Esercizi

E1. Consideriamo l'AFD $\boldsymbol{D}_0$ su $\Sigma = \{\texttt{a}, \texttt{b}\}$ la cui tabella delle transizioni è qui di seguito riportata.

q	a	b
s	s	1
1	2	3
2	2	3
3	3	3

Tabella delle transizioni di $\boldsymbol{D}_0$

Lo stato di partenza di $\boldsymbol{D}_0$ è s mentre gli stati di accettazione di $\boldsymbol{D}_0$ sono 1 e 2.

(a) Trovare per via algoritmica un AFD $\boldsymbol{D}$ di dimensione minima equivalente a $\boldsymbol{D}_0$.

(b) Facendo uso del Lemma di Arden trovare un'espressione regolare **E** che denoti il linguaggio accettato dall'AFD $\boldsymbol{D}$ del punto (a).

(c) Costruire l'automa di Thomson associato all'espressione **E** del punto (b).

(d) A partire dall'espressione regolare **E** del punto (b), costruire per via algoritmica un AFD $\boldsymbol{D}_1$ che accetti il linguaggio denotato da **E** e verificare con l'algoritmo di Moore che $\boldsymbol{D}_0$ e $\boldsymbol{D}_1$ sono equivalenti.

(e) Dato l'AFD $\boldsymbol{D}$ del punto (a), trovare una grammatica regolare $\boldsymbol{G}$ che generi il linguaggio accettato da $\boldsymbol{D}$.

(f) Data la grammatica regolare $\boldsymbol{G}$ del punto (e), trovare una grammatica acontestuale $\boldsymbol{G}^*$ in forma normale di Chomsky equivalente a $\boldsymbol{G}$.

(g) Data la grammatica acotenstuale $\boldsymbol{G}^*$ del punto (f), applicare l'algorimo CYK per decidere se la stringa `abaa` è una frase di $\boldsymbol{G}^*$; se lo è, trovare tutti i suoi alberi di dervazione.

Soluzione

(a) Dato l'AFD $\boldsymbol{D}_0$ con tabella delle transizioni

q	a	b
s	s	1
1	2	3
2	2	3
3	3	3

in cui s è lo stato di partenza ed 1 e 2 sono stati di accettazione, si ottiene la partizione in stati indistinguibili $Q' = \{s', 1', 2'\}$ dove $s' = \{s\}$, $1' = \{1, 2\}$ e $3' = \{3\}$. Pertanto, un AFD di dimensione minima equivalente a $\boldsymbol{D}_0$ è l'AFD $\boldsymbol{D}$ con tabella delle transizioni

q'	a	b
s'	s'	$1'$
$1'$	$1'$	$2'$
$2'$	$2'$	$2'$

in cui s' è lo stato di partenza ed $1'$ è l'unico stato di accettazione. Si noti che $2'$ uno stato morto.

(b) Per l'AFD $\boldsymbol{D}$ abbiamo il sistema di equazioni

$$\lambda_0 = \{\texttt{a}\} \cdot \lambda_0 \cup \{\texttt{b}\} \cdot \lambda_1$$
$$\lambda_1 = \{\texttt{a}\} \cdot \lambda_1 \cup (\{\texttt{b}\} \cdot \lambda_2 \cup \{\varepsilon\})$$
$$\lambda_2 = \{\texttt{a}, \texttt{b}\} \cdot \lambda_2 \cup \emptyset$$

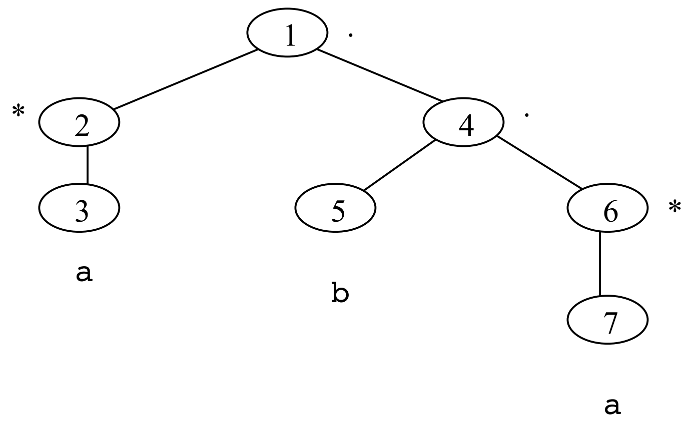
Il Lemma di Arden fornisce la seguente soluzione:

$$\lambda_2 = \{\texttt{a}, \texttt{b}\}^* \cdot \emptyset = \emptyset$$
$$\lambda_1 = \{\texttt{a}\}^* \cdot \{\varepsilon\} = \{\texttt{a}\}^*$$
$$\lambda_0 = \{\texttt{a}\}^* \cdot (\{\texttt{b}\} \cdot \{\texttt{a}\}^*)$$

Pertanto il linguaggio accettato da $\boldsymbol{D}$ è denotato dall'espressione regolare

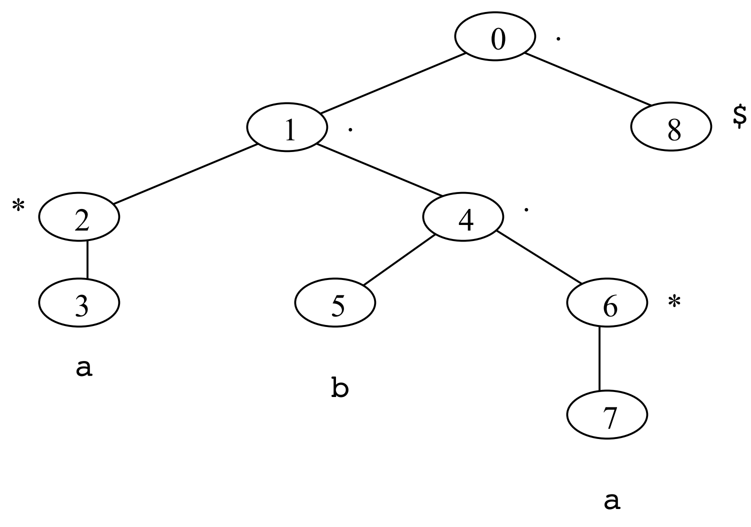
$$\mathbf{E} = ((\texttt{a})^*)((\texttt{b})((\texttt{a})^*)).$$

(*c*) La struttura sintattica dell'espressione regolare **E** è l'albero ordinato mostrato in figura.



A questo punto, con le regole di Thompson è facile costruire un automa finito che accetti il linguaggio denotato da **E**.

(*d*) Consideriamo l'espressione regolare **E**′ = (**E**)(`$`) = (((`a`)*)((`b`)((`a`)*)))(`$`) e la sua struttura sintattica è l'albero ordinato mostrato in figura.



Le foglie sono nell'ordine i vertici

$f_1 = 3$ $f_2 = 5$ $f_3 = 7$ $f_4 = 8.$

La seguente tabella riporta la lista di adiacenza dei nodi 1, 2, 3 e 4.

nodo i	etichetta di i	nodi adiacenti ad i
1	a	1, 2
2	b	3, 4
3	a	3, 4
4	$	Ø

L'AFD che si ottiene è esattamente l'AFD del punto (a).

La verifica dell'equivalenza avviene costruendo la tabella di Moore

(q, q')	$(f(q,\mathtt{a}), f'(q',\mathtt{a}))$	$(f(q,\mathtt{b}), f'(q',\mathtt{b}))$
(s, s')	(s, s')	$(1, 1')$
$(1, 1')$	$(2, 1')$	$(3, 2')$
$(2, 1')$	$(2, 1')$	$(3, 2')$
$(3, 2')$	$(3, 2')$	$(3, 2')$

che è completa, il che dimostra che $\boldsymbol{D}_0$ e $\boldsymbol{D}$ sono equivalenti.

(*e*) All'AFD $\boldsymbol{D}$ resta associata la grammatica regolare $\boldsymbol{G} = (T, N, \mathtt{S}, P_1 \cup P_2)$ con $T = \{\mathtt{a}, \mathtt{b}\}$, $N = \{\mathtt{S}, \mathtt{A}, \mathtt{B}\}$ e

$P_1 = \{\mathtt{S} \to \mathtt{a\ S} \mid \mathtt{b\ A}, \mathtt{A} \to \mathtt{a\ A} \mid \mathtt{b\ B}, \mathtt{B} \to \mathtt{a\ B} \mid \mathtt{b\ B}\}$
$P_2 = \{\mathtt{A} \to \varepsilon\}$.

(*f*) La grammatica $\boldsymbol{G}$ contiene un simbolo improduttivo di prima specie (il simbolo B) e non contiene simboli improduttivi di seconda specie. Pertanto le produzioni $\mathtt{A} \to \mathtt{b\ B}$, $\mathtt{B} \to \mathtt{a\ B}$ e $\mathtt{B} \to \mathtt{b\ B}$ sono inutili e possono essere eliminate da P_1. Ottenianmo così una grammatica equivalente a $\boldsymbol{G}$ con insieme di produzioni

$P = \{\mathtt{S} \to \mathtt{a\ S} \mid \mathtt{b\ A}, \mathtt{A} \to \mathtt{a\ A}, \mathtt{A} \to \varepsilon\}$

che, con l'eliminazione della produzione nulla, diventa

$P' = \{\mathtt{S} \to \mathtt{a\ S} \mid \mathtt{b\ A} \mid \mathtt{b}, \mathtt{A} \to \mathtt{a\ A} \mid \mathtt{a}\}$.

La riduzione in forma normale di Chomsky genera la grammatica $\boldsymbol{G}^* = (T, N^*, \mathtt{S}, P^*)$ con $T = \{\mathtt{a}, \mathtt{b}\}$, $N^* = \{\mathtt{S}, \mathtt{A}, \mathtt{X}, \mathtt{Y}\}$ e

$P^* = \{\mathtt{S} \rightarrow \mathtt{X\ S} \mid \mathtt{Y\ A} \mid \mathtt{b}, \mathtt{A} \rightarrow \mathtt{X\ A} \mid \mathtt{a}, \mathtt{X} \rightarrow \mathtt{a}, \mathtt{Y} \rightarrow \mathtt{b}\}$.

Vogliamo infine decidere se la stringa x = `abaa` è una frase di ***G****. Le sottostringhe $x_{i,j}$ di x sono di seguito riportati.

11 a	12 ab	13 aba	14 aba a
21 b	22 ba	23 baa	-
31 a	32 aa	-	-
41 a	-	-	-

Tabella delle sottostringhe $x_{i,j}$

Con l'algorimo CYK calcoliamo gli insiemi $N_{i,j}$:

11 A, X	12 S	13 S	14 S
21 S, Y	22 S	23 S	-
31 A, X	32 A	-	-
41 A, X	-	-	-

Tabella degli insiemi $N_{i,j}$

e, visto che il simbolo speciale `S` appartiene ad $N_{1,4}$, possiamo concludere che la stringa `abaa` è una frase di ***G**** ed esiste infine un solo albero di derivazione.

E2. Consideriamo l'automa finito $\boldsymbol{A}$ su $\Sigma = \{\texttt{a}, \texttt{b}\}$ la cui tabella delle transizioni è qui di seguito riportata.

stato	a	b
s	$\{s, q\}$	$\{q\}$
q	$\emptyset$	$\{s, q\}$

Tabella delle transizioni di $\boldsymbol{A}$

Lo stato di partenza di $\boldsymbol{A}$ è s mentre q è l'unico stato di accettazione di $\boldsymbol{A}$.

(a) Trovare una grammatica acontestuale $\boldsymbol{G}$ in forma normale di Chomsky che genera il linguaggio accettato da $\boldsymbol{A}$; quindi, verificare che la stringa `bba` è una frase di $\boldsymbol{G}$ e trovare i possibili alberi di derivazione.

(b) Trovare per via algoritmica un AFD $\boldsymbol{D}$ equivalente ad $\boldsymbol{A}$; quindi, a partire da $\boldsymbol{D}$, determinare un'espressione regolare che ne denoti il linguaggio.

Soluzione

(*a*) All'automa finito $\boldsymbol{A}$ resta associata la grammatica regolare $\boldsymbol{G} = (T, N, \texttt{S}, P_1 \cup P_2)$ con $T = \{\texttt{a}, \texttt{b}\}$, $N = \{\texttt{S}, \texttt{Q}\}$ e

$$P_1 = \{\texttt{S} \rightarrow \texttt{aS} \mid \texttt{aQ} \mid \texttt{bQ}, \texttt{Q} \rightarrow \texttt{bS} \mid \texttt{bQ}\}$$
$$P_2 = \{\texttt{Q} \rightarrow \varepsilon\}.$$

Eliminando la produzione nulla si ottiene l'insieme delle produzioni

$$\{\texttt{S} \rightarrow \texttt{aS} \mid \texttt{aQ} \mid \texttt{bQ} \mid \texttt{a} \mid \texttt{b}, \texttt{Q} \rightarrow \texttt{bS} \mid \texttt{bQ} \mid \texttt{b}\}$$

Pertanto, una grammatica acontestuale in forma normale di Chomsky che genera il liguaggio accettato da $\boldsymbol{A}$ è $\boldsymbol{G} = (T, N, \texttt{S}, P)$ con $T = \{\texttt{a}, \texttt{b}\}$, $N = \{\texttt{S}, \texttt{Q}, \texttt{A}, \texttt{B}\}$ e

$$P = \{\texttt{S} \rightarrow \texttt{AS} \mid \texttt{AQ} \mid \texttt{BQ} \mid \texttt{a} \mid \texttt{b}, \texttt{Q} \rightarrow \texttt{BS} \mid \texttt{BQ} \mid \texttt{b}, \texttt{A} \rightarrow \texttt{a}, \texttt{B} \rightarrow \texttt{b}\}\ .$$

Vogliamo ora decidere se la stringa $x = \texttt{bba}$ è una frase di $\boldsymbol{G}$. Le sottostringhe $x_{i,j}$ di x sono di seguito riportati.

