Esercizi di Informatica Teorica

Macchine di Turing

a cura di
Luca Cabibbo e Walter Didimo

Sommario

- macchine di Turing a nastro singolo
- macchine di Turing multinastro
- macchine di Turing trasduttrici
- macchine di Turing non deterministiche
- composizione di macchine di Turing

notazioni sul livello degli esercizi: (*) facile, (**) non difficile
(***) media complessità, (****) difficile, (*****) quasi impossibile

Macchina di Turing

macchina di Turing (MT) :

$M = <\Sigma, \underline{b}, K, q_0, F, \delta>$ dove

- Σ è l'alfabeto (finito) di simboli
- $\underline{b} \notin \Sigma$ è il carattere speciale "spazio bianco" (blank)
- K è un insieme finito e non vuoto di stati interni
- $q_0 \in K$ è lo stato iniziale
- $F \subseteq K$ è l'insieme degli stati finali
- $\delta : K \times (\Sigma \cup \{\underline{b}\}) \rightarrow K \times (\Sigma \cup \{\underline{b}\}) \times \{s, d, i\}$ è la funzione (parziale) di transizione;

$\delta(q,a) = <p, c, m>$ vuol dire che quando M è nello stato 'q' e la testina è posizionata sul simbolo 'a', M passa allo stato 'p', scrive il simbolo 'c' al posto di 'a' sul nastro, ed esegue uno spostamento 'm' della testina, dove 'm' può equivalere a fare un passo a sinistra (s), fare un passo a destra (d), o restare fermo (i).

Configurazioni e transizioni

- configurazione di una MT: contenuto del nastro + posizione della testina + stato corrente
- rappresentazione di una configurazione: $\alpha q \beta$

dove $\alpha \in (\Sigma \cup \{\underline{b}\})^*$ è la porzione di nastro a sinistra della testina, q è lo stato corrente, $\beta = a\beta' \in (\Sigma \cup \{\underline{b}\})^+$, dove 'a' è il simbolo su cui si trova la testina e β' è la porzione di nastro a destra della testina

- configurazione iniziale: $q_0\beta$ (oppure $\underline{b}q_0\beta$)
- configurazione finale: $\alpha q \beta$ con $q \in F$

- transizione (o passo o mossa): applicazione della funzione di transizione ad una configurazione ($c_i \vdash c_{i+1}$)

Computazioni

• computazione di una MT: sequenza (finita o infinita) di transizioni

$c_1 \vdash c_2 \vdash \ldots\ldots \vdash c_i \vdash \ldots\ldots$

• una computazione finita $c_1 \vdash c_2 \vdash \ldots\ldots \vdash c_n$ si indica anche al modo $c_1 \vdash^* c_n$, dove ($\vdash^*$ è la chiusura riflessiva e transitiva di $\vdash$)

• convenzione: in ogni computazione può esistere al più una configurazione finale (cioè se la macchina raggiunge uno stato finale la computazione termina)

• computazione (finita) massimale $c_1 \vdash^* c_n \Leftrightarrow$ non esiste una configurazione 'c' tale che $c_n \vdash c$

• computazione (finita) accettante $c_0 \vdash^* c_n \Leftrightarrow c_0$ è iniziale e c_n è finale

• computazione (massimale) rifiutante $c_0 \vdash^* c_n \Leftrightarrow c_0$ è iniziale e c_n non è finale

• computazione non terminante $\Leftrightarrow$ né accettante né rifiutante

Esercizi svolti sulle MT

Esercizio 1(**) sia M la seguente macchina di Turing:

$\Sigma=\{0,1,2\}$ $K=\{q_0, q_1, q_2, q_F\}$ $F=\{q_F\}$

$\delta(q_0,0) = \langle q_0,0,\rightarrow\rangle$ $\delta(q_0,2) = \langle q_1,2,\rightarrow\rangle$ $\delta(q_0,\underline{b}) = \langle q_F,\underline{b},\leftrightarrow\rangle$

$\delta(q_1,0) = \langle q_1,1,\rightarrow\rangle$ $\delta(q_1,1) = \langle q_1,0,\rightarrow\rangle$

$\delta(q_1,2) = \langle q_2,2,\leftarrow\rangle$ $\delta(q_1,\underline{b}) = \langle q_F,\underline{b},\leftrightarrow\rangle$

$\delta(q_2,0) = \langle q_2,0,\leftarrow\rangle$ $\delta(q_2,1) = \langle q_2,1,\leftarrow\rangle$

$\delta(q_2,2) = \langle q_1,2,\rightarrow\rangle$ $\delta(q_2,\underline{b}) = \langle q_F,\underline{b},\leftrightarrow\rangle$

(dove '$\rightarrow$', '$\leftarrow$', '$\leftrightarrow$' sono usati in luogo di 's', 'd' e 'i'); mostrare le computazioni sugli input "002101", "012101" e "00210212", specificando per ciascuna di esse se si tratta di una computazione accettante, rifiutante o non terminante

Esercizi svolti sulle MT

Soluzione

• computazione su “002101”: $q_0002101 \vdash 0q_002101 \vdash 00q_02101$ $\vdash 002q_1101 \vdash 0020q_101 \vdash 00201q_11 \vdash 002010q_1\underline{b} \vdash 002010q_F\underline{b}$

computazione accettante

• computazione su “012101”: $q_0012101 \vdash 0q_012101$

computazione rifiutante

• computazione su “00210212”: $q_000210212 \vdash 0q_00210212 \vdash$ $00q_0210212 \vdash \underline{002q_110212} \vdash 0020q_10212 \vdash 00201q_1212 \vdash$ $0020q_21212 \vdash 002q_201212 \vdash 00q_2201212 \vdash 002q_101212 \vdash$ $0021q_11212 \vdash 00210q_1212 \vdash 0021q_20212 \vdash 002q_210212 \vdash$ $00q_2210212 \vdash \underline{002q_110212}$ ...... (cicla all'infinito)

computazione non terminante

Macchina di Turing e linguaggi

• una MT (con alfabeto Σ) riconosce (decide) un linguaggio $L \subseteq \Sigma^* \Leftrightarrow$
  - $\forall x \in L$ la computazione su x è accettante
  - $\forall x \notin L$ la computazione su x è rifiutante

osservazione: ciò vuol dire che $\forall x \in \Sigma^*$ la MT è in grado di decidere se $x \in L$ (x accettata) o se $x \notin L$ (x rifiutata)

