Esercizio 7.1

Determinare una grammatica lineare destra che genera il linguaggio descritto dalla seguente espressione regolare:

$$b^* + (ab)^*$$

Il linguaggio relativo all'espressione regolare è:

$$S\left(b^* + (ab)^*\right) = S(b^*) \cup S\left((ab)^*\right) = \left(S(b)\right)^* \cup \left(S(ab)\right)^* = \{b\}^* \cup \left(S(a) \cdot S(b)\right)^* =$$
$$= \{b\}^* \cup \left(\{a\} \cdot \{b\}\right)^* = \{b\}^* \cup \{ab\}^*$$

Costruiamo dapprima la grammatica G_1 tale che

$$L(G_1) = S(b^*) = \{b\}^*$$

$$G_1 = (X_1, V_1, S_1, P_1)$$

ove:

$$X_1 = \{b\} \qquad V_1 = \{S_1\} \qquad P_1 = \{S_1 \to bS_1 \mid \lambda\}\ .$$

Costruiamo ora la grammatica G_2 tale che

$$L(G_2) = S(ab) = \{ab\}$$

$$G_2 = (X_2, V_2, S_2, P_2)$$

ove:

$$X_2 = \{a, b\} \qquad V_2 = \{S_2, B_2\} \qquad P_2 = \{S_2 \to aB_2,\ B_2 \to b\}\ .$$

La grammatica G_3 tale che

$$L(G_3) = S\left((ab)^*\right) = \{ab\}^*$$

è dunque (si veda Capitolo 5, Tavola 1):

$$G_3 = (X_3, V_3, S_3, P_3)$$

ove:

$$X_3 = X_2 = \{a, b\} \qquad V_3 = V_2 \cup \{S_3\}$$
$$P_3 = \{S_3 \to \lambda\} \cup (P_2 - \{S_2 \to \lambda\}) \cup \{S_3 \to w \mid S_2 \to w \in P_2\} \cup$$
$$\cup \{A \to bS_3 \mid A \to b \in P_2\} =$$
$$= \{S_3 \to \lambda\} \cup P_2 \cup \{S_3 \to aB_2\} \cup \{B_2 \to bS_3\} =$$
$$= \{S_3 \to \lambda,\ S_2 \to aB_2,\ B_2 \to b,\ S_3 \to aB_2,\ B_2 \to bS_3\} =$$
$$= \{S_3 \to \lambda \mid aB_2,\ S_2 \to aB_2,\ B_2 \to b \mid bS_3\}.$$

Si osservi che il nonterminale S_2 non compare nella parte destra di nessuna produzione in P_3 (S_2 è un nonterminale inutile - si veda Definizione 8.12), dunque l'intera produzione ($S_2 \to aB_2$) può essere rimossa da P_3 senza alterare il linguaggio generato da G_3.

Quindi P_3 diventa:

$$P_3 = \{S_3 \to \lambda \mid aB_2,\ B_2 \to b \mid bS_3\}\ .$$

La grammatica G tale che:

$$L(G) = S(b^*) \cup S\left((ab)^*\right) = S\left(b^* + (ab)^*\right)$$

è dunque:

$$G = (X, V, S, P)$$

ove:

$$X = X_1 \cup X_3 = \{a, b\}$$
$$V = V_1 \cup V_3 = V_1 \cup V_2 \cup \{S_3\} \cup \{S\} = \{S, S_1, B_2, S_3\}$$
$$\begin{aligned} P &= \{S \to w \mid S_1 \to w \in P\} \cup \{S \to w \mid S_3 \to w \in P_3\} \cup P_1 \cup P_3 = \\ &= \{S \to bS_1 \mid \lambda\} \cup \{S \to aB_2 \mid \lambda\} \cup \{S_1 \to bS_1 \mid \lambda\} \cup \\ &\quad \cup \{S_3 \to \lambda \mid aB_2,\ B_2 \to b \mid bS_3\} = \\ &= \{S \to bS_1 \mid aB_2 \mid \lambda,\ S_1 \to bS_1 \mid \lambda,\ S_3 \to aB_2 \mid \lambda,\ B_2 \to bS_3 \mid b\} \end{aligned}$$

<u>Esercizio 7.2</u>

Data la seguente grammatica lineare destra:

$$G = (X, V, S, P)$$

ove:

$$X = \{a, b, c\} \qquad V = \{S, A, B\}$$
$$P = \left\{ S \overset{(1)}{\to} bA \mid aS \mid b,\ \ A \overset{(2)}{\to} aB \mid cS \mid a,\ \ B \overset{(3)}{\to} bA \mid cB \mid c \right\}$$

Determinare un'espressione regolare che denota il linguaggio $L(G)$.

Senza rischio di confusione, denotiamo con A, B ed S gli insiemi delle stringhe derivabili in G dai nonterminali A, B ed S, rispettivamente.

Dalla produzione (2) risulta:

$$A = \{a\} \cdot B \cup \{c\} \cdot S \cup \{a\}$$

che, per brevità, scriviamo nella forma:

$$A = aB \cup cS \cup a \qquad (2')$$

Sostituendo in (1) la (2′), si ha:

$$S = b \cdot (aB \cup cS \cup a) \cup aS \cup b$$

che, per la proprietà distributiva della concatenazione rispetto all'unione, è uguale a:

$$S = baB \cup bcS \cup ba \cup aS \cup b$$

Applicando la proprietà commutativa per l'unione, la proprietà associativa per la concatenazione e la distributività della concatenazione rispetto all'unione si ha:

$$S = baB \cup (bc \cup a)S \cup (ba \cup b) \qquad (1')$$

Sostituendo in (3) la (2′), si ottiene:

$$B = b \cdot (aB \cup cS \cup a) \cup cB \cup c = (ba \cup c)B \cup (bcS \cup ba \cup c) \qquad (3')$$

Osserviamo la (1′) e la (3′).

