Esempio 7.1 pag.184

b) $R = a^* \; b \; a^* \; b \; a^*$

Determinare una grammatica *G* di tipo 3 corretta per il linguaggio denotato dall'espressione regolare *R*, cioè tale che *L(G)=S(R)*.

Costruire un automa a stati finiti deterministico *M* con funzione δ totale che riconosce il linguaggio denotato dall'espressione regolare *R*, ossia tale che *T(M)=S(R)*.

Descriviamo il linguaggio $S(R)$ denotato dall'espressione regolare *R*.

$$S(a^* \; b \; a^* \; b \; a^*) = S(a^*) \cdot S(b) \cdot S(a^*) \cdot S(b) \cdot S(a^*) = \left(S(a)\right)^* \cdot S(b) \cdot \left(S(a)\right)^* \cdot S(b) \cdot \left(S(a)\right)^*$$
$$= \{a\}^*\{b\}\,\{a\}^*\{b\}\,\{a\}^*$$

Il linguaggio denotato dall'espressione regolare *R* è

$$S(R) = \{w \in X^* | \#(b, w) = 2\}, \;\; \text{con}\; X = \{a, b\}$$

S(R) può essere riguardato come una sequenza di concatenazione di due linguaggi:

$$L = L_1 L_2 L_1 L_2 L_1$$

$$L_1 = \{a\}^*, \;\; L_2 = \{b\}$$

Per i quali possiamo individuare altrettante grammatiche corrette:

$G_1 = (X, \{S_1\}, S_1, \{S_1 \to \lambda | aS_1\})$ tale che $L(G_1) = L_1$

$G_2 = (X, \{S_2\}, S_2, \{S_2 \to b\})$ tale che $L(G_2) = L_2$

L_1 potrebbe, a sua volta, essere riguardato come iterazione del linguaggio $\{a\}$, ma per brevità e per semplicità del linguaggio evitiamo di applicare il teorema di chiusura di $\mathcal{L}_3$ rispetto all'iterazione.

Osserviamo che le due grammatiche G_1 e G_2 sono di tipo 3. Per il teorema di chiusura di $\mathcal{L}_3$ rispetto alla concatenazione, la grammatica G' corretta per L fornita dalla dimostrazione costruttiva del teorema è ancora una grammatica di tipo 3.

Per la proprietà associativa dell'operazione di concatenazione, possiamo riguardare *L* come segue:

$$L = (L_1 L_2)(L_1 L_2) L_1 = (L_1 L_2)^2 L_1 = L'^2 L_1$$

Possiamo ora determinare una grammatica generativa corretta per il linguaggio $L' = L_1 L_2$:

$G' = (X, \{S_1\}, S_1, P')$, con $P' = \{S_1 \to aS_1 | b\}$ tale che $L(G') = L'$

Consideriamo ora $L'' = L'L' = L'^2$.
Per il teorema di chiusura di $\mathcal{L}_3$ rispetto alla concatenazione, la grammatica G''' corretta per L'' fornita dalla dimostrazione costruttiva del teorema è ancora una grammatica di tipo 3. Per determinare G''' dobbiamo preventivamente 'standardizzare separatamente' G'. Otteniamo così 2 grammatiche equivalenti G' e G'' prive di variabili in comune, ossia tali che tali che $V' \cap V''=\emptyset$.

$$G' = (X, \{S_1'\}, S_1', P')$$

$$P' = \{S_1' \to aS_1' | b\}$$

$$G'' = (X, \{S_1''\}, S_1'', P'')$$

$$P'' = \{S_1'' \to aS_1''|b\}$$

La grammatica G''' corretta per L'' fornita dalla dimostrazione costruttiva del teorema di chiusura di $\mathcal{L}_3$ rispetto alla concatenazione è dunque:

$$G''' = (X, \{S_1', S_1''\}, S_1{}', P''')$$

$$P''' = \{S_1' \to aS_1'\} \cup \{S_1' \to bS_1''\} \cup \emptyset \cup \emptyset \cup P'' = \{S_1' \to aS_1'|bS_1'', S_1'' \to aS_1''|b\}$$

