(t)

Esercizio 2

1.

Il problema non è né decidibile, né semidecidibile. Non è decidibile per il teorema di Rice (l'insieme delle MT che riconoscono il linguaggio vuoto non è l'insieme vuoto, né è l'insieme universo).

Non è neanche semidecidibile, in quanto è semidecidibile il suo complemento: enumerando opportunamente le stringhe in ingresso, con la solita tecnica diagonale è possibile simulare le esecuzioni della MT in modo che, se una stringa viene accettata, prima o poi questa viene trovata; in caso contrario, la computazione prosegue all'infinito.

2.

Il problema è decidibile (e quindi anche semidecidibile). Il numero di computazioni di una MT senza cicli è finito ed ognuna di esse è fatta di un numero finito di passi. E' quindi possibile enumerarle tutte, e determinare se almeno una di queste si porta in stato finale oppure no.

3.

Il problema non è decidibile, ma è semidecidibile. Data una MT qualunque (anche con autoanelli) è possibile costruirne in modo algoritmico una che sia invece priva di autoanelli (è sufficiente "sdoppiare" gli stati su cui ci sono autoanelli). Di conseguenza, è facile ridurre il problema della *emptiness* del linguaggio di una MT generica (cioè il determinare se il linguaggio accettato è vuoto o no) a quello di una MT senza autoanelli.

Se quindi fosse decidibile il problema in oggetto, lo sarebbe anche quello della *emptiness* di una MT generica (che è in effetti il complemento di quello in oggetto), che non è decidibile per quanto detto al punto 1.

Il problema è invece semidecidibile, in quanto, sempre per quanto detto al punto 1, per una generica MT (e quindi a maggior ragione per una senza autoanelli) è possibile, enumerando le stringhe in ingresso e le varie computazioni con meccanismo diagonale, determinare se questa accetta almeno una stringa (in caso contrario il procedimento algoritmico non termina).

Esercizio 3

1.

E' possibile scrivere una MT che effettua la traduzione desiderata senza fare uso di nastri di memoria, e in tempo Θ (n).

Tale macchina deve semplicemente scorrere il nastro di input 2 volte: una per produrre in uscita $a^{n/2}$, ed una per produrre in uscita b^m. Più precisamente, nella prima passata, la macchina produce, ogni 2 a incontrate, una a in uscita. Arrivata in fondo al nastro di input, essa lo scorre al contrario, producendo una b per ogni b letta.

2.

Una macchina RAM, non potendo scorrere al contrario il nastro di input, dovrà necessariamente ricorrere alla memoria per effettuare la traduzione desiderata.

Più precisamente, essa si può comportare nella seguente maniera: legge la stringa in ingresso e, per ogni a (risp. b) letta, incrementa un opportuno contatore memorizzato in M[1] (risp. M[2]). Alla fine della lettura divide M[1] per 2, quindi ripete M[1] volte la scrittura in uscita di a, ed M[2] volte la scrittura in uscita di b.

Le complessità (a criterio di costo logaritmico) di una simile macchina sono Θ (n log(n)) per quella temporale, e Θ (log(n)) per quella spaziale.

3.

Una Macchina di Turing a nastro singolo può realizzare la traduzione desiderata con un meccanismo simile a quello dell'insertion sort per portare prima tutte le a in testa alla stringa, farle seguire dalle b, e quindi chiudere con le c.

Una volta fatto questo, è sufficiente cancellare le c in coda per ottenere la traduzione desiderata.

La complessità temporale di una simile macchina di Turing è, come per l'algoritmo di insertion sort, Θ (n^2); quella spaziale è Θ (n).