Informatica Teorica

Sezione Cremona + Como

Appello del 20 Luglio 2004

Coloro che recuperano la I prova risolvano gli esercizi 1 e 2 tra quelli indicati qui sotto entro un'ora.

Coloro che recuperano la II prova risolvano gli esercizi 3 e 4 tra quelli indicati qui sotto entro un'ora.

Coloro che recuperano entrambe le prove o che sostengono l'appello della semiunità di Informatica Teorica VO risolvano tutti gli esercizi entro due ore.

Coloro che sostengono l'esame del corso integrato con Algebra risolvano uno a scelta tra gli esercizi 1, 2, e l'esercizio 3 entro un'ora.

NB: i punteggi indicati per i vari esercizi fanno riferimento esclusivamente al corso di Informatica Teorica; non hanno valore per il corso integrato.

Esercizio 1 (punti 9/15)

Si specifichi, mediante una formula del prim'ordine un apparato che funziona nel modo seguente:

All'istante 0 esso emette un segnale s, che può essere uno 0 o un 1. Se, *dopo* l'emissione di s e *prima* dello scadere di 3 secondi, viene ricevuto lo stesso segnale in risposta, allora allo scadere del terzo secondo viene emesso un nuovo segnale invertendone il valore (0 se prima si era emesso 1 e viceversa); in caso contrario viene riemesso lo stesso segnale inviato in precedenza. Il comportamento dell'apparato si ripete poi periodicamente con periodo di 3 secondi applicando sempre la stessa regola.

L'apparato emette un segnale soltanto in istanti che sono multipli di 3 secondi.

Si suggerisce di usare i predicati emette(s,t) e riceve(s,t) per indicare, rispettivamente l'emissione e la ricezione del segnale s all'istante t.

NB: il domino temporale può essere sia discreto (ad esempio, l'insieme dei naturali) che continuo (ad esempio, l'insieme dei reali).

Esercizio 2 (punti 8/15)

Con riferimento all'esercizio precedente, si assuma un tempo discreto (la cui unità sia il secondo). Si costruisca un'opportuna macchina astratta, preferibilmente a potenza minima, che abbia come alfabeto di ingresso (rispettivamente, di uscita) la ricezione (risp., emissione) di 0, la ricezione (risp., emissione) di 1, oppure l'assenza di segnale e si comporti come specificato nell'esercizio 1.

NB: essendo il comportamento dell'apparato periodico, senza che sia prevista una sua terminazione, la macchina astratta, di conseguenza, non dovrà prevedere una situazione di arresto, a meno che non debba bloccarsi a causa di errori.

Esercizio 3 (punti 7/15)

Si dica, giustificando brevemente la risposta, quali di queste affermazioni sono vere e quali false:

1. La funzione $g(y,x) = (1$ se $f_y(x)$ è pari, 0 altrimenti) è calcolabile.
2. La funzione $g(y,x) = (1$ se $f_y(x)$ è pari, $\perp$ altrimenti) è calcolabile
3. La funzione $g(y) = (1$ se $f_y(x)$ è pari $\forall x$, 0 altrimenti) è calcolabile
4. La funzione $g(y) = (1$ se $f_y(x)$ è pari $\forall x$, $\perp$ altrimenti) è calcolabile

NB: Nel caso la risposta a qualcuna delle domande precedenti fosse "Falsa" e se ne fornisse una dimostrazione mediante tecnica diagonale, il punteggio acquisito verrebbe maggiorato di punti 3.

Esercizio 4 (punti 8/15)

Si descriva con sufficiente precisione, ma senza necessariamente specificare ogni dettaglio, come una Macchina di Turing a nastro singolo deterministica (si ricordi che una Macchina "a nastro singolo" è diversa da una macchina a k nastri con k = 1) possa riconoscere il linguaggio $L = \{ww, w \in \{a,b\}^*\}$ analizzandone la complessità spaziale e temporale (a meno dell'ordine di grandezza determinato dalla Θ–equivalenza).

