Esercizi: Macchine di Turing

1 Problemi

Problema 2.1: Sia L l'insieme delle stringhe $s = \langle x_1 x_2 \dots x_n \rangle$ di lunghezza pari e tali che:

- $x_i \in \{a,b\}$, per $i = 1, \dots, n/2$;
- $x_i \in \{c,d\}$, per $i = n/2+1, \dots, n$;
- $x_i = a \Leftrightarrow x_{n-i+1} = c$, per $i = 1, \dots, n/2$;
- $x_i = b \Leftrightarrow x_{n-i+1} = d$, per $i = 1, \dots, n/2$.

Esempio. Le stringhe *abacdc* e *aabbddcc* appartengono a L, mentre le stringhe *abadc* e *aabbadddcc* non appartengono a L.

Definire una macchina di Turing deterministica che accetti tutte e sole le parole contenute in L.

Problema 2.2: Sia k un valore costante (ad esempio, $k = 5$). Scegliere opportunamente un modello di macchina di Turing e progettare una macchina rispondente alle caratteristiche di tale modello che, ricevendo in input k parole binarie $p_1 = x_{11}x_{12}\dots x_{1n}$, $p_2 = x_{21}x_{22}\dots x_{2n}$, ... , $p_k = x_{k1}x_{k2}\dots x_{kn}$, tutte aventi la stessa lunghezza n (non costante), esegue su tali parole il *controllo di parità orizzontale e verticale*, ossia, calcola le due parole o e v definite come segue:

- $o = o_1 o_2 \dots o_k$, dove, per ogni $i = 1, \dots, k$ $o_i = 1$ se p_i contiene un numero dispari di 1, $o_i = 0$ altrimenti (controllo orizzontale);
- $v = v_1 v_2 \dots v_n$, dove, per ogni $i = 1, \dots, n$ $v_i = 1$ se la parola $x_{1i}x_{2i}\dots x_{ki}$ contiene un numero dispari di 1, $v_i = 0$ altrimenti (controllo verticale).

Problema 2.3: Progettare una macchina di Turing che calcoli le due funzioni $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ e $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ di seguito descritte:

$$f(n,k) = \lceil \frac{n}{k} \rceil; \qquad g(n,k) = \lfloor \frac{n}{k} \rfloor \qquad \text{con } k > 0.$$

Problema 2.4: Siano $\Sigma = \{a, b\}$ e $L = \{a^n b^{2n} : n \in \mathbb{N}\}$ ove, ricordiamo, a^n è la parola costituita dalla concatenazione di n caratteri a. Progettare una macchina di Turing che accetta tutte e sole le parole contenute in L.

Problema 2.5: Sia T_Σ una macchina di Turing definita sull'alfabeto $\Sigma = \{a, b, c, d\}$. Derivare da T_Σ una macchina T_B, equivalente a T_Σ, che opera sull'alfabeto $B = \{0, 1\}$.

Nota: non viene richiesta la prova formale di equivalenza delle due macchine, è sufficiente limitarsi ad osservazioni intuitive.

Problema 2.6: Sia $k \in \mathbb{N}$ una costante, e sia NT_k una macchina di Turing non deterministica con grado di non determinismo pari a k. Definire una macchina di Turing non deterministica NT_2 con grado di non determinismo pari a 2 che sia equivalente a NT_k.

Problema 2.7: Sia T una macchina di Turing di tipo riconoscitore, ad un nastro, definita sull'alfabeto $\{0, 1\}$. Definire una nuova macchina T_0, a due nastri e definita su un alfabeto opportunamente introdotto, che utilizza i soli stati interni q_0, q_A e q_R e che è equivalente a T.

Suggerimento: utilizzare il secondo nastro di T_0 per memorizzare lo stato interno in cui si trova T.

Problema 2.8: Sia $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ e sia $x = x_1 x_2 \ldots x_n$ una parola in Σ^*.
Si consideri una caccia al tesoro in cui il tesoro, rappresentato dal numero intero 0, può esistere o meno e in cui la parola x contiene la catena di indizi che portano a scoprire il tesoro, se esiste, o a concludere che il tesoro non esiste nel caso contrario. In particolare

- il primo carattere x_1 di x (un intero compreso fra 0 e 9 oppure un $\square$) è il primo indizio: se $x_1 = 0$ allora il tesoro è stato trovato, se $x_1 = \square$ allora il tesoro non esiste, altrimenti il prossimo indizio della caccia al tesoro è nella posizione $1 + x_1$ della parola x (ossia, il prossimo indizio è il carattere x_{1+x_1});
- in generale, se dopo un certo numero di passi non è ancora stato trovato il tesoro e non si è capito che esso non esiste, e, dunque, si è arrivati a leggere il carattere x_i, allora: se $x_i = 0$ allora il tesoro è stato trovato, se $x_i = \square$ allora il tesoro non esiste, altrimenti il prossimo indizio della caccia al tesoro è nella posizione $i + x_i$ della parola x (ossia, il prossimo indizio è il carattere x_{i+x_i}).

Si chiede, dunque, di progettare una macchina di Turing che, con input $x \in \Sigma^*$, decide se, in accordo alle regole appena descritte, x contiene il tesoro.