Se denotiamo con *A*, *B* ed *S* tre espressioni regolari (oltre che gli insiemi di stringhe derivabili dai nonterminali *A*, *B* ed *S*), si ha che l'identità tra espressioni regolari corrispondente alla (1′) è:

$$S = baB + (bc + a)S + (ba + b) \qquad (1'')$$

mentre l'identità tra espressioni regolari corrispondente alla (3′) è:

$$B = (ba + c)B + (bcS + ba + c) \qquad (3'')$$

In entrambi i casi ci troviamo di fronte ad una identità del tipo:

$$R_1 = R_2 \cdot R_1 + R_3 \text{ con } R_2 \neq \lambda .$$

Per la proprietà 20) sulle espressioni regolari, si ha:

$$R_1 = R_2^* \cdot R_3 .$$

Dunque la (3′′) diventa:

$$B = (ba + c)^* \cdot (bcS + ba + c) \qquad (3''')$$

Sostituendo la (3′′′) nella (1′′), si ottiene:

$$S = \left[ba \cdot (ba + c)^* \cdot (bcS + ba + c)\right] + (bc + a)S + (ba + b) =$$
$$= ba \cdot (ba + c)^* bcS + (a + bc)S + ba \cdot (ba + c)^* \cdot (ba + c) + ba + b =$$
$$= \left(ba \cdot (ba + c)^* bc + a + bc\right)S + ba \cdot (ba + c)^* \cdot (ba + c) + ba + b =$$

per la proprietà 20) sulle espressioni regolari

$$= \left(ba \cdot (ba + c)^* bc + a + bc\right)^* \cdot \left(ba \cdot (ba + c)^* (ba + c) + ba + b\right).$$

<u>Esercizio 7.3</u>

Sia *L* il linguaggio denotato dalla seguente espressione regolare:

$$(aa + aaa)^*$$

1) Trovare un automa a stati finiti che riconosce *L*.
2) Trasformare l'automa non deterministico trovato al punto 1) in un automa deterministico equivalente.

1) Determiniamo innanzitutto due grammatiche lineari destre G_1 e G_2 tali che:

$$L(G_1) = S(aa) = \{aa\} \qquad\qquad L(G_2) = S(aaa) = \{aaa\}$$
$$G_1 = (X, V_1, S_1, P_1) \qquad\qquad G_2 = (X, V_2, S_2, P_2)$$
$$X = \{a\}$$
$$V_1 = \{S_1, A\} \qquad\qquad V_2 = \{S_2, B, C\}$$
$$P_1 = \{S_1 \to aA,\ A \to a\} \qquad\qquad P_2 = \{S_2 \to aB,\ B \to aC,\ C \to a\}$$

Determiniamo poi la grammatica lineare destra G_3 tale che:

$$L(G_3) = L(G_1) \cup L(G_2) = S(aa) \cup S(aaa) = S(aa + aaa)$$
$$G_3 = (X, V_3, S_3, P_3)$$

dove:

$$V_3 = V_1 \cup V_2 \cup \{S_3\} = \{S_1, S_2, S_3, A, B, C\}.$$

Le produzioni di G_3 si ottengono come segue:

$$P_3 = \{S_3 \to w \mid S_1 \to w \in P_1\} \cup \{S_3 \to w \mid S_2 \to w \in P_2\} \cup P_1 \cup P_2.$$

Dunque si ha:

$$P_3 = \{S_3 \to aA \mid aB,\ S_1 \to aA,\ A \to a,\ S_2 \to aB,\ B \to aC,\ C \to a\}$$

da cui possiamo eliminare le produzioni che presentano S_1 o S_2 nella parte sinistra (S_1 ed S_2 sono nonterminali inutili perché non sono presenti nella parte destra di nessuna produzione e né S_1 né S_2 sono il simbolo distintivo di G_3). Per cui P_3 diventa:

$$P_3 = \{S_3 \to aA \mid aB,\ A \to a,\ B \to aC,\ C \to a\}$$

Determiniamo ora la grammatica lineare destra G tale che:

$$L(G) = \left(L(G_3)\right)^* = S\left((aa + aaa)^*\right)$$
$$G = (X, V, S, P)$$

dove:

$$V = V_3 \cup \{S\} = \{S, S_3, A, B, C\}$$

Le produzioni di G si ottengono come segue (si veda Tavola 1):

$$\begin{aligned} P = {} & \{S \to \lambda\} \cup (P_3 - \{S_3 \to \lambda\}) \cup \{S \to w \mid S_3 \to w \in P_3\} \cup \\ & \cup \{A \to bS \mid A \to b \in P_3\} \cup \\ & \cup \{A \to bS \mid A \to bB \in P_3,\ b \neq \lambda,\ B \to \lambda \in P_1\}. \end{aligned}$$

Dunque si ha:

$$\begin{aligned} P = {} & \{S \to \lambda\} \cup \{S_3 \to aA \mid aB,\ A \to a,\ B \to aC,\ C \to a\} \cup \\ & \cup \{S \to aA \mid aB\} \cup \{A \to aS,\ C \to aS\} \end{aligned}$$

da cui possiamo eliminare le produzioni che presentano S_3 nella parte sinistra, ottenendo:

$$P = \{S \to aA \mid aB \mid \lambda,\ A \to aS \mid a,\ B \to aC,\ C \to aS \mid a\}.$$

L'automa non deterministico che riconosce $L(G)$ si costruisce come segue:

$$M = (Q, \delta, q_0, F) \qquad \text{con alfabeto di ingresso } X$$

ove, per il relativo algoritmo di trasformazione (Algoritmo 7.1), si ha:

1) $Q = V \cup \{q\} = \{S, A, B, C, q\}$;
2) $q_0 = S$;
3) $F = \{q, S\}$;
4) δ è definita dalla seguente tavola di transizione:

δ	S	A	B	C	q
a	$\{A, B\}$	$\{S, q\}$	$\{C\}$	$\{S, q\}$	$-$

Quindi il grafo degli stati è come in Figura 7.4.