Infine otteniamo una grammatica corretta per il linguaggio L applicando ancora una volta la parte costruttiva della dimostrazione del teorema di chiusura di $\mathcal{L}_3$ rispetto alla concatenazione applicato a L'''=L(G''') e L_1:

$$G = \left(X, \{S_1', S_1'', S_1\}, S_1'{}_1, P\right),$$

$$\begin{aligned} P &= \{S_1' \to aS_1'|bS_1'', S_1'' \to aS_1''\} \cup \{S_1'' \to bS_1\} \cup \emptyset \cup \emptyset \cup \{S_1 \to \lambda|aS_1\} = \\ &= \{S_1' \to aS_1'|bS_1'', S_1'' \to aS_1''|bS_1, S_1 \to \lambda|aS_1\} \end{aligned}$$

Quindi L(G)=L. Un esempio di derivazione in G; deriviamo la parola *bb*:

$$S_1' \Rightarrow bS_1'' \Rightarrow bbS_1 \Rightarrow bb$$

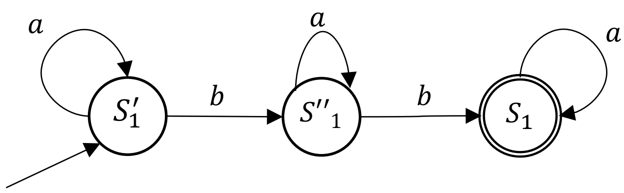
Si vuole adesso costruire un NDA che riconosca il linguaggio L.
Per il teorema di Kleene, $\mathcal{L}_3 \equiv \mathcal{L}_{FSL} \equiv \mathcal{L}_{REG}$. In particolare utilizziamo la dimostrazione relativa a $\mathcal{L}_3 \subset \mathcal{L}_{FSL}$, per cui, ricordando che $\mathcal{L}_{FSL} \equiv \mathcal{L}_{NDL} = \{L \in 2^{X^*} | \exists M, M \text{ è } un\ NDA: L = T(M)\}$, possiamo affermare che esiste sicuramente un NDA che riconosce il linguaggio L. La dimostrazione costruttiva del teorema consiste nell'Algoritmo 7.1 consente di costruire il seguente NDA:

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

con alfabeto di ingresso $X = \{a, b\}$ e definito come segue:

- $Q = V \cup \{q\}, q \notin V = \{S_1', S_1'', S_1, q\}$
- $q_0 = S_1'$
- $F = \{q\} \cup \{S_1\} = \{q, S_1\}$
- $\delta: Q \times X \to 2^Q$ così definito:
  - $S_1' \to aS_1'$ dà luogo a $S_1' \in \delta(S_1', a)$
  - $S_1' \to bS_1''$ dà luogo a $S_1'' \in \delta(S_1', b)$
  - $S_1'' \to aS_1''$ dà luogo a $S_1'' \in \delta(S_1'', a)$
  - $S_1'' \to bS_1$ dà luogo a $S_1 \in \delta(S_1'', b)$
  - $S_1 \to aS_1$ dà luogo a $S_1 \in \delta(S_1, a)$

Il diagramma di transizione per M è il seguente:



Lo stato finale q non è raggiunto da alcuna transizione (è un nodo sconnesso), dunque è possibile ignorarlo. Osserviamo però che la funzione δ non è totale in quanto ci sono transizioni non definite.