Problema 2.9: Sia $\Sigma = \{a, b, c, d\}$. Si consideri la macchina di Turing T definita sull'alfabeto Σ descritta dal seguente insieme di quintuple:

$$
\begin{array}{ll}
\langle q_0, a, a, q_1, d \rangle & \langle q_0, x, x, q_R, f \rangle \ \forall x \in \Sigma \cup \{\square\} - \{a\} \\
\langle q_1, b, b, q_2, d \rangle & \langle q_1, x, x, q_R, f \rangle \ \forall x \in \Sigma \cup \{\square\} - \{b\} \\
\langle q_2, x, x, q_2, d \rangle \ \forall x \in \Sigma - \{\square\} & \langle q_2, \square, \square, q_3, s \rangle \\
\langle q_3, d, d, q_4, s \rangle & \langle q_3, x, x, q_R, f \rangle \ \forall x \in \Sigma \cup \{\square\} - \{d\} \\
\langle q_4, c, c, q_A, f \rangle & \langle q_4, x, x, q_R, f \rangle \ \forall x \in \Sigma \cup \{\square\} - \{c\},
\end{array}
$$

dove q_0, q_A e q_R sono, rispettivamente gli stati iniziale, di accettazione e di rigetto di T.

Dopo aver definito il linguaggio $L \subseteq \Sigma^*$ deciso da T, si trasformi T in una macchina T' definita sull'alfabeto $\{0, 1\}$ equivalente a T.

2 Soluzioni

Soluzione del problema 2.1

Sia $s = \langle x_1 x_2 \dots x_n \rangle \in \{a,b,c,d\}^n$ e $\sigma(s) = \langle y_1 y_2 \dots y_n \rangle \in \{0,1\}^n$ la stringa binaria associata ad s secondo le regole seguenti:

- $y_i = 0 \Leftrightarrow x_i = a \vee x_i = c$, per $1 \leq i \leq n$;
- $y_i = 1 \Leftrightarrow x_i = b \vee x_i = d$, per $1 \leq i \leq n$.

Segue dalle definizioni di L e di $\sigma(s)$ che $s \in L$ se e soltanto se $\sigma(s)$ è una stringa palindroma. Pertanto, la macchina di Turing richiesta è una banale modifica di quella vista a lezione per il linguaggio PALINDROMIA: T è definita sull'alfabeto $\{a,b,c,d,\square\}$ ($\square$ è il carattere blank) e sull'insieme degli stati $Q = \{q_0, q_a, q_b, q_c, q_d, q_{ind}, q_{acc}, q_{rig}\}$ in cui q_0 è lo stato iniziale, q_{acc} lo stato finale di accettazione e q_{rig} lo stato finale di rigetto. L'insieme delle quintuple di T è il seguente (per chiarezza di notazione, indichiamo con dx e sx, rispettivamente, lo spostamento a destra e a sinistra della testina)

1. $\langle q_0, a, \square, q_a, dx \rangle$;
2. $\langle q_a, a, a, q_a, dx \rangle$, $\langle q_a, b, b, q_a, dx \rangle$, $\langle q_a, c, c, q_a, dx \rangle$, $\langle q_a, d, d, q_a, dx \rangle$;
3. $\langle q_a, \square, \square, q_c, sx \rangle$;
4. $\langle q_c, c, \square, q_{ind}, sx \rangle$;
5. $\langle q_c, a, a, q_{rig}, f \rangle$, $\langle q_c, b, b, q_{rig}, f \rangle$, $\langle q_c, d, d, q_{rig}, f \rangle$, $\langle q_c, \square, \square, q_{rig}, f \rangle$;
6. $\langle q_0, b, \square, q_d, dx \rangle$;
7. $\langle q_b, a, a, q_b, dx \rangle$, $\langle q_b, b, b, q_b, dx \rangle$, $\langle q_b, c, c, q_b, dx \rangle$, $\langle q_b, d, d, q_b, dx \rangle$;
8. $\langle q_b, \square, \square, q_d, sx \rangle$;
9. $\langle q_d, d, \square, q_{ind}, sx \rangle$;
10. $\langle q_d, a, a, q_{rig}, f \rangle$, $\langle q_d, b, b, q_{rig}, f \rangle$, $\langle q_d, c, c, q_{rig}, f \rangle$, $\langle q_d, \square, \square, q_{rig}, f \rangle$;
11. $\langle q_{ind}, a, a, q_{ind}, sx \rangle$, $\langle q_{ind}, b, b, q_{ind}, sx \rangle$, $\langle q_{ind}, c, c, q_{ind}, sx \rangle$, $\langle q_{ind}, d, d, q_{ind}, sx \rangle$;
12. $\langle q_{ind}, \square, \square, q_0, dx \rangle$;
13. $\langle q_0, \square, \square, q_{acc}, f \rangle$;
14. $\langle q_0, c, c, q_{rig}, f \rangle$, $\langle q_0, d, d, q_{rig}, f \rangle$.

Soluzione del problema 2.2

Viene utilizzata una macchina di Turing a $k+2$ nastri a testine indipendenti: i primi k nastri contengono le k parole p_1, p_2, …, p_k, sul nastro $k+1$ viene scritta la parola o e sul nastro $k+2$ viene scritta la parola v.

La macchina opera in due fasi: durante la prima fase calcola la parola v e la scrive sul nastro $k+2$, durante la seconda fase calcola la parola o e la scrive sul nastro $k+1$.

Inizialmente, i primi k nastri contengono l'input, scritti a partire dalle cella di indica 0, la macchina si trova nello stato iniziale q_0 e le testine sono posizionate sulle celle di indice 0 dei rispettivi nastri.