11 `b`	12 `bb`	13 `bba`
21 `b`	22 `ba`	-
31 `a`	-	-

Tabella delle sottostringhe $x_{i,j}$

Con l'algoritmo CYK calcoliamo gli insiemi $N_{i,j}$:

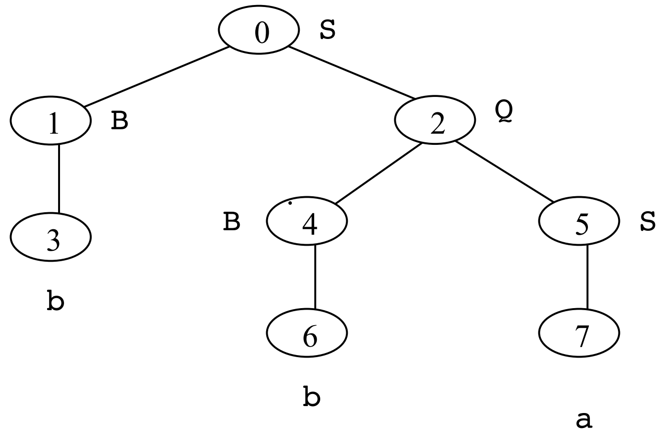
11 S, Q, B	12 S, Q	13 S, Q
21 S, Q, B	22 Q	-
31 S, A	-	-

Tabella degli insiemi $N_{i,j}$

e, visto che il simbolo speciale `S` appartiene ad $N_{1,3}$, possiamo concludere che la stringa `bba` è una frase di ***G***. Con la variante dell'algoritmo CYK calcoliamo anche i possibili alberi di derivazione:

11 S: b Q: b B: b	12 S: B^{11}, Q^{21} Q: B^{11}, S^{21} Q: B^{11}, Q^{21}	13 S: B^{11}, Q^{22} Q: B^{11}, Q^{22}
21 S: b Q: b B: b	22 Q: B^{21}, S^{31}	-
31 S: a A: a	-	-

Pertanto, esiste un solo albero di derivazione della frase `bba` di ***G***:



(*b*) Per trovare un AFD ***D*** equivalente ad ***A***, utilizziamo il seguente algoritmo

1. $\bar{s}$:= $\{s\}$; $\overline{Q}$:= $\{\bar{s}\}$; dichiara $\bar{s}$ "aperto".

2. Fintantoché esiste uno stato aperto $\bar{q}$ in $\overline{Q}$, ripetere

    dichiarare $\bar{q}$ "chiuso";
    per ogni $a \in \Sigma$, ripetere

        $\bar{p} := F(\bar{q}, a)$;
        se $\bar{p} \notin \overline{Q}$, allora
            aggiungere $\bar{p}$ a $\overline{Q}$;
            dichiarare $\bar{p}$ "aperto"

        $f(\bar{q}, a) := \bar{p}$

3. $\overline{R} = \{\bar{q} \in \overline{Q} : \bar{q} \cap R \neq \emptyset\}$.

ed otteniamo l'AFD ***D*** = (Σ, $\overline{Q}$, q_0, $\overline{R}$, f) con solo quattro stati:

stato di D	sottoinsieme di Q
q_0	$\{s\}$
q_1	$\{s, q\}$
q_2	$\{q\}$
q_3	Ø

Stati di ***D***

con stato di partenza q_0 e stati di accettazione q_1 e q_2 e con la seguente tabella delle transizioni

stato	a	b
q_0	q_1	q_2
q_1	q_1	q_1
q_2	q_3	q_1
q_3	q_3	q_3

Tabella delle transizioni di ***D***

Il sistema di equazioni associato ad ***D*** è

$$\lambda_0 = \{\mathtt{a}\} \cdot \lambda_1 \cup \{\mathtt{b}\} \cdot \lambda_2$$
$$\lambda_1 = \{\mathtt{a}, \mathtt{b}\} \cdot \lambda_1 \cup \{\varepsilon\}$$
$$\lambda_2 = \{\mathtt{a}\} \cdot \lambda_3 \cup \{\mathtt{b}\} \cdot \lambda_1 \cup \{\varepsilon\}$$
$$\lambda_3 = \{\mathtt{a}, \mathtt{b}\} \cdot \lambda_3$$

Applicando il Lemma di Arden si ottiene la soluzione

$$\lambda_0 = \{\mathtt{a}, \mathtt{bb}\} \cdot \{\mathtt{a}, \mathtt{b}\}^* \cup \{\mathtt{b}\}$$
$$\lambda_1 = \{\mathtt{a}, \mathtt{b}\}^*$$
$$\lambda_2 = \{\mathtt{b}\} \cdot \{\mathtt{a}, \mathtt{b}\}^* \cup \{\varepsilon\}$$
$$\lambda_3 = \emptyset$$

Pertanto un'espressione regolare che denota il linguaggio accettato da ***A*** è

((a|bb)(a|b)*)|b

E3. Data la grammatica acontestuale $\boldsymbol{G}_o = (T, N_o, \mathtt{S}, P_o)$, dove $T = \{\mathtt{a}, \mathtt{b}\}$, $N_o = \{\mathtt{S}\}$ e $P_o = \{\mathtt{S} \rightarrow \mathtt{aSb} \mid \varepsilon\}$, trovare una grammatica acontestuale $\boldsymbol{G} = (T, N, \mathtt{S}, P)$ in forma normale di Chomsky equivalente a $\boldsymbol{G}_o$, verificare con l'algoritmo CYK che la stringa `aabb` è una frase di ***G*** e trovare i possibili alberi di derivazione di *x*.

Soluzione

$\boldsymbol{G} = (T, N, \mathtt{S}, P)$ dove $N = \{\mathtt{S}, \mathtt{A}, \mathtt{B}, \mathtt{C}\}$ e *P* contiene le produzioni

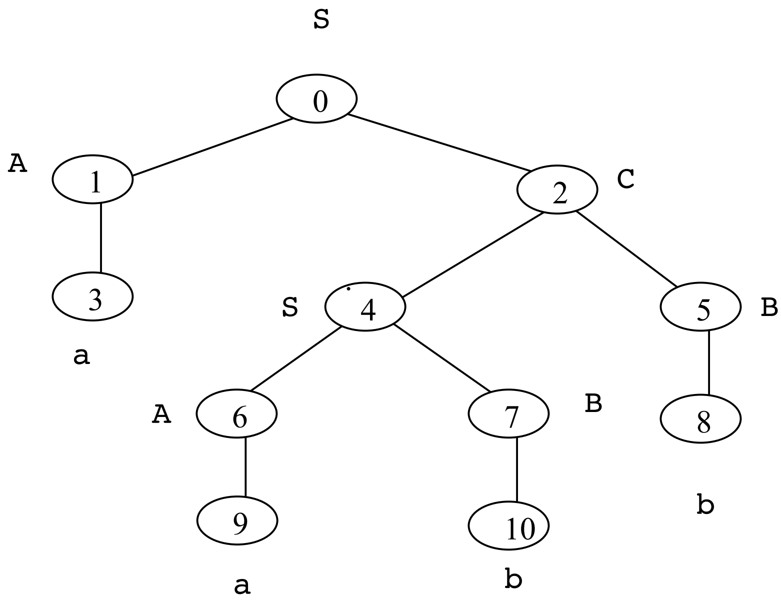
$$\mathtt{S} \rightarrow \mathtt{AC} \mid \mathtt{AB} \mid \varepsilon \qquad \mathtt{C} \rightarrow \mathtt{SB} \qquad \mathtt{A} \rightarrow \mathtt{a} \qquad \mathtt{B} \rightarrow \mathtt{b}$$

L'applicazione dell'algoritmo CYK

11 a	12 aa	13 aab	14 aab b
21 a	22 ab	23 abb	-
31 b	32 bb	-	-
41 b	-	-	-

11 A: a	12 Ø	13 Ø	14 S: A^{11} C^{23}
21 A: a	22 S: A^{21} B^{31}	23 C: S^{22} B^{41}	-
31 B: b	Ø	-	-
41 B: b	-	-	-

e, visto che il simbolo speciale `S` appartiene ad $N_{1,4}$, possiamo concludere che la stringa `aabb` è una frase di ***G***. Inoltre, esiste un solo albero di derivazione della frase `abba` di ***G***:


S
0
A
1
2
C
3
S
4
5
B
a
A
6
7
B
8
9
10
b
a
b

E4. Si consideri AFN $\boldsymbol{A}$ su $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ il cui diagramma delle transizioni è qui di seguito riportato, con 0 stato di partenza e 2 stato di accettazione.

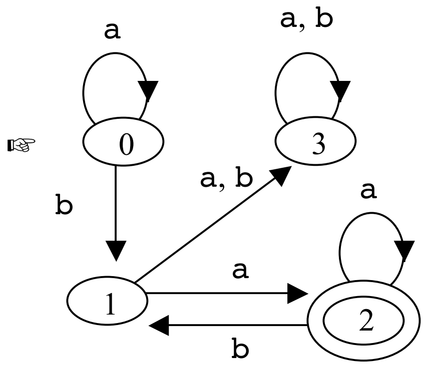


Diagramma delle transizioni di $\boldsymbol{A}$

(*a*) Usando il Lemma di Arden, trovare un'espressione regolare che denoti il linguaggio accettato da $\boldsymbol{A}$.

(*b*) Trovare una rappresentazione canonica del linguaggio accettato da $\boldsymbol{A}$ e, quindi, una grammatica regolare in forma standard che generi il linguaggio accettato da $\boldsymbol{A}$.

(*c*) Dopo aver semplificato e normalizzato la grammatica trovata al punto (2), vista questa volta come grammatica acontestuale, costruire (con l'algoritmo CYK) tutti gli alberi di derivazione della stringa `ababa`.

Soluzione

(*a*) Abbiamo il sistema di equazioni

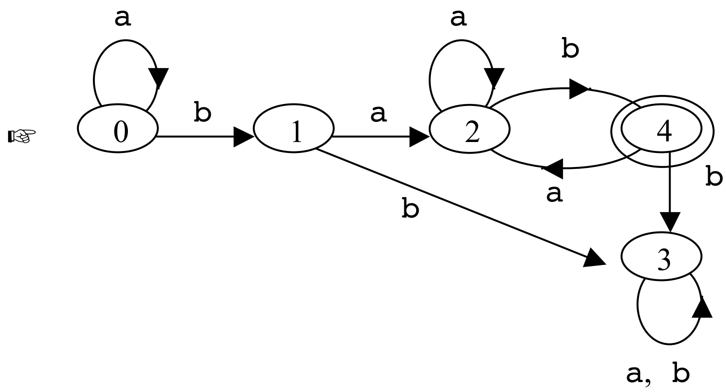
$$\lambda_0 = \{\mathtt{a}\} \cdot \lambda_0 \cup \{\mathtt{b}\} \cdot \lambda_1$$
$$\lambda_1 = \{\mathtt{a}\} \cdot \lambda_2 \cup \{\mathtt{a}, \mathtt{b}\} \cdot \lambda_3$$
$$\lambda_2 = \{\mathtt{b}\} \cdot \lambda_1 \cup \{\mathtt{a}\} \cdot \lambda_2 \cup \{\varepsilon\}$$
$$\lambda_3 = \{\mathtt{a}, \mathtt{b}\} \cdot \lambda_3$$

che ammette la soluzione

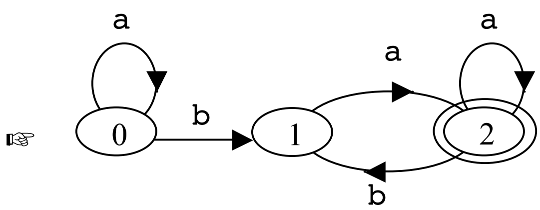
$\lambda_3 = \emptyset$ $\quad\lambda_2 = \{\mathtt{a}, \mathtt{ba}\}^*$ $\quad\lambda_1 = \{\mathtt{a}\}\ \{\mathtt{a}, \mathtt{ba}\}^*$ $\quad\lambda_0 = \{\mathtt{a}\}^*\ \{\mathtt{ba}\}\ \{\mathtt{a}, \mathtt{ba}\}^*$

da cui l'espressione regolare `a*ba(a|ba)*`.

(*b*) L’AFD ***D*** riportato in figura èequivalente ad ***A***.



