

Informatica Teorica

Prova d'esame - 13 Luglio 2005, Sezione Pradella

Attenzione! I voti proposti verranno pubblicati sul sito:

<http://www.elet.polimi.it/upload/pradella/IT.html>

Gli studenti avranno tempo due giorni (48 ore, per la precisione) dalla data della pubblicazione per rifiutare il voto proposto, se sufficiente. L'eventuale rifiuto deve essere comunicato via email, facendo uso dell'indirizzo ufficiale del Poliself, non di indirizzo privato! Trascorsi i due giorni, i voti verranno registrati e trasmessi alla segreteria.

Per evitare disagi dovuti a malfunzionamenti del servizio di posta elettronica, il docente spedisce a ogni studente "rinunciatario" un esplicito messaggio di "ricevuta". In caso di mancata ricezione di tale ricevuta si consiglia di contattare il docente telefonicamente.

Esercizio 1 (punti 8/30-esimi)

Si scriva un automa che riconosca il linguaggio L le cui stringhe sono costruite sull'alfabeto $A=\{0, 1\}$ e sono fatte nella seguente maniera: le stringhe hanno lunghezza dispari; se il primo e l'ultimo carattere della stringa sono entrambi uguali ad '1', il carattere di centro è anch'esso uguale ad '1', altrimenti il carattere di centro è uguale a '0'.

NB: il punteggio massimo verrà assegnato solo se l'automa ideato sarà a potenza riconoscitiva minima tra quelli che riconoscono il linguaggio desiderato.

Esercizio 2 (punti 8/30-esimi)

Descrivere (senza necessariamente codificarla nei dettagli) una macchina RAM che riconosce il linguaggio L descritto nell'esercizio 1, e se ne dia la complessità spaziale e temporale a costo costante e a costo logaritmico.

Esercizio 3 (punti 8/30-esimi)

Scriva una formula logica che descrive un segnale fatto nella seguente maniera: dal momento in cui il segnale viene emesso la prima volta (che potrebbe anche non essere l'istante 0), esso viene emesso ad intervalli che si raddoppiano continuamente. L'intervallo tra i primi due istanti di emissione può essere qualunque.

In altre parole, se la distanza fra la $(k-1)$ -esima emissione e la k -esima emissione è d , la distanza tra la k -esima e la $(k+1)$ -esima emissione è $2d$.

Un esempio possibile tra i tanti di evoluzione temporale del segnale è la seguente (vengono indicati gli istanti in cui il segnale viene emesso; si noti che l'intervallo tra le prime due emissioni è pari a 3 istanti di tempo):

2, 5, 11, 23, 47, 95, 191, ...

(continua dietro)

Esercizio 4 (punti 8/30-esimi)

parte a.

Siano date le seguenti pre- e post- condizioni di un metodo Java che ha un parametro in ingresso *arg* di tipo *String*, ritorna un *int*, e può sollevare un'eccezione *ComputationException*:

Pre: $arg \neq \text{null} \wedge 1 \leq \text{length}(arg) \leq 50 \wedge$
 $\forall i, j (1 \leq i < j \leq \text{length}(arg) \rightarrow arg[i] \neq arg[j])$

Post: $\text{result} > 0 \vee \text{raise } \text{ComputationException}$

laddove *result* indica il valore ritornato dal metodo (se questo ritorna senza sollevare eccezioni) e *raise ComputationException* indica il fatto che viene sollevata un'eccezione *ComputationException*; inoltre, *length(arg)* indica la lunghezza della stringa *arg*, e *arg[i]* indica l'i-esimo carattere della stringa.

E' decidibile il problema di stabilire se un'implementazione I soddisfa la specifica data sopra?

Si motivi adeguatamente la risposta.

parte b.