Soluzioni

Esercizio 1

$\forall s$

$(s \in \{0,1\}$

$\wedge$

$emette(0,0) \vee emette(1,0) \wedge$

$\forall t, k(k \in N \wedge k > 0 \wedge t = 3.k \rightarrow$

$\quad (emette(s,(k-1).t) \rightarrow$

$\quad\quad (\exists t_1((k-1).t < t_1 < k.t \wedge riceve(s,t_1)) \rightarrow emette((s+1) \bmod 2, t)) \wedge$

$\quad\quad (\neg\exists t_1((k-1).t < t_1 < k.t \wedge riceve(s,t_1)) \rightarrow emette(s,t))$

$\quad))$

$\quad)$

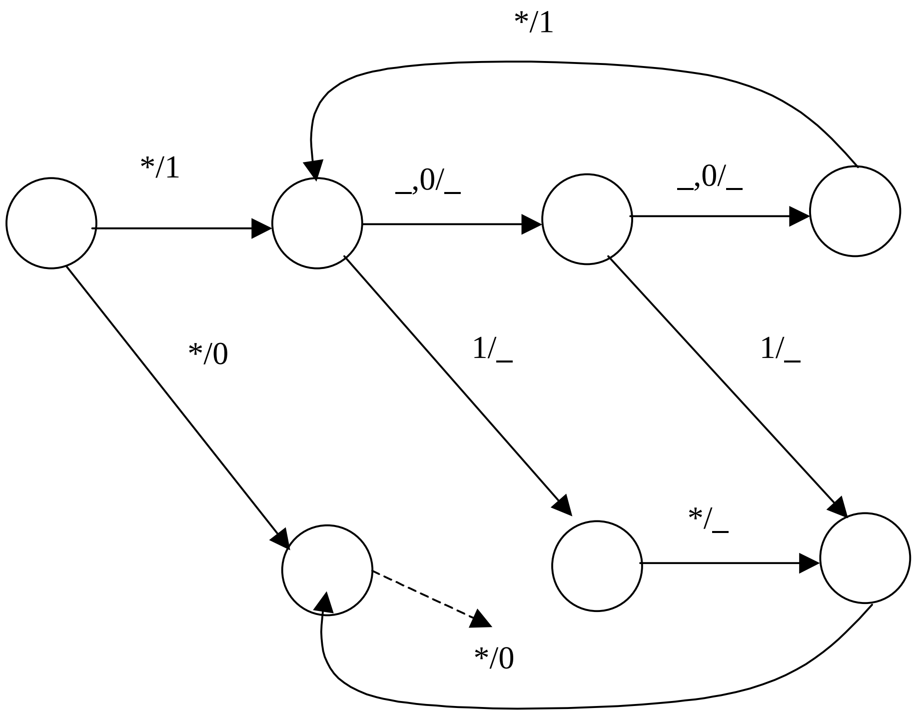
$\wedge$

$emette(s,t) \rightarrow \exists k \in N \wedge t = 3.k$

$)$

Esercizio 2

La figura sottostante descrive parzialmente un automa a stati finiti che si comporta secondo le specifiche. La parte omessa, cui si accede attraverso la freccia tratteggiata, tratta il caso dell'emissione iniziale di uno 0 in modo simmetrico.



Legenda: * indica un qualsiasi simbolo; _ indica assenza di segnale; a, b/ etichetta una transizione che avviene in conseguenza dll'input a o dell'input b.

Esercizio 3

1. Falsa. E' possibile ridurre il problema della terminazione al problema dato assegnando al risultato della funzione il valore 2 a valle del calcolo di una funzione generica $f_z(w)$di cui si voglia stabilire la terminazione.

   Si può sfruttare per la dimostrazione anche il teorema di Rice nel seguente modo: si fissi un certo $\underline{x}$ (qualunque, per esempio 7) e si definisca la funzione h(y) = (1 se $f_y(\underline{x})$ è pari, 0 altrimenti). h(y) è la funzione caratteristica dell'insieme delle macchine di Turing che calcolano funzioni che danno risultato pari quando hanno come input il numero $\underline{x}$. Questo insieme non è né l'insieme vuoto, né l'insieme universo, e non è dunque decidibile, quindi la sua funzione caratteristica non è calcolabile. Se g(x,y) fosse calcolabile, lo sarebbe anche h(y),il che porterebbe a una contraddizione.

   La dimostrazione può anche essere costruita in forma diretta usando la tecnica diagonale nel modo seguente:
   Si definisca h(x) = Se $f_x(x)$ non è pari allora 2, $\perp$ altrimenti. Se g fosse calcolabile lo sarebbe anche h. Quindi h dovrebbe essere = f_{x0} per qualche x0.Ma h(x0) non pari implicherebbe h(x0) = 2 e h(x0) = 2 implicherebbe h(x0) non pari.
2. Vera: basta fare il run di $f_y(x)$
3. Falsa grazie al teorema di Rice: infatti la g(y) è la funzione caratteristica dell'insieme di tutte (e sole) le macchine di Turing che computano funzioni totali con valore sempre pari. Questo insieme non è né l'insieme vuoto, né l'insieme universo, non è quindi decidibile, e la sua funzione caratteristica non è computabile.
4. Falsa. Infatti questa è la funzione semicaratteristica dell'insieme di tutte (e sole) le macchine di Turing che computano funzioni totali con valore sempre pari. Se tale funzione fosse calcolabile, ciò vorrebbe dire che tale insieme di funzioni è RE. Questo è tuttavia impossibile, Si prenda infatti una funzione calcolabile f(x) generica; si può facilmente calcolare la funzione f'(x) = 2f(x), la quale ha lo stesso dominio di f(x) ed ha, laddove è definita, valore pari. Di conseguenza, a partire dall'insieme di tutte le funzioni computabili a valori solo pari è possibile costruire l'insieme di tutte le funzioni computabili semplicemente dividendo per 2 le immagini. Da questo deriva che se l'insieme di tutte e sole le funzioni totali a valori pari fosse RE, anche l'insieme di tutte e sole le funzioni totali lo sarebbe. E' tuttavia ben noto (da lezione) che l'insieme di tutte e sole le funzioni totali non è RE, di conseguenza non lo è neanche l'insieme di funzioni di partenza.

Esercizio 4

La macchina M deve in primo luogo individuare la metà della stringa in ingresso. Per ottenere ciò, mediante una prima passata memorizza –ad esempio in unario- la lunghezza della stringa, alla sua destra (durante questa passata la macchina dovrà tenere traccia del carattere fino a dove si è contato, per esempio cambiandolo da a ad a', e da b a b', e riconvertendolo nel carattere originale prima di passare a contare il prossimo carattere). Ciò richiede un tempo $\Theta(n)$ per ogni carattere letto e quindi $\Theta(n^2)$ per l'intera stringa. Indi, per ogni coppia di caratteri della stringa che memorizza la lunghezza dell'input, sposta di una posizione un'opportuna marca all'interno della stringa di input (ad esempio sostituendo il carattere a con il carattere a' e b con b'). Al termine la marca si troverà a metà della stringa di ingresso. Anche questa macro-operazione richiede un tempo $\Theta(n^2)$ ($\Theta(n)$ per ogni coppia di caratteri).

A questo punto M può procedere a confrontare, uno per uno il primo carattere dell'input con il carattere in corrispondenza della marca che segna la metà; il secondo con il successivo. Anche questo confronto richiede un tempo $\Theta(n)$ per ogni carattere e quindi $\Theta(n^2)$ per l'intera stringa.

In conclusione l'intero procedimento ha una complessità temporale $\Theta(n^2)$ e spaziale $\Theta(n)$.