Descriviamo la prima fase, che utilizza gli stati q_0, q_2^{v0}, q_2^{v1}, q_3^{v0}, q_3^{v1}, …, q_k^{v0}, q_k^{v1} e q_1^o. Nello stato q_0, se la testina del primo nastro legge 0 allora tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_2^{v0} e (soltanto) la testina del nastro 1 si muove a destra; se la testina del primo nastro legge 1 allora tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_2^{v1} e (soltanto) la testina del nastro 1 si muove a destra: $\forall x_2, \ldots, x_k$

$$\langle q_0, (0, x_2, \ldots x_k, \square, \square), (0, x_2, \ldots x_k, \square, \square), q_2^{v0}, (d, f, \ldots, f) \rangle$$

$$\langle q_0, (1, x_2, \ldots x_k, \square, \square), (1, x_2, \ldots x_k, \square, \square), q_2^{v1}, (d, f, \ldots, f) \rangle.$$

Poi, in generale, per $i = 2, \ldots, k-1$:

- nello stato q_i^{v0}, se la testina del nastro i legge 0 allora tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_{i+1}^{v0} e (soltanto) la testina del nastro i si muove a destra; se la testina del nastro i legge 1 allora tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_{i+1}^{v1} e (soltanto) la testina del nastro i si muove a destra;
- nello stato q_i^{v1}, se la testina del nastro i legge 0 allora tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_{i+1}^{v1} e (soltanto) la testina del nastro i muove a destra; se la testina del nastro i legge 1 allora tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_{i+1}^{v0} e (soltanto) la testina del nastro i si muove a destra.

Infine,

- nello stato q_k^{v0}, se la testina del nastro k legge 0, allora tutte le testine tranne la $k+2$ riscrivono quello che hanno letto, la testina $k+2$ scrive 0, la macchina entra nello stato q_0 e (soltanto) la testina del nastro $k+2$ si muove a destra; se la testina del nastro k legge 1 allora tutte le testine tranne la $k+2$ riscrivono quello che hanno letto, la testina $k+2$ scrive 1, la macchina entra nello stato q_0 e (soltanto) la testina del nastro $k+2$ si muove a destra;
- nello stato q_k^{v1}, se la testina del nastro k legge 0, allora tutte le testine tranne la $k+2$ riscrivono quello che hanno letto, la testina $k+2$ scrive 1, la macchina entra nello stato q_0 e (soltanto) la testina del nastro $k+2$ si muove a destra; se la testina del nastro k legge 1 allora tutte le testine tranne la $k+2$ riscrivono quello che hanno letto, la testina $k+2$ scrive 0, la macchina entra nello stato q_0 e (soltanto) la testina del nastro $k+2$ si muove a destra.

La prima fase termina quando, nello stato q_0 la testina del nastro 1 legge $\square$: in questo caso, tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_1 e le testine dei nastri 1, …, k si muovono a sinistra.

La seconda fase utilizza gli stati q_1^o, q_1^{o0}, q_1^{o1}, …, q_k^o, q_k^{o0}, q_k^{o1}, q_{k+1}^o. Negli stati q_i^o, q_i^{o0}, q_i^{o1} viene calcolato il bit o_i della parola o, per $i = 1, \ldots, k$:

- nello stato q_i^o, se la testina del nastro i legge 0, allora tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_i^{o0} e (soltanto) la testina del nastro i si muove a sinistra; se la testina del nastro i legge 1, allora tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_i^{o1} e (soltanto) la testina del nastro i si muove a sinistra;
- nello stato q_i^{o0}, se la testina del nastro i legge 0, allora tutte le testine riscrivono quello che hanno letto, la macchina rimane nello stato q_i^{o0} e (soltanto) la testina del nastro i si muove a sinistra; se la testina del nastro i legge 1, allora tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_i^{o1} e (soltanto) la testina del nastro i si muove a sinistra; se la testina del nastro i legge $\square$ allora tutte le testine tranne la $k+1$ riscrivono quello che hanno letto, la testina $k+1$ scrive 0, la macchina entra nello stato q_{i+1}^o e (soltanto) la testina del nastro $k+1$ si muove a destra;
- nello stato q_i^{o1}, se la testina del nastro i legge 0, allora tutte le testine riscrivono quello che hanno letto, la macchina rimane nello stato q_i^{o1} e (soltanto) la testina del nastro i si muove a sinistra; se la testina del nastro i legge 1, allora tutte le testine riscrivono quello che hanno letto, la macchina entra nello stato q_i^{o0} e (soltanto) la testina del nastro i si muove a sinistra; se la testina del nastro i legge $\square$ allora tutte le testine tranne la $k+1$ riscrivono quello che hanno letto, la testina $k+1$ scrive 1, la macchina entra nello stato q_{i+1}^o e (soltanto) la testina del nastro $k+1$ si muove a destra.

La parola o è stata calcolata e lo stato q^o_{k+1} è lo stato finale.

Soluzione del problema 2.3

Definiamo una macchina di Turing T a 4 nastri (a testine indipendenti) che calcola simultaneamente $\lceil \frac{n}{k} \rceil$ e $\lfloor \frac{n}{k} \rfloor$ come di seguito descritto:

- il nastro N_1 contiene l'input n ed il nastro N_2 contiene l'input k, ivi memorizzati in unario all'inizio della computazione, preceduti e seguiti da $\square$;
- il nastro N_3 è il nastro di lavoro e di output per la funzione $\lceil \frac{n}{k} \rceil$, inizialmente vuoto, sul quale, al termine della computazione, si troverà il valore $\lceil \frac{n}{k} \rceil$;
- il nastro N_4 è il nastro di lavoro e di output per la funzione $\lfloor \frac{n}{k} \rfloor$, inizialmente vuoto, sul quale, al termine della computazione, si troverà il valore $\lfloor \frac{n}{k} \rfloor$.