Sia data la seguente implementazione

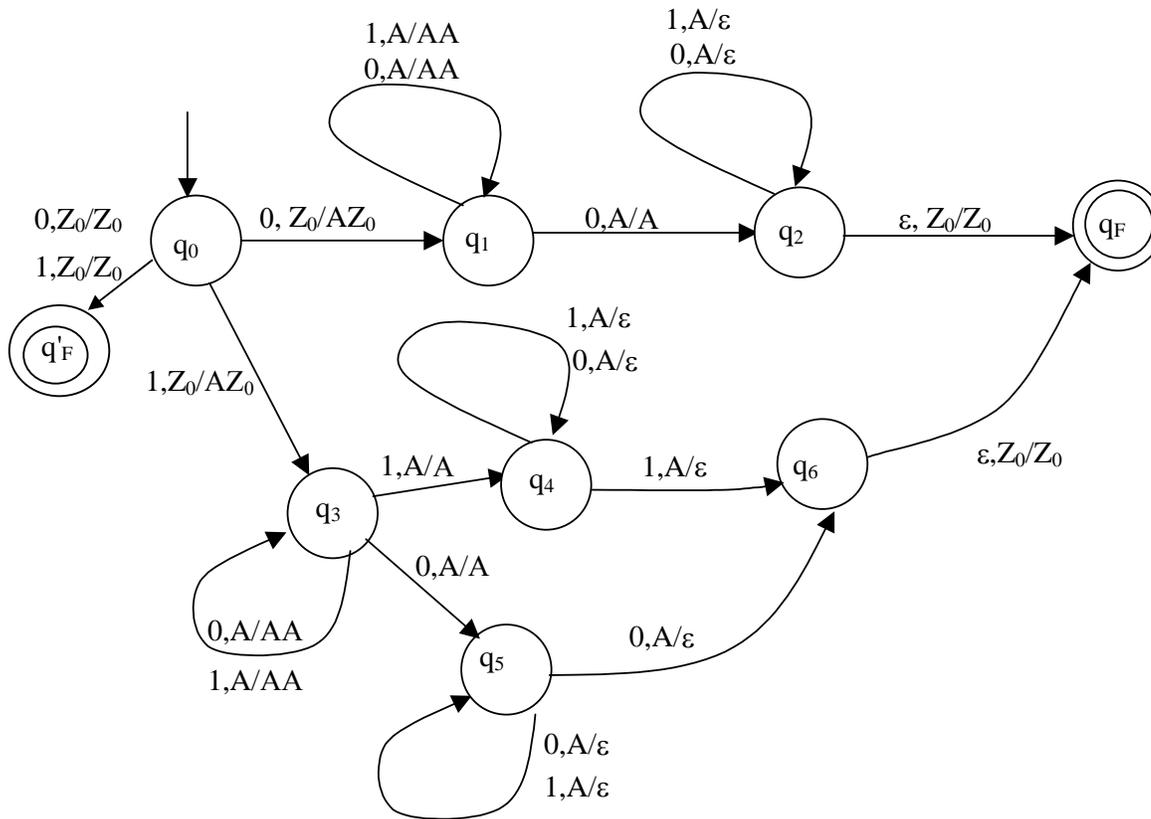
```
int myImpl(String arg) throws ComputationException {
    int sum = 0, i = 1;
    try {
        while(sum < (int)Math.pow(23.5, 4.21)){
            int diff = (int)(arg.charAt(i-1)-arg.charAt(0));
            sum += (int)(diff*Math.pow(4.67, 2.9));
            if (i == arg.length()) i = 1;
            else i++;
        } catch (Exception e) {
            throw new ComputationException();
        }
    }
    return sum;
}
```

E' decidibile il problema di stabilire se *myImpl* soddisfa la specifica descritta sopra?

Soluzioni

Esercizio 1

Il linguaggio L è riconosciuto dal seguente automa a pila nondeterministico:



Evidentemente non è possibile riconoscere L con una macchina a stati finiti, visto che le stringhe di L possono essere arbitrariamente lunghe, nè con un automa a pila deterministico, che non è in grado di “indovinare” il punto centrale della stringa.

Esercizio 2

Una semplice macchina RAM che riconosce il linguaggio è fatta nella seguente maniera.

Innanzitutto, essa legge una volta i caratteri in ingresso; ogni carattere letto viene memorizzato, e contemporaneamente si incrementa un contatore che mi dice quanti caratteri ho letto fino a quel momento. Alla fine della lettura, se ho contato un numero di caratteri pari, segnalo sul nastro di uscita che la stringa non è accettata. Se invece ho contato un numero di caratteri dispari, vado a leggere il primo carattere, l'ultimo, e quello di mezzo (per recuperare l'offset del quale rispetto al primo carattere basta dividere per 2 il numero di caratteri letti). Se questi sono tutti uguali ad 1, oppure se quello di centro ed almeno uno tra il primo e l'ultimo sono uguali a 0, accetto la stringa, segnalandolo sul nastro di uscita, altrimenti rifiuto.

A costo costante, sia la complessità spaziale che quella temporale sono $\Theta(n)$.

A costo logaritmico, la complessità spaziale è ancora $\Theta(n)$, mentre quella temporale è $\Theta(n \cdot \log(n))$.

Esercizio 3

Indicando, come al solito, col predicato $e(t)$ che il segnale viene emesso al tempo t , una possibile soluzione è:

$$\begin{aligned} & \forall t_1, t_2 (t_1 < t_2 \wedge e(t_1) \wedge e(t_2) \wedge \forall t' (t_1 < t' < t_2 \rightarrow \neg e(t')) \\ & \quad \rightarrow \\ & \quad e(t_2 + 2(t_2 - t_1)) \wedge \forall t' (t_2 < t' < t_2 + 2(t_2 - t_1) \rightarrow \neg e(t'))) \end{aligned}$$

Esercizio 4

parte a.

Non è decidibile.

Una maniera per dimostrarlo potrebbe essere tramite il Teorema di Rice: l'insieme delle implementazioni (cioè delle MT, visto che si può tradurre un programma Java in una MT equivalente) che soddisfano la specifica non è certamente l'insieme vuoto (esiste almeno una funzione che soddisfa la specifica), e non è certamente neanche l'insieme universo (è banale pensare ad una funzione che non soddisfa la specifica), quindi il problema di stabilire se una data MT appartiene all'insieme non è decidibile.

Una maniera alternativa potrebbe riducendo il problema della terminazione al problema del soddisfacimento della specifica. Preso un programma qualunque P (facendo attenzione che in P non compaia il parametro arg) basta in effetti costruire un metodo siffatto:

```
int Pridotto(String arg) throws ComputationException {
    P
    return 1;
}
```

Tale metodo soddisfa la specifica se e solo se P termina.

Se sapessi risolvere il problema di partenza, saprei anche risolvere il problema della terminazione di un programma qualunque, che è un assurdo.

parte b.

In questo caso, il problema è banalmente decidibile.

Infatti, l'implementazione è in questo caso fissata a *myImpl*, quindi si danno 2 soli casi: o *myImpl* soddisfa le specifiche, oppure non le soddisfa. La risposta è quindi in questo caso binaria, o sì o no, e la MT che risolve il problema o è quella costante uguale ad 1, oppure è quella costante uguale a 0. In entrambi i casi essa è banalmente calcolabile.