• una MT (con alfabeto Σ) accetta un linguaggio $L \subseteq \Sigma^* \Leftrightarrow$
  - $\forall x \in L$ la computazione su x è accettante
  - $\forall x \notin L$ la computazione su x è o rifiutante o non terminante

osservazione: ciò vuol dire che $\forall x \in \Sigma^*$ la MT è in grado di stabilire se $x \in L$ (x accettata), mentre non garantisce alcun comportamento nel caso in cui $x \notin L$

nota bene: la MT riconosce L $\Rightarrow$ la MT accetta L (non viceversa)

Esercizi svolti sulle MT

Esercizio 2(***) definire una MT che riconosce il linguaggio

$L = \{0^n 1^n 2^n : n > 0\}$

Soluzione

• strategia di riconoscimento:

- una stringa x di L è del tipo 00...011...122...2 e la configurazione iniziale della MT è $q_0$00...011...122...2;
- al primo passo effettuo nell'ordine le seguenti operazioni:

  (a) cerco il primo 0 muovendomi a destra e lo rimpiazzo con una Z

  (b) cerco il primo 1 muovendomi a destra e lo rimpiazzo con una U

  (c) cerco il primo 2 muovendomi a destra e lo rimpiazzo con una D

  (se qualche ricerca tra le (a), (b) e (c) fallisce allora la computazione sarà rifiutante)

  (d) mi riposiziono a destra della prima Z che trovo muovendomi a sinistra

Esercizi svolti sulle MT

- al generico passo cerco il primo 0 muovendomi a destra:
  - se trovo lo 0 allora lo rimpiazzo con una Z ed eseguo i passi (b), (c) e (d); se durante i passi (b) e (c) non trovo un 1 o un 2, allora la computazione sarà rifiutante;
  - se non trovo lo 0 allora verifico che non ci siano più 1 e 2 a destra; se la verifica va a buon fine allora la computazione sarà accettante, altrimenti sarà rifiutante

• definizione dei simboli e degli stati:

$\Sigma=\{0,1,2,Z,U,D\}$, $K = \{q_0, q_1, q_2, q_R, q_V, q_F\}$, $F = \{q_F\}$

$\{q_0, q_1, q_2\}$ = ricerca di uno 0, di un 1 e di un 2 verso destra

q_R = riposizionamento a destra della prima Z che incontro andando da destra verso sinistra; q_V = verifica che non ci siano più 1 e 2 a destra

Esercizi svolti sulle MT

• la funzione di transizione

$\delta(q_0,0) = <q_1,Z,\rightarrow>$ $\delta(q_0,U) = <q_V,U,\rightarrow>$ non ci sono più 0

$\delta(q_1,0) = <q_1,0,\rightarrow>$ $\delta(q_1,U) = <q_1,U,\rightarrow>$

$\delta(q_1,1) = <q_2,U,\rightarrow>$

$\delta(q_2,1) = <q_2,1,\rightarrow>$ $\delta(q_2,D) = <q_2,D,\rightarrow>$

$\delta(q_2,2) = <q_R,D,\leftarrow>$

$\delta(q_R,0) = <q_R,0,\leftarrow>$ $\delta(q_R,1) = <q_R,1,\leftarrow>$

$\delta(q_R,U) = <q_R,U,\leftarrow>$ $\delta(q_R,D) = <q_R,D,\leftarrow>$

$\delta(q_R,Z) = <q_0,Z,\rightarrow>$ ricomincia il conteggio

$\delta(q_V,U) = <q_V,U,\rightarrow>$ $\delta(q_V,D) = <q_V,D,\rightarrow>$

$\delta(q_V,\underline{b}) = <q_F,\underline{b}, \leftrightarrow>$

Esercizi svolti sulle MT

Esercizio 3(***) definire una MT che riconosce il linguaggio delle stringhe su {0,1,2} tali che #0 = #1 = #2

Soluzione

• strategia di riconoscimento: la strategia è simile a quella dell'Esercizio 2, ma stavolta, poiché l'ordine dei simboli nella stringa può essere qualunque, occorre riposizionare la testina all'inizio della stringa prima di ogni ricerca di simbolo:

  • cerca uno 0, marcalo e torna all'inizio della stringa
  • cerca un 1, marcalo e torna all'inizio della stringa
  • cerca un 2, marcalo e torna all'inizio della stringa
  • cerca uno 0, ...

• definizione dei simboli e degli stati: come per l'Esercizio 2, ma bisogna aggiungere uno stato di riavvolgimento per ogni simbolo da ricercare

Esercizi svolti sulle MT

- la funzione di transizione

$\delta(q_0,0) = <q_{0R},Z, \leftarrow>$ $\delta(q_0,\underline{b}) = <q_V,\underline{b}, \leftarrow>$

$\delta(q_0,\alpha) = <q_0,\alpha,\rightarrow>$ $\forall\alpha\in \Sigma - \{0\}$

$\delta(q_{0R},\underline{b}) = <q_1,\underline{b}, \rightarrow>$ $\delta(q_{0R},\beta) = <q_{0R},\beta,\leftarrow>$ $\forall \beta \in \Sigma$

$\delta(q_1,1) = <q_{1R},U, \leftarrow>$ $\delta(q_1,\alpha) = <q_1,\alpha,\rightarrow>$ $\forall\alpha\in \Sigma - \{1\}$

$\delta(q_{1R},\underline{b}) = <q_2,\underline{b}, \rightarrow>$ $\delta(q_{1R},\beta) = <q_{1R},\beta,\leftarrow>$ $\forall \beta \in \Sigma$

$\delta(q_2,2) = <q_{2R},D, \leftarrow>$ $\delta(q_2,\alpha) = <q_2,\alpha,\rightarrow>$ $\forall\alpha\in \Sigma - \{2\}$