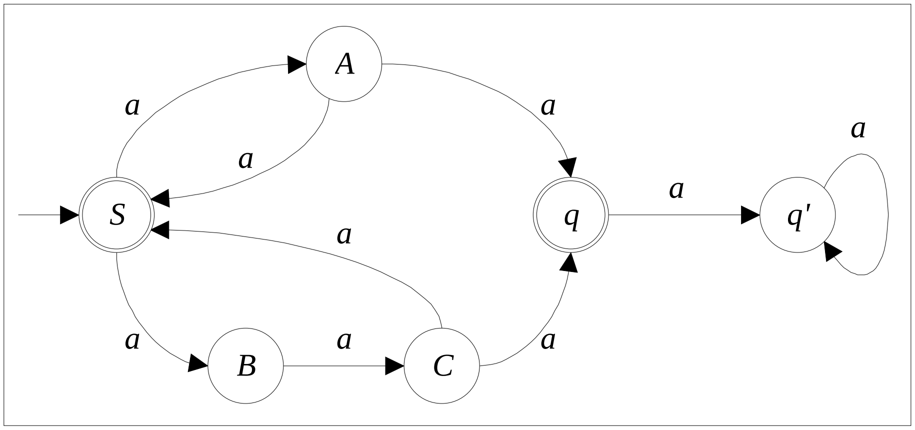

Figura 7.4

2) L'automa accettore deterministico M' equivalente ad M (ossia tale che $T(M') = T(M)$) si costruisce, secondo l'Algoritmo 6.1, come segue:

$$M' = (Q', \delta', q_0', F')$$

ove:

a) $Q' = 2^Q$;

b) $q_0' = \{q_0\}$;

c) $F' = \{p \subseteq Q \mid p \cap F \neq \varnothing\}$;

d) $\delta' : Q' \times X \to Q' \quad \ni'$

$\forall q' = \{q_1, q_2, ... q_i\} \in Q', \ \forall x \in X$:

$\delta'(q', x) = \delta'(\{q_1, q_2, ..., q_i\}, x) =$

$$= \delta(q_1, x) \cup \delta(q_2, x) \cup ... \cup \delta(q_i, x) = \bigcup_{j=1}^{i} \delta(q_j, x)$$

Dunque si ha:

$$M' = (Q', \delta', q_0', F')$$

con:

a) $Q' = 2^Q = 2^{\{S,A,B,C,q\}}, \ |Q'| = 32$;

b) $q_0' = \{S\}$;

c) $F' = \left\{\{q\}, \{S\}, \{q, A\}, \{q, B\}, \{q, C\}, ..., \{S, A\}, \{S, B\}, \{S, C\}, ..., \{q, S\}\right\}$.

Nella pratica, non è necessario considerare tutti gli stati di Q', in quanto molti sono irraggiungibili. Per questo motivo, iniziamo a definire δ' dal nuovo stato iniziale e proseguiamo definendo δ' per un nuovo stato di Q' non appena esso viene generato.

Per cui si ha:

$\delta'(\{S\},a)=\delta(S,a)=\{A,B\}$

$\delta'(\{A,B\},a)=\delta(A,a)\cup\delta(B,a)=\{S,q\}\cup\{C\}=\{S,C,q\}$

$\delta'(\{S,C,q\},a)=\delta(S,a)\cup\delta(C,a)\cup\delta(q,a)=$

$=\{A,B\}\cup\{S,q\}\cup\varnothing=\{S,A,B,q\}$

$\delta'(\{S,A,B,q\},a)=\delta(S,a)\cup\delta(A,a)\cup\delta(B,a)\cup\delta(q,a)=$

$=\{A,B\}\cup\{S,q\}\cup\{C\}\cup\varnothing=$

$=\{S,A,B,C,q\}$

$\delta'(\{S,A,B,C,q\},a)=\delta(\{S,A,B,q\},a)\cup\delta(C,a)=\{S,A,B,C,q\}$

Il grafo degli stati di M' è dato in Figura 7.5.

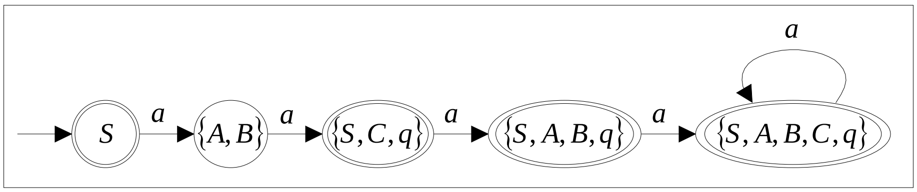

Figura 7.5

Quanto illustrato in precedenza costituisce il modo “standard” di svolgere l’esercizio.
Un modo alternativo è il seguente.

$$\left((aa+aaa)^*\right)$$

Osserviamo che:

$$S\left((aa+aaa)^*\right)=\{\lambda,aa,aaa,aaaa,aaaaa,\ldots\}=\{a\}^*-\{a\}$$

L’automa (minimo e deterministico) che riconosce $\{a\}^*-\{a\}$ è (Figura 7.6):

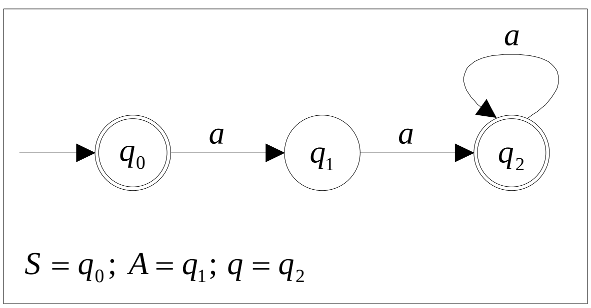

Figura 7.6

corrispondente alla grammatica:

$$S\rightarrow aA\,|\,\lambda$$
$$A\rightarrow aA\,|\,a$$

Si osservi che, data la grammatica G, le cui produzioni sono:

$$S \to aA \mid \lambda$$
$$A \to aA \mid a$$

l'automa (non deterministico) che riconosce $L(G)$ è (Figura 7.7):

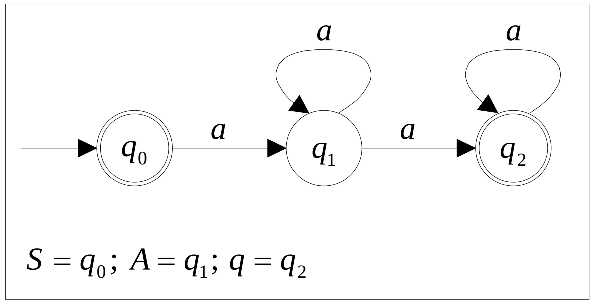

Figura 7.7

che può essere comunque trasformato nell'automa deterministico equivalente in modo molto più semplice:

$\delta'(\{q_0\}, a) = \delta(q_0, a) = \{q_1\}$

$\delta'(\{q_1\}, a) = \delta(q_1, a) = \{q_1, q_2\}$

$\delta'(\{q_1, q_2\}, a) = \delta(q_1, a) \cup \delta(q_2, a) = \{q_1, q_2\}$

per cui l'automa equivalente è (Figura 7.8):

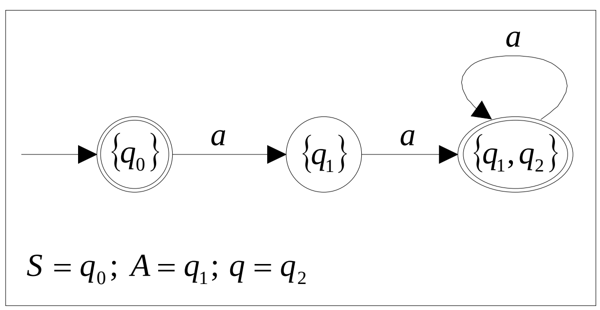

Figura 7.8

Un terzo modo di risolvere l'esercizio:

$$\{a\}^* - \{a\} = \overline{\{a\}}$$

L'automa che riconosce $L = \{a\}$ è dato in Figura 7.9.

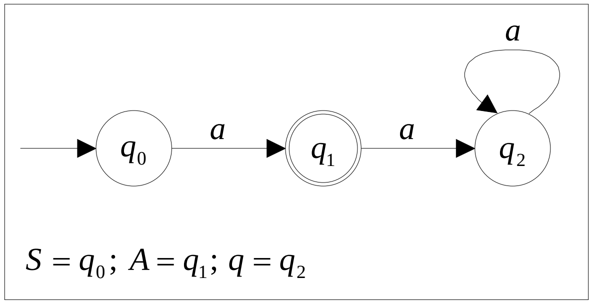

Figura 7.9

L'automa che riconosce $\overline{L} = \overline{\{a\}}$ si costruisce dall'automa che riconosce L considerando come stati di accettazione il complementare di F (Figura 7.10).

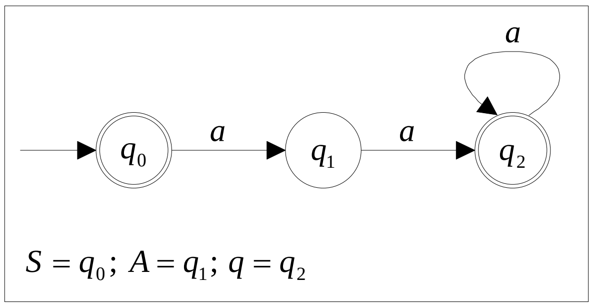

Figura 7.10

Cosa accade se consideriamo l'automa che riconosce $L = \{a\}$ con funzione di transizione δ definita parzialmente?

<u>Esercizio 7.4</u>

Con riferimento all'Esercizio 6.1, determinare una grammatica lineare destra che genera il linguaggio:

$$L = \left\{ w \,\middle|\, w \in \{a,b\}^*,\ w \text{ ha un numero pari di } a \text{ ed un numero dispari di } b \right\}$$

Una grammatica lineare destra che genera il linguaggio L può essere costruita a partire dall'automa accettore a stati finiti determinato nell'Esercizio 6.1 utilizzando l'Algoritmo 7.2, come segue:

$$G = (X, V, S, P)$$

dove:

1) $X = \{a,b\}$;
2) $V = Q = \{q_0, q_1, q_2, q_3\}$;
3) $S = q_0$;
4) $P = \{q_0 \to aq_2 \mid bq_1 \mid b,\ q_1 \to aq_3 \mid bq_0,\ q_2 \to aq_0 \mid bq_3,\ q_3 \to aq_1 \mid bq_2 \mid a\}$.