Per trovare una grammatica regolare generativa del linguaggio *L*(***D***) conviene utilizzare una sua rappresentazione canonica. Siccome gli stati 1 e 4 sono indistinguibili (Algoritmo di Moore) e lo stato 3 è uno stato morto, una rappresentazione canonica di *L*(***D***) è



ed una grammatica regolare generativa di *L*(***D***) è

S → aS | bX
X → aY
Y → aY | bX | ε

che è già in forma normale.

(*c*) Una grammatica acontestuale generativa di *L*(***D***) in forma normale di Chomsky è

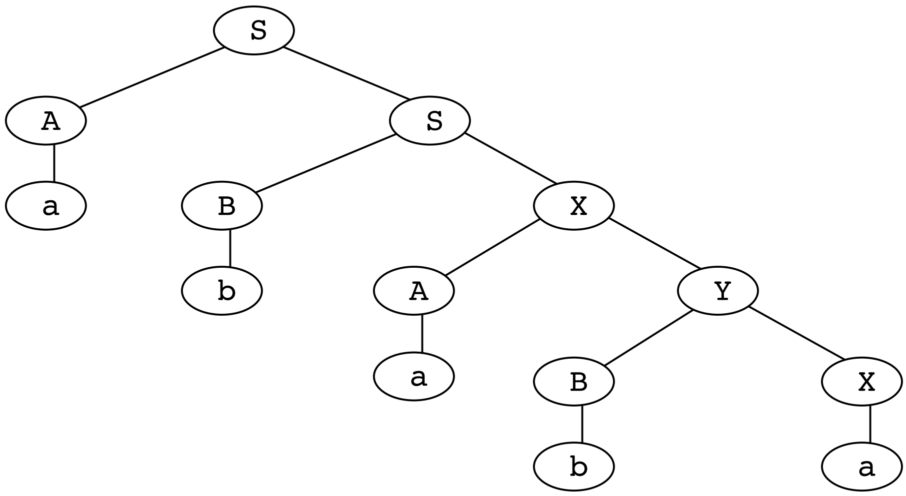
S → AS | BX
X → a | AY
Y → a | AY | BX
A → a
B → b

Quando si applica l'algoritmo CYK alla stringa `ababa` si ottiene la tabella degli insiemi $N_{i,j}$

11 X Y A	12 Ø	13 S X Y	14 Ø	15 S X Y
21 B	22 S Y	23 Ø	24 S Y	
31 X Y A	32 Ø	33 S X Y		
41 B	42 S Y			
51 X Y A				

Tabella degli insiemi $N_{i,j}$

A questo punto, con l'ausilio delle tabelle delle foreste si ottiene il solo albero di derivazione:



E5. Data la grammatica ***G*** con le due produzioni

$S \rightarrow \varepsilon \mid \mathtt{a}S\mathtt{b}$

1. Provare che ***G*** appartiene alla classe *LL*(1). Con il metodo deduttivo costruire un albero di derivazione per la stringa `aabb`.

2. Provare che ***G*** appartiene alla classe *LR*(1). Con il metodo induttivo provare che la stringa vuota e la stringa `aabb` sono frasi di ***G***.

Soluzione

1. Per i corpi delle due produzioni di ***G*** abbiamo

$\mathcal{I}(\varepsilon) = \{\varepsilon\}$ $\qquad\qquad$ $\mathcal{I}(\mathtt{a}S\mathtt{b}) = \{\mathtt{a}\}$

e per l'unico simbolo nonterminale di ***G*** abbiamo

$\mathcal{J}(S) = \{\mathtt{b}, \$\}$

Il criterio di appartenenza ad $LL(1)$ applicato alle due produzioni di ***G*** richiede che

$\mathcal{I}(\varepsilon) \cap \mathcal{I}(\mathtt{a}S\mathtt{b}) = \varnothing$

se $\varepsilon \in \mathcal{I}(\mathtt{a}S\mathtt{b})$ allora $\mathcal{I}(\varepsilon) \cap \mathcal{J}(S) = \varnothing$

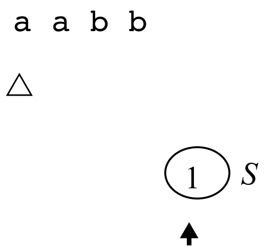
se $\varepsilon \in \mathcal{I}(\varepsilon)$ allora $\mathcal{I}(\mathtt{a}S\mathtt{b}) \cap \mathcal{J}(S) = \varnothing$

Tutte e tre le condizioni sono soddisfatte e quindi ***G*** appartiene alla classe $LL(1)$. Esplicitamente, la tabella di controllo è

simbolo	a	b	$
S	aSb	ε	ε

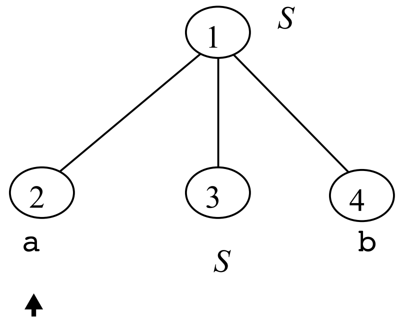
e non contiene celle multiple.

Consideriamo ora la stringa aabb. Inizialmente la situazione è la seguente



Cominciamo ad applicare la procedura di *sviluppo*:

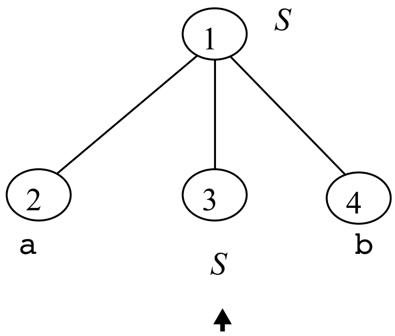
Passo 1. Il nodo di controllo viene sviluppato utilizzando la produzione $S \rightarrow \mathtt{a}S\mathtt{b}$



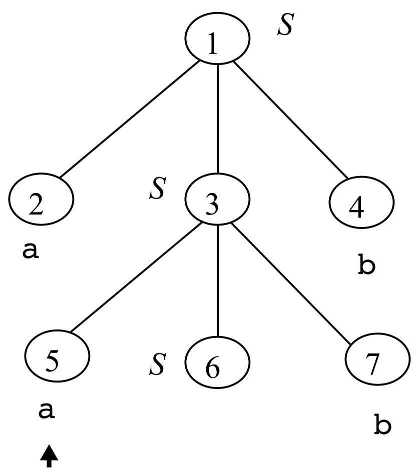
Passo 2. Siccome il nodo di controllo ha per etichetta un simbolo terminale (a) e questo coincide con il simbolo corrente della stringa di ingresso, allora il cursore △ e la sonda ⬆ vengono entrambi fatti avanzare: il nuovo simbolo corrente è a ed il nuovo nodo di controllo è il vertice 3:

a a b b

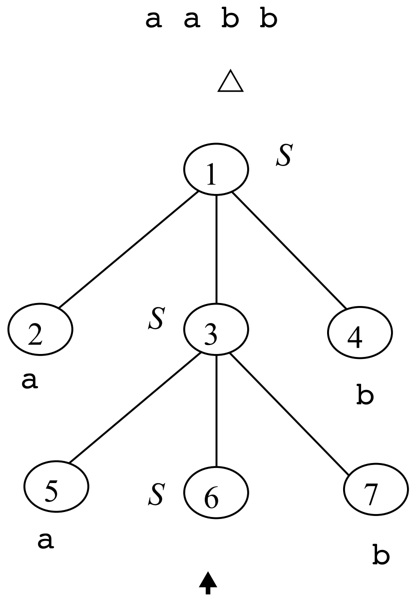
△



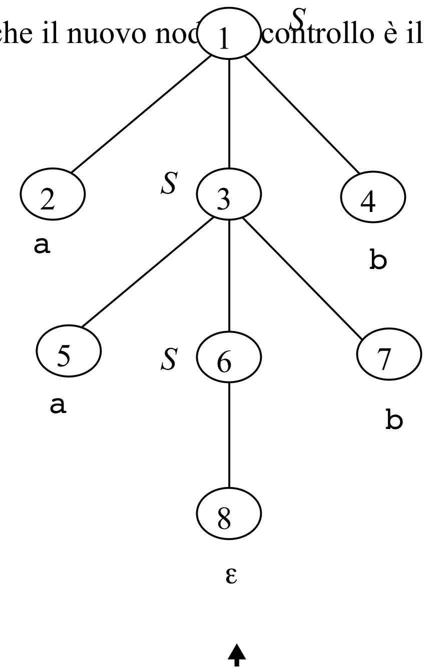
Passo 3. Il nodo di controllo viene sviluppato utilizzando la produzione $S \rightarrow \mathtt{a}S\mathtt{b}$



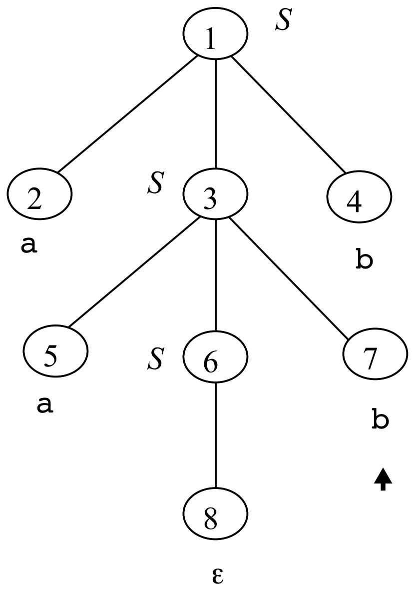
Passo 4. Il nodo di controllo ha per etichetta un simbolo terminale (a) che coincide con il simbolo corrente della stringa di ingresso; così, il cursore △ e la sonda 🠉 vengono entrambi fatti avanzare: il nuovo simbolo corrente è b ed il nuovo nodo di controllo è il vertice 6.



Passo 5. Il nodo di controllo viene sviluppato utilizzando la produzione $S \to \varepsilon$ e la sonda viene fatta avanzare sì che il nuovo nodo di controllo è il vertice 8.



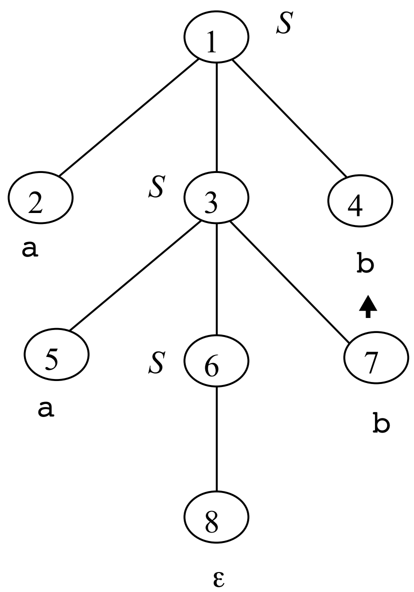
Passo 6. Il nodo di controllo ha per etichetta la stringa vuota (ε) e la sonda viene fatta risalire al vertice 7.



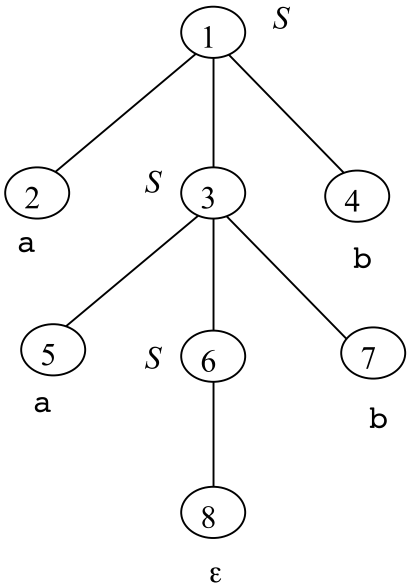
Passo 7. Il nodo di controllo ha per etichetta un simbolo terminale (b) che coincide con il simbolo corrente della stringa di ingresso; così, il cursore △ e la sonda 🠉 vengono entrambi fatti avanzare: il nuovo simbolo corrente è b ed il nuovo nodo di controllo è il vertice 4.

a a b b

△



Passo 8. Il nodo di controllo ha per etichetta un simbolo terminale (`b`) che coincide con il simbolo corrente della stringa di ingresso. Essendo il simbolo corrente l'ultimo della stringa d'ingresso ed essendo il nodo di controllo l'ultima foglia, il processo termina. Così, si è ottenuto l'albero di derivazione



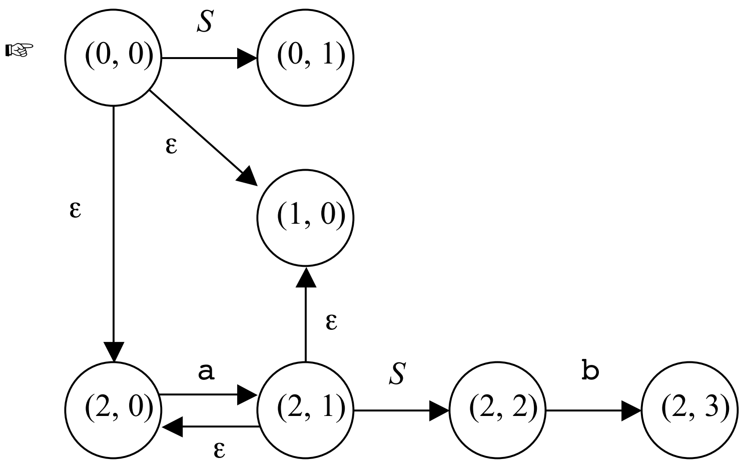
2. Innanzitutto si costruisce la grammatica estesa $\boldsymbol{G'}$

