Esercizi sui modelli computazionali

Roberto Cordone

3 novembre 2007

1 Confronto fra alcuni modelli computazionali

Da un punto di vista funzionale, i diversi modelli computazionali sono caratterizzati, fra le altre cose, da due aspetti fondamentali:

1. il modo in cui accedono all'istanza:
   - scorrendola passo per passo dal primo all'ultimo simbolo (automa a stati finiti, automa con pila, rete di Petri)
   - scorrendola un passo alla volta, ma liberamente, senza limiti sul numero di accessi a ciascun simbolo (macchina di Turing)
   - accedendo direttamente a qualsiasi simbolo (macchina RAM)
2. il modo in cui conservano informazioni:
   - nello stato, avendo un insieme di stati finito e quindi potendo conservare una quantità di informazioni finita (macchina di Turing)
   - nello stato, avendo un insieme di stati infinito (rete di Petri, dove lo stato è la marcatura, cioè il numero di token in ogni posto)
   - nello stato, avendo un insieme di stati finito, e in strutture ausiliarie infinite; la struttura determina la modalità di accesso alle informazioni
     - se si tratta di una pila, l'accesso è rigorosamente *LIFO* (*last in/first out*) (automa con pila)
     - se si tratta di due pile, l'accesso è libero, ma costa tempo giostrare i simboli fra le due pile (automa con due pile)
     - se si tratta di un nastro, l'accesso è libero, ma costa tempo scorrere il nastro (macchina di Turing)
     - se si tratta di registri ad accesso diretto, l'accesso è libero e istantaneo (macchina RAM)

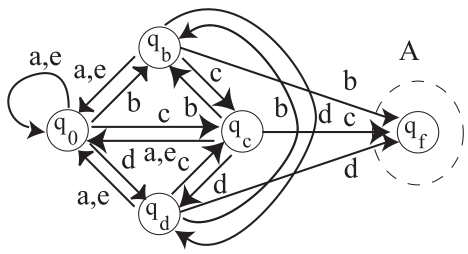
2 Esercizio 1: Consonanti doppie

P: La stringa $I \in \{a, b, c, d, e\}^*$ contiene consonanti doppie?

Con consonanti doppie si intende una coppia di consonanti identiche contigue. Sono accettabili anche sequenze più lunghe di due consonanti identiche (nel senso che, se esiste una tale sequenza, ne esiste anche una di due).

Gli stati devono conservare l'informazione sull'ultimo carattere letto dall'istanza. Precisamente, è necessario poter distinguere l'una dall'altra le singole consonanti, ma non le vocali (quindi servono tre stati q_b, q_c e q_d). E non è neppure necessario distinguere la situazione in cui si è appena letta una vocale da quella in cui non si è ancora letto nulla (quindi basta lo stato iniziale q_0). Infine, occorre distinguere lo stato q_f in cui si sono lette le desiderate due consonanti. Quindi, bastano 5 stati:

1. q_0: non si è ancora letto nulla o si è letta una vocale
2. q_b: si è letta una sola b
3. q_c: si è letta una sola c
4. q_d: si è letta una sola d
5. q_f: si sono lette due consonanti identiche



In ogni stato, la lettura del *blank* fa arrestare la macchina. L'unico stato accettante è q_f. Non occorrendo altro, ci basta un automa a stati finiti. Se vogliamo usare una macchina di Turing, basta definire l'uscita come sempre identica all'ingresso e lo spostamento lungo il nastro come sempre identico a D (*destra*).

3 Esercizio 2: Divisibilità per tre

P: La stringa $I \in \{0, \dots, 9\}^*$ rappresenta un numero divisibile per 3?

Si tratta di sommare le cifre e verificare se la somma sia divisibile per 3, eventualmente riapplicando al risultato lo stesso algoritmo, sinché non si rimane con una cifra sola, per la quale sia banale determinare la divisibilità.

Non è necessario conservare la somma nello stato: basta conservare il resto della divisione per 3, che può essere solo 0, 1 o 2. Qui basta un automa a stati finiti, con 4 stati.

1. q_0: non si è ancora letto nulla
2. q_1: la somma delle cifre lette, divisa per 3, dà resto 1
3. q_2: la somma delle cifre lette, divisa per 3, dà resto 2
4. q_3: le cifre lette hanno somma divisibile per 3

Bisogna distinguere il caso in cui non si è letto nulla (q_0) da quello in cui la somma delle cifre lette è divisibile per 3 (q_3) perché se ci si ferma nel primo la soluzione è *falso*, se ci si ferma nel secondo è *vero*. Il sottoinsieme degli stati accettanti è $A = \{q_3\}$.

4 Esercizio 3: Prevalenza di 1

Ci vuole l'automa con pila, e l'idea e' che:

- se la pila ha in cima γ_0, gli sostituiamo il simbolo in ingresso seguito da γ_0 e passiamo in uno stato 0 o 1 secondo il simbolo letto
- se la pila ha in cima 0 e si legge 1 o ha in cima 1 e si legge 0, gli sostituiamo una stringa vuota e restiamo nello stato corrente
- se la pila ha in cima 0 e si legge 0 o ha in cima 1 e si legge 1, gli sostituiamo rispettivamente 00 o 11 e restiamo nello stato corrente
- se si legge il fine nastro, δ non è definita, per cui ci si ferma: se lo stato è quello iniziale o 1 la risposta è *vero*, altrimenti *falso*

5 Esercizio 4: Palindromicità

Ci vuole una macchina di Turing, così organizzata:

- $Q = \{q_S, q_1, q_0, q_{1f}, q_{0f}, q_R\}$, con stato iniziale q_S e stati accettanti $A = \{q_S, q_{1f}, q_{0f}\}$
- $\Sigma = \{0, 1\}$ e $\Gamma = \{0, 1, -\}$
- δ è definita dalla seguente tabella

Q	$\times$	Γ	Q	$\times$	Γ	$\times$	$\{S, D\}$	Commento
q_S		0	q_0		0		D	q_0 significa "letto 0"
q_S		1	q_1		1		D	q_1 significa "letto 1"
q_S		$-$	q_T		$-$		D	stringa vuota: palindroma (lunghezza pari)
q_1		0	q_1		0		D	proseguire verso destra
q_1		1	q_1		1		D	proseguire verso destra
q_1		$-$	q_{1f}		$-$		S	fine stringa: tornare indietro
q_0		0	q_0		0		D	proseguire verso destra
q_0		1	q_0		1		D	proseguire verso destra
q_0		$-$	q_{0f}		$-$		S	fine stringa: tornare indietro
q_{1f}		1	q_R		$-$		S	letto 1: cancellarlo e tornare indietro
q_{1f}		$-$	q_T		$-$		S	trovato blank: fine (lunghezza dispari)
q_{0f}		0	q_R		$-$		S	letto 0: cancellarlo e tornare indietro
q_{0f}		$-$	q_T		$-$		S	trovato blank: fine (lunghezza dispari)
q_R		0	q_R		0		S	retrocedere verso sinistra
q_R		1	q_R		1		S	retrocedere verso sinistra
q_R		$-$	q_S		$-$		D	trovato l'inizio della stringa

6 Esercizio 5: Anagrammi

Anche per riconoscere anagrammi, occorre la macchina di Turing