Esercizio 7.5

Sia L il linguaggio denotato dalla seguente espressione regolare:

$$ab(bb)^* c$$

1) Trovare un automa a stati finiti che riconosce L.
2) Trasformare l'automa non deterministico al punto 1) in un automa deterministico equivalente.

1) Determiniamo una grammatica lineare destra che genera L.

Poiché:

$$\begin{aligned} S\left(ab(bb)^* c\right) &= S(a) \cdot S(b) \cdot S\left((bb)^*\right) \cdot S(c) = \\ &= \{a\} \cdot \{b\} \cdot \{bb\}^* \cdot \{c\} = \\ &= \{ab\} \cdot \{bb\}^* \cdot \{c\} \end{aligned}$$

possiamo determinare una grammatica che genera L sfruttando le proprietà di chiusura dei linguaggi di tipo '3'.

Una grammatica che genera $\{ab\}$ è:

$$G_1 = (X, V_1, S_1, P_1)$$

dove:

- $X = \{a, b\}$;
- $V_1 = \{S_1, A\}$;
- $P_1 = \{S_1 \to aA,\ A \to b\}$.

Una grammatica G_3 che genera $\{bb\}^*$ si ottiene per iterazione dalla grammatica G_2 che genera $\{bb\}$:

$$G_2 = (X, V_2, S_2, P_2)$$

dove:

- $X = \{b\}$;
- $V_2 = \{S_2, B\}$;
- $P_2 = \{S_2 \to bB,\ B \to b\}$.

$$G_3 = (X, V_3, S_3, P_3)$$

dove:

- $X = \{b\}$;
- $V_3 = V_2 \cup \{S_3\} = \{S_2, B, S_3\}$;
- $P_3 = \{S_3 \to bB \mid \lambda,\ B \to bS_3 \mid b,\ \cancel{S_2 \to bB}\}$.[1]

Una grammatica che genera $\{c\}$ è:

$$G_4 = (X, V_4, S_4, P_4)$$

---

[1] La produzione $S_2 \to bB$ viene cancellata in quanto S_2 è un nonterminale inutile.

dove:

- $X = \{c\}$;
- $V_4 = S_4$;
- $P_4 = \{S_4 \rightarrow c\}$.

Una grammatica che genera $\{ab\} \cdot \{bb\}^*$ è:

$$G_5 = (X, V_5, S_5, P_5)$$

dove:

- $X = \{a, b\}$;
- $V_5 = V_1 \cup V_3 = \{S_1, A, S_3, B\}$;
- $S_5 = S_1$;
- $P_5 = \{S_1 \rightarrow aA,\ \ A \rightarrow bS_3,\ \ S_3 \rightarrow bB \mid \lambda,\ \ B \rightarrow bS_3 \mid \mathrm{b}\}$.

Infine, una grammatica che genera il linguaggio $L = \{ab\} \cdot \{bb\}^* \cdot \{c\}$ è:

$$G = (X, V, S, P)$$

dove:

- $X = \{a, b, c\}$;
- $V = V_4 \cup V_5 = \{S_4, S_1, A, S_3, B\}$;
- $S = S_1$;
- $P = \{S_1 \rightarrow aA,\ \ B \rightarrow bS_3 \mid bS_4,\ \ A \rightarrow bS_3 \mid bS_4,\ \ S_3 \rightarrow bB,\ \ S_4 \rightarrow c\}$.

L'automa che riconosce $L(G)$ si costruisce come segue, in base all'Algoritmo 7.1:

$$M = (Q, \delta, q_0, F) \text{ con alfabeto di ingresso } X$$

- $Q = V \cup \{q\},\ \ q \notin V$;
- $q_0 = S_1$;
- $F = \{q\}$;

e il diagramma di transizione è dato in Figura 7.11.

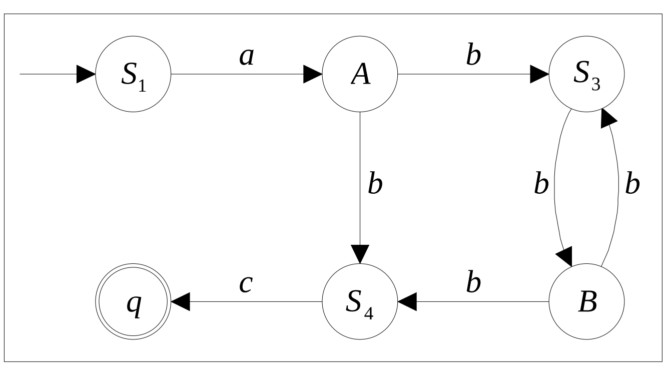

Figura 7.11

2) L'automa deterministico M' equivalente ad M si costruisce come segue, in base all'Algoritmo 6.1:

$$M' = (Q', \delta', q_0', F')$$

dove:

