Curs 6

II.2.6 Automate-continuare

Teorema II.2.6.1. *Orice automat finit nedeterminist este echivalent cu un automat finit determinist.*

Demonstraţie. Fie $\mathcal{A} = (X, A, I, F, E)$ un automat nedeterminist, unde:

- X este mulţimea stărilor;
- A este alfabetul input-urilor;
- $I \subseteq X$ este mulţimea stărilor iniţiale;
- $F \subseteq X$ este mulţimea stărilor finale (acceptoare);
- $E \subseteq X \times A \times X$ este relaţia de tranziţie.

Construim un automat determinist astfel: $Det(\mathcal{A}) = (\mathcal{P}(X), A, \mathcal{I}, \mathcal{F}, \delta)$, unde:

- $\mathcal{I} = \{I\} \subseteq \mathcal{P}(X)$;
- $\mathcal{F} = \{X' \subseteq X \mid X' \cap F \neq \emptyset\}$;
- $\delta : \mathcal{P}(X) \times A \longrightarrow \mathcal{P}(X), \delta(X', a) = \{y \in X \mid \exists\ x \in X', (x, a, y) \in E\}$.

Trebuie să arătăm că cele două automate acceptă acelaşi limbaj. Fie $\alpha \in A^*$ un string acceptat de $\mathcal{A}$. Dacă $\alpha = \epsilon$, atunci $\exists x \in X$ stare iniţială şi finală. Deci $I \cap F \neq \emptyset$, adică $I \in \mathcal{I} \cap \mathcal{F}$ este stare iniţială şi finală în automatul $Det(\mathcal{A})$. Rezultă că ϵ este acceptat de $Det(\mathcal{A})$. Reciproc, dacă ϵ este acceptat de $Det(\mathcal{A})$, starea iniţială I în $Det(\mathcal{A})$ este şi finală, adică $I \cap F \neq \emptyset$. Rezultă că există $x \in I \cap F$ stare iniţială şi finală în $\mathcal{A}$, de unde ϵ este string acceptat de $\mathcal{A}$.

Fie acum $\alpha = a_1 \dots a_n$. Atunci: α este acceptat de $\mathcal{A} \Leftrightarrow \exists x_1, \dots, x_n \in X$ astfel încât (x_0, a_1, x_1), (x_1, a_2, x_2), …, $(x_{n-1}, a_n, x_n) \in E$ şi $x_0 \in I, x_n \in F$ (altfel scris: $x_0 \stackrel{a_1}{\to} x_1 \stackrel{a_2}{\to} \dots \stackrel{a_n}{\to} x_n$).

Fie atunci în $Det(\mathcal{A})$ următoarea secvenţă de stări, definită recursiv:

$$\begin{cases} X_0 = I \\ X_{k+1} = \delta(X_k, a_{k+1}), \text{ pentru } k = \overline{0, n-1} \end{cases}$$

Prin construcţie, această secvenţă de stări $X_0, \ldots, X_n \subseteq X$ admite tranziţiile:

$$X_0 \xrightarrow{a_1} X_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} X_n$$

şi $X_0 = I$ e stare iniţială. Rămâne de văzut că X_n e stare finală. Pentru aceasta, vom arăta prin inducţie după $k = \overline{0, n-1}$ că $x_k \in X_k$. Pentru $k = 0$, avem $x_0 \in I = X_0$. Presupunem $x_0 \in X_0, \ldots, x_k \in X_k$ şi arătăm $x_{k+1} \in X_{k+1}$: cum $(x_k, a_{k+1}, x_{k+1}) \in E$ şi $x_k \in X_k$ din ipoteza inductivă, rezultă $x_{k+1} \in \delta(X_k, a_{k+1}) = X_{k+1}$.

Deci pentru orice $k = \overline{0, n-1}, x_k \in X_k$; în particular $x_n \in X_n$; dar $x_n \in F$, de unde $X_n \cap F \neq \emptyset$ şi X_n e stare finală în $Det(\mathcal{A})$.