(0) $S' \rightarrow S$
(1) $S \rightarrow \varepsilon$
(2) $S \rightarrow \mathtt{a}S\mathtt{b}$

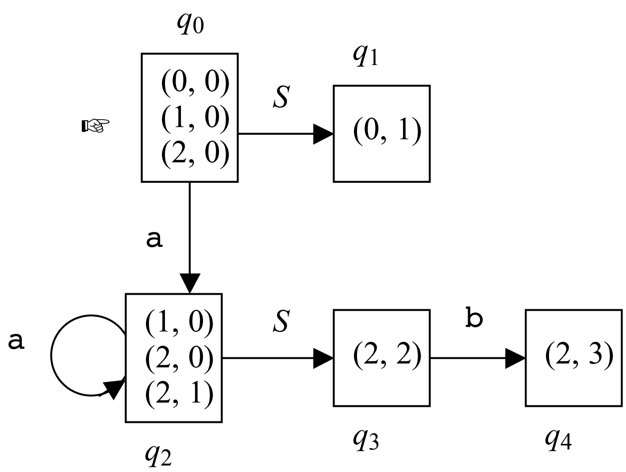
e si determinano le fasi delle tre produzioni:

$(0, 0) = (S' \rightarrow S, 0)$ $(0, 1) = (S' \rightarrow S, 1)$
$(1, 0) = (S \rightarrow \varepsilon, 0)$
$(2, 0) = (S \rightarrow \mathtt{a}S\mathtt{b}, 0)$ $(2, 1) = (S \rightarrow \mathtt{a}S\mathtt{b}, 1)$ $(2, 2) = (S \rightarrow \mathtt{a}S\mathtt{b}, 2)$ $(2, 3) = (S \rightarrow \mathtt{a}S\mathtt{b}, 3)$

A questo punto si costruiscono l'automa ***A*** delle fasi



e l'automa ***D*** delle preformule



Infine viene costruita la tabella delle azioni tendendo presente che $\mathcal{J}(S) = \{\texttt{b}, \texttt{\$}\}$. Così, agli stati di ***D*** vengono associate le azioni come appresso specificato.

(q_0) Lo stato q_0 di ***D*** contiene i tre stati di ***A***

$$(0, 0) = (S' \rightarrow S, 0) \quad (1, 0) = (S \rightarrow \varepsilon, 0) \quad (2, 0) = (S \rightarrow \texttt{a}S\texttt{b}, 0)$$

Lo stato (0, 0) di ***A*** è inefficace. Lo stato (1, 0) di ***A*** fa sì che alle coppie (q_0, b) e (q_0, \$) venga associata l'azione $\mathsf{R}(S\rightarrow\varepsilon)$. Lo stato (2, 0) di ***A*** fa sì che alla coppia (q_0, a) venga associata l'azione T.

(q_1) Lo stato q_1 di ***D*** contiene il solo stato (0, 1) = ($S' \rightarrow S$, 1) di ***A*** che fa sì che alla coppia (q_1, $) venga associata l'azione A.

(q_2) Lo stato q_2 di ***D*** contiene i tre stati di ***A***

(1, 0) = ($S \rightarrow \varepsilon$, 0) (2, 0) = ($S \rightarrow$ aSb, 0) (2, 1) = ($S \rightarrow$ aSb, 1)

Lo stato (1, 0) di ***A*** fa sì che alle coppie (q_2, b) e (q_2, $) venga associata l'azione R($S \rightarrow \varepsilon$). Lo stato (2, 0) di ***A*** fa sì che alla coppia (q_2, a) venga associata l'azione T. Lo stato (2, 1) di ***A*** è inefficace.

(q_3) Lo stato q_3 di ***D*** contiene il solo stato (2, 2) = ($S \rightarrow$ aSb, 2) di ***A*** che fa sì che alla coppia (q_3, b) venga associata l'azione T.

(q_4) Lo stato q_4 di ***D*** contiene il solo stato (2, 3) = ($S \rightarrow$ aSb, 3) di ***A***

che fa sì che alle coppie (q_4, b) e (q_4, $) venga associata l'azione R($S \rightarrow$ aSb).

Dunque, la tabella delle azioni è

stato	S	a	b	$
q_0		T	R(1)	R(1)
q_1				A
q_2		T	R(1)	R(1)
q_3			T	
q_4			R(2)	R(2)

Riduzione sinistrorsa-destrorsa della stringa vuota

Il processo di riduzione si sviluppa nei seguenti passi.

pila	*buffer*	*azione*
q_0	$	R(1)
q_0Sq_1	$	A

Riduzione sinistrorsa-destrorsa della stringa aabb

Il processo di riduzione si sviluppa nei seguenti passi.

pila	*buffer*	*azione*
q_0	aabb$	T
q_0aq_2	abb$	T
q_0aq_2aq_2	bb$	R(1)
q_0aq_2aq_2Sq_3	bb$	T
q_0aq_2aq_2Sq_3bq_4	b$	R(2)
q_0aq_2Sq_3	b$	T
q_0aq_2Sq_3bq_4	$	R(2)
q_0Sq_1	$	A

Testi d'esame

T.1 La grammatica acontestuale ***G*** con produzioni

$$\mathtt{S} \to \mathtt{AaAb} \mid \mathtt{BbBa},\ \mathtt{A} \to \varepsilon,\ \mathtt{B} \to \varepsilon$$

(nessuna delle quali è inutile) genera il linguaggio $L = \{\mathtt{ab}, \mathtt{ba}\}$.

(1) Sia ***G′*** la grammatica risultante dall'eliminazione delle produzioni nulle. Si riporti l'algoritmo che elimina le produzioni inutili e lo si applichi a ***G′***. Sia ***G″*** la grammatica in forma normale di Chomsky equivalente a ***G′***. Si applichi l'algoritmo CYK per determinare gli alberi di derivazione di `ab` e `ba`.

(2) Verificare che ***G*** appartiene alla classe *LL*(1) ed applicare il metodo deduttivo per costruire alberi di derivazione per le stringhe `ab` e `ba`.

(3) Verificare che ***G*** non appartiene alla classe *LR*(1). Utilizzando il Lemma di Arden, determinare il linguaggio accettato dall'automa deterministico ***D*** delle preformule di ***G***.

(4: *facoltativo*) Verificare che ***G″*** appartiene alla classe *LR*(1) ed applicare il metodo induttivo per costruire alberi di derivazione per le stringhe `ab` e `ba`.

Soluzione

(1) La grammatica ***G′*** risultante dall'eliminazione delle produzioni nulle di ***G*** contiene solo le due produzioni $\mathtt{S} \to \mathtt{ab} \mid \mathtt{ba}$. La grammatica ***G″*** in forma normale di Chomsky equivalente a ***G′*** ha le quattro produzioni

$$\mathtt{S} \to \mathtt{XY} \mid \mathtt{YX} \qquad \mathtt{X} \to \mathtt{a} \qquad \mathtt{Y} \to \mathtt{b}$$

L'applicazione dell'algoritmo CYK alle due stringhe `ab` e `ba` è ovvia.

(2)

Algoritmo 10.1

1. Per ogni simbolo grammaticale X di $\boldsymbol{G}$, porre $\mathcal{I}(X) := \{X\}$ se X è un simbolo terminale e $\mathcal{I}(X) =: \emptyset$ altrimenti.

2. Finché per nessun simbolo nonterminale A l'insieme $\mathcal{I}(A)$ possa essere ulteriormente ampliato, ripetere

   per ogni produzione nonnulla $A \to \alpha$ di $\boldsymbol{G}$, sia β il più lungo prefisso di α che contiene solo simboli nonterminali annullabili;

   se $\beta \neq \varepsilon$ allora, per ogni simbolo B in β, porre
   $\mathcal{I}(A) := \mathcal{I}(A) \cup \mathcal{I}(B)$;

   se $\beta \neq \alpha$ allora, detto X il $(|\beta|+1)$.esimo simbolo in α, porre
   $\mathcal{I}(A) := \mathcal{I}(A) \cup \mathcal{I}(X)$.

3. Per ogni simbolo nonterminale A annullabile, porre $\mathcal{I}(A) := \mathcal{I}(A) \cup \{\varepsilon\}$.

I simboli nonterminali annullabili di $\boldsymbol{G}$ sono `A` e `B`. Le produzioni nonnulle sono

`S` → `AaAb`
`S` → `BbBa`.

Il più lungo prefisso di `AaAb` che contiene solo simboli nonterminali annullabili è `A` ed il più lungo prefisso di `BbBa` che contiene solo simboli nonterminali annullabili è `B`. Così, il risultato dell'applicazione dell'Algoritmo 10.1 è il seguente:

X	$\mathcal{I}(X)$
a	{a}
b	{b}
A	{ε}
B	{ε}
S	{a, b}

Algoritmo 10.2

1. Se $\alpha = \varepsilon$, allora porre $\mathcal{I}(\alpha) := \{\varepsilon\}$; altrimenti, $\mathcal{I}(\alpha) := \emptyset$.

2. Se $\alpha \neq \varepsilon$, allora

   sia β il più lungo prefisso di α che contiene solo simboli nonterminali annullabili;

   se $\beta \neq \varepsilon$ allora, per ogni simbolo B in β, porre
   $\mathcal{I}(\alpha) := \mathcal{I}(\alpha) \cup (\mathcal{I}(B)\setminus\{\varepsilon\})$;

   se $\beta \neq \alpha$ allora, detto X il $(|\beta|+1)$.esimo simbolo in α, porre
   $\mathcal{I}(\alpha) := \mathcal{I}(\alpha) \cup \mathcal{I}(X)$; altrimenti, porre $\mathcal{I}(\alpha) := \mathcal{I}(\alpha) \cup \{\varepsilon\}$.

Il risultato dell'applicazione dell'Algoritmo 10.2 è il seguente:

α	$\mathcal{I}(\alpha)$
ε	$\{\varepsilon\}$
`a`	{`a`}
`b`	{`b`}
`aAb`	{`a`}
`bBa`	{`b`}
`AaAb`	{`a`}
`BbBa`	{`b`}
…	…

Algoritmo 10.3

1. Porre $\mathcal{J}(S) := \{\$\}$. Per ogni altro simbolo nonterminale A di $\mathbf{G}$, $\mathcal{J}(A) := \emptyset$.

2. Finché per nessun simbolo nonterminale A insieme $\mathcal{J}(A)$ possa essere ulteriormente ampliato, ripetere:

per ogni produzione $A \rightarrow \alpha$ di ***G*** tale che α contenga almeno un simbolo nonterminale per ogni simbolo nonterminale B contenuto in α per ogni suffisso γ di α preceduto da B (cioè $A \rightarrow \beta B\gamma$) $\mathcal{J}(B) := \mathcal{J}(B) \cup (\mathcal{I}(\gamma)\setminus\{\varepsilon\})$; se $\varepsilon \in \mathcal{I}(\gamma)$ allora $\mathcal{J}(B) := \mathcal{J}(B) \cup \mathcal{J}(A)$.

Le produzioni $A \rightarrow \alpha$ tali che α contenga almeno un simbolo nonterminale sono

`S` → `AaAb`
`S` → `BbBa`.

Il corpo della prima produzione contiene il solo simbolo nonterminale `A` ed abbiamo due suffissi preceduti da `A`: γ = `aAb` e γ = `b`. In entrambi i casi, $\varepsilon \notin \mathcal{I}(\gamma)$ cosicché andremo ad applicare $\mathcal{J}(\texttt{A}) := \mathcal{J}(\texttt{A}) \cup \mathcal{I}(\gamma)$.

Il corpo della seconda produzione contiene il solo simbolo nonterminale `B` ed abbiamo due suffissi preceduti da `B`: γ = `bBa` e γ = `a`. In entrambi i casi, $\varepsilon \notin \mathcal{I}(\gamma)$ cosicché andremo ad applicare $\mathcal{J}(\texttt{B}) := \mathcal{J}(\texttt{B}) \cup \mathcal{I}(\gamma)$.

Così, il risultato dell'applicazione dell'Algoritmo 10.3 è il seguente:

A	$\mathcal{J}(A)$
`A`	{`a`, `b`}
`B`	{`a`, `b`}
`S`	{`$`}

Condizione necessaria e sufficiente perché ***G*** sia una grammatica *LL*(1) è che, per ogni coppia di produzioni distinte $A \rightarrow \alpha$ ed $A \rightarrow \beta$ in P, siano soddisfatte tutte e tre le condizioni:

— $\mathcal{I}(\alpha) \cap \mathcal{I}(\beta) = \emptyset$,
— se $\varepsilon \in \mathcal{I}(\beta)$ allora $\mathcal{I}(\alpha) \cap \mathcal{J}(A) = \emptyset$
— se $\varepsilon \in \mathcal{I}(\alpha)$ allora $\mathcal{I}(\beta) \cap \mathcal{J}(A) = \emptyset$.