- $Q' = 2^Q$;
- $q_0' = \{S_1\}$;
- $F' = \{p \subseteq Q \mid p \cap F \neq \emptyset\}$ con $|F'| = 32$;
- δ' è definita come segue:

$$\delta' : Q' \times X \to Q'$$

- ♦ $\delta'(\{S_1\}, a) = \delta(S_1, a) = \{A\}$;
- ♦ $\delta'(\{S_1\}, b) = \delta(S_1, b) =$ non è definita;
- ♦ $\delta'(\{S_1\}, c) = \delta(S_1, c) =$ non è definita;
- ♦ $\delta'(\{A\}, a) = \delta(A, a) =$ non è definita;
- ♦ $\delta'(\{A\}, b) = \delta(A, b) = \{S_3, S_4\}$;
- ♦ $\delta'(\{A\}, c) = \delta(A, c) =$ non è definita;
- ♦ $\delta'(\{S_3, S_4\}, a) = \delta(S_3, a) \cup \delta(S_4, a) =$ non è definita;
- ♦ $\delta'(\{S_3, S_4\}, b) = \delta(S_3, b) \cup \delta(S_4, b) = \{B\}$;
- ♦ $\delta'(\{S_3, S_4\}, c) = \delta(S_3, c) \cup \delta(S_4, c) = \{q\}$;
- ♦ $\delta'(\{B\}, a) = \delta(B, a) =$ non è definita;
- ♦ $\delta'(\{B\}, b) = \delta(B, b) = \{S_3, S_4\}$;
- ♦ $\delta'(\{B\}, c) = \delta(B, c) =$ non è definita;

e il diagramma degli stati è dato in Figura 7.12.

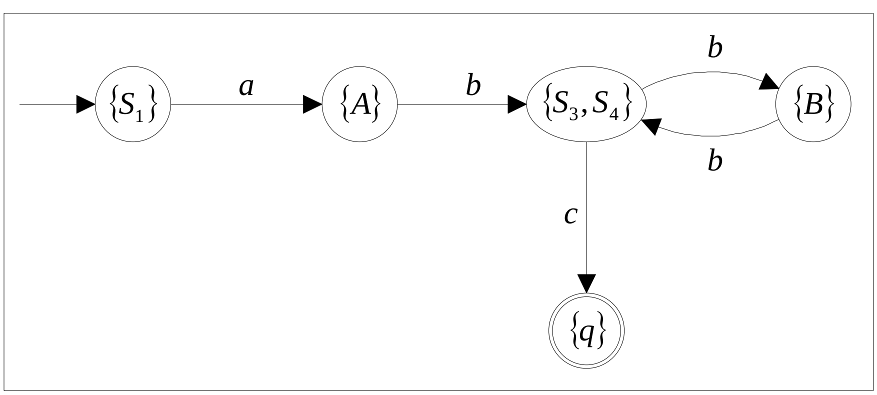

Figura 7.12

Esercizio 7.6

Si consideri la seguente grammatica lineare destra:

$$G = (X, V, S, P)$$

con

$$X = \{a, b\} \qquad V = \{S, B\}$$

$$P = \{S \to aB,\ B \to aB \mid bS \mid a\}$$

Determinare un automa deterministico M tale che:

$$L(G) = T(M)$$

Facciamo riferimento all'algoritmo per la costruzione dell'automa accettore a stati finiti non deterministico equivalente ad una grammatica lineare destra (Algoritmo 7.1).

$$M = (Q, \delta, q_0, F) \text{ con}$$

1) $X = \{a, b\}$ alfabeto di ingresso;

2) $Q = \{S, B, q\}$;

3) $q_0 = S$;

4) $F = \{q\}$;

δ è definita come segue:

5) $\delta : Q \times X \to 2^Q$

5.a) $S \to aB$ dà origine a: $B \in \delta(S, a)$;
$B \to aB$ dà origine a: $B \in \delta(B, a)$;
$B \to bS$ dà origine a: $S \in \delta(B, b)$;

5.b) $B \to a$ dà origine a: $q \in \delta(B, a)$.

Per cui, la tavola di transizione che riassume la definizione della funzione δ è la seguente:

δ	S	B	q
a	$\{B\}$	$\{B, q\}$	$-$
b	$-$	$\{S\}$	$-$

Il grafo degli stati di M è dato in Figura 7.13.

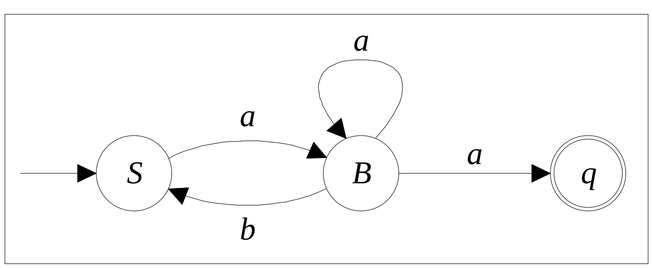

Figura 7.13

L'automa accettore a stati finiti deterministico M' equivalente ad M si costruisce come segue (Algoritmo 6.1):

$$M' = (Q', \delta', q_0', F') \qquad X = \{a, b\} \text{ alfabeto di ingresso}$$

con:

i) $Q' = 2^Q = 2^{\{S,B,q\}}$;

ii) $q_0' = \{S\}$;

iii) $F' = \{\{q\}, \{q, S\}, \{q, B\}, \{q, S, B\}\}$;

iv) δ' è definita come segue:

$$\delta' : Q' \times X \to Q'$$

- $\delta'(\{S\}, a) = \delta(S, a) = \{B\}$;
- $\delta'(\{S\}, b) = \delta(S, b) =$ non definita;
- $\delta'(\{B\}, a) = \delta(B, a) = \{B, q\}$;
- $\delta'(\{B\}, b) = \delta(B, b) = \{S\}$;
- $\delta'(\{B, q\}, a) = \delta(B, a) \cup \delta(q, a) = \{B, q\}$;
- $\delta'(\{B, q\}, b) = \delta(B, b) \cup \delta(q, b) = \{S\}$.