Rezultă că stringul $\alpha = a_1 \ldots a_n$ e acceptat de $Det(\mathcal{A})$. Reciproc, dacă $\alpha = a_1 \ldots a_n$ e acceptat de $Det(\mathcal{A})$, atunci există o secvenţă de stări în $Det(\mathcal{A})$,

$$X_0 \xrightarrow{a_1} X_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} X_n$$

unde $X_0 = I, X_n \in \mathcal{F}$ şi $X_{k+1} = \delta(X_k, a_k), \forall k = \overline{0, n-1}$.

De remarcat că prin construcţia funcţiei de tranziţie δ, toate stările $X_0, \ldots, X_n$ sunt nevide. Fie atunci $x_n \in X_n \cap F$ (X_n e stare finală); cum $X_n = \delta(X_{n-1}, a_n)$, rezultă că există $x_{n-1} \in X_{n-1}$ cu

$(x_{n-1}, a_n, x_n) \in E(x_{n-1} \xrightarrow{a_n} x_n)$.

Analog construim $x_{n-2} \in X_{n-2}, \ldots, x_0 \in X_0 = I$ cu $(x_{n-2}, a_{n-1}, x_{n-1}) \in E, \ldots, (x_0, a_1, x_1) \in E$.

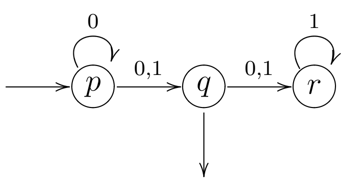
Am obţinut deci $x_0 \xrightarrow{a_1} x_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} x_n$ în $\mathcal{A}$ şi $x_0 \in I, x_n \in F$.

Rezultă că $\alpha = a_1 \ldots a_n$ e acceptat de $\mathcal{A}$. □

Observaţia II.2.6.2. *În practică, se poate eficientiza determinizarea unui automat nedeterminist cu n stări astfel: în loc de $\mathcal{P}(X)$ pentru mulţimea stărilor în $Det(\mathcal{A})$ se consideră doar mulţimea acelor submulţimi pentru care există o secvenţă de tranziţii pornind de la I (vezi exemplul următor).*

Observaţia II.2.6.3. *Din teoremă rezultă că orice automat finit este echivalent cu un automat cu o singură stare de intrare; în continuare toate automatele vor avea doar o singură stare de intrare.*

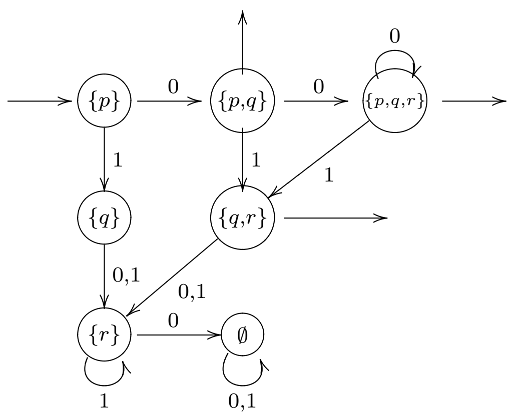
Exemplul II.2.6.4. *Determinizaţi urmatorul NDFA:*



Avem $X = \{p, q, r\}$, $A = \{0, 1\}$, $I = \{p\}$, $F = \{q\}$, *de unde rezultăîn particular că* $\mathcal{P}(X)$ *are 8 elemente. Funcţia de tranziţie* $\delta : \mathcal{P}(X) \times A \to \mathcal{P}(X)$ *este descrisă în tabelul urmator:*

		0	1
	$\emptyset$	$\emptyset$	$\emptyset$
$\to$	$\{p\}$	$\{p,q\}$	$\{q\}$
$\leftarrow$	$\{q\}$	$\{r\}$	$\{r\}$
	$\{r\}$	$\emptyset$	$\{r\}$
$\leftarrow$	$\{p,q\}$	$\{p,q,r\}$	$\{q,r\}$
	$\{p,r\}$	$\{p,q\}$	$\{q,r\}$
$\leftarrow$	$\{q,r\}$	$\{r\}$	$\{r\}$
$\leftarrow$	$\{p,q,r\}$	$\{p,q,r\}$	$\{q,r\}$

În coloana din stânga a tabelului, am indicat cu o săgeată "$\to$" starea de intrare şi cu o săgeată orientată invers "$\leftarrow$" stările finale. Automatul determinist asociat va fi:



Nu am mai reprezentat starea $\{p, r\}$ *pentru că nu este accesibilă (nu există nicio secvenţă de tranziţie de la starea de intrare* $\{p\}$ *care să conducă la* $\{p, r\}$*).* $\square$

II.2.7 Limbaje raţionale

Definiţia II.2.7.1. *Fie A un alfabet. Atunci*

(i) $\emptyset$, ϵ *şi orice* $a \in A$ *sunt* expresii regulate, *numite expresii regulate primitive.*

(ii) Dacă r_1 *şi* r_2 *sunt expresii regulate, la fel sunt* $r_1 + r_2$, $r_1 \cdot r_2$, r_1^* *şi* (r_1).

(iii) Un string este o expresie regulată dacă poate fi obţinut din expresii regulate primitive în urma unui număr finit de aplicări ale regulii precedente.

Exemplul II.2.7.2. *Fie* $A = \{0, 1\}$. *Atunci* $\emptyset, \epsilon, 0, 1, 0+1, 0\cdot 1, 0+0\cdot 1, (0+(0\cdot 1))^*$ *sunt expresii regulate. Dar* $(0 + 1 \cdot 1+)$ *nu este o expresie regulată.*

Definiţia II.2.7.3. Limbajul raţional *este limbajul asociat unei expresii regulate* r, *notat* $L(r)$, *după regulile următoare:*

(i) $L(\emptyset) = \emptyset$, $L(\epsilon) = \{\epsilon\}$, $L(a) = \{a\}$, $\forall a \in A$.

(ii) $L(r_1 + r_2) = L(r_1) \cup L(r_2)$, $L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$, $L((r_1)) = L(r_1)$, $L(r_1^*) = L(r_1)^*$, $\forall r_1, r_2$ *expresii regulate.*

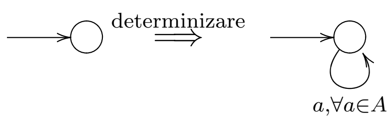
Exemplul II.2.7.4. $L(0^* \cdot (0+1)) = L(0^*) \cdot L(0+1) = L(0)^* \cdot (L(0) \cup L(1)) = \{0\}^* \cdot (\{0\} \cup \{1\}) = \{0\}^* \cdot \{0, 1\} = \{\epsilon, 0, 00, \ldots\}$.

Teorema II.2.7.5. *Fie* $L \subseteq A^*$ *un limbaj raţional. Atunci există un automat finit determinist care acceptă* L.

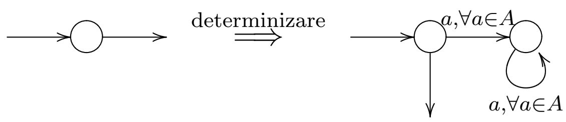
Nu vom da demonstraţia acestei teoreme, ci un algoritm pentru construcţia unui automat determinist (nu este unic!) pentru o expresie regulată dată.

(i) Pentru fiecare dintre limbajele asociate expresiilor regulate $\emptyset$, ϵ, a, unde $a \in A$, am construit în cursul trecut câte un automat nedeterminist care acceptă acest limbaj. Prin determinizare, rezultă:

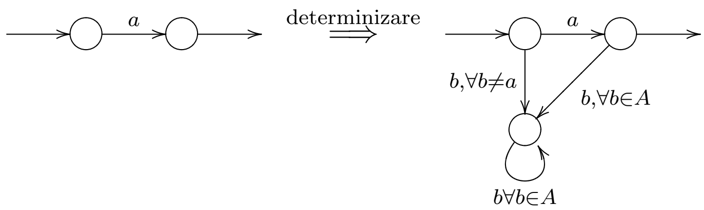
(a) Automat care acceptă $L(\emptyset) = \emptyset$:



(b) Automat care acceptă $L(\epsilon) = \{\epsilon\}$:



(c) Automat care acceptă $L(a) = \{a\}$:



(ii) Presupunem că am construit deja automate deterministe pentru $L(r_1)$ şi $L(r_2)$ cu r_1 şi r_2 expresii regulate; fie acestea $\mathcal{A}_1 = (X_1, A, x_1, F_1, \delta_1)$ şi $\mathcal{A}_2 = (X_2, A, x_2, F_2, \delta_2)$.

(a) Automat care acceptă $L(r_1+r_2) = L(r_1) \cup L(r_2)$: $\mathcal{A} = (X, A, x, F, \delta)$, unde: $X = X_1 \times X_2$, starea iniţială este $x = (x_1, x_2)$, $F = \{(y_1, y_2) \in X_1 \times X_2 \mid y_1 \in F_1 \text{ sau } y_2 \in F_2\}$. Funcţia de tranziţie este:

$$\delta : X_1 \times X_2 \times A \longrightarrow X_1 \times X_2,\ \ \delta((y_1, y_2), a) = (\delta(y_1, a), \delta(y_2, a))$$

Atunci un string $\alpha = a_1 \dots a_n \in A^*$ este acceptat de $\mathcal{A} \Longleftrightarrow$ există o secvenţă finită de tranziţii

$$(y_1^{(0)}, y_2^{(0)}) \xrightarrow{a_1} (y_1^{(1)}, y_2^{(1)}) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (y_1^{(n)}, y_2^{(n)})$$

unde

$$(y_1^{(0)}, y_2^{(0)}) = (x_1, x_2)$$

este starea iniţială,

$$(y_1^{(n)}, y_2^{(n)}) \in F$$

şi

$$(y_1^{(k+1)}, y_2^{(k+1)}) = \delta((y_1^{(k)}, y_2^{(k)}), a_k), k = \overline{0, n-1}$$

adică

$$y_1^{(k+1)} = \delta_1(y_1^{(k)}, a_k)$$

şi

$$y_2^{(k+1)} = \delta_2(y_2^{(k)}, a_k), \forall k = \overline{0, n-1}$$

Dar

$$(y_1^{(n)}, y_2^{(n)}) \in F \Leftrightarrow y_1^{(n)} \in F \text{ sau } y_2^{(n)} \in F$$

de unde rezultă $\alpha \in L(r_1)$ sau $\alpha \in L(r_2)$.

Reciproc, se arată analog că orice $\alpha \in L(r_1) \cap L(r_2)$ e acceptat de automatul $\mathcal{A}$.

(b) Automat care acceptă $L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2) : \mathcal{A} = (X, A, x, F, E)$, mai întâi nedeterminist şi apoi determinizat, unde: $X = X_1 \cup X_2$ (presupunem X_1, X_2 distincte; dacă nu, le redenumim), $x = x_1$,

$$F = \begin{cases} F_2, & \text{dacă } x_2 \notin F_2 \\ F_1 \cup F_2, & \text{dacă } x_2 \in F_2 \end{cases}$$

şi relaţia de tranziţie $E \subseteq (X_1 \cup X_2) \times A \times (X_1 \cup X_2)$ descrisă prin:

$$\begin{array}{rl} (y, a, \delta_1(y, a)) \in E, & y \in X_1 \setminus F_1 \\ (y, a, \delta_1(y, a)), (y, a, \delta_2(x_2, a)) \in E, & y \in F_1 \\ (y, a, \delta_2(y, a)) \in E, & y \in X_2 \end{array}$$

Atunci $\alpha = a_1...a_n \in L(r_1)$ şi $\beta = b_1...b_m \in L(r_2)$ $(\beta \neq \epsilon) \iff \alpha\beta = a_1...a_n b_1...b_m$ este acceptat de automatul $\mathcal{A}$:

$$\begin{array}{rl} \mathcal{A}_1 : & x_1 = y_1^{(0)} \stackrel{a_1}{\to} y_1^{(1)} \stackrel{a_2}{\to} ... \stackrel{a_n}{\to} y_1^{(n)} \in F_1 \\ \mathcal{A}_2 : & x_2 = y_2^{(0)} \stackrel{b_1}{\to} y_2^{(1)} \stackrel{b_2}{\to} ... \stackrel{b_m}{\to} y_2^{(m)} \in F_2 \\ \mathcal{A} : & x_1 = y_1^{(0)} \stackrel{a_1}{\to} y_1^{(1)} \stackrel{a_2}{\to} ... \stackrel{a_n}{\to} y_1^{(n)} \stackrel{b_1}{\to} y_2^{(1)} \stackrel{b_2}{\to} ... \stackrel{b_m}{\to} y_2^{(m)} \in F_2 \subseteq F \end{array}$$

Dacă $\beta = \epsilon$, atunci $\alpha \in L(r_1) \iff \alpha$ e acceptat de automatul $\mathcal{A}$.
În final, prin determinizare se obţine un automat determinist care acceptă $L(r_1 \cdot r_2)$.

(c) Automat care acceptă $L(r_1^*) = L(r_1)^*$: din nou în doi paşi: mai întâi un automat nedeterminist: $\mathcal{A} = (X, A, x, F, E)$, unde $X = X_1 \cup \{x\}$ (adăugăm o nouă stare x care va deveni stare iniţială), $F = F_1 \cup \{x\}$ şi tranziţiile:

- Din starea x:

$$\begin{array}{rl} (x, a, \delta_1(x_1, a)), (x, a, x_1) \in E, & \text{dacă } \delta_1(x_1, a) \in F_1 \\ (x, a, \delta_1(x, a)) \in E, & \text{în caz contrar} \end{array}$$

- Dintr-o stare arbitrară $y \in X_1$:

$$\begin{array}{rl} (y, a, \delta_1(y, a)), (y, a, x) \in E, & \text{dacă } \delta_1(y, a) \in F_1 \\ (y, a, \delta_1(y, a)), & \text{în caz contrar} \end{array}$$

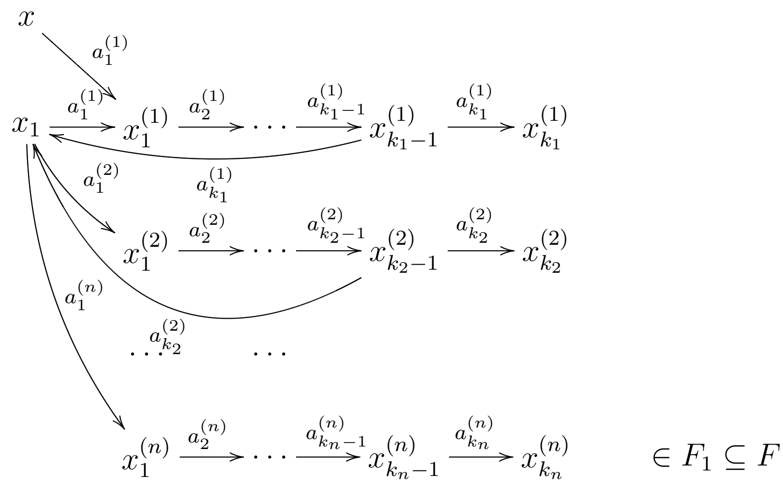
Atunci ϵ este acceptat de $\mathcal{A}$ (x e stare iniţială şi finală) şi un produs de stringuri $\alpha_1...\alpha_n$ este acceptat de $\mathcal{A} \iff \alpha_1, ..., \alpha_n \in L(r_1*)$. Pentru a vedea aceasta, fie $\alpha_1 = a_1^{(1)}...a_{k_1}^{(1)}$, $\alpha_2 = a_1^{(2)}...a_{k_2}^{(2)}$, $\ldots$, $\alpha_n = a_1^{(n)}...a_{k_n}^{(n)}$. Atunci fiecărui string îi corespunde o secvenţă de stări, astfel:

$$\mathcal{A}_1: \quad \underbrace{x_1 \overset{a_1^{(1)}}{\to} x_1^{(1)} \overset{a_2^{(1)}}{\to} \ldots \overset{a_{k_1}^{(1)}}{\to} x_{k_1}^{(1)}}_{\alpha_1} \in F_1$$

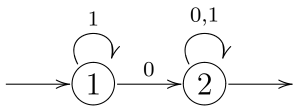
$$\ldots\ldots\ldots$$

$$\underbrace{x_1 \overset{a_1^{(n)}}{\to} x_1^{(n)} \overset{a_2^{(n)}}{\to} \ldots \overset{a_{k_n}^{(1)}}{\to} x_{k_n}^{(n)}}_{\alpha_n} \in F_1$$

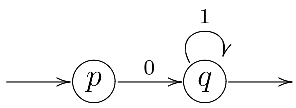
$\mathcal{A}:$



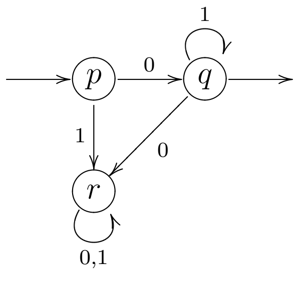
Exemplul II.2.7.6. *Fie* $L_1 = L(1^*0(0+1)^*)$ *şi* $L_2 = L(01^*)$. *Un exemplu de automat* $\mathcal{A}_1$ *care acceptă* L_1 *este*



Pentru L_2, *considerăm mai întâi un automat nedeterminist* $\widetilde{\mathcal{A}}_2$, *cum ar fi*

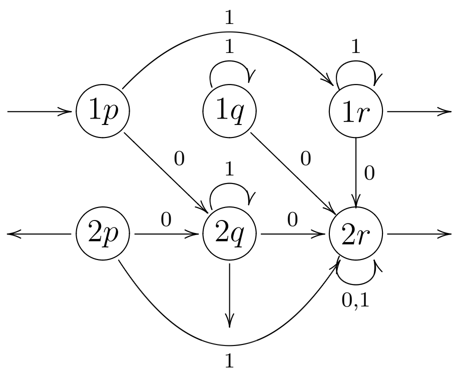


care prin determinizare conduce la automatul $\mathcal{A}_2$ *dat de*



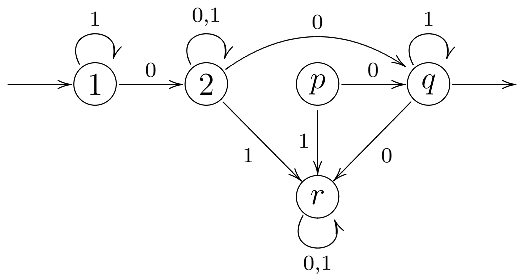
unde r corespunde mulţimii vide şi am omis scrierea acoladelor.

Construim acum automatul care acceptă $L_1 \cup L_2 = L(1^*0(0+1)^* + 01^*)$:



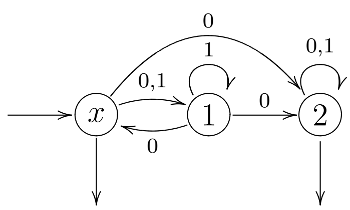
De remarcat că stările $1q$ şi $2p$ nu sunt accesibile, deci se poate renunţa la ele.

Automatul (nedeterminist) care acceptă $L_1 \cdot L_2 = L(1^*0(0+1)^*01^*)$ va fi:



În acest automat, starea p devine inutilă.

În final, să construim şi automatul care acceptă $(L_1)^*$:



Teorema II.2.7.7. *Fie L un limbaj acceptat de un automat finit. Atunci L este raţional (există o expresie regulată r astfel încât $L = L(r)$).*

Algoritm de determinare a limbajului acceptat de un automat finit.

Fie $A = (X, a, x_0, F, \delta)$ automat finit cu n stări, determinist, care acceptă L. Redenumind stările, putem presupune $X = \{1, 2, ..., n\}$.

Fie $R_{ij}^{(k)}$ numele unei expresii regulate al cărei limbaj asociat este mulţimea stringurilor α astfel încât există un drum de la starea i la starea j cu inputul α astfel încât stările intermediare nu sunt mai mari strict decât k.

Vom construi $R_{ij}^{(k)}$ inductiv pentru $k = \overline{0,n}$ (din construcţie va rezulta $R_{ij}^{(k)}$ expresie regulată).

Dacă $k = 0$, atunci nu vor fi stări intermediare.

Cazul I: $i \neq j \Rightarrow$ putem avea doar tranziţie $i \to j$

- dacă nu există $a \in A$ astfel încât $\delta(i,a) = j$, atunci $R_{ij}^{(0)} = \emptyset$ (expresie regulată)

- dacă există $a_1, ..., a_l \in A$ cu $\delta(i,a_1) = ... = \delta(i,a_l) = j$, atunci $R_{ij}^{(0)} = a_1 + ... + a_l$ (expresie regulată)

Cazul II: $i = j \Rightarrow$

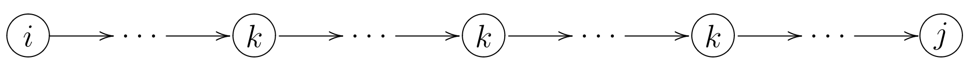
- dacă nu există $a \in A$ astfel încât $\delta(i,a) = i$, atunci $R_{ij}^{(0)} = \epsilon$ (expresie regulată)

- dacă există $a_1, ..., a_l \in A$ cu $\delta(i,a_1) = ... = \delta(i,a_l) = i$, atunci $R_{ij}^{(0)} = \epsilon + a_1 + ... + a_l$ (expresie regulată)

Presupunem acum că există un drum $i \to j$ pentru care toate stările intermediare sunt mai mici sau egale cu k.

Cazul I: Drumul nu trece deloc prin starea k. Atunci toate stările intermediare sunt strict mai mici decât k si obţinem expresia $R_{ij}^{(k-1)}$

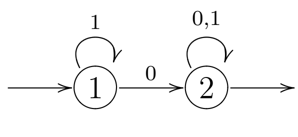
Cazul II: Drumul trece prin starea k cel puţin o dată. Atunci putem împărţi drumul atfel:



şi obţinem expresia regulată $R_{ik}^{(k-1)}(R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$. Rezultă $R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)}(R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$ expresie regulată.

În final, limbajul acceptat de automat va fi limbajul asociat expresiei $\sum\limits_{j \in F} R_{ij}^{(n)}$ (sumă finită), deci un limbaj raţional.

Exemplul II.2.7.8. *Considerăm automatul determinist $\mathcal{A}$ dat prin*



Atunci $L = L(R_{12}^{(2)})$. *Calculăm:*

$$\begin{array}{rcl} R_{12}^{(2)} & = & R_{12}^{(1)} + R_{12}^{(1)}(R_{22}^{(1)})^* R_{22}^{(1)} \\ R_{12}^{(1)} & = & R_{12}^{(0)} + R_{11}^{(0)}(R_{11}^{(0)})^* R_{12}^{(0)} \\ R_{22}^{(2)} & = & R_{22}^{(0)} + R_{21}^{(0)}(R_{11}^{(0)})^* R_{12}^{(0)} \end{array}$$

Avem

$$\begin{array}{rcl} R_{11}^{(0)} & = & \epsilon + 1 \\ R_{12}^{(0)} & = & 0 \\ R_{21}^{(0)} & = & \emptyset \\ R_{22}^{(0)} & = & \epsilon + 0 + 1 \end{array}$$

De aici rezultă

$$\begin{array}{rcl} R_{12}^{(1)} & = & 0 + (\epsilon + 1) \cdot (\epsilon + 1)^* \cdot 0 = \\ & = & 0 + (\epsilon + 1)^* \cdot (\epsilon + 1) \cdot 0 \\ & = & 0 + (\epsilon + 1)^* \cdot (0 + 10) \\ & = & 0 + 1^* \cdot (0 + 10) \\ & = & 0 + 1^* \cdot 0 + 1^* 10 \\ & = & 0 + (1^* + 1^* 1) 0 \\ & = & 0 + 1^* \cdot 0 \\ & = & 1^* \cdot 0 \end{array}$$

şi

$$\begin{array}{rcl} R_{22}^{(1)} & = & \epsilon + 0 + 1 + \emptyset \cdot (\epsilon + 1)^* \cdot 0 = \\ & = & \epsilon + 0 + 1 + \emptyset \\ & = & \epsilon + 0 + 1 \end{array}$$

dar şi

$$\begin{array}{rcl} R_{12}^{(2)} & = & 1^* \cdot 0 + 1^* \cdot 0 \cdot (\epsilon + 0 + 1)^* \cdot (\epsilon + 0 + 1) = \\ & = & 1^* \cdot 0 + 1^* \cdot 0 \cdot (\epsilon + 0 + 1)^* \\ & = & 1^* \cdot 0 + 1^* \cdot 0 \cdot (0 + 1)^* \\ & = & 1^* \cdot 0 \cdot (0 + 1)^* \end{array}$$

În final, rezultă $L = L(1^* \cdot 0 \cdot (0 + 1)^*)$