Ora, siccome per `S` → `AaAb` | `BbBa` abbiamo

— I(AaAb) ∩ I(BbBa) = {a} ∩ {b} = ∅,
— ε ∉ I(BbBa),
— ε ∉ I(AaAb),

possiamo concludere che ***G*** è una grammatica *LL*(1). La tabella di controllo è

simbolo	a	b	$
S	AaAb	BbBa	
A	ε	ε	
B	ε	ε	

L'applicazione del metodo deduttivo per costruire alberi di derivazione per le stringhe ab e ba è ovvia.

(2) Consideriamo la grammatica ***G′*** che si ottiene con l'aggiunta della produzione S′ → S. Si osservi che esistono solo due derivazioni destrorse dal simbolo speciale S′:

(S′, S, AaAb, Aab, ab) (S′, S, BbBa, Bba, ba)

Pertanto le formule grammaticali destrorse di ***G′*** sono:

S′, S, AaAb, BbBa, Aab, Bba, ab, ba

Sia σ un termine di una coderivazione distinto da S′, sia σ′ il termine successivo a σ ed *A* → α la produzione riducente utilizzata per la riduzione (σ, σ′) di σ. Così, σ è della forma λαμ e σ′ della forma λ*A*μ. Chiamiamo λα il *prefisso riducibile* di σ e un qualsiasi prefisso di λα un *prefisso* di σ *coerente* con la riduzione (σ, σ′). Così per la prima coderivazione (ab, Aab, AaAb, S, S′) abbiamo

σ	produzione riducente	prefisso riducibile	prefissi coerenti
ab	A → ε	ε	ε
Aab	A → ε	Aa	ε, A, Aa
AaAb	S → AaAb	AaAb	ε, A, Aa, AaA, AaAb
S	S′ → S	S	ε, S

e per la seconda coderivazione (ba, Bba, BbBa, S, S′) abbiamo

σ	produzione riducente	prefisso riducibile	prefissi coerenti
ba	B → ε	ε	ε
Bba	B → ε	Bb	ε, B, Bb
BbBa	S → BbBa	BbBa	ε, B, Bb, BbB, BbBa
S	S′ → S	S	ε, S

In generale, chiamiamo *preformula* di ***G*** una qualsiasi stringa che sia un prefisso di una formula grammaticale destrorsa σ di ***G′*** con σ ≠ S′ e che sia coerente con la riduzione di σ in una qualche coderivazione che contenga σ.

Per ottenere l'automa deterministico che accetta le preformule di ***G′***, si ordinano le produzioni della grammatica ***G′***

(0) S′ → S
(1) S → AaAb
(2) S → BbBa
(3) A → ε
(4) B → ε

e se ne determinano le fasi:

(0, 0) = (S′ → S, 0)   (0, 1) = (S′ → S, 1)

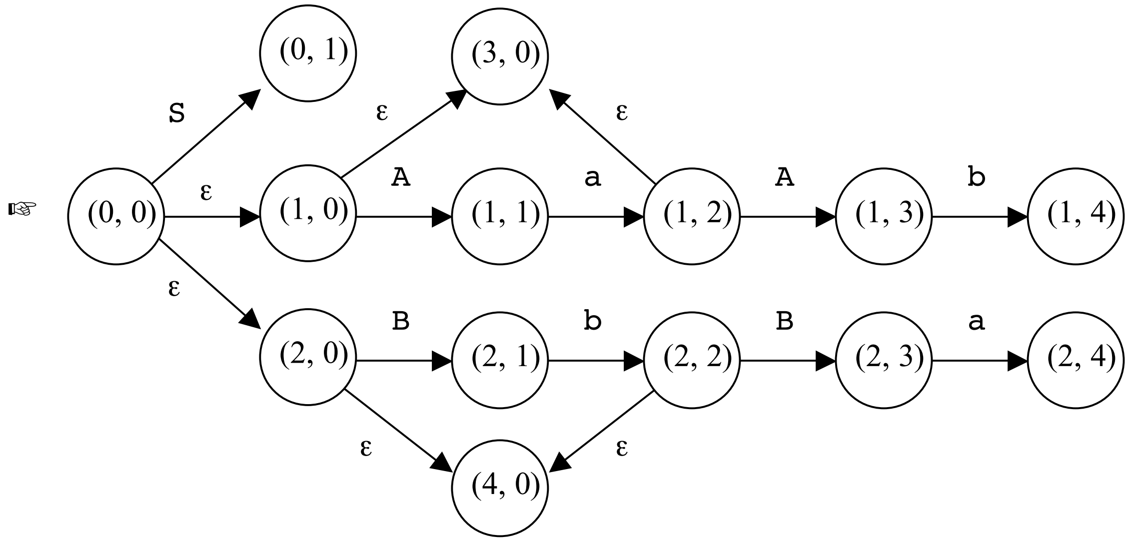
(1, 0) = (S → AaAb, 0)   (1, 1) = (S → AaAb, 1)   (1, 2) = (S → AaAb, 2)
(1, 3) = (S → AaAb, 3)   (1, 4) = (S → AaAb, 4)

(2, 0) = (S → BbBa, 0)   (2, 1) = (S → BbBa, 1)   (2, 2) = (S → BbBa, 2)
(2, 3) = (S → BbBa, 3)   (2, 4) = (S → BbBa, 4)

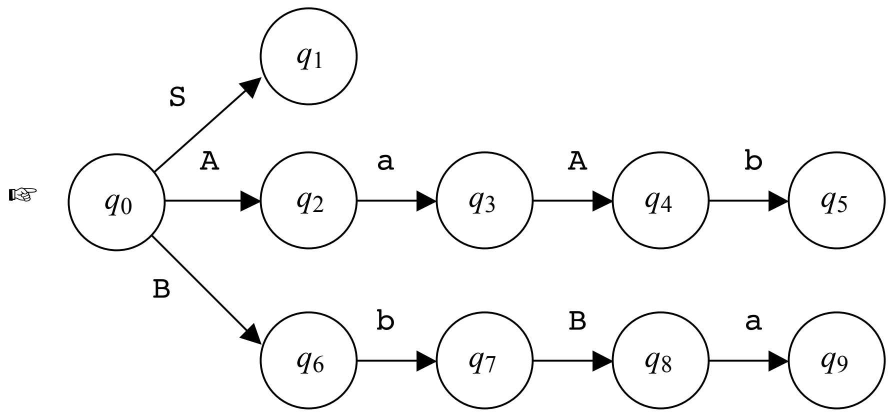
(3, 0) = (A → ε, 0)

(4, 0) = (B → ε, 0)

A questo punto si costruisce l'automa nondeterministico ***A*** che accetta le preformule di ***G′***



cosicché l'automa deterministico ***D*** equivalente ad ***A*** ha il seguente diagramma delle transizioni:



con

$q_0 = \{(0, 0), (1, 0), (2, 0), (3, 0), (4, 0)\}$ $q_1 = \{(0, 1)\}$

$q_2 = \{(1, 1)\}$ $q_3 = \{(1, 2), (3, 0)\}$

$q_4 = \{(1, 3)\}$ $q_5 = \{(1, 4)\}$

$q_6 = \{(2, 1)\}$ $q_7 = \{(2, 2), (4, 0)\}$

$q_8 = \{(2, 3)\}$ $q_9 = \{(2, 4)\}$

Il sistema di equazioni per l'automa deterministico ***D*** è:

$\lambda_0 = \{\mathtt{S}\} \cdot \lambda_1 \cup \{\mathtt{A}\} \cdot \lambda_2 \cup \{\mathtt{B}\} \cdot \lambda_6 \cup \{\varepsilon\}$
$\lambda_1 = \{\varepsilon\}$
$\lambda_2 = \{\mathtt{a}\} \cdot \lambda_3 \cup \{\varepsilon\}$
$\lambda_3 = \{\mathtt{A}\} \cdot \lambda_4 \cup \{\varepsilon\}$
$\lambda_4 = \{\mathtt{b}\} \cdot \lambda_5 \cup \{\varepsilon\}$
$\lambda_5 = \{\varepsilon\}$
$\lambda_6 = \{\mathtt{b}\} \cdot \lambda_7 \cup \{\varepsilon\}$
$\lambda_7 = \{\mathtt{B}\} \cdot \lambda_8 \cup \{\varepsilon\}$
$\lambda_8 = \{\mathtt{a}\} \cdot \lambda_9 \cup \{\varepsilon\}$
$\lambda_9 = \{\varepsilon\}$

cosicché il linguaggio accettato da ***D*** è

$$\lambda_0 = \{\varepsilon, \mathtt{S}, \mathtt{A}, \mathtt{B}, \mathtt{Aa}, \mathtt{Bb}, \mathtt{AaA}, \mathtt{BbB}, \mathtt{AaAb}, \mathtt{BbBa}\}.$$

Infine viene costruita la tabella delle azioni tendendo presente che $\mathcal{J}(\mathtt{S}) = \{\$\}$ e $\mathcal{J}(\mathtt{A}) = \mathcal{J}(\mathtt{B}) = \{\mathtt{a}, \mathtt{b}\}$. Così, agli stati di ***D*** vengono associate le azioni come appresso specificato.

(q_0) Lo stato q_0 di ***D*** contiene i cinque stati di ***A***

$(0, 0) = (\mathtt{S}' \rightarrow \mathtt{S}, 0)$ $(1, 0) = (\mathtt{S} \rightarrow \mathtt{AaAb}, 0)$ $(2, 0) = (\mathtt{S} \rightarrow \mathtt{BbBa}, 0)$
$(3, 0) = (\mathtt{A} \rightarrow \varepsilon, 0)$ $(4, 0) = (\mathtt{B} \rightarrow \varepsilon, 0)$

Gli stati (0, 0), (1, 0) e (2, 0) di ***A*** sono inefficaci. Lo stato (3, 0) di ***A*** fa sì che alle coppie (q_0, a) e (q_0, b) venga associata l'azione R($\mathtt{A} \rightarrow \varepsilon$). Lo stato (4, 0) di ***A*** fa sì che alle coppie (q_0, a) e (q_0, b) venga associata l'azione R($\mathtt{B} \rightarrow \varepsilon$).

(q_1) Lo stato q_1 di ***D*** contiene il solo stato $(0, 1) = (\mathtt{S}' \rightarrow \mathtt{S}, 1)$ di ***A*** che fa sì che alla coppia (q_1, \$) venga associata l'azione A.

(q_2) Lo stato q_2 di ***D*** contiene il solo stato $(1, 1) = (\mathtt{S} \rightarrow \mathtt{AaAb}, 1)$ di ***A*** che fa sì che alla coppia (q_1, a) venga associata l'azione T.

(q_3) Lo stato q_3 di ***D*** contiene gli stati $(1, 2) = (\mathtt{S} \rightarrow \mathtt{AaAb}, 2)$ e $(3, 0) = (\mathtt{A} \rightarrow \varepsilon, 0)$ di ***A***. Lo stato (1, 2) di ***A*** è inefficace mentre lo stato (3, 0) di ***A*** fa sì che alle coppie (q_3, a) e (q_3, b) venga associata l'azione R($\mathtt{A} \rightarrow \varepsilon$).

(q_4) Lo stato q_4 di ***D*** contiene il solo stato $(1, 3) = (\mathtt{S} \rightarrow \mathtt{AaAb}, 3)$ di ***A*** che fa sì che alla coppia (q_4, b) venga associata l'azione T.

(q_5) Lo stato q_5 di ***D*** contiene il solo stato (1, 4) = (`S` → `AaAb`, 4) di ***A*** che fa sì che alla coppia (q_5, `$`) venga associata l'azione R(`S` → `AaAb`).

(q_6) Lo stato q_6 di ***D*** contiene il solo stato (2, 1) = (`S` → `BbBa`, 1) di ***A*** che fa sì che alla coppia (q_6, `b`) venga associata l'azione T.

(q_7) Lo stato q_7 di ***D*** contiene gli stati (2, 2) = (`S` → `BbBa`, 2) e (4, 0) = (`B` → ε, 0) di ***A***. Lo stato (2, 2) di ***A*** è inefficace mentre lo stato (4, 0) di ***A*** fa sì che alle coppie (q_7, `a`) e (q_7, `b`) venga associata l'azione R(`B` → ε).

(q_8) Lo stato q_8 di ***D*** contiene il solo stato (2, 3) = (`S` → `BbBa`, 3) di ***A*** che fa sì che alla coppia (q_8, `a`) venga associata l'azione T.

(q_9) Lo stato q_9 di ***D*** contiene il solo stato (2, 4) = (`S` → `BbBa`, 4) di ***A*** che fa sì che alla coppia (q_9, `$`) venga associata l'azione R(`S` → `BbBa`).