La tavola di transizione che riassume la definizione della funzione δ' è la seguente:

δ'	$\{S\}$	$\{B\}$	$\{B,q\}$
a	$\{B\}$	$\{B,q\}$	$\{B,q\}$
b	$-$	$\{S\}$	$\{S\}$

Il grafo degli stati è indicato in Figura 7.14.

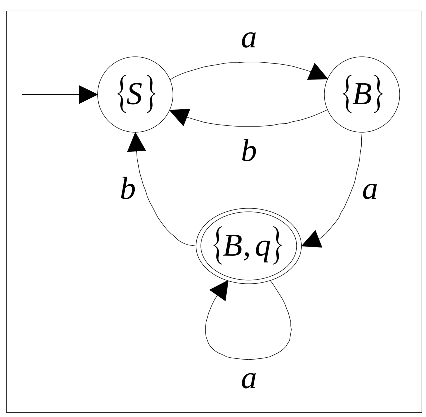

Figura 7.14

Esercizio 7.7

Con riferimento all'Esercizio 6.2, determinare una grammatica lineare destra che genera il linguaggio:

$$L = \left\{ w \,\middle|\, \begin{array}{l} w \in \{a,b\}^*,\ w \neq \alpha aa\beta,\ \alpha,\beta \in \{a,b\}^* \\ (w \text{ non contiene due } a \text{ consecutive}) \end{array} \right\}$$

Una grammatica lineare destra che genera il linguaggio L può essere costruita a partire dall'automa accettore a stati finiti determinato nell'Esercizio 6.2, utilizzando l'Algoritmo 7.2, come segue:

$$G = (X, V, S, P)$$

con:

- $X = \{a,b\}$;
- $V = Q = \{q_0, q_1, q_2\}$;
- $S = q_0$;
- $P = \{q \to xq' \mid q' \in \delta(q,x)\} \cup \{q \to x \mid \delta(q,x) \in F\}$
  $= \{q_0 \to bq_0 \mid aq_1 \mid a \mid b \mid \lambda,\ q_1 \to bq_0 \mid \cancel{aq_2} \mid b,\ q_2 \to \cancel{bq_2} \mid \cancel{aq_2}\} =$
  $= \{q_0 \to bq_0 \mid aq_1 \mid a \mid b \mid \lambda,\ q_1 \to bq_0 \mid b\}$.

Esercizio 7.8

Si utilizzi il Pumping Lemma per i linguaggi regolari (Teorema 7.2) per dimostrare che i seguenti linguaggi non sono regolari:

1) $L = \{a^k b^k \mid k > 0\}$

2) $L = \{a^n b^m c^k \mid n > k,\ m,n,k \geq 1\}$

1) Supponiamo, per assurdo, che L sia regolare. Questo implica che $\exists\ M = (Q, \delta, q_0, F) :\ T(M) = L$. Supponiamo che $|Q| = n$ con $n > 0$. Consideriamo la seguente parola di L:

$$a^n b^n$$

$$\overbrace{aaaa...a}^{n}bbbb...b$$

L'automa M parte dallo stato q_0 e legge una a per volta. Dopo aver letto la prima a si porta in q_1, dopo la seconda a in q_2, ..., dopo la n-esima a si porta in q_n. Abbiamo dunque $n+1$ stati q_0, $q_1, \dots q_n$, in cui M transita.

Si ha:

$$q_0 \xrightarrow{a} q_1$$
$$q_1 \xrightarrow{a} q_2$$
$$\dots$$
$$q_{n-1} \xrightarrow{a} q_n$$

Poiché M ha solo n stati, due tra gli stati $q_0, q_1, \ldots q_n$ devono coincidere.

Siano, ad esempio, q_i e q_j gli stati coincidenti:

$$q_i = q_j, \qquad i < j$$

Si ha dunque un ciclo nel grafo degli stati di M (Figura 7.15).

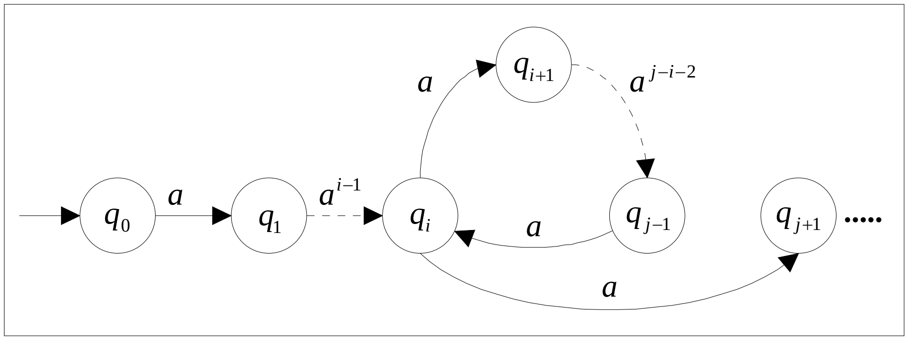

Figura 7.15

Tale ciclo ha lunghezza $j - i$.

Poiché esiste tale ciclo, possiamo aggiungere indefinitamente un numero arbitrario di a nella parola in ingresso, senza modificare l'esito del riconoscimento (vale a dire che la parola viene comunque accettata) se tale numero è un multiplo di $j - i$.

Dunque anche $a^{n+k(j-i)}b^n \in T(M),\ k = 0,1,2,\ldots$.

Ma ciò è in contraddizione con l'ipotesi $T(M) = L$. Dunque L non è regolare.