T utilizza gli stati q_0 (stato iniziale), q_1, q_2 e lo stato finale q_F: q_0 (oltre ad essere stato iniziale) indica che sul nastro N_1 è stato letto un numero di 1 pari ad un multiplo di k, q_1 indica che sul nastro N_1 è stato letto un numero di 1 pari ad un multiplo di k più qualche ulteriore 1 (in numero inferiore a k), e q_2 indica che la scansione del nastro 2 è terminata avendo letto per ciascun 1 sul nastro N_2 un corrispondente 1 sul nastro N_1 (ossia, se la macchina entra per la h-esima volta nello stato q_2, allora n vale almeno hk) ed occorre riposizionare la testina di N_2 sul simbolo 1 più a sinistra.
Il valore $\lceil \frac{n}{k} \rceil$ viene calcolato scrivendo un 1 sul nastro N_3 ogni volta che, nello stato q_0, viene letto un 1 sul nastro N_1: questo significa che n è almeno un multiplo di k (perché la macchina è nello stato q_0) più 1 (perché viene letto un 1). Così, se $n = hk + m$, con $m < k$, al termine della scansione dell'input, risultano scritti sul nastro N_3 h 1 più un eventuale ulteriore 1 se $m > 0$.
Analogamente, il valore $\lfloor \frac{n}{k} \rfloor$ viene calcolato scrivendo un 1 sul nastro N_4 ogni volta che la macchina entra nello stato q_2: questo significa che n è almeno un multiplo di k. Così, se $n = hk + m$, con $m < k$, al termine della scansione dell'input, risultano scritti sul nastro N_4 h 1.

Formalmente, le quintuple utilizzate sono:

$$\begin{array}{r}
\langle q_0, (1,1,\square,\square), (1,1,1,\square), q_1, (d,d,d,f) \rangle \\
\langle q_0, (\square,1,\square,\square), (\square,1,\square,\square), q_F, (f,f,f,f) \rangle \\
\langle q_1, (1,1,\square,\square), (1,1,\square,\square), q_1, (d,d,f,f) \rangle \\
\langle q_1, (1,\square,\square,\square), (1,\square,\square,1), q_2, (f,s,f,d) \rangle \\
\langle q_1, (\square,1,\square,\square), (\square,\square,\square,\square), q_F, (f,f,f,f) \rangle \\
\langle q_1, (\square,\square,\square,\square), (\square,\square,\square,1), q_F, (f,f,f,f) \rangle \\
\langle q_2, (1,1,\square,\square), (1,1,\square,\square), q_2, (f,s,f,f) \rangle \\
\langle q_2, (1,\square,\square,\square), (1,\square,\square,\square), q_0, (f,d,f,f) \rangle
\end{array}$$

Soluzione del problema 2.4

La macchina T che decide L opera come di seguito descritto. Nello stato q_0, T legge il simbolo nella cella scandita dalla testina e, se tale simbolo è a, lo cancella, entra nello stato q_a, raggiunge la fine della parola e, se gli ultimi due caratteri sono una coppia di b, li cancella e torna all'inizio della parola. Se invece la parola termina con qualcosa di diverso, T entra nello stato di rigetto q_R. Analogamente, se nello stato q_0 T legge b allora entra nello stato q_R. Se, infine, nello stato q_0 T legge $\square$ allora entra nello stato di accettazione q_A (in quanto la parola input è la parola vuota, che appartiene ad L, oppure tutti i suoi caratteri sono stati cancellati). Formalmente, T è definita dalle quintuple seguenti:

$$\begin{array}{llll}
\langle q_0, a, \Box, q_a, destra\rangle & \langle q_0, b, b, q_R, fermo\rangle & \langle q_0, \Box, \Box, q_A, fermo\rangle & (2.1)\\
\langle q_a, a, a, q_a, destra\rangle & \langle q_a, b, b, q_{ab}, destra\rangle & \langle q_a, \Box, \Box, q_R, fermo\rangle & (2.2)\\
\langle q_{ab}, a, a, q_R, fermo\rangle & \langle q_{ab}, b, b, q_{ab}, destra\rangle & \langle q_{ab}, \Box, \Box, q_b, sinistra\rangle & (2.3)\\
\langle q_b, b, \Box, q_{b1}, sinistra\rangle & \text{gli altri casi non sono possibili} & & (2.4)\\
\langle q_{b1}, a, a, q_R, fermo\rangle & \langle q_{b1}, b, \Box, q_1, sinistra\rangle & \langle q_{b1}, \Box, \Box, q_R, fermo\rangle & (2.5)\\
\langle q_1, a, a, q_1, sinistra\rangle & \langle q_1, b, b, q_1, sinistra\rangle & \langle q_1, \Box, \Box, q_0, destra\rangle & (2.6)
\end{array}$$

Soluzione del problema 2.5

Data l'equivalenza fra macchine di Turing ad un nastro e macchine di Turing che utilizzano più nastri, senza perdita di generalità possiamo assumere che T_Σ sia una macchina ad un nastro.
Innanzi tutto, fissiamo una codifica $\rho : \{a,b,c,d,\Box\} \to \{0,1,\Box\}^2$ dei simboli dell'alfabeto di lavoro $\Sigma \cup \Box$ di T_Σ nell'alfabeto di lavoro $B \cup \Box$ di T_B: scegliamo

$$\rho(a) = 00 \qquad \rho(b) = 01 \qquad \rho(c) = 10 \qquad \rho(d) = 11 \qquad \rho(\Box) = \Box\Box.$$

Codifichiamo, inoltre ogni simbolo $\Box$ eventualmente presente, o che scriveremo, all'interno della stringa input di T_Σ mediante una coppia di caratteri $\Box$: ossia, il blank viene rappresentato in T_B nella forma $\Box\Box$.