Dunque, la tabella delle azioni è

stato	a	b	$
q_0	R(3) R(4)	R(3) R(4)	
q_1			A
q_2	T		
q_3	R(3)	R(3)	
q_4		T	
q_5			R(1)
q_6		T	
q_7	R(4)	R(4)	
q_8	T		
q_9			R(2)

e contiene due celle, ciascuna con due azioni, la qual cosa prova che ***G*** non appartiene alla classe *LR*(1). Possiamo verificarlo, considerando la frase `ab` di ***G***. Alla coppia (q_0, `a`) sono associate non una ma due azioni: R(3) ed R(4).
Ora, se applichiamo R(3), allora `ab` viene accettata mentre, se applichiamo R(4), `ab` viene rifiutata.

pila	*buffer*	*azione*
q_0	`ab$`	R(3)
q_0Aq_2	`ab$`	T
q_0Aq_2aq_3	`b$`	R(3)
q_0Aq_2aq_3Aq_4	`b$`	T
q_0Aq_2aq_3Aq_4bq_5	`$`	R(1)
q_0Sq_1	`$`	A

pila	*buffer*	*azione*
q_0	`ab$`	R(4)
q_0Bq_6	`ab$`	E

T.2 Si consideri la grammatica acontestuale ***G*** con produzioni

```
S → SA | A
A → 10
```

1. Dopo aver trasformato ***G*** in forma normale di Chomsky, applicare l'algoritmo CYK per verificare che la stringa `1010` è una frase di ***G*** che ammette un solo albero di derivazione. Quindi, costruire la derivazione sinistrorsa e destrorsa della stringa `1010`.

2. Trovare una grammatica ***G′*** equivalente a ***G*** che sia priva di produzioni ricorsive a sinistra. Verificare che ***G′*** appartiene alla classe $LL(1)$ ed applicare il metodo deduttivo per costruire un albero di derivazione per la stringa `1010`.

3. Verificare che ***G*** appartiene alla classe $LR(1)$ ed in caso affermativo applicare il metodo induttivo per costruire un albero di derivazione per la stringa `1010`. Utilizzando il Lemma di Arden, determinare il linguaggio accettato dall'automa deterministico ***D*** delle preformule di ***G***.

T.3 La grammatica acontestuale ***G*** con produzioni

$$\{\mathtt{S} \rightarrow \mathtt{aS} \mid \mathtt{A},\ \mathtt{A} \rightarrow \mathtt{bAc} \mid \varepsilon\}$$

genera il linguaggio $L = \{\mathtt{a}^n\mathtt{b}^m\mathtt{c}^m\colon n, m \geq 0\}$.

1. Si fornisca una la grammatica ***G'*** in forma normale di Chomsky equivalente a ***G***. Si applichi l'algoritmo CYK per determinare gli alberi di derivazione di `abc`.

2. Decidere se ***G'*** appartiene alla classe *LL*(1) e, in caso affermativo, applicare il metodo deduttivo per costruire un albero di derivazione per la stringa `abc`.

3. Decidere se ***G'*** appartiene alla classe *LR*(1) e, in caso affermativo, applicare il metodo induttivo per costruire un albero di derivazione per la stringa `abc` e determinare il linguaggio accettato dall'automa deterministico delle preformule di ***G'***.

T4. (1) È noto che, date due qualsiasi espressioni **e** ed **f** di Kleene, sono equivalenti le due espressioni

$$\mathbf{E} = \mathbf{e}(\mathbf{fe})^* \qquad\qquad \mathbf{E}' = (\mathbf{ef})^*\mathbf{e}.$$

Per verificare l'equivalenza di **E** ed **E**′, provare l'equivalenza degli automi ***A*** e ***A′*** di Thompson associati rispettivamente alle due espressioni **E** ed **E**′ prendendo per "alfabeto" l'inseme {**e**, **f**}.

(2) Si consideri la seguente grammatical ***G*** con produzioni

`S` → `A = 1`
`A` → `A • 1`
`A` → `1`

2a) Provare che ***G*** non appartiene alla classe *LL*(1). Trovare una grammatica ***G′*** che appartiene alla classe *LL*(1) ed è equivalente a ***G***, e costruire col metodo deduttivo un albero di derivazione per la frase `1 • 1 = 1`.

2b) Provare che ***G*** appartiene alla classe *LR*(1), elencare le preformule della grammatica estesa ***G′*** e applicare la procedura di riduzione alla frase `1 • 1 = 1`.

Soluzione

(2a) Per i corpi delle due produzioni con testa `A` si ha

$$\mathcal{I}(\mathtt{A} \cdot \mathtt{1}) = \{\mathtt{1}\} \qquad\qquad \mathcal{I}(\mathtt{1}) = \{\mathtt{1}\}$$

la qual cosa prova che ***G*** non appartiene alla classe *LL*(1). Per ottenere una grammatica ***G**** che appartenga alla classe *LL*(1) e sia equivalente a ***G***, basta eliminare la produzione macina `A` → `A • 1`. Così ***G**** contiene le produzioni

`S` → `A = 1`
`A` → `1 B`
`B` → `• 1 B`
`B` → ε

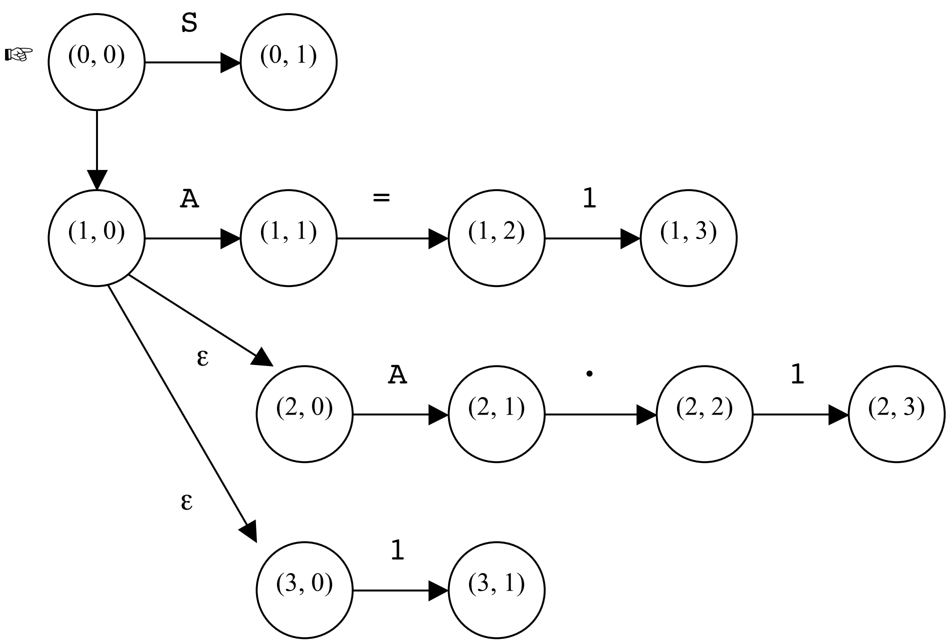
Tenuto conto che $\mathcal{I}$(S) = $\mathcal{I}$(A) = {1}, $\mathcal{I}$(B) = {•, ε}, $\mathcal{J}$(S) = {$}, $\mathcal{J}$(A) = $\mathcal{J}$(B) = {=}, la tabella di controllo è

	1	•	=	$
S	A = 1			
A	1 B			
B		• 1 B	ε	

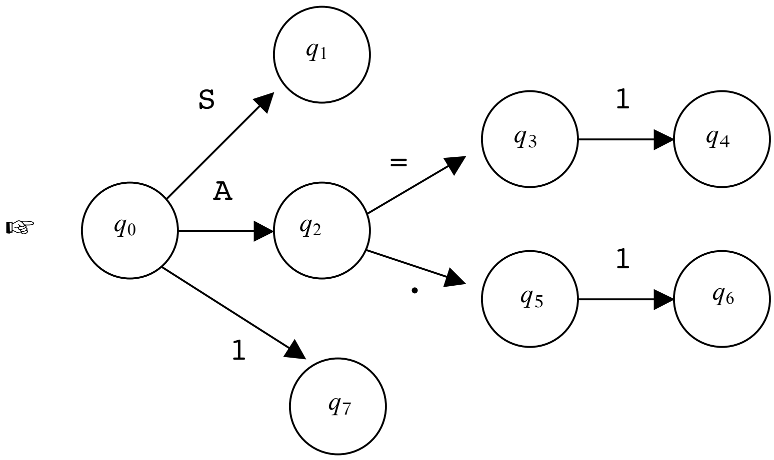
(2b) A partire dalla grammatica estesa ***G′***

(0) S′ → S
(1) S → A = 1
(2) A → A • 1
(3) A → 1

si costruisce l'AFN ***A***



e quindi l'AFD ***D***



Tenuto conto che $\mathcal{J}$(`S`) = {`$`}, $\mathcal{J}$(`A`) = {`•`, `=`}, la tabella delle azioni è

	S	A	1	•	=	$
q_0			T			
q_1						A
q_2				T	T	
q_3			T			
q_4						R(1)
q_5			T			
q_6				R(2)	R(2)	
q_7				R(3)	R(3)	

e, quindi, ***G*** appartiene alla classe *LR*(1).

Risolvendo con il lemma di Arden il sistema di equazioni associato a ***D*** si ottiene che le preformule di ***G*** sono: `S`, `A=1`, `A=`, `A•1`, `A•`, `A`, `1`, ε.

La procedura di riduzione della stringa `1•1=1` si sviluppa come segue.

pila	*buffer*	*azione*
q_0	1•1=1$	T
$q_0$1q_7	•1=1$	R(3)
q_0Aq_2	•1=1$	T
q_0Aq_2•q_5	1=1$	T
q_0Aq_2•$q_5$1q_6	=1$	R(2)
q_0Aq_2	=1$	T
q_0Aq_2=q_3	1$	T
q_0Aq_2=$q_3$1q_4	$	R(1)
q_0Sq_1	$	A

T5. (1) È noto che, date due qualsiasi espressioni **e** ed **f** di Kleene, sono equivalenti le due espressioni

$$\mathbf{E} = (\mathbf{e}\,\mathbf{f}^*)^* \qquad\qquad \mathbf{E}' = \boldsymbol{\varepsilon} \cup \mathbf{e}\,(\mathbf{e} \cup \mathbf{f})^*.$$

Per verificare l'equivalenza di **E** ed **E′**, basta provare l'equivalenza degli automi di Thompson associati alle due espressioni **E** ed **E′** prendendo per "alfabeto" l'insieme {**e**, **f**}. A tale scopo, facendo uso dei grafi delle adiacenze, si costruiscano gli *automi di Thompson ridotti* ***A*** e ***A′*** associati alle due espressioni **E** ed **E′** e, quindi, due AFD ***D*** e ***D′*** equivalenti ad ***A*** e ***A′***.

(2) Si consideri la seguente grammatica ***G*** con produzioni

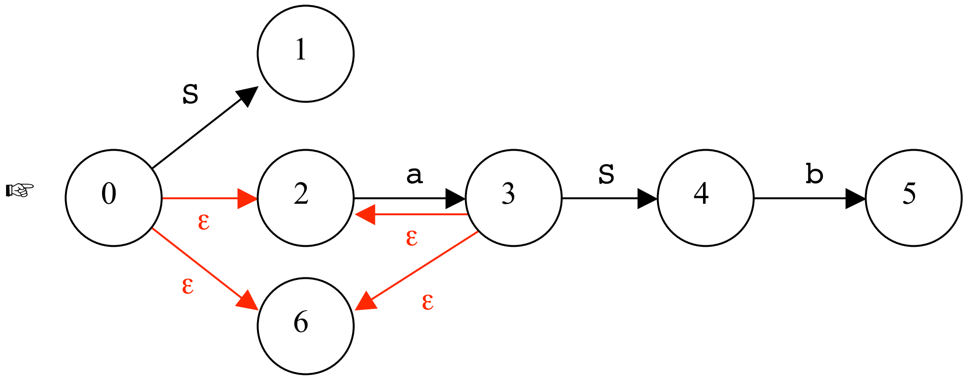
$$\mathtt{S} \to \mathtt{AS} \mid \mathtt{b}$$
$$\mathtt{A} \to \mathtt{SA} \mid \mathtt{a}$$

2a) Costruire l'AFD ***D*** delle preformule di ***G*** e calcolare il linguaggio accettato da ***D***.

2b) Costruire la tabella delle azioni.

2c) Applicare la procedura di riduzione alla frase `ab` e verificare che le stringhe contenute nella pila durante il processo di riduzione sono tutte preformule di ***G***.

T6. Si consideri l'automa finito ***A*** mostrato in Figura, in cui tutti gli stati sono di accettazione.



1. Trovare un AFD $\boldsymbol{D}_{min}$ equivalente ad ***A*** e di dimensione minima. Quindi, determinare il linguaggio accettato da ***A*** applicando a $\boldsymbol{D}_{min}$ il lemma di Arden.