2) 1° Modo.

Supponiamo per assurdo che:

$$L = \{a^n b^m c^k \mid n > k,\ m,n,k \geq 1\}$$

sia regolare.

Allora esiste un automa accettore a stati finiti $M = (Q, \delta, q_0, F)$ tale che $L = T(M)$. Supponiamo $|Q| = n$ con $n > 0$. Consideriamo le sottostringhe:

$$a^2, a^3, \ldots, a^n, a^{n+1}, a^{n+2}$$

della stringa $a^{n+2}bc$, e gli stati in cui M si porta quando ha in ingresso una delle suddette sottostringhe.

Si ha:

$$a^2 \to q_{a^2}$$
$$a^3 \to q_{a^3}$$
$$\ldots$$
$$a^n \to q_{a^n}$$
$$a^{n+1} \to q_{a^{n+1}}$$
$$a^{n+2} \to q_{a^{n+2}}$$

Se $q_{a^2}, \ldots q_{a^{n+1}}, q_{a^{n+2}}$ fossero tutti distinti tra loro, avremmo $n+1$ stati nell'automa (contro l'ipotesi fatta).

Dunque almeno due stati devono coincidere. Siano a^i e a^j le stringhe per cui M si porta in uno stesso stato e supponiamo $i < j$. Si ha dunque:

$$q_{a^i} = q_{a^j} = q$$

Consideriamo ora la parola di L: $a^j bc^{j-1}$. Poiché $L = T(M)$, tale parola deve essere accettata da M. Ma allora la parola $a^i bc^{j-1}$, $i < j$, viene accettata da M.

Ma questo è un assurdo in quanto $a^i bc^{j-1} \notin L$, $i < j$.

2° Modo.

Supponiamo, per assurdo, che L sia regolare. Questo implica che $\exists\, M = (Q, \delta, q_0, F)$: $T(M) = L$. Supponiamo che $|Q| = n$ con $n > 0$. Consideriamo la seguente parola di L:

$$a^{n+1}bc^n$$

$$aaaa\ldots ab\,\overbrace{cccc\ldots c}^{n}$$

L'automa M parte dallo stato q_0, legge le $n+1$ a e si porta in q, poi legge l'unica b e si porta in q^1. Di seguito, legge la prima c e si porta in q^2, legge la seconda c e si porta in q^3,…, legge la n-esima c e si porta in q^{n+1}. Abbiamo dunque $n+1$ stati $q^1, q^2, \ldots, q^{n+1}$ in cui M transita. Poiché M ha solo n stati, due tra gli stati $q^1, q^2, \ldots, q^{n+1}$ devono coincidere.

Siano q^i e q^j tali stati:

$$q^i = q^j = q, \quad i < j$$

Nel grafo degli stati di M esiste dunque un ciclo di lunghezza $j-i$. L'esistenza di tale ciclo nel grafo di M ci permette di aggiungere altre c nella parola in ingresso, ottenendo ancora parole accettate da M se il numero di c aggiunte è un multiplo di $j-i$. Dunque anche: $a^{n+1}bc^{n+k(j-i)} \in T(M)$. Ma $a^{n+1}bc^{n+k(j-i)} \notin L,\ k = 1,2,\ldots$.

Da cui l'assurdo. Dunque L non è regolare.

Esercizio 7.9

Dimostrare che

$$L = \{a^n b^m c^k \mid m > k,\ \ n, m, k \geq 1\}\ ;$$

non è un linguaggio regolare.

Supponiamo per assurdo che L sia un linguaggio regolare.

Allora esiste un automa accettore a stati finiti $M = (Q, \delta, q_0, F)$ di alfabeto di ingresso $X = \{a, b, c\}$ tale che $L = T(M)$. Supponiamo che $|Q| = n$ con $n > 0$. Consideriamo le seguenti sottostringhe:

$$ab^2, ab^3, \ldots, ab^{n+1}, ab^{n+2}$$

della parola $ab^{n+2}c^{n+1}$ e gli stati in cui M si porta quando ha in ingresso una delle suddette sottostringhe:

$$\begin{gathered} ab^2 \rightarrow q_{ab^2} \\ ab^3 \rightarrow q_{ab^3} \\ \ldots \\ ab^n \rightarrow q_{ab^n} \\ ab^{n+1} \rightarrow q_{ab^{n+1}} \\ ab^{n+2} \rightarrow q_{ab^{n+2}} \end{gathered}$$

Se $q_{ab^2}, q_{ab^3}, \ldots, q_{ab^{n+2}}$ fossero tutti stati distinti di M, avremmo $n+1$ stati nell'automa (contro l'ipotesi che $|Q| = n$). Dunque almeno due stati devono coincidere.

Siano ab^i e ab^j le stringhe per cui M si porta in uno stesso stato e supponiamo $2 \leq i < j \leq n+2$. Dunque si ha:

$$q_{ab^i} = q_{ab^j} = q$$

Consideriamo ora la stringa:

$$ab^j c^{j-1}$$

Evidentemente:

$$ab^j c^{j-1} \in L$$

Poiché $L = T(M)$, tale stringa deve essere accettata da M.

Ma allora anche la stringa $ab^i c^{j-1}$, $i < j$, viene accettata da M (lo stato in cui M si porta quando ha in ingresso $ab^j c^{j-1}$ e $ab^i c^{j-1}$ è lo stesso).

Siamo dunque giunti ad una contraddizione:

$$ab^i c^{j-1} \in T(M) \qquad \text{e} \qquad ab^i c^{j-1} \notin L$$

Ne consegue che L non è regolare.