Expresii regulate

In acest capitol vom arata faptul ca un limbaj regulat (acceptat de un automat finit) poate fi descris prin expresii regulate. O expresie regulata se descrie prin string-uri formate cu simboluri din Σ, folosind paranteze () pentru grupare, precum si operatorii +, · si *. Operatorul + este folosit pentru reuniune, operatorul · este folosit pentru concatenare, iar operatorul * inseamna ñde ori cate oriñ.

Formal, operatorii + si · se definesc pentru doua multimi de simboluri *A* si *B* astfel:

A+B = A∪B
A·B = {vw | v ∈ A, w ∈ B}

Practic, pentru A·B se calculeaza produsul cartezian dintre A si B, cu deosebirea ca fiecare element din A·B nu reprezinta o pereche ordonata (intre paranteze) de elemente din A si B, ci reprezinta un nou cuvant obtinut prin alipirea a doua cuvinte, unul din prima multime, al doilea din a doua.

Daca A = {λ, ab} si B = {a, b, ab}, atunci avem:

A+B = {λ, ab, a, b}
A·B = {a, b, ab, aba, abb, abab}

Formal, operatorul * (*Kleene star*) se defineste pentru o multime de simboluri *M* astfel:

$$M^* = \bigcup_{i=0}^{+\infty} M_i \text{, unde } M_i = \begin{cases} \{\lambda\},\ daca\ i = 0 \\ \{wv \mid w \in M_{i-1},\ v \in M\},\ altfel \end{cases}, \forall i \geq 0.$$

Evident, in definitia de mai sus, M_i contine cuvintele de lungime *i* formate cu simboluri din multimea *M*. Multimea *M** este infinita, daca M nu este vida.

Daca *M* = {a, b}, atunci avem:

*M** = {λ, a, aa, aaa, é ., b, bb, bbb, ab, aab, é ., abb, abbb, é .}.

De exemplu, expresia (a+b·c)* reprezinta cuvintele formate cu simbolul *a* si/sau simbolurile alipite *bc* luate de oricate ori. Expresia descrie limbajul format din cuvintele λ, a, bc, abc, bca, aa, bcbc etc.

Definitia 1: Pornind de la un alfabet dat Σ, o **expresie regulata** se construieste astfel:

1. $\varnothing$, $\{\lambda\}$ si $\{a\}$ ($a \in \Sigma$) sunt considerate expresii regulate. Aceste se numesc **expresii regulate primitive**.
2. Daca r_1 si r_2 sunt expresii regulate, atunci $r_1 + r_2$, $r_1 \cdot r_2$ sunt expresii regulate.
3. Daca r este expresie regulata, atunci r* si (r) sunt expresii regulate.
4. Un string este o expresie regulata daca si numai daca poate fi derivata din expresii regulate primitive in urma aplicarii de un numar finit de ori a regulilor 2 si 3.

Propozitia 1: Cateva proprietati imediate ale operatorilor +, · si *:

1. Operatia + este comutativa si asociativa.
2. Operatia · nu este comutativa, dar este asociativa.
3. $r + \varnothing = \varnothing + r = r$, pentru oricare expresie regulata r.
4. $r \cdot \varnothing = \varnothing \cdot r = \varnothing$, pentru oricare expresie regulata r.
5. $r \cdot \{\lambda\} = \{\lambda\} \cdot r = r$, pentru oricare expresie regulata r.
6. $\varnothing^* = \{\lambda\}$.
7. Operatia · este distributiva fata de +, adica $r_1 \cdot (r_2 + r_3) = r_1 \cdot r_2 + r_1 \cdot r_3$, pentru orice expresii regulate r_1, r_2, r_3.

Pentru o expresie regulata *r* vom nota cu *L(r)* limbajul asociat expresiei *r*.

Definitia 2: Limbajul *L(r)* asociat expresiei *r* este definit dupa regulile:

1. $\varnothing$ este expresia regulata cu care se noteaza multimea vida ($L(\varnothing) = \varnothing$).
2. λ este expresia regulata cu care se noteaza multimea $\{\lambda\}$ ($L(\{\lambda\}) = \lambda$).
3. Pentru fiecare $a \in \Sigma$ cu a se noteaza expresia regulata $\{a\}$ ($L(\{a\}) = a$).

   Pentru fiecare expresii regulate r_1 si r_2 avem:
4. $L(r_1 + r_2) = L(r_1) \cup L(r_2)$
5. $L(r_1 \cdot r_2) = L(r_1)L(r_2)$.

   Pentru fiecare expresie regulata r avem:
6. $L((r)) = L(r)$
7. $L(r^*) = L(r)^*$.

Exemplu: $L(b^* \cdot (a + b)) = L(b^*)L(a + b) = L(b)^*(L(a) \cup L(b)) = b^*\{a, b\} = \{a, ba, bba, bbba, é\ ., b, bb, bbb, bbbb\}$, adica este limbajul format cu acele cuvintele ce incep cu oricate simboluri b si se termina cu a sau b. Prin ñoricateò se intelege niciunul sau unul sau doua sau trei sau patru etc.