$\delta(q_{2R},\underline{b}) = <q_0,\underline{b}, \rightarrow>$ $\delta(q_{2R},\beta) = <q_{2R},\beta,\leftarrow>$ $\forall \beta \in \Sigma$

$\delta(q_V,Z) = <q_V,Z, \leftarrow>$ $\delta(q_V,U) = <q_V,U, \leftarrow>$

$\delta(q_V,D) = <q_V,D, \leftarrow>$ $\delta(q_V,\underline{b}) = <q_F,\underline{b}, \leftrightarrow>$

Macchina di Turing multinastro

macchina di Turing multinastro (MTM) :

sia n = numero di nastri

$M = <\Sigma, \underline{b}, K, q_0, F, \delta>$ dove

- $\Sigma, \underline{b}, K, q_0$ ed F sono definiti come per una MT
- $\delta : K \times (\Sigma \cup \{\underline{b}\})^n \rightarrow K \times (\Sigma \cup \{\underline{b}\})^n \times \{s, d, i\}^n$

$\delta(q, a_1, a_2 ,\dots,a_n) = <p, c_1, c_2 ,\dots, c_n , m_1, m_2 ,\dots, m_n>$ vuol dire che quando M è nello stato ‘q’, e le n testine sono posizionate sui simboli $a_1, a_2 ,\dots,a_n$, M passa allo stato ‘p’, scrive i simboli $c_1, c_2 ,\dots,c_n$ rispettivamente al posto di $a_1, a_2 ,\dots,a_n$, ed esegue gli spostamenti $m_1, m_2 ,\dots,m_n$ delle n testine sugli n nastri

Configurazioni e transizioni

• configurazione di una MTM: $q\#\alpha_1\uparrow\beta_1\#\alpha_2\uparrow\beta_2\#\ldots\#\alpha_n\uparrow\beta_n$
dove q è lo stato corrente, il primo carattere di β_i è quello su cui si trova la testina dell'i-esimo nastro, ed α_i può anche essere vuota

• classe rappresentativa delle MTM con n nastri:
  • il primo nastro è di input (sola lettura)
  • gli altri n-1 nastri sono di lavoro (scrittura e lettura)
  • configurazione iniziale: $q_0\#\uparrow\beta_1\#\uparrow Z_0\#\ldots\#\uparrow Z_0$
  dove β_1 è la stringa sul nastro di input, e Z_0 è l'unico simbolo che si trova inizialmente sugli altri nastri
  • configurazione finale: $q\#\alpha_1\uparrow\beta_1\#\alpha_2\uparrow\beta_2\#\ldots\#\alpha_n\uparrow\beta_n$ con $q\in F$

• le transizioni e le computazioni sono analoghe al caso di MT

Esercizi svolti sulle MTM

Esercizio 4(***) definire una MTM che riconosce il linguaggio delle stringhe su {0,1,2} tali che #0 = #1 = #2

Soluzione

• strategia di riconoscimento: consideriamo una MTM con un nastro di input (sola lettura) monodirezionale (scorrimento sempre a destra dopo ogni transizione):
  • si scandisce la stringa sul nastro di input e si copiano gli 0 sul primo nastro di lavoro, gli 1 sul secondo ed i 2 sul terzo
  • si verifica che il numero di 0, 1 e 2 sui tre nastri lavoro sia lo stesso

• definizione dei simboli e degli stati:
$\Sigma = \{0,1,2, Z_0\}$, $K=\{q_0, q_V, q_F\}$, $F = \{q_F\}$
q_0 = copia gli 0, 1 e 2 sui tre nastri
q_V = verifica che i tre nastri abbiano lo stesso numero di simboli

Esercizi svolti sulle MTM

• la funzione di transizione

poiché il nastro di input è in sola lettura e monodirezionale, si semplifica la funzione di transizione, evitando di scrivere cosa avviene sul nastro di input

$\delta(q_0,0, Z_0, Z_0, Z_0) = <q_0,0, \underline{b}, \underline{b}, \rightarrow, \leftrightarrow, \leftrightarrow>$

$\delta(q_0,1, Z_0, Z_0, Z_0) = <q_0,\underline{b}, 1, \underline{b}, \leftrightarrow, \rightarrow, \leftrightarrow>$

$\delta(q_0,2, Z_0, Z_0, Z_0) = <q_0,\underline{b}, \underline{b}, 2, \leftrightarrow, \leftrightarrow, \rightarrow>$

copia al primo passo

$\delta(q_0,0, \underline{b}, \underline{b}, \underline{b}) = <q_0,0, \underline{b}, \underline{b}, \rightarrow, \leftrightarrow, \leftrightarrow>$

$\delta(q_0,1, \underline{b}, \underline{b}, \underline{b}) = <q_0,\underline{b}, 1, \underline{b}, \leftrightarrow, \rightarrow, \leftrightarrow>$

$\delta(q_0,2, \underline{b}, \underline{b}, \underline{b}) = <q_0,\underline{b}, \underline{b}, 2, \leftrightarrow, \leftrightarrow, \rightarrow>$

copia a regime

$\delta(q_0,\underline{b}, \underline{b}, \underline{b}, \underline{b}) = <q_V,\underline{b}, \underline{b}, \underline{b}, \leftarrow, \leftarrow, \leftarrow>$

$\delta(q_V,\underline{b}, 0, 1, 2) = <q_V,0, 1, 2, \leftarrow, \leftarrow, \leftarrow>$

$\delta(q_V,\underline{b}, \underline{b}, \underline{b}, \underline{b}) = <q_F,\underline{b}, \underline{b}, \underline{b}, \leftrightarrow, \leftrightarrow, \leftrightarrow>$

verifica

Esercizi svolti sulle MTM

Esercizio 5(****) definire una macchina di Turing che riconosce il linguaggio delle stringhe su {x, y} tali che il numero di ‘x’ è una potenza di due, più la stringa vuota.