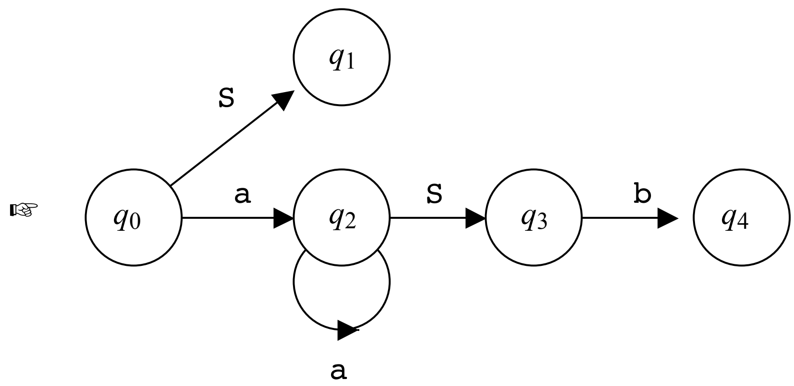
2. Si interpreti ***A*** come l'automa finito delle preformule di una grammatica acontestuale con N = {`S`} e T = {`a`, `b`}. Sia questa ***G***.

2.1) Trovare una grammatica acontestuale ***C*** equivalente a ***G*** e in forma normale di Chomsky. Verificare con l'algoritmo CYK che le stringhe ε e `aabb` sono frasi di ***C*** e, quindi, di ***G***.

2.2) Applicare il metodo induttivo (tecnica di riduzione destrorsa) per verificare che le stringhe ε e `aabb` sono frasi di ***G***. (A tale scopo, a partire da ***G***′ costruire la *tabella delle azioni*). Verificare che in entrambi i casi le stringhe contenute nella pila nel corso del processo di riduzione sono tutte stringhe accettate da ***A*** (cioè preformule di ***G***).

Soluzione

1. Con la costruzione dei sottoinsemi, troviamo il seguente AFD ***D*** equivalente ad ***A***:



dove

$q_0 = \{0, 2, 6\}$ $\quad$ $q_1 = \{1\}$

$q_2 = \{2, 3, 6\}$ $\quad$ $q_3 = \{4\}$ $\quad$ $q_4 = \{5\}$

Con l'aggiunta di uno stato non di accettazione q_5 che sia stato finale per tutte le transizioni assenti in ***D***, otteniamo un AFD completo ***D***′ equivalente ad ***A***. A questo punto possiamo ottenere un automo finito deterministico completo ***D***″ equivalente ad ***A*** e di dimensione minima per raffinamento della partizione iniziale degli stati di ***D***′ in $R = \{q_0, q_1, q_2, q_3, q_4\}$ ed $S = \{q_5\}$. Consideriamo la sottotabella della tabella delle transizioni di ***D***′ indotta da R.

stato	S	a	b
q_0	R	R	S
q_1	S	S	S
q_2	R	R	S
q_3	S	S	R
q_4	S	S	S

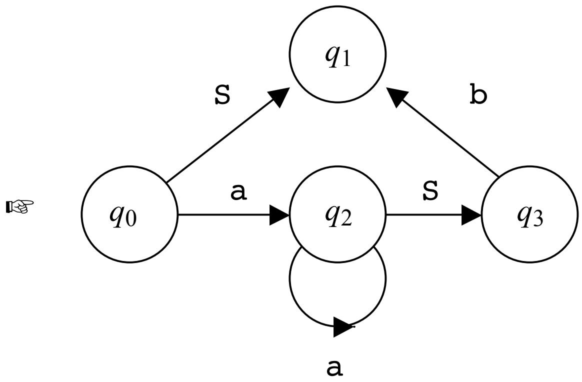
Così la classe S viene scomposta nelle tre classi $X = \{q_0, q_2\}$, $Y = \{q_1, q_4\}$ e $Z = \{q_3\}$. Consideriamo prima X e poi Y. Consideriamo la sottotabella della tabella delle transizioni di ***D***′ indotta da X.

stato	S	a	b
q_0	Y	X	S
q_2	Z	X	S

Così la classe X viene scomposta nelle due classi $\{q_0\}$ e $\{q_2\}$. Consideriamo ora la sottotabella della tabella delle transizioni di $\boldsymbol{D}'$ indotta da Y.

stato	S	a	b
q_1	S	S	S
q_4	S	S	S

Così la classe Y non può essere scomposta. In definitiva, $\boldsymbol{D}''$ ha cinque stati, corrispondenti a $\{q_0\}$, $\{q_2\}$, Y, Z, S, di cui S è uno stato morto. Pertanto, $\boldsymbol{D}_{min}$ ha il seguente diagramma delle transizioni e tutti i suoi stati sono di accettazione.



Finalmente, possiamo applicare il lemma di Arden a $\boldsymbol{D}_{min}$ e si ottiene il sistema di equazioni

$$\lambda_0 = \{\mathtt{S}\} \cdot \lambda_1 \cup \{\mathtt{a}\} \cdot \lambda_2 \cup \{\varepsilon\}$$
$$\lambda_1 = \{\varepsilon\}$$
$$\lambda_2 = \{\mathtt{a}\} \cdot \lambda_2 \cup \{\mathtt{S}\} \cdot \lambda_3 \cup \{\varepsilon\}$$
$$\lambda_3 = \{\mathtt{b}\} \cdot \lambda_1 \cup \{\varepsilon\}$$

che ha per soluzione

$$\lambda_3 = \{\mathtt{b}, \varepsilon\} \qquad \lambda_2 = \{\mathtt{a}\}^* \{\mathtt{Sb}, \mathtt{S}, \varepsilon\} \qquad \lambda_1 = \{\varepsilon\}$$

$$\begin{aligned}\lambda_0 &= \{\mathtt{a}\}\ \{\mathtt{a}\}^*\ \{\mathtt{Sb}, \mathtt{S}, \varepsilon\} \cup \{\mathtt{S}, \varepsilon\} = \\ &= \{\mathtt{a}\}^*\ \{\mathtt{a}\}\ \{\mathtt{Sb}, \mathtt{S}, \varepsilon\} \cup \{\mathtt{S}, \varepsilon\} = \\ &= \{\mathtt{a}\}^*\ \{\mathtt{aSb}\} \cup \{\mathtt{a}\}^+\ \{\mathtt{S}, \varepsilon\} \cup \{\mathtt{S}, \varepsilon\} = \\ &= \{\mathtt{a}\}^*\ \{\mathtt{aSb}\} \cup (\{\mathtt{a}\}^+ \cup \{\varepsilon\})\ \{\mathtt{S}, \varepsilon\} =\end{aligned}$$

$= \{\texttt{a}\}^* \{\texttt{aSb}\} \cup \{\texttt{a}\}^* \{\texttt{S}, \varepsilon\} =$
$= \{\texttt{a}\}^* \{\texttt{aSb}, \texttt{S}, \varepsilon\}$.

Pertanto $L(\boldsymbol{A}) = L(\boldsymbol{D}) = L(\boldsymbol{D}') = L(\boldsymbol{D}'') = L(\boldsymbol{D}_{min}) = \{\texttt{a}\}^* \{\texttt{aSb}, \texttt{S}, \varepsilon\}$.

2. Gli stati di ***A*** corrispondono alle fasi delle produzioni della grammatica estesa ***G′*** di ***G***:

P0: $\texttt{S}' \rightarrow \texttt{S} \mid \varepsilon$ P1: $\texttt{S} \rightarrow \texttt{aSb} \mid \texttt{ab}$ P2: $\texttt{S}' \rightarrow \varepsilon$

Esplicitamente si ha:

*stato di **A***	*fase*
0	(P0, 0)
1	(P0, 1)
2	(P1, 0)
3	(P1, 1)
4	(P1, 2)
5	(P1, 3)
6	(P2, 0)

Pertanto, la grammatica ***G*** ha due sole produzioni: $\texttt{S} \rightarrow \texttt{aSb} \mid \varepsilon$.

2.1) La grammatica ***C*** ha produzioni

$\texttt{S} \rightarrow \texttt{AC} \mid \texttt{AB} \mid \varepsilon$ $\texttt{C} \rightarrow \texttt{SB}$ $\texttt{A} \rightarrow \texttt{a}$ $\texttt{B} \rightarrow \texttt{b}$

La stringa vuota è banalmente una frase di ***C***. Quanto alla stringa x = `aabb` le sue sottostringhe $x_{i,j}$ sono di seguito riportati.

11 a	12 aa	13 aab	14 aabb
21 a	22 ab	23 abb	-
31 b	32 bb	-	-
41 b	-	-	-

Tabella delle sottostringhe $x_{i,j}$

Con l'algorimo CYK calcoliamo gli insiemi $N_{i,j}$:

11 A	12 ∅	13 ∅	14 S
21 A	22 S	23 C	-
31 B	32 ∅	-	-
41 B	-	-	-

Tabella degli insiemi $N_{i,j}$

e, visto che il simbolo speciale S appartiene ad $N_{1,4}$, possiamo concludere che la stringa aabb è una frase di ***C*** e, quindi, di ***G***.

2.2) Per costruire la tabella delle azioni, va ripreso l'automa finito ***D*** i cui stati corrispondono agli insiemi

q_0 = {(P0, 0), (P1, 0), (P2, 0)} q_1 = {(P0, 1)}
q_2 = {(P1, 0), (P1, 1), (P2, 0)} q_3 = {(P1, 2)} q_4 = {(P1, 3)}

cosicché, dopo aver calcolato $\mathcal{J}$(S) = {b, $}, otteniamo la tabella delle azioni:

stato	a	b	$
q_0	T	R (S → ε)	R (S → ε)
q_1			A
q_2	T	R (S → ε)	R (S → ε)
q_3		T	
q_4		R (S → aSb)	R (S → aSb)

Per la stringa vuota abbiamo il processo di riduzione

pila	*buffer*	*azione*
q_0	$	R (S → ε)
q_0Sq_1	$	A

e le stringhe contenute nella pila nel corso della riduzione sono le preformule ε ed S.

Per la stringa `aabb` abbiamo il processo di riduzione

pila	*buffer*	*azione*
q_0	`aabb$`	T
q_0aq_2	`abb$`	T
q_0aq_2aq_2	`bb$`	R (S → ε)
q_0aq_2aq_2Sq_3	`bb$`	T
q_0aq_2aq_2Sq_3bq_4	`b$`	R (S → aSb)
q_0aq_2Sq_3	`b$`	T
q_0aq_2Sq_3bq_4	`$`	R (S → aSb)
q_0Sq_1	`$`	A

e le stringhe contenute nella pila nel corso della riduzione sono le preformule: ε, `a`, `aa`, `aaS`, `aaSb`, `aS`, `aSb` ed `S`.

T.7 Un frammento della grammatica generativa delle espressioni dell'algebra relazionale (per basi di dati) ha la forma seguente $\boldsymbol{G} = (T, N, \mathtt{E}, P)$ dove

$T = \{$**rel**, **set**, **join**, (,)$\}$ $\qquad$ $N = \{\mathtt{E}\}$

P: $\quad \mathtt{E} \rightarrow$ **rel** | ($\mathtt{E}$) **join** ($\mathtt{E}$)

Si consideri la stringa x = (**rel**) **join** (**rel**)

(1) Trovare una grammatica acontestuale $\boldsymbol{G}'$ equivalente a $\boldsymbol{G}$ in forma normale di Chomsky. Verificare con l'algoritmo CYK che la stringa x è una frase di $\boldsymbol{G}'$ e, quindi, di $\boldsymbol{G}$.

(2) Applicare il metodo induttivo (tecnica di riduzione destrorsa) per verificare che la stringa x sia una frase di $\boldsymbol{G}$.

(3) Determinare il linguaggio accettato dall'AFD delle preformule e verificare che le stringhe contenute nella pila nel corso della riduzione sono tutte preformule.

Soluzione

(1) La grammatica $\boldsymbol{G}'$ ha produzioni

$\mathtt{E} \rightarrow$ **rel** | $\mathtt{AX_1}$

$\mathtt{X_1} \rightarrow \mathtt{EX_2} \quad \mathtt{X_2} \rightarrow \mathtt{CX_3} \quad \mathtt{X_3} \rightarrow \mathtt{JX_4} \quad \mathtt{X_4} \rightarrow \mathtt{AX_5} \quad \mathtt{X_5} \rightarrow \mathtt{EC}$

$\mathtt{J} \rightarrow$ **join** $\quad \mathtt{A} \rightarrow$ **(** $\quad \mathtt{C} \rightarrow$)

Quanto alla stringa x le sue sottostringhe $x_{i,j}$ sono di seguito riportati.

11 (	12 (rel	13 (rel)	14 (rel)join	15 (rel)join(	16 (rel)join(rel	17 (rel)join(rel)
21 rel	22 rel)	23 rel)join	24 rel)join(	25 rel)join(rel	26 rel)join(rel)	
31)	32)join	33)join(	34)join(rel	35)join(rel)		
41 join	42 join(	43 join(rel	44 join(rel)			

51 (	52 **(rel**	53 **(rel)**				
61 **rel**	62 **rel**)					
71)						

Tabella delle sottostringhe $x_{i,j}$

Con l'algoritmo CYK calcoliamo gli insiemi $N_{i,j}$:

11 A	12 ∅	13 X_4	14 ∅	15 ∅	16 ∅	17 E
21 E	22 X_5	23 ∅	24 ∅	25 ∅	26 X_1	
31 C	32 ∅	33 ∅	34 ∅	35 X_2		
41 J	42 ∅	43 ∅	44 X_3			
51 A	52 ∅	53 X_4				
61 E	62 X_5					
71 C						

Tabella degli insiemi $N_{i,j}$

e, visto che il simbolo speciale E appartiene ad $N_{1,7}$, possiamo concludere che la stringa x è una frase di $\boldsymbol{G}$.