Observatie: Pentru operatorii +, · si * stabilim oridinea de aplicare (precedenta) astfel: * se aplica primul, apoi · si in final + (daca lipsesc parantezele). De asemenea, putem omite · in definirea expresiilor regulate. In exemplul de mai sus puteam descrie expresia regulata asociata limbajului fara · astfel b*(a+b).

Definitia 3: Doua expresii regulate se numesc **echivalente** daca desemneaza acelasi limbaj.

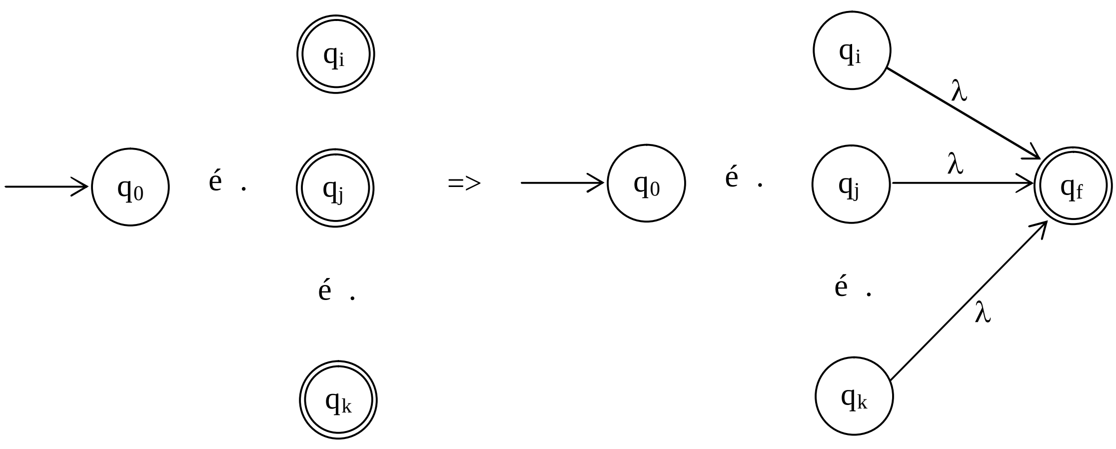
Teme:
1. Gasiti toate cuvintele de lungime mai mica decat 4 ale limbajului L((0+1)*b(a+ab)*)
2. Gasiti expresiile regulate ale limbajelor:
   $L_1 = \{a^n b^m \mid m, n \geq 0\}$
   $L_2 = \{a^n b^m \mid m, n \geq 1\}$
   $L_3 = \{a^n b^m \mid n \geq 4, m \leq 4\}$
   $L_4 = \{a^n b^m \mid n < 4, m < 4\}$.

Lema 1: Pentru fiecare automat finit $A = (Q, \Sigma, \delta, q_0, F)$ exista un automat finit $A' = (Q', \Sigma, \delta', q_0, F')$ echivalent cu A care are o singura stare finala, adica $|F'| = 1$.

Demonstratie: A' se construieste pornind de la A astfel:

1. Adaugam o stare suplimentara q_f la multimea Q, adica: $Q' = Q \cup \{q_f\}$.
2. In A' singura stare finala este q_f, adica $F' = \{q_f\}$.
3. Pentru δ' se preiau toate tranzitiile lui δ, in plus se adauga λ-tranzitiile:
   $\delta'(q, \lambda) = \{q_f\}$, pentru oricare $q \in F$.

In consecinta, graful de tranzitie atasat automatului A' se obtine pornind de la graful de tranzitie atasat automatului A adaugand arce etichetate λ de la starile finale din A la singura stare finala din A':



Automatul A　　　　　　　　Automatul A'

L(A)⊆L(A'):

Este usor de observat ca orice cuvant $w \in \Sigma^*$ acceptat de A este acceptat si de A'. Intr-adevar, cuvantului w in G_A ii corespunde un drum etichetat w de la q_0 la o stare finala q. La acest drum in $G_{A'}$ adaugam arcul (q, q_f) etichetat λ, ceea ce face ca w sa fie acceptat si de A'.

L(A')⊆L(A):

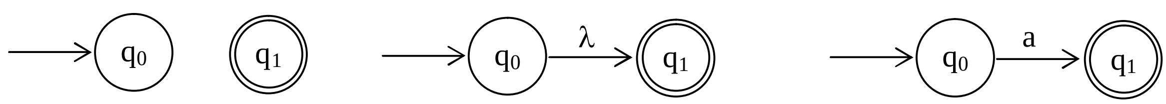
Reciproc, consideram un drum etichetat w in $G_{A'}$ de la q_0 la q_f (adica acceptat de A'). Din acest drum eliminam ultimul arc (etichetat λ) si astfel obtinem un drum etichetat w de la q_0 la un nod q care este stare finala in A, ceea ce inseamna ca w este acceptat de automatul A (q.e.d.).

Vom arata in continuare echivalenta intre notiunile de limbaj regulat si expresie regulata.