Ciò premesso, presentiamo due soluzioni differenti al problema in questione, la prima delle quali definisce una macchina T_B a due nastri a testine solidali, la seconda una macchina ad un solo nastro.

Soluzione 1. La macchina T_B proposta in questa soluzione è dotata di due nastri, N_1 ed N_2 sui quali vengono scritti, rispettivamente, il primo simbolo e il secondo simbolo della codifica binaria sopra descritta degli elementi di $\Sigma \cup \{\Box\}$ che costituiscono l'input: ad esempio, se l'i-esimo carattere dell'input di T_Σ è il carattere 'c' (ossia, la cella i-esima del nastro di T_Σ al tempo $t = 0$ contiene il carattere 'c'), allora nella cella $N_1[i]$ viene scritto il simbolo 1 e nella cella $N_2[i]$ viene scritto il simbolo 0. Assumiamo, quindi, che, all'istante iniziale, la codifica dell'input di T_Σ sia scritta sui due nastri di T_B in accordo alla regola appena descritta.

Mostriamo, ora, come derivare le quintuple di T_B da quelle di T_Σ; cominciamo con l'osservare che, per costruzione, se all'istante iniziale la testina di T_Σ legge un carattere $\sigma \in \Sigma$ allora, all'istante iniziale, le due testine di T_B leggono i due bit $b_1(\sigma)$ e $b_2(\sigma)$ corrispondenti alla codifica di σ. Allora, per ogni quintupla di T_Σ del tipo $\langle q_0, \sigma, \tau, q_2, m\rangle$, la macchina T_B contiene la quintupla

$$\langle q_0, (b_1(\sigma), b_2(\sigma)), (b_1(\tau), b_2(\tau)), q_2, m\rangle.$$

Intuitivamente, l'osservazione appena fatta può essere generalizzata (la dimostrazione formale è non richiesta): se all'istante t la testina di T_Σ legge un carattere $\sigma \in \Sigma$ allora, all'istante t, le due testine di T_B leggono i due bit $b_1(\sigma)$ e $b_2(\sigma)$ corrispondenti alla codifica di σ. Allora, per ogni quintupla di T_Σ del tipo $\langle q_1, \sigma, \tau, q_2, m\rangle$, la macchina T_B contiene la quintupla

$$\langle q_1, (b_1(\sigma), b_2(\sigma)), (b_1(\tau), b_2(\tau)), q_2, m\rangle.$$

Soluzione 2. La macchina T_B proposta in questa soluzione è dotata di un solo nastro sul quale viene scritta inizialmente la codifica binaria dell'input di T_Σ.
Consideriamo, ora, una quintupla $\langle q_1, s_1, s_2, q_2, m\rangle$ di T_Σ, con $s_1 \in \Sigma$ e $s_2 \in \Sigma \cup \{\Box\}$, e trasformiamola in una serie di quintuple di T_B. Mostriamo questa trasformazione nel caso particolare in cui $s_1 = a$, $s_2 = b$ e $m = S$ (gli altri casi sono analoghi, ricordando che il simbolo $\Box$ di T_Σ è codificato in T_B mediante $\Box\Box$):

$$\langle q_1, 0, 0, q_1^0, D\rangle, \qquad \langle q_1^0, 0, 1, q_2^0, S\rangle, \qquad \langle q_2^0, 0, 0, q_2^{sin,1}, S\rangle, \qquad \forall x \in \{0,1\}\ \langle q_2^{sin,1}, x, x, q_2, S\rangle$$

La prima quintupla sopra, a partire dallo stato q_1 verifica che il carattere letto possa essere il primo caratere della codifica di a: se questo è vero, entra nello stato q_1^0 (che tiene traccia dello stato di partenza q_1 e del carattere letto 0) e si muove a destra per controllare se il carattere successivo è proprio il secondo carattere della codifica di a; in tal caso, la seconda quintupla sopra inizia l'esecuzione della quintupla $\langle q_1, s_1, s_2, q_2, m\rangle$ di T_Σ scrivendo il secondo carattere di b (1), entrando nello stato q_2^0 (che tiene traccia del primo carattere di b che deve essere scritto nella cella immediatamente a sinistra e dello stato in cui dovrà entrare al termine della esecuzione della quintupla) e muovendosi a sinistra. La terza quintupla sopra, scrive il carattere 0 (il primo carattere della codifica di b, specificato all'interno dello stato attuale q_2^0) e si prepara a spostare la testina a sinistra di due posizioni (perché il movimento della quintupla di T_Σ è S) entrando nello stato $q_2^{sin,1}$ e spostandosi a sinistra. Infine, la quarta quintupla, indipendentemente dal carattere letto, sposta la testina a sinistra ed entra nello stato q_2, predisponendosi a simulare un'altra quintupla di T_Σ.

Consideriamo, infine, una quintupla $\langle q_3, \square, s, q_4, m\rangle$ di T_Σ, con $s \in \Sigma \cup \{\square\}$, e trasformiamola in una serie di quintuple di T_B. Mostriamo questa trasformazione nel caso particolare in cui $s_2 = c$ e $m = D$ (gli altri casi sono analoghi):

$$\langle q_3, \square, \square, q_3^\square, D\rangle, \qquad \langle q_3^\square, \square, 0, q_4^1, S\rangle, \qquad \langle q_4^1, \square, 1, q_4^{des,1}, D\rangle, \qquad \forall x \in \{0,1\}\ \langle q_4^{des,1}, x, x, q_4, D\rangle$$

La spiegazione di questo ultimo gruppo di quintuple è analoga a quella del gruppo precedente.

Per affermare che T_B simula T_Σ, consideriamo la computazione $T_\Sigma(x)$, per una qualsiasi parola $x \in \Sigma^*$, e la computazione $T_B(\rho(x)$, dove $\rho(x)$ è la codifica di x mediante ρ. Osserviamo, allora, che, per come abbiamo costruito le quintuple di T_B, se ad un dato istante $t \geq 1$ della computazione $T_\Sigma(x)$[1] viene eseguita la quintupla $\langle q_1, s_1, s_2, q_2, m\rangle$ e che all'istante $t+1$ T_Σ si trova nello stato q_2 e la sua testina è posizionata su una cella che contiene il carattere $s \in \Sigma \cup \{\square\}$, allora all'istante $4t$ della computazione $T_B(x)$ inizia l'esecuzione delle quattro quintuple corrispondenti descritte sopra e all'istante $4t+4$ T_B si trova nello stato q_2 e la sua testina è posizionata su una cella che contiene il primo carattere della codifica di $\rho(s)$ di s.

Una dimostrazione formale di quanto affermato sopra richiede un semplice ragionamento induttivo che, comunque, non era necessario ai fini dell'esame.

Soluzione del problema 2.6

Indichiamo con Σ, con Q_k, e con P_k, rispettivamente, l'alfabeto, l'insieme degli stati, e l'insieme delle quintuple che definiscono NT_k. Poiché NT_k è una macchina non deterministica, è possibile che, per qualche $x \in \Sigma_k$ e per qualche $q \in Q_k$, P_k contenga più di una quintupla i cui primi due elementi siano, rispettivamente, q e x. Indichiamo, dunque, per ogni $x \in \Sigma_k$ e per ogni $q \in Q_k$, con $P_k(q,x)$ l'insieme delle quintuple in P_k i cui primi due elementi sono q e x (si osservi che tale insieme può essere vuoto). D'altra parte, poiché il grado di non deteminismo di NT_k è k, per ogni $x \in \Sigma_k$ e per ogni $q \in Q_k$, $|P_k| \leq k$.
Siano, dunque, $x \in \Sigma$ e $q \in Q_k$ tali che

$$P_k(q,x) = \langle q, x, x_1, q_1, m_1\rangle,\ \langle q, x, x_2, q_2, m_2\rangle,\ \ldots\ \langle q, x, x_h, q_h, m_h\rangle$$

con $h > 2$ (e, ovviamente, $h \leq k$). Ricordiamo che il significato dell'insieme $P_k(q,x)$ di quintuple è il seguente: *se* la macchina si trova nello stato q *e* legge sul nastro il simbolo x *allora* deve eseguire *o* la quintupla $\langle q, x, x_1, q_1, m_1\rangle$ *o* la quintupla $\langle q, x, x_2, q_2, m_2\rangle$,... *o* la quintupla $\langle q, x, x_h, q_h, m_h\rangle$. Per ottenere lo stesso comportamento con una macchina che abbia grado di non determinismo 2, ragioniamo nel modo seguente: *se* la nuova macchina si trova nello stato q *e* legge sul nastro il simbolo x *allora* deve eseguire la quintupla $\langle q, x, x_1, q_1, m_1\rangle$ *oppure* non deve eseguirla - ossia, deve eseguire un'altra quintupla $\langle q, x, x, q'_1(x), \texttt{ferma}\rangle$; a questo punto, nello stato $q'_1(x)$ e leggendo il simbolo x, deve eseguire la quintupla $\langle q'_1(x), x, x_2, q_2, m_2\rangle$ *oppure* non deve eseguirla - ossia, deve eseguire un'altra quintupla $\langle q, x, x, q'_2(x), \texttt{ferma}\rangle$, e così via.

Quanto appena descritto è illustrato in Figura 2.1: la parte (a) mostra il comportamento della macchina NT_k quando si trova nello stato interno q e legge sul nastro il simbolo x (e, dunque, esegue una quintupla scelta in $P_k(q,x)$), mentre la parte (b) mostra la sequenza di passi che devono essere eseguiti dalla macchina NT_2 per ottenere un un comportamento equivalente.

[1] Assumiamo che la computazione inizi al tempo $t = 1$.

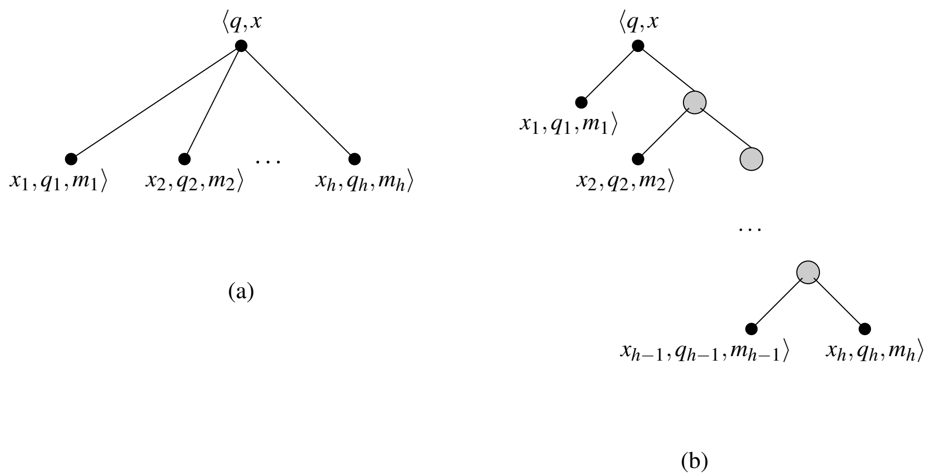

Figura 2.1: Un passo non deterministico di grado $h > 2$ in (a) e gli $h-1$ passi non deterministici di grado 2 ad esso equivalenti in (b).

Lo schema illustrato in Figura 2.1 può essere implementato in linguaggio **PascalMinimo**. A questo scopo, assumiamo. come di consueto, che lo stato interno della macchina sia memorizzato nella variabile q, il contenuto del nastro nell'array$N$, la posizione della testina della variabile intera t e che il movimento della testina sia indicato da un intero in $\{-1,0,1\}$. Allora, il passo non deterministico di grado k è implementato nel seguente frammento di codice

scegli una quintupla $\langle q,N[t],x_i,q_i,m_i\rangle$ nell'insieme $P_k(q,n[t]$:
$N[t] \to x_i$;
$q \to q_i$;
$t \to t+m_i$;

mentre la sequenza di passi non deterministici di grado 2 è implementato nel seguente frammento di codice

$i \to 1$; scelta $\to$ `falso`;
while $(i \leq |P_k(q,N[t]-1) \wedge$ scelta $\to$ `falso`) **do begin**
    scegli se eseguire la quintupla $\langle q,N[t],x_i,q_i,m_i\rangle$ oppure no:
    if (hai scelto di eseguire la quintupla $\langle q,N[t],x_i,q_i,m_i\rangle$) **then begin**
        $N[t] \to x_i$;
        $q \to q_i$;
        $t \to t+m_i$;
        scelta $\to$ `vero`;
    end;
    else $i \to i+1$;
end;
if (scelta $\to$ `falso`) **then begin**
    $N[t] \to x_h$;
    $q \to q_h$;
    $t \to t+m_h$;
end.

Definiamo, ora, formalmente, la macchina NT_2, il cui alfabeto di lavoro è ancora Σ ed il cui insieme degli stati Q_2

contiene propriamente Q_k. Indichiamo con P_2 l'insieme delle quintuple di NT_2 e, per ogni $x \in \Sigma_k$ e per ogni $q \in Q_k$, con $P_2(q,x)$ l'insieme delle quintuple in P_2 i cui primi due elementi sono q e x.
Se

$$P_k(q,x) = \langle q,x,x_1,q_1,m_1\rangle,\ \langle q,x,x_2,q_2,m_2\rangle,\ \dots\ \langle q,x,x_h,q_h,m_h\rangle$$

e $2 < h \leq k$, allora introduciamo l'insieme dei nuovi stati interni $Q_2(q,x) = \{q'_1(q,x),\dots,q'_{h-2}(q,x)\}$ e definiamo l'insieme $P_2(q,x)$ come costituito dalle quintuple seguenti:

$$\begin{array}{rl}
\langle q,x,x_1,q_1,m_1\rangle & \langle q,x,x,q'_1(q,x),\texttt{ferma}\rangle \\
\langle q'_1(q,x),x,x_2,q_2,m_2\rangle & \langle q'_1(q,x),x,x,q'_2(q,x),\texttt{ferma}\rangle \\
\langle q'_2(q,x),x,x_3,q_3,m_3\rangle & \langle q'_2(q,x),x,x,q'_3(q,x),\texttt{ferma}\rangle \\
 & \dots \\
\langle q'_{h-2}(q,x),x,x_{h-1},q_{h-1},m_{h-1}\rangle & \langle q'_{h-2}(q,x),x,x_h,q_h,m_h\rangle.
\end{array}$$

Se, invece $|P_k(q,x)| \leq 2$, allora definiamo $P_2(q,x) = P_k(q,x)$ e $Q_2(q,x) = \emptyset$. Infine, poniamo $P_2 = \cup_{q \in Q_k \,\wedge\, x \in \Sigma} P_2(q,x)$. Allora, per costruzione, per ogni $q \in Q_k$ e per ogni $x \in \Sigma$, il numero delle quintuple in $P_2(q,x)$ che iniziano con la stessa coppia stato-simbolo è al più 2. Infine, sempre per costruzione, $Q_2(q,x) \cap Q_2(q',x') = \emptyset$ se $q \neq q'$ o $x \neq x'$, e questo completa la prova che il grado di non determinismo di NT_2 è 2.

Soluzione del problema 2.7

In quanto segue, descriviamo una macchina T_0 a due nastri a testine indipendenti.
Poiché, come indicato nel suggerimento, il secondo nastro di T_0 contiene gli stati interni di T,, allora l'alfabeto di lavoro di T_0 è $\{0,1\} \cup Q$, dove Q è l'insieme degli stati di Q.

La macchina T_0, non avendo possibilità di cambiare stato (se non quando entra in uno stato finale), deve utilizzare il secondo nastro per tener traccia dei cambiamenti di stato di T, durante le sue computazioni, e per scegliere in base ad essi le quintuple da eseguire.

All'inizio della computazione, il nastro di T contiene l'input $x \in \{0,1\}^*$ e, quindi, corrispondentemente, il nastro 1 di T_0 contiene x e il nastro 2 è vuoto. Quando T esegue la prima quintupla, è possibile che essa cambi stato: in corrispondenza, quando T_0 esegue la prima quintupla, leggendo $\square$ sul secondo nastro, scrive lo stato di arrivo della corrispondente quintupla di T sul secondo nastro. Formalmente: ad ogni quintupla $\langle q_0,a,b,q',m\rangle$ di T (con $m \in \{\texttt{sinistra},\texttt{fermo},\texttt{destra}$ e $a,b \in \{0,1\})$ corrisponde in T_0 la quintupla

$$\langle q_0,(a,\square),(b,q'),q_0,(m,\texttt{fermo})\rangle.$$

Successivamente, il contenuto del secondo nastro di T_0 sarà utilizzato per capire quale quintupla di T eseguire. Quindi, ad ogni quintupla $\langle q,a,b,q',m\rangle$ di T corrisponde in T_0 la quintupla

$$\langle q_0,(a,q),(b,q'),q_0,(m,\texttt{fermo})\rangle.$$

Infine, quando T_0 legge sul nastro 2 lo stato q_a o q_R, entra nello stao corrispondente e termina: quindi, anche le seguenti due quintuple fanno parte delle istruzioni di T_0

$$\langle q_0,(a,q_A),(a,q_A),q_A,(\texttt{fermo},\texttt{fermo})\rangle,$$

$$\langle q_0,(a,q_R),(a,q_R),q_R,(\texttt{fermo},\texttt{fermo})\rangle,$$

per ogni $a \in \{0,1\}$.

Soluzione del problema 2.8

Ad ogni passo, leggendo il carattere c nella cella scandita dalla testina, la macchina T che decide il problema deve operare come segue:

- se $c = 0$, allora T entra nello stato di accettazione q_A e termina;
- se $c = \square$, allora T entra nello stato di rigetto q_R e termina;
- se c è un valore compreso fra 1 e 9, allora T sposta la sua testina a destra di c posizioni.

Per eseguire quanto indicato nel terzo punto sopra, dotiamo T, oltre che dello stato iniziale q_0, dello stato di accettazione q_A e dello stato di rigetto q_R, degli stati q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8 e q_9: quando T è nello stato q_i, con $1 \leq i \leq 9$, indipendentemente da quello che legge la sua testina, sposta la testina a destra di una posizione ed entra nello stato q_{i-1}.
Quindi, la macchina T è descritta dalle quintuple seguenti:

$$\langle q_0, 0, 0, q_A, ferma \rangle, \qquad \langle q_0, \square, \square, q_R, ferma \rangle, \qquad \langle q_0, i, i, q_i ferma \rangle \ \forall \ 1 \leq i \leq 9,$$

$$\langle q_i, x, x, q_{i-1}, destra \rangle \ \forall \ 1 \leq i \leq 9, \ \forall \ x \in \{0,1,2,3,4,5,6,7,8,9\} \cup \{\square\}.$$

Soluzione del problema 2.9

Il linguaggio $L(T)$, deciso dalla macchina T, è costituito dalle parole in Σ^* che iniziano con *ab* e terminano con *cd* ed è definito formalmente nel modo seguente

$$L(T) = \{abxcd : x \in \Sigma^*\}.$$

Definiamo, innanzi tutto, una codifica $\chi : \Sigma \to \{0,1\}^*$: sia, dunque, $\chi(a) = 00$, $\chi(b) = 01$, $\chi(c) = 10$ e $\chi(d) = 11$.
La macchina T_{01} definita sull'alfabeto $\{0,1\}$ che decide il linguaggio $L(T)$ codificato secondo la codifica χ utilizza l'insieme di stati $Q_{01} = \{q_0, q_0(0), q_1, q_1(0), q_2, q_3, q_3(1), q_4, q_4(0), q_A, q_R\}$, ove q_0 è lo stato iniziale, ed è descritta dal seguente insieme di quintuple:

$$\begin{array}{ll}
\langle q_0, 0, 0, q_0(0), d \rangle & \langle q_0, x, x, q_R, f \rangle \ \forall x \in \{1, \square\} \\
\langle q_0(0), 0, 0, q_1, d \rangle & \langle q_0(0), x, x, q_R, f \rangle \ \forall x \in \{1, \square\} \\
\langle q_1, 0, 0, q_1(0), d \rangle & \langle q_1, x, x, q_R, f \rangle \ \forall x \in \{1, \square\} \\
\langle q_1(0), 1, 1, q_2, d \rangle & \langle q_1(0), x, x, q_R, f \rangle \ \forall x \in \{0, \square\} \\
\langle q_2, x, x, q_2, d \rangle \ \forall x \in \{0,1\} & \langle q_2, \square, \square, q_3, s \rangle \\
\langle q_3, 1, 1, q_3(1), s \rangle & \langle q_3, x, x, q_R, f \rangle \ \forall x \in \{0, \square\} \\
\langle q_3(1), 1, d, q_4, s \rangle & \langle q_3(1), x, x, q_R, f \rangle \ \forall x \in \{0, \square\} \\
\langle q_4, 0, 0, q_4(0), s \rangle & \langle q_4, x, x, q_R, f \rangle \ \forall x \in \{1, \square\}, \\
\langle q_4(0), 1, 1, q_A, f \rangle & \langle q_4(0), x, x, q_R, f \rangle \ \forall x \in \{0, \square\},
\end{array}$$