Se M si trova nello stato S_1 e legge b, l'automa deve rigettare la parola in quanto nessuna parola del linguaggio $S(R)$ contiene tre b. Pertanto definiamo un nuovo stato q_p, che sarà uno stato pozza e le relative transizioni:

$$\delta(S_1, b) = \{q_p\}, \qquad \delta(q_p, a) = \{q_p\}, \qquad \delta(q_p, b) = \{q_p\}$$

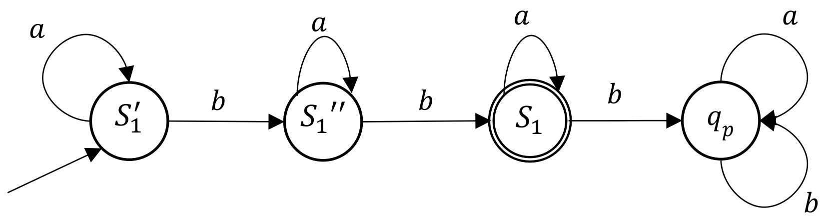
Pertanto l'FSA M' con funzione δ' totale che riconosce L risulta così definito:

$$M' = (\{S'_1, S''_1, S_1, q_p\}, \delta', S_1', \{S_1\})$$

Con funzione δ totale definita come segue:

$\delta'(S'_1, a) = S_1'$	$\delta'(S''_1, a) = S_1''$	$\delta'(S_1, a) = S_1$	$\delta'(q_p, a) = q_p$
$\delta'(S'_1, b) = S_1''$	$\delta'(S_1'', b) = S_1$	$\delta'(S_1, b) = q_p$	$\delta'(q_p, b) = q_p$

Il diagramma di transizione per M' è il seguente:



c) $R = (a + b)^*$

Determinare una grammatica *G* di tipo 3 corretta per il linguaggio denotato dall'espressione regolare *R*, cioè tale che *L(G)=S(R)*.

Costruire un automa a stati finiti deterministico *M* con funzione δ totale che riconosce il linguaggio denotato dall'espressione regolare *R*, ossia tale che *T(M)=S(R)*.

Descriviamo il linguaggio denotato da *R*:

$$S(R) = S((a + b)^*) = \big(S(a + b)\big)^* = \big(S(a) \cup S(b)\big)^* = (\{a\} \cup \{b\})^* = \{a, b\}^*$$

Per cui il linguaggio denotato è $S(R) = \{a, b\}^*$

Una grammatica di tipo 3 per il linguaggio $S(R)$ è

$$G = (X = \{a, b\}, V = \{S\}, S, P = \{S \underset{(1)}{\rightarrow} \lambda, S \underset{(2)}{\rightarrow} aS, S \underset{(3)}{\rightarrow} bS\})$$

Passiamo adesso a costruire un NDA che riconosca il linguaggio *L*. Per il teorema di Kleene, $\mathcal{L}_3 \equiv \mathcal{L}_{FSL} \equiv \mathcal{L}_{REG}$. In particolare risulta che $\mathcal{L}_3 \subset \mathcal{L}_{FSL}$, per cui, ricordando che $\mathcal{L}_{FSL} \equiv \mathcal{L}_{NDL} = \{L \in 2^{X^*} | \exists M, M \text{ è } un\ NDA: L = T(M)\}$, possiamo affermare che esiste sicuramente un NDA che riconosce le parole del linguaggio L. La dimostrazione costruttiva del teorema consiste nell'Algoritmo 7.1 consente di costruire il seguente NDA:

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

con alfabeto di ingresso $X = \{a, b\}$ e definito come segue:

- $Q = V \cup \{q\}, q \notin V = \{S, q\}$
- $q_0 = S$
- $F = \{q\} \cup \{S\} = \{q, S\}$
- $\delta: Q \times X \rightarrow 2^Q$ così definito:
  - $S \in \delta(S, a)$
  - $S \in \delta(S, b)$

$$\delta(S, a) = \{S\} \qquad\qquad \delta(S, b) = \{S\}$$

Osserviamo che lo stato *q* non viene raggiunto da alcuna transizione, dunque è un nodo disconnesso e si può eliminare.

Il diagramma di transizione per *M* è il seguente: