TUTORIAT LFA 6: EXPRESII REGULATE

RADU COSTACHE, MARIA PREDA

1. Breviar Teoretic

Notaţie 1.1. *Familia expresiilor regulate se noteaza $RegEx$.*

Observaţie 1.2. *Avem urmatoarele operatii posibile peste $RegEx$, care seamana cu operatiile pentru cuvinte si limbaje pe care le-am facut pana acum:*

- *"+" = echivalentul reuniunii*
- *"·" = echivalentul concatenarii*
- *"*" = stelarea*
- *"()" = care se comporta ca parantezele normale*

Definiţie 1.3. *Expresiile regulate peste Σ fac parte din familia expresiilor regule. Aceasta familie se defineste recursiv in felul urmator:*

(1) $\emptyset$, $\lambda \in RegEx$ *si* $a \in RegEx$, $\forall a \in \Sigma$ *(cazul de baza)*

(2) $e_1, e_2 \in RegEx \Rightarrow (e_1 + e_2) \in RegEx$

(3) $e_1, e_2 \in RegEx \Rightarrow (e_1 \cdot e_2) \in RegEx$

(4) $e \in RegEx \Rightarrow e^* \in RegEx$

Observaţie 1.4. *Ca de fiecare data cand sunt definiti operatori, avem nevoie si de o prioritate a lor, numita precedenta. Ordinea acestora, de la precedenta cea mai mare la cea mai mica, este:*

$$"()" > "*" > "\cdot" > "+"$$

De multe ori, urmarirea unui limbaj (definit ca pana acum) sau a unui automat care accepta un limbaj este complicata, mai ales in cazurile in care automatul are foarte multe stari.

Observaţie 1.5. *Putem simplifica automatele si limbajele cu ajutorul expresiilor regulate.*

Observaţie 1.6. *Pentru orice automat finit exista o expresie regulata echivalenta.*

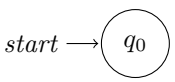
Observaţie 1.7. *Pentru orice expresie regulata exista un automat finit care o accepta.*

Scopul acestui tutoriat este sa vedem cum putem transforma expresii regulate in automate si automate in expresii regulate.

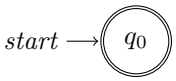
Vom porni de la cazurile de baza, prezente in Definitia 1.3.

(1) *Avem cele trei situatii de expresii regulate:*

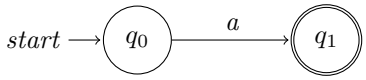
- $e = \emptyset \Rightarrow L = \emptyset$



- $e = \lambda \Rightarrow L = \{\lambda\}$

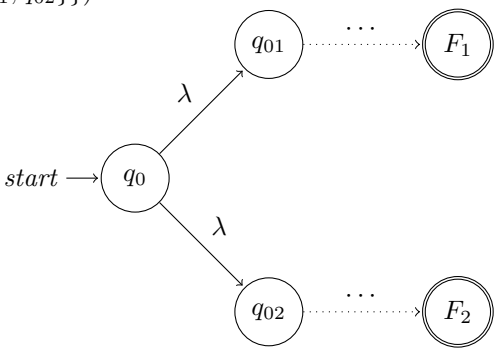


- $e = a, a \in \Sigma \Rightarrow L = \{a\}$



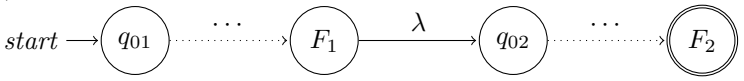
(2) $e = e_1 + e_2 \Rightarrow L(e) = L(e_1) \cup L(e_2)$

Automatul finiti pentru $L(e_1 + e_2)$ *este* $(Q_1 \cup Q_2 \cup \{q_0\}, \Sigma_1 \cup \Sigma_2, q_0, F_1 \cup F_2, \delta_1 \cup \delta_2 \cup \{\delta(q_0, \lambda) = \{q_{01}, q_{02}\}\})$



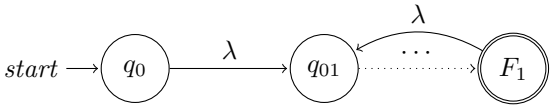
(3) $e = e_1 \cdot e_2 \Rightarrow L(e) = L(e_1) \cdot L(e_2)$

Automatul finiti pentru $L(e_1 \cdot e_2)$ *este* $(Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, q_{01}, F_2, \delta_1 \cup \delta_2 \cup \{\delta(q_{f_1}, \lambda) \ni q_{02} | \forall f_1 \in F_1\})$



(4) $e = (e_1)^* \Rightarrow L(e) = (L(e_1))^*$

Automatul finiti pentru $(L(e_1))^*$ *este* $(Q_1 \cup \{q_0\}, \Sigma_1, q_0, F_1 \cup \{q_0\}, \delta_1 \cup \{\delta(q_0, \lambda) = q_{01}\} \cup \{\delta(q_{f_1}, \lambda) \ni q_{01} | \forall f_1 \in F_1\})$



Definiţie 1.8. *Se numeste automat finit extins (AFE) un automat in care, spre deosebire de cele obisnuite, functia de tranzitie este inlocuita de o functie de etichetare et* : $Q \times Q \to RegEx(\Sigma)$.

1.1. **Algoritmul de transformare λ-NFA intr-o expresie regulata.** ***Intuitie:*** *Automatul finit extins are pe muchii expresii regulate. Atunci, am putea transforma intai automatul finit initial intr-unul extins, pentru ca ulterior, eliminand stare cu stare, sa ajungem cu doar doua stari, cea initiala si cea finala, intre care avem o expresie regulata.*

In continuare, vom prezenta pasii algoritmului:

Pas 1: *Daca intre doua stari exista mai mult de o singura tranzitie, le vom comprima intr-o singura expresie regulata.*

$$et(q_x, q_y) = \{w \in REX(\Sigma) | w = a_1 + a_2 + \cdots + a_n; q_y \in \delta(q_x, a_i), a_i \in (\Sigma \cup \{\lambda\}), \forall i \in \{1, \cdots, n\}\}$$

Pas 2: *Se va adauga o noua stare initiala daca se intampla cel putin unul dintre cele doua cazuri:*

- *Starea initiala este si finala.*
- *Exista tranzitii care vin catre starea initiala dinstre alte stari sau bucle.*

Noua stare initiala va pleca spre o singura stare cu λ*, aceasta starea fiind vechea stare initiala.*

Pasul 3: *Trebuie sa avem o singura stare finala, din care nu mai putem pleca. Daca aceste conditii nu sunt indeplinite, vom adauga o noua stare finala, spre care vom avea λ-tranzitie din toate vechile stari finale (acestea nu mai sunt finale acum).*

Pasul 4: *Se elimina pe rand, una cate una, toate starile, pana cand ajungem sa mai avem doar starea initiala si starea finala.*

Observaţie 1.9. *Ordinea in care alegem sa eliminam starile nu va afecta rezultatul. De obicei se elimină starea cu cele mai puţine săgeţi.*

Distingem urmatoarele cazuri la eliminarea unei stari:

- *Pentru fiecare pereche de tranzitii (una care intra in starea pe care o eliminam si una care pleaca din ea), vom pune o noua tranzitie intre starea din care ajungeam si cea spre care ne duceam, pe care vom pune expresia regulata obtinuta prin concatenarea expresiilor regulate de pe muchiile vechi. Daca mai exista o tranzitie intre aceste doua stari, vom reuni expresiile regulate.*

Pasul 5: *Cand raman doar starea initiala si cea finala, expresia regulata de pe tranzitia intre ele este cea echivalenta limbajului acceptat de automat.*

Observaţie 1.10. *La **pasul 4** se poate elimina orice stare, prin unirea oricărui predecesor cu oricare succesor al acesteia.*

Observaţie 1.11. *Reuniunea se include între paranteze, pentru a se executa înainte de concatenare şi se încearcă simplificarea cât mai mult a expresiilor rezultate.*

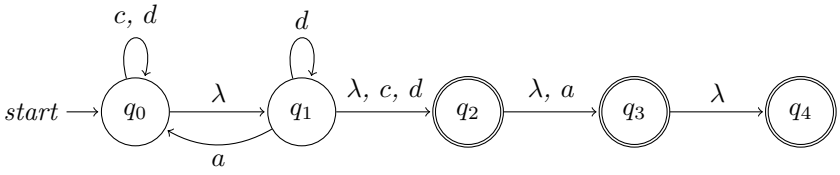
2. Exerciții

(1) (a) *Daţi un automat finit nedeterminist (NFA) care nu este λ-NFA şi nici DFA. Automatul trebuie sa aibă cel putin 4 stari accesibile din starea iniţială.*

(b) *Transformaţi automatul de la punctul a. într-o expresie regulată arătând paşii succesivi de obţinere a expresiei regulate.*

(c) *Daţi un automat finit λ-nedeterminist (λ-NFA) care nu este NFA şi nici DFA. Automatul trebuie sa aibă cel putin 4 stari accesibile din starea iniţială cu λ mişcari, şi macar una din aceste stări nu are muchie directă de la starea iniţială. În plus automatul are 2 stări finale şi cel puţin 2 muchii de la o stare la ea însăşi.*

Soluţie:

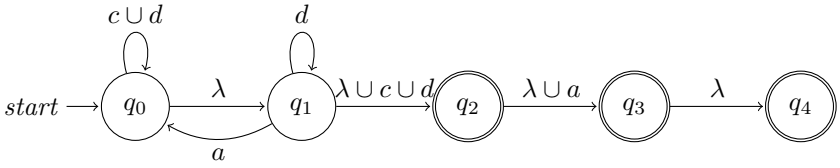


(d) *Transformaţi automatul de la punctul c. într-o expresie regulată arătând paşii succesivi de obţinere a expresiei regulate.*

Soluţie:

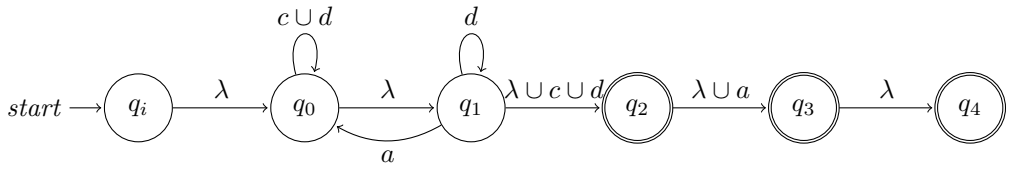
Pasul 1:

Trebuie sa avem maxim o tranzitie intre oricare doua stari. Tranzitiile pe care avem caractere despartite prin virgula sunt considerate mai multe tranzitii.

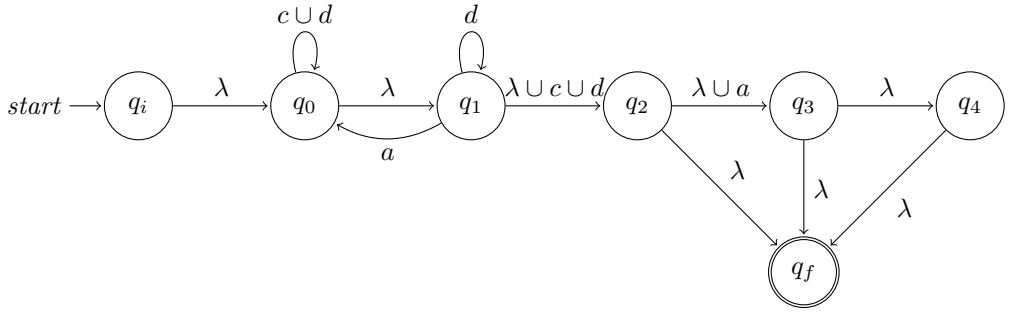


Pasul 2:

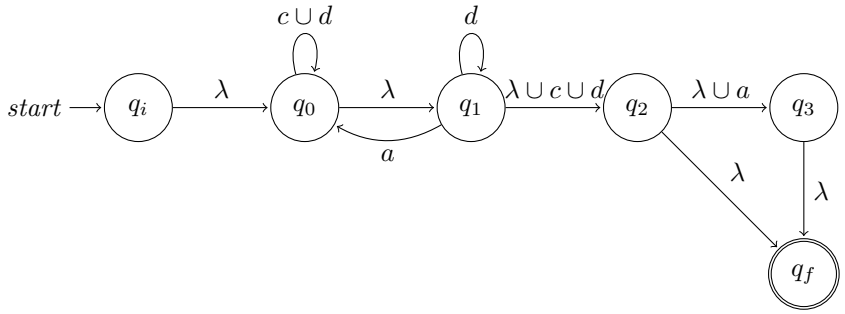
Starea initiala trebuie sa fie unica si sa nu aiba muchii de intrare ⇒ trebuie sa adaugam o noua stare initiala. q_0 nu mai este acum stare initiala.



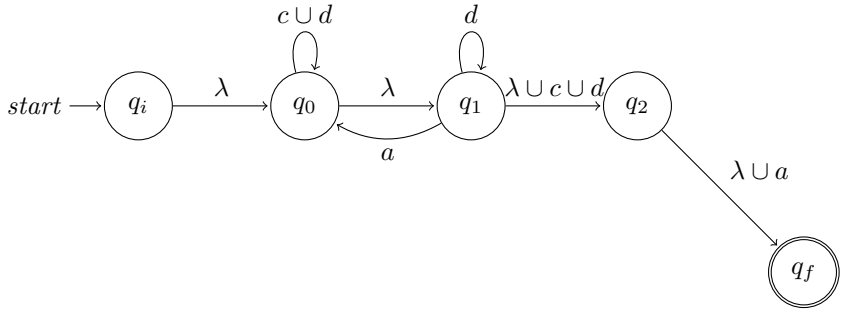
Pasul 3:
Observam ca avem mai mult de o singura stare finala. In plus, exista si muchii care sa iasa dintr-o stare finala, dar acest lucru nu mai este important in acest caz, deoarece aveam deja nevoie de o stare finala noua.



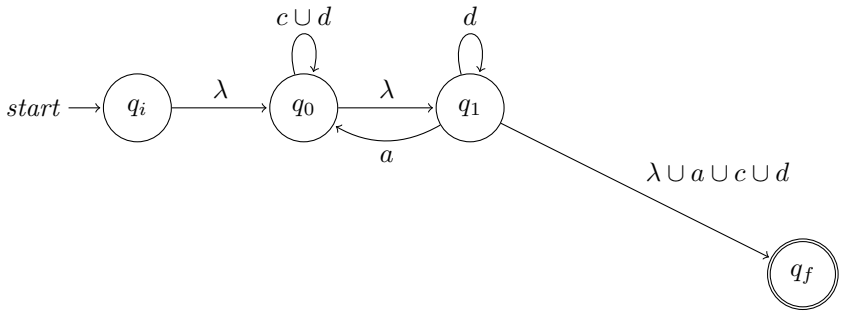
Pasul 4:
La acest pas, trebuie sa incepem sa eliminam stare cu stare, pana cand ramanem doar cu starea initiala, starea finala si o singura sageata cu expresia regulata echivalenta automatului. Vom incepe prin eliminarea starii q_4. Observam ca singura muchie de intrare este dinspre q_3 cu λ si singura de iesire este spre starea finala tot cu λ. Teoretic, ar trebui sa adaugam de la q_3 la q_f o muchie cu λ. O muchie exista deja, deci vom avea o reuniune: $\lambda \cup \lambda \Rightarrow \lambda$



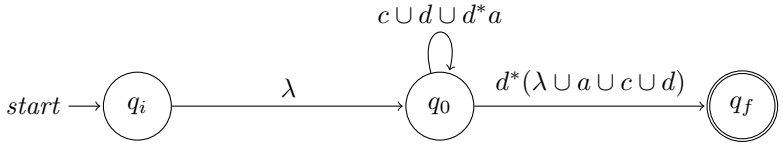
In mod similar, eliminam starea q_3:



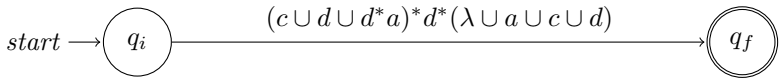
Eliminam q_2:



Eliminam starea q_1:

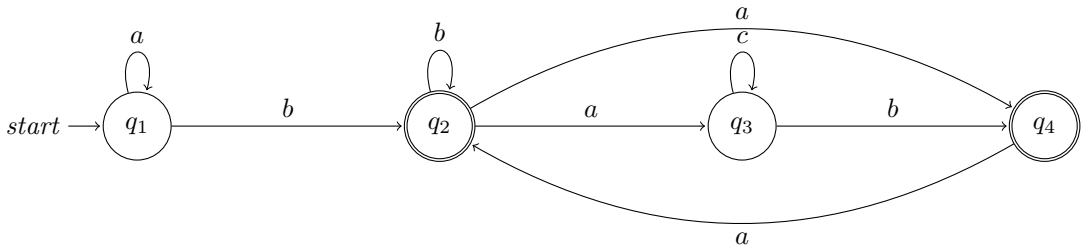


Ultima stare care mai trebuie eliminata este q_0. Putem sa gandim in felul urmator. Cand ajungem in starea q_0, avem optiunea de a ne invarti pe bucla sau de a pleca spre starea q_1, din care putem sa ne invartim pe bucla cu d, iar ulterior putem sa mergem ori inapoi, ori sa avansam, fara a ne mai putea intoarce. Practic, Putem privi muchiile dintre q_0 si q_1 ca o bucla desupra lui q_0 odata ce eliminam starea q_1.

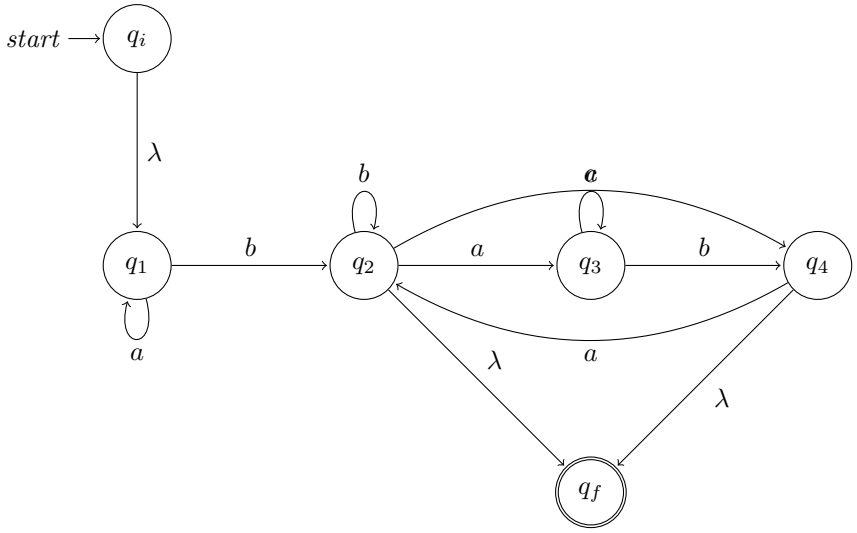


(2) *Pentru următorul automat daţi fiecare pas din algoritmul de construire al expresiei regulate echivalente. 4 stări (stările $1, 2, 3, 4$), stare iniţială starea 1, finale staruile 2 si 4. Tranzitiile: de la 1 cu a la 1 $(1, a, 1)$; de la 1 cu b la 2 $(1, b, 2)$; $(2, b, 2); (2, a, 3); (2, a, 4); (3, c, 3); (3, b, 4); (4, a, 2)$.*
Soluţie:

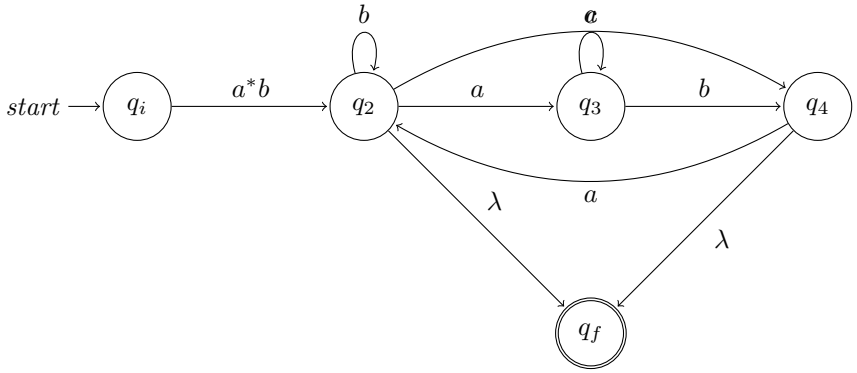
Automatul initial este:



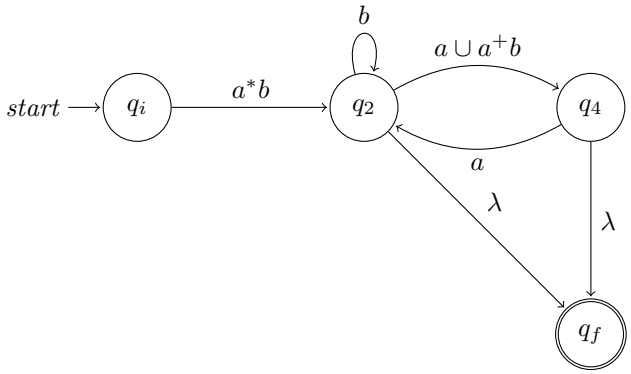
Pasul 1:
Nu avem mai multe muchii intre o pereche de stari, deci nu avem nimic de facut la acest pas.
Pasii 2 si 3:
Observam ca avem nevoie de o noua stare initiala, deoarece avem bucla pe cea de acum, si de o noua stare finala, pentru ca avem doua stari finale.



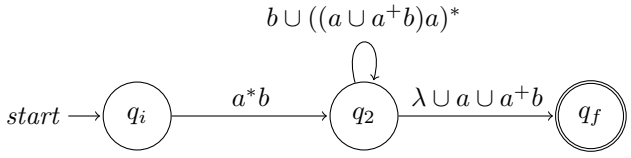
Pasul 4:
Vom incepe prin eliminarea starii q_1.



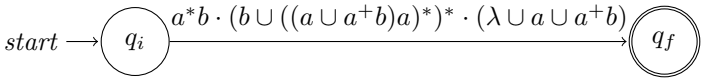
Se observa ca starea q_3 *este cea mai simplu de eliminat, avand o singura muchie de intrare si o singura muchie de iesire.*



Eliminam starea q_4. *Cele doua muchii dintre* q_2 *si* q_4 *se comporta ca o bucla desupra lui* q_2 *sau, daca nu ne intoarcem in* q_2 *dupa ce am ajuns in* q_4, *atunci este ca si cum am aveam inca o muchie de la* q_2 *la* q_f.



Ultima stare care mai trebuie eliminata este q_2.



(3) (a) *Construiti un DFA care sa accepte limbajul* $L = \{0^{3k}1^1 \mid k \geq 2\}$*;*
(b) *Să se construiască o expresie regulată pentru automatul de la punctul a.*

(4) (a) *Construiţi o expresie regulată pentru limbajul* $L = \{w \mid w \in \{0,1,2\}^*\}$*, cu condiţia că* w *are în componenţă* 01010110*;*
(b) *Să se construiască automatul DFA pentru expresia regulată de la punctul a.*

(5) *Folosind algoritmul de la curs/seminar, transformaţi următorul automat finit într-o expresie regulată echivalentă (faceţi desenele şi pentru paşii intermediari).*