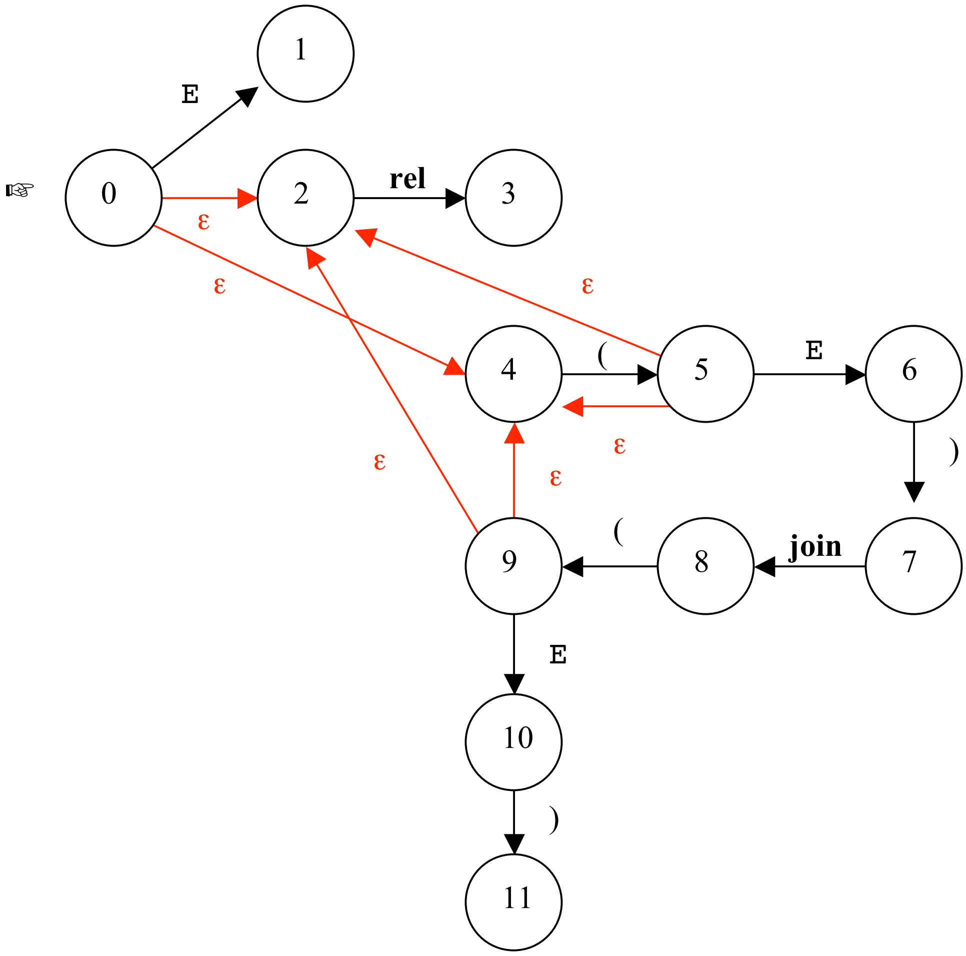
(2)

$\boldsymbol{G'}$ P0: $E' \rightarrow E$

P1: $E \rightarrow$ **rel**

P2: $E \rightarrow$ (E)**join**(E)

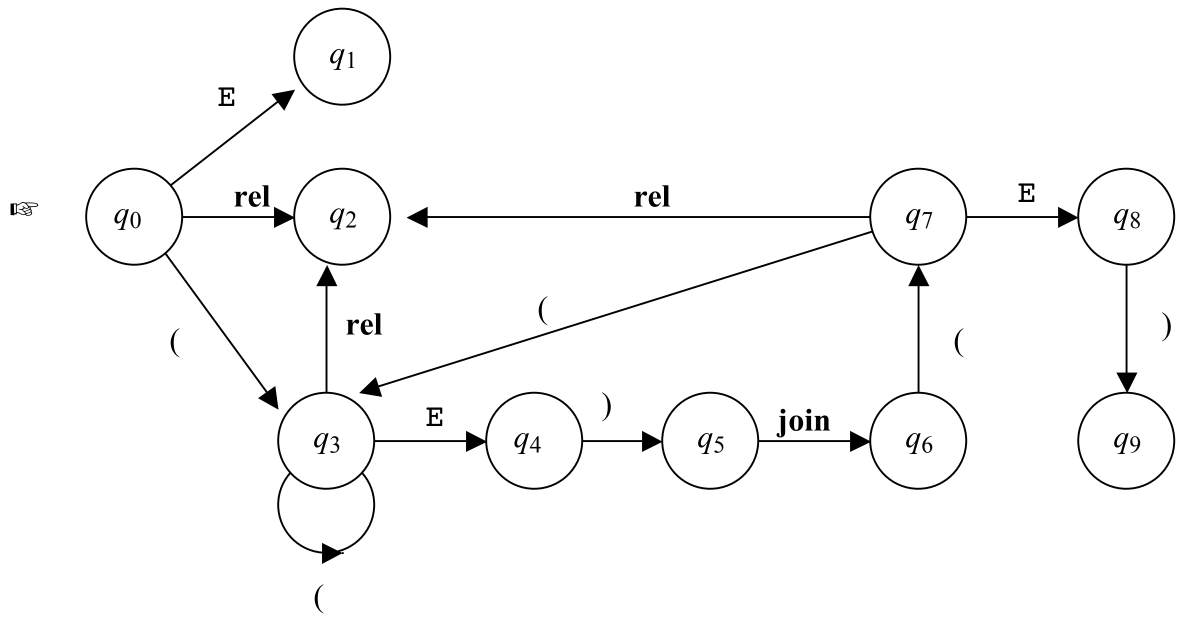
L'AFN $\boldsymbol{A}$ delle preformule è mostrato in Figura (tutti gli stati sono di accettazione).



dove

$0 = (P0, 0)$	$1^A = (P0, 1)$		
$2^T = (P1, 0)$	$3^R = (P1, 1)$		
$4^T = (P2, 0)$	$5 = (P2, 1)$	$6^T = (P2, 2)$	$7^T = (P2, 3)$
$8^T = (P2, 4)$	$9 = (P2, 5)$	$10^T = (P2, 6)$	$11^R = (P2, 7)$

Con la costruzione dei sottoinsemi, troviamo il seguente automa finito quasideterministico ***D*** equivalente ad ***A***:



dove

$q_0 = \{0, 2^T, 4^T\}$ $q_1 = \{1^A\}$ $q_2 = \{3^R\}$
$q_3 = \{2^T, 4^T, 5\}$ $q_4 = \{6^T\}$ $q_5 = \{7^T\}$
$q_6 = \{8^T\}$ $q_7 = \{2^T, 4^T, 9\}$ $q_8 = \{10^T\}$ $q_9 = \{11^R\}$

A questo punto, dopo aver calcolato $\mathcal{J}$(`E`) = {), `$`}, otteniamo la tabella delle azioni:

stato	**rel**	**set**	**join**	(	)	$
q_0	T			T		
q_1						A
q_2					R (`E` → **rel**)	R (`E` → **rel**)
q_3	T			T		
q_4					T	
q_5			T			
q_6				T		
q_7	T			T		
q_8					T	
q_9					R (`E` → (`E`) **join** (`E`))	R (`E` → (`E`) **join** (`E`))

Per la stringa (**rel**) **join** (**rel**) abbiamo il processo di riduzione

pila	*buffer*	*azione*
q_0	(**rel**)**join**(**rel**)$	T
$q_0(q_3$	**rel**)**join**(**rel**)$	T
$q_0(q_3$**rel**q_2	)**join**(**rel**)$	R (E → **rel**)
$q_0(q_3$Eq_4	)**join**(**rel**)$	T
$q_0(q_3$E$q_4)q_5$	**join**(**rel**)$	T
$q_0(q_3$E$q_4)q_5$**join**q_6	(**rel**)$	T
$q_0(q_3$E$q_4)q_5$**join**$q_6(q_7$	**rel**)$	T
$q_0(q_3$E$q_4)q_5$**join**$q_6(q_7$**rel**q_2	)$	R (E → **rel**)
$q_0(q_3$E$q_4)q_5$**join**$q_6(q_7$Eq_8	)$	T
$q_0(q_3$E$q_4)q_5$**join**$q_6(q_7$E$q_8)q_9$	$	R (E → (E) **join** (E))
q_0Eq_1	$	A

e le stringhe contenute nella pila nel corso della riduzione sono nell'ordine:

ε ((**rel** (E (E) (E)**join**

(E)**join**((E)**join**(**rel** (E)**join**(E (E)**join**(E) E

(3) Se applichiamo il lemma di Arden all'automa ***D*** otteniamo il sistema di equazioni

$\lambda_0 = \{\mathtt{E}\} \cdot \lambda_1 \cup \{\mathbf{rel}\} \cdot \lambda_2 \cup \{(\} \cdot \lambda_3 \cup \{\varepsilon\}$

$\lambda_1 = \{\varepsilon\}$

$\lambda_2 = \{\varepsilon\}$

$\lambda_3 = \{(\} \cdot \lambda_3 \cup \{\mathtt{E}\} \cdot \lambda_4 \cup \{\varepsilon\}$

$\lambda_4 = \{)\} \cdot \lambda_5 \cup \{\varepsilon\}$

$\lambda_5 = \{\mathbf{join}\} \cdot \lambda_6 \cup \{\varepsilon\}$

$\lambda_7 = \{\mathbf{rel}\} \cdot \lambda_2 \cup \{(\} \cdot \lambda_3 \cup \{\mathtt{E}\} \cdot \lambda_8 \cup \{\varepsilon\}$

$\lambda_8 = \{)\} \cdot \lambda_9 \cup \{\varepsilon\}$

$\lambda_9 = \{\varepsilon\}$

che ha per soluzione

$\lambda_3 = \{\mathtt{b}, \varepsilon\}$ $\lambda_2 = \{\mathtt{a}\}^* \{\mathtt{Sb}, \mathtt{S}, \varepsilon\}$ $\lambda_1 = \{\varepsilon\}$

$$
\begin{aligned}
\lambda_0 &= \{\mathtt{a}\}\ \{\mathtt{a}\}^*\ \{\mathtt{Sb}, \mathtt{S}, \varepsilon\} \cup \{\mathtt{S}, \varepsilon\} = \\
&= \{\mathtt{a}\}^*\ \{\mathtt{a}\}\ \{\mathtt{Sb}, \mathtt{S}, \varepsilon\} \cup \{\mathtt{S}, \varepsilon\} = \\
&= \{\mathtt{a}\}^*\ \{\mathtt{aSb}\} \cup \{\mathtt{a}\}^+\ \{\mathtt{S}, \varepsilon\} \cup \{\mathtt{S}, \varepsilon\} = \\
&= \{\mathtt{a}\}^*\ \{\mathtt{aSb}\} \cup (\{\mathtt{a}\}^+ \cup \{\varepsilon\})\ \{\mathtt{S}, \varepsilon\} = \\
&= \{\mathtt{a}\}^*\ \{\mathtt{aSb}\} \cup \{\mathtt{a}\}^*\ \{\mathtt{S}, \varepsilon\} = \\
&= \{\mathtt{a}\}^*\ \{\mathtt{aSb}, \mathtt{S}, \varepsilon\}.
\end{aligned}
$$

Pertanto $L(\boldsymbol{A}) = L(\boldsymbol{D}) = L(\boldsymbol{D}') = L(\boldsymbol{D}'') = L(\boldsymbol{D}_{min}) = \{\mathtt{a}\}^*\ \{\mathtt{aSb}, \mathtt{S}, \varepsilon\}$.

T.8 Le parole di senso compiuto che sono anagrammi di `amor` sono tutte e solo le frasi della grammatica ***G*** = (T, N, `S`, P}, dove T = {`a`, `m`, `o`, `r`}, N = {`S`, `U`, `V`, `X`, `Y`, `Z`} e P è l'insieme delle produzioni

$$\mathtt{S} \rightarrow \mathtt{UV} \mid \mathtt{XY} \mid \mathtt{YZ}$$

$$\mathtt{U} \rightarrow \mathtt{ro} \mid \mathtt{or} \qquad \mathtt{V} \rightarrow \mathtt{ma} \qquad \mathtt{X} \rightarrow \mathtt{Z} \mid \mathtt{ar} \qquad \mathtt{Y} \rightarrow \mathtt{mo} \qquad \mathtt{Z} \rightarrow \mathtt{ra}$$

1) Trovare una grammatica acontestuale **C** equivalente a ***G*** in forma normale di Chomsky. Verificare che la stringa `roma` è una frase di ***G*** applicando l'algoritmo CYK con input **C**.

2) Applicare il metodo induttivo (tecnica di riduzione destrorsa) per verificare che la stringa `roma` è una frase di ***G***. (A tale scopo, a partire da ***G′*** costruire la *tabella delle azioni*). Verificare che, nel corso del processo di riduzione, le stringhe della pila sono tutte stringhe accettate dall'AFD utilizzato dal metodo induttivo.