Soluzione

• strategia di riconoscimento:

- uso una MTM con 2 nastri, uno di input ed uno di lavoro;
- scandisco tutta la stringa sul nastro di input, conto il numero di ‘x’ e memorizzo il risultato del conteggio sul nastro di lavoro (utilizzo il nastro di lavoro anche per memorizzare i conteggi parziali);
- leggo la stringa (risultato del conteggio) sul nastro di lavoro e verifico che sia una potenza di due.

Esercizi svolti sulle MTM

osservazione: conviene contare in notazione binaria, perché è facile poi verificare se il numero è una potenza di due

algoritmo per contare in binario: dal decimale n (in notazione binaria) al decimale n+1 (in notazione binaria)

- posizionarsi sulla cifra all'estrema destra del numero
- se la cifra su cui ci si trova è un 1, trasformarla in uno 0 e muoversi a sinistra di un passo
- iterare il processo di sostituzione di 1 in 0 con spostamento a sinistra fino a quando una tra le due condizioni seguenti è verificata:
  - si incontra uno 0 ⇒ trasformarlo in 1 e terminare
  - sono finite le cifre ⇒ aggiungere un 1 a sinistra (che diviene la prima cifra del numero incrementato)

Esercizi svolti sulle MTM

esempi di conteggio con l'algoritmo proposto:

- dal numero 23 (10111) al numero 24 (?)

10111 ⇒ 10110 ⇒ 10100 ⇒ 11000 (24)

- dal numero 31 (11111) al numero 32 (?)

11111 ⇒ 11110 ⇒ 11100 ⇒ 11000 ⇒ 10000

⇒ 10000 ⇒ 100000 (32)

Esercizi svolti sulle MTM

- definizione dei simboli e degli stati:

$\Sigma = \{0, 1, Z_0\}$, $K=\{q_0, q_1, q_R, q_V, q_{V1}, q_F\}$, $F = \{q_F\}$

q_0 = scansione della stringa di input

q_1 = incremento del numero di x sul nastro di lavoro

q_R = riposizionamento alla fine del numero sul nastro lavoro

q_V = verifica che il numero sul nastro lavoro sia una potenza di due

q_{V1} = stato di supporto alla verifica (trovato un 1 verifica che non ci siano più cifre a sinistra)

q_F = accettazione (verifica andata a buon fine)

osservazione: anche la computazione su una stringa di input senza x deve terminare nello stato di accettazione

Esercizi svolti sulle MTM

- la macchina di Turing multinastro

$\delta(q_0,y,Z_0) = <q_0,y,Z_0,\rightarrow,\leftrightarrow>$	$\delta(q_0,y,\underline{b}) = <q_0,y,\underline{b},\rightarrow,\leftrightarrow>$
$\delta(q_0,x,Z_0) = <q_0,x,1,\rightarrow,\rightarrow>$	(inizializza il nastro lavoro con un 1)
$\delta(q_0,x,\underline{b}) = <q_1,x,\underline{b},\leftrightarrow,\leftarrow>$	(stato di incremento sul nastro lavoro)
$\delta(q_0,\underline{b},\underline{b}) = <q_V,\underline{b},\underline{b},\leftrightarrow,\leftarrow>$	(stato di verifica del numero di x contate)
$\delta(q_0,\underline{b},Z_0) = <q_F,\underline{b},Z_0,\leftrightarrow,\leftrightarrow>$	(caso di sole y, cioè zero x)
$\delta(q_1,x,1) = <q_1,x,0,\leftrightarrow,\leftarrow>$	(trasforma gli 1 in 0 e va a sinistra)
$\delta(q_1,x,0) = <q_R,x,1,\leftrightarrow,\rightarrow>$	(fine increm. e ritorno a fine numero)
$\delta(q_1,x,\underline{b}) = <q_R,x,1,\leftrightarrow,\rightarrow>$	(fine increm. e ritorno a fine numero)
$\delta(q_R,x,0) = <q_R,x,0,\leftrightarrow,\rightarrow>$	(scorrimento per ritorno a fine nastro)
$\delta(q_R,x,1) = <q_R,x,1,\leftrightarrow,\rightarrow>$	(scorrimento per ritorno a fine nastro)
$\delta(q_R,x,\underline{b}) = <q_0,x,\underline{b},\rightarrow,\leftrightarrow>$	(ricomincia a scandire il nastro di input)
$\delta(q_V,\underline{b},0) = <q_V,\underline{b},0,\leftrightarrow,\leftarrow>$	(scorrimento per verifica)
$\delta(q_V,\underline{b},1) = <q_{V1},\underline{b},1,\leftrightarrow,\leftarrow>$	$\delta(q_{V1},\underline{b},\underline{b}) = <q_F,\underline{b},\underline{b},\leftrightarrow,\leftrightarrow>$

Macchina di Turing trasduttrici

macchina di Turing trasduttrice:

- serve per calcolare una funzione (parziale) f
- la configurazione iniziale ha la forma: q_0x
- la configurazione finale ha la forma: $xq_Ff(x)$

osservazioni:

- si può pensare a convenzioni diverse per le configurazioni iniziali e finali
- i valori di x per cui la macchina di Turing non termina o termina in una configurazione non finale sono quelli per i quali la funzione f non è definita
- si può pensare ad MTM trasduttrici con un nastro di input (per memorizzare x), uno di output (per memorizzare f(x)) e *k* nastri di lavoro

Esercizi svolti sulle MT trasduttrici

Esercizio 6(***) definire una MTM trasduttrice che calcola la funzione prodotto di due interi positivi in notazione unaria, secondo le seguenti convenzioni:

- sul nastro di input è memorizzata la stringa: 1^n b 1^m, dove le due sequenze (non vuote) di 1 rappresentano i numeri da moltiplicare in notazione unaria
- sul nastro di output verrà memorizzata la stringa: 1^{nm}

...bbb1111b111bbb....... nastro di input

...bbb111111111111bbb....... nastro di output

Esercizi svolti sulle MT trasduttrici

Soluzione

• strategia di calcolo

  • utilizzo un nastro di lavoro su cui copio la stringa 1^n

  • per ogni 1 della stringa 1^m copio (accodandolo) tutto il contenuto del nastro lavoro sul nastro di output

• definizione dei simboli e degli stati:

$\Sigma = \{1, Z_0\}$, $K=\{q_0, q_1, q_R, q_V, q_{V1}, q_F\}$, $F = \{q_F\}$

q_0 = copia della stringa 1^n sul nastro di lavoro

q_1 = scansione della stringa 1^m dal nastro di input

q_C = copia del nastro di lavoro sul nastro di output

q_R = riposizionamento sul nastro di lavoro

q_F = stato di fine computazione

Esercizi svolti sulle MT trasduttrici

• la funzione di transizione

$\delta(q_0,1,Z_0,Z_0) = \langle q_0,1,1,Z_0,\rightarrow,\rightarrow,\leftrightarrow\rangle$ (copia di 1^n, primo passo)

$\delta(q_0,1,\underline{b},Z_0) = \langle q_0,1,1,Z_0,\rightarrow,\rightarrow,\leftrightarrow\rangle$ (copia di 1^n, a regime)

$\delta(q_0,\underline{b},\underline{b},Z_0) = \langle q_1,\underline{b},\underline{b},\underline{b},\rightarrow,\leftrightarrow,\leftrightarrow\rangle$ (inizio scansione di 1^m)

$\delta(q_1,1,\underline{b},\underline{b}) = \langle q_C,1,\underline{b},\underline{b},\leftrightarrow,\leftarrow,\leftrightarrow\rangle$ (inizio copia di 1^n su output)

$\delta(q_1,\underline{b},\underline{b},\underline{b}) = \langle q_F,\underline{b},\underline{b},\underline{b},\leftrightarrow,\leftrightarrow,\leftrightarrow\rangle$ (fine calcolo)

$\delta(q_C,1,1,\underline{b}) = \langle q_C,1,1,1,\leftrightarrow,\leftarrow,\rightarrow\rangle$ (copia di 1^n su output)

$\delta(q_C,1,\underline{b},\underline{b}) = \langle q_R,1,\underline{b},\underline{b},\leftrightarrow,\rightarrow,\leftrightarrow\rangle$ (inizio riposiz. su nastro lavoro)

$\delta(q_R,1,1,\underline{b}) = \langle q_R,1,1,\underline{b},\leftrightarrow,\rightarrow,\leftrightarrow\rangle$ (riposiz. su nastro lavoro)

$\delta(q_R,1,\underline{b},\underline{b}) = \langle q_1,1,\underline{b},\underline{b},\rightarrow,\leftrightarrow,\leftrightarrow\rangle$ (ripresa della scansione di 1^m)

Macchina di Turing non deterministica

macchina di Turing non deterministica (MTND) :

$M = <\Sigma, \underline{b}, K, q_0, F, \delta_N>$ dove

- $\Sigma, \underline{b}, K, q_0, F$ sono definiti come per le MT
- $\delta_N : K \times (\Sigma \cup \{\underline{b}\}) \rightarrow \mathbf{P}(K \times (\Sigma \cup \{\underline{b}\}) \times \{s, d, i\})$ è la funzione (parziale) di transizione;

considerazioni:

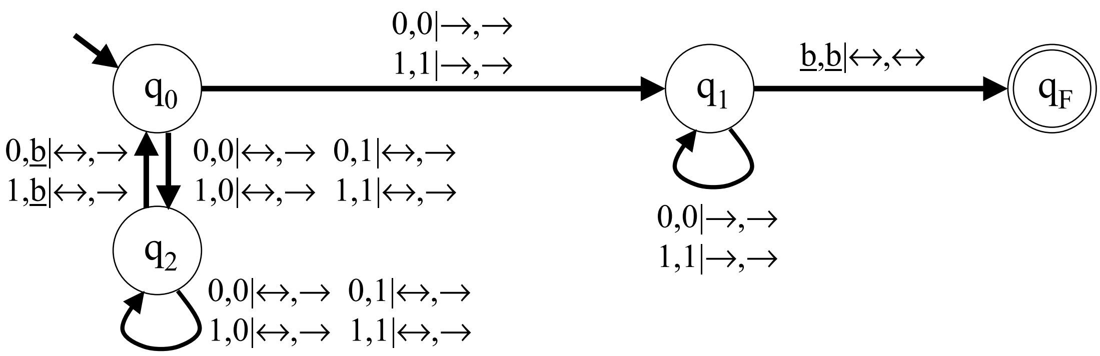
- per un dato input x, M esegue un albero di computazioni
- M accetta x $\Leftrightarrow$ esiste una computazione dell'albero che è accettante
- M rifiuta x $\Leftrightarrow$ ci sono nell'albero solo computazioni rifiutanti
- una MTND può solo essere utilizzata come riconoscitore, non come trasduttore

Esercizi svolti sulle MTND

Esercizio 7(***) si consideri una MTND M con due nastri di sola lettura, così configurati:

- primo nastro: ...$\underline{bb}\alpha\underline{bb}$..., con $\alpha \in \{0,1\}^+$
- secondo nastro: ... $\underline{bb}\alpha_1\underline{b}\alpha_2....\underline{b}\alpha_n$ $\underline{bb}$... con $\alpha_i \in \{0,1\}^+$

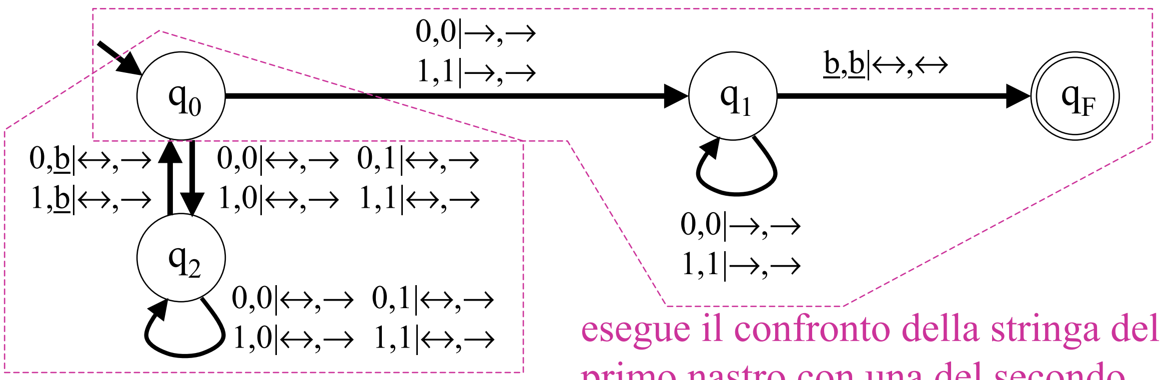
nella configurazione iniziale, M ha le testine posizionate all'inizio rispettivamente di α ed α_1; dire cosa fa M, sapendo che la sua funzione di transizione è definita dal seguente diagramma



Esercizi svolti sulle MTND

Soluzione

M effettua il confronto della stringa α con le stringhe $\alpha_1, \ldots, \alpha_n$, e termina nello stato finale (cioè accetta l'input) se e solo se α coincide con almeno una delle stringhe $\alpha_1, \ldots, \alpha_n$; M effettua dunque la ricerca di una stringa in una lista di stringhe date

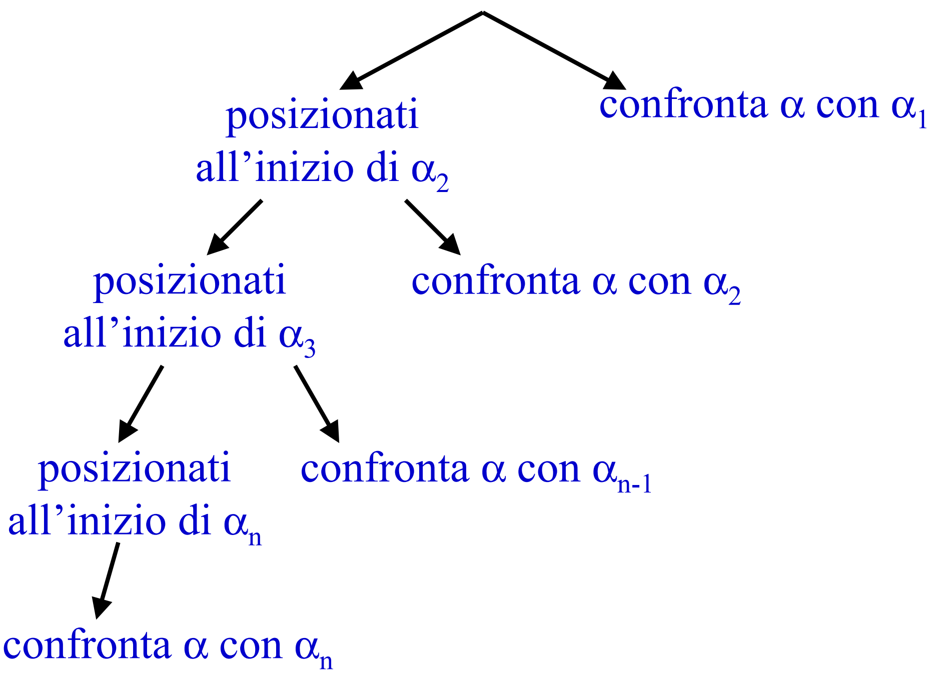


esegue il confronto della stringa del primo nastro con una del secondo

si posiziona all'inizio di una qualunque stringa del secondo nastro

Esercizi svolti sulle MTND

struttura ad alto livello dell'albero delle computazioni



Esercizi svolti sulle MTND

Esercizio 8(****) definire una MTND che riconosce il linguaggio
$L = \{ww : w \in \{0,1\}^+\}$ (*suggerimento*: utilizzare un nastro di input, su cui è scritta la stringa iniziale, ed un nastro di lavoro)

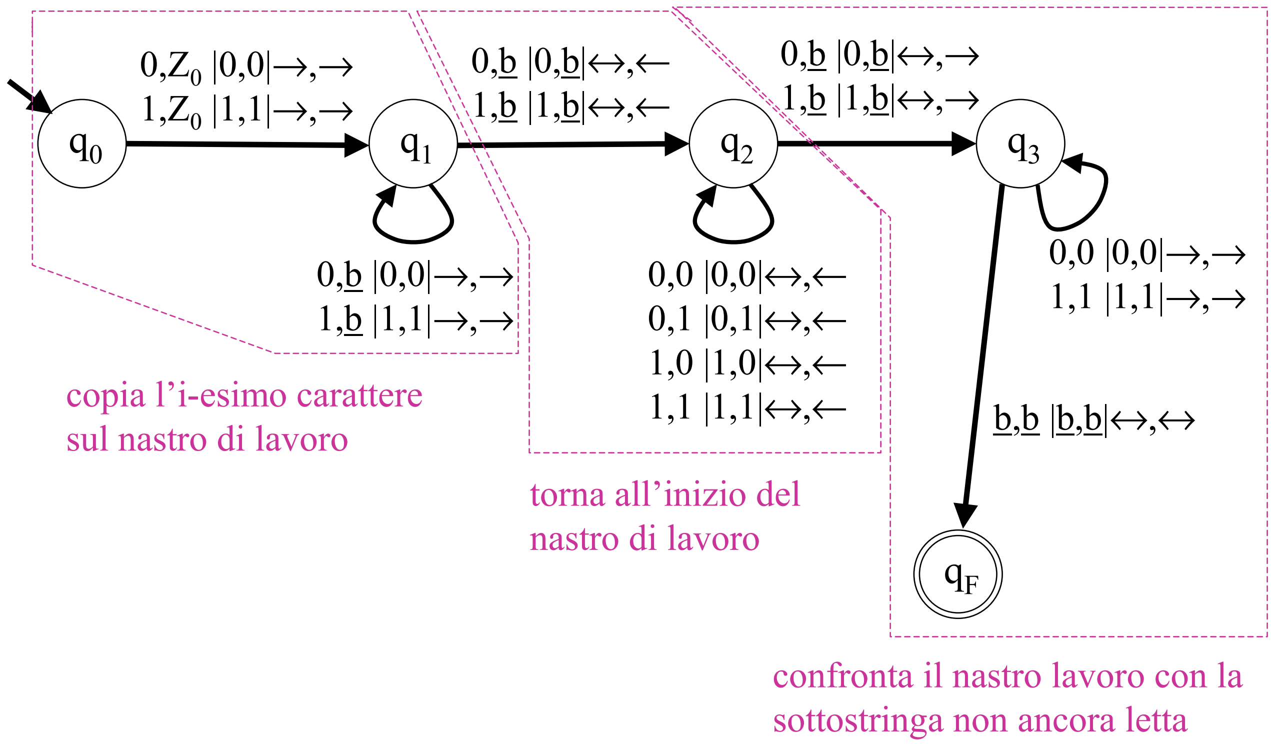
Soluzione

• strategia di riconoscimento

all'i-esimo passo si effettuano non deterministicamente due possibili operazioni:

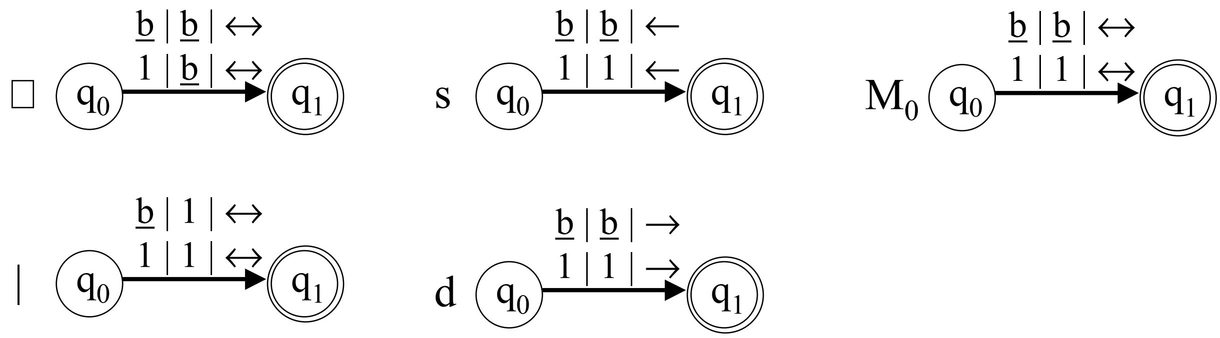
- si copia l'i-esimo carattere della stringa di input sul nastro di lavoro
- si confronta la stringa di input dall'i-esimo carattere in poi con la sottostringa già copiata sul nastro di lavoro

Esercizi svolti sulle MTND



Composizione di MT

- ipotesi non restrittiva: MT con alfabeto $\Sigma=\{1\}$
- MT elementari:



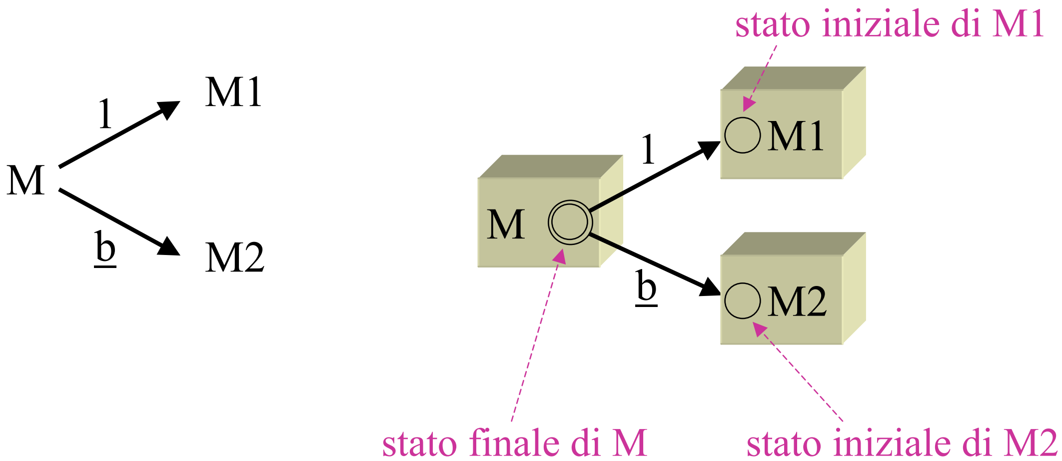
□ = scrive un b
| = scrive un 1

s = mossa a sinistra
d = mossa a destra

M_0 = ferma

Composizione di MT

- ogni MT può essere definita per composizione di MT elementari, usando diramazioni condizionate



se la testina di M si ferma su un 1 allora parte M1, se si ferma su un b parte M2

Esercizi svolti su composizione di MT

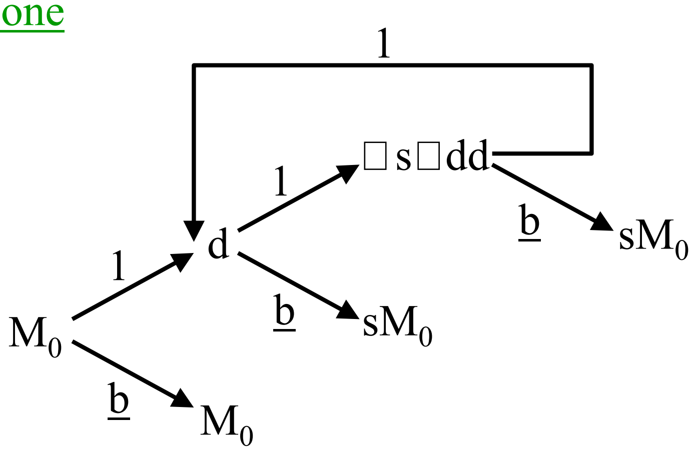
Esercizio 9(***) definire una MT M per composizione di MT elementari tale che:

- M ha un solo nastro di input/output ed alfabeto {1}
- sul nastro è scritto un numero naturale *n* in notazione unaria (input)
- la testina di M si trova inizialmente sulla prima cifra di *n*
- M calcola e scrive sul nastro il resto *r* (in notazione unaria) della divisione di *n* per 2
- al termine della computazione, sul nastro deve esserci solo *r* e la testina di M deve essere sul resto (che ha ovviamente una sola cifra)

↓ ...bbb1111bbb...	↓ ...bbb1bbb...
conf. iniziale	conf. finale

Esercizi svolti su composizione di MT

Soluzione



↓ ...bbb111bbb...	↓ ...bbb111bbb...	↓ ...bbb1b1bbb...	↓ ...bbb1b1bbb...
↓ ...bbbbb1bbb...	↓ ...bbbbb1bbb...	↓ ...bbbbb1bbb...	↓ ...bbbbb1bbb...

Esercizi svolti su composizione di MT

Esercizio 10(****) definire una MT M per composizione di MT elementari tale che:

- M ha un solo nastro di lettura/scrittura ed alfabeto {1}
- sul nastro è scritto un solo 1
- la testina di M si trova inizialmente in un punto qualunque del nastro
- M deve cercare l'uno sul nastro e terminare la computazione con la testina posizionata su tale simbolo

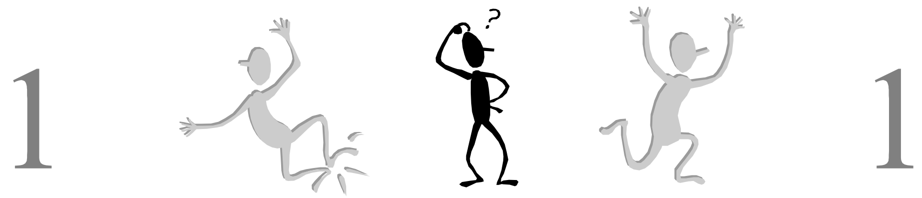
...bbbbbbbbb1bbb...
conf. iniziale

...bbbbbbbbb1bbb...
conf. finale

Esercizi svolti su composizione di MT

Soluzione

- il problema è.... capire se l'uno si trova a destra o a sinistra



- non si può procedere sempre in una direzione scelta a caso, perché si rischia di non terminare la computazione
- occorre procedere a zig-zag (faccio un passo a sinistra poi due a destra, poi tre a sinistra, poi quattro a destra....)

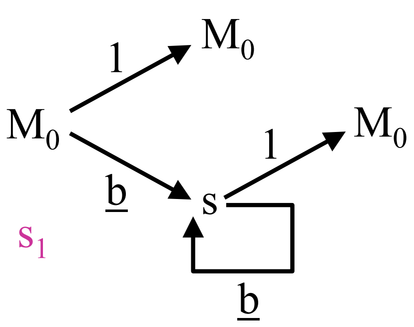
Esercizi svolti su composizione di MT

• non si può utilizzare un altro nastro lavoro per contare i passi fatti fino ad ora in una qualunque direzione, quindi occorre marcare l'ultima posizione raggiunta in ogni direzione

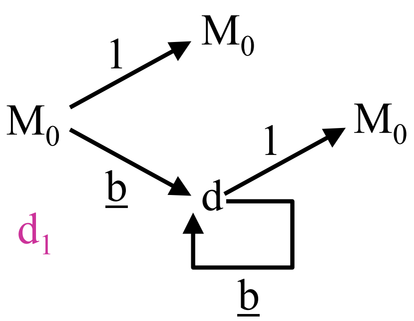
...bbbbbbb1bbb...	...bbbb1bb1bbb...	...bbb11bb1bbb...
...bbb1bbb1bbb...	...bbb1b1b1bbb...	...bbb1b1b1bbb...
...bbbbb1b1bbb...	...bb1bb1b1bbb...	...bb1bb1b1bbb...
...bb1bbbb1bbb...	...bb1bbb11bbb...	...bb1bbb11bbb...
...bbbbbb11bbb...	...b1bbbb11bbb...	...b1bbbbb1bbb...

Esercizi svolti su composizione di MT

definiamo prima le due seguenti MT

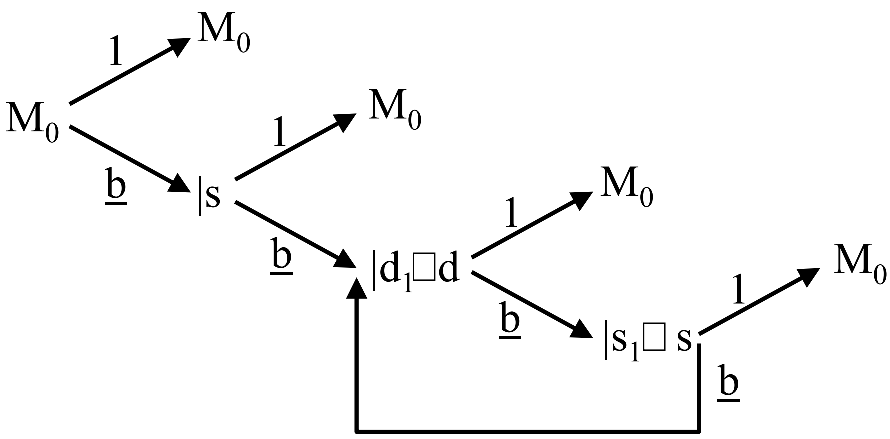


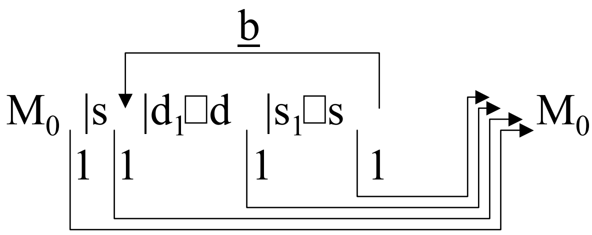
s_1= MT che si sposta a sinistra fino a quando non incontra un 1



d_1= MT che si sposta a sinistra fino a quando non incontra un 1

Esercizi svolti su composizione di MT





descrizione linearizzata