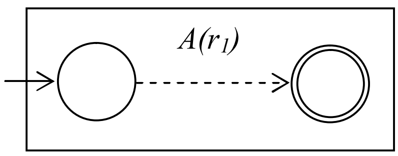
Teorema 1: Fie *r* o expresie regulata. *L(r)* este un limbaj regulat.

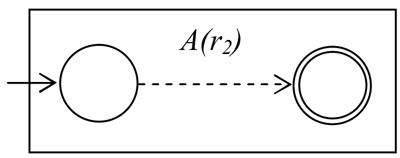
Demonstratie: Vom arata ca exista automat finit nedeterminist care accepta limbajul *L(r)*.

Construim urmatoarele automate finite nedeterministe care accepta expresiile regulate $\emptyset$, λ si $a \in \Sigma$:

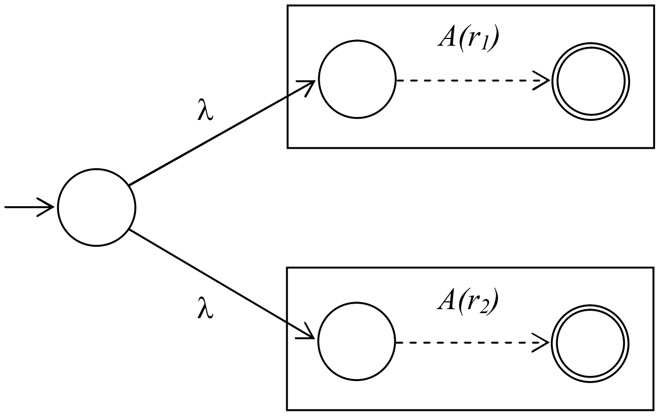


Fie doua expresii regulate r_1 si r_2 pentru care presupunem ca exista automatele finite nedeterministe $A(r_1)$ si $A(r_2)$ care accepta limbajele $L(r_1)$ si $L(r_2)$. Mai mult, folosind lema 1 presupunem (fara a reduce generalitatea problemei) ca automatele $A(r_1)$ si $A(r_2)$ au fiecare cate o singura stare finala. Avand fiecare cate o singura stare finala, automatele $A(r_1)$ si $A(r_2)$ le putem reprezenta in felul urmator:

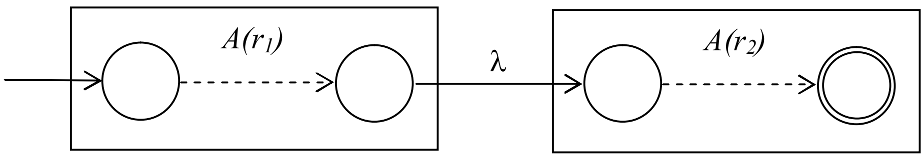




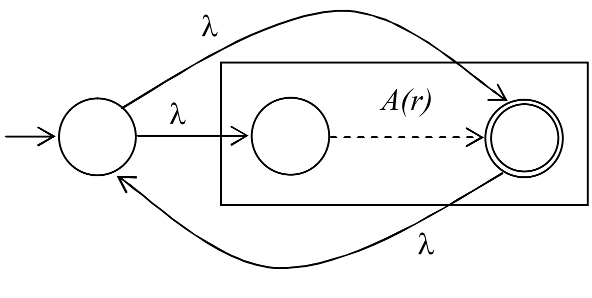
Pentru expresia regulata $r_1 + r_2$ construim automatul $A(r_1 + r_2)$ care este usor de observat ca accepta limbajul $L(r_1 + r_2)$:



Pentru expresia regulata $r_1 \cdot r_2$ construim automatul $A(r_1 \cdot r_2)$ care este usor de observat ca accepta limbajul $L(r_1 \cdot r_2)$:



Pentru expresia regulata r^* construim automatul $A(r^*)$ pornind de la automatul $A(r)$ (care are o singura stare finala):



Este usor de observat ca automatul $A(r^*)$ de mai sus accepta limbajul $L(r^*)$.

Evident, expresiile regulate r si (r) sunt echivalente.

In concluzie, am prezentat modul in care se poate construi un automat finit nedeterminist care accepta limbajul $L(r)$ pentru orice expresie regulata r (am prevazut toate situatiile din definitia 1). Asadar, teorema 1 este demonstrata.

Tema: Construiti cate un automat finit pentru fiecare dintre urmatoarele expresii regulate:

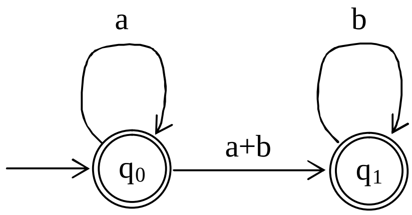
1. (a + b · c)*
2. b* · (a + b)
3. a · (b + c*)*
4. a* · (a* · b + c)

Pentru a arata in continuare faptul ca pentru orice limbaj regulat *L* exista o expresie regulata *r* astfel incat *L* = *L(r)*, introducem mai intai notiunea de graf de tranzitie generalizat.

Definitia 4: Un **graf de tranzitie generalizat** se defineste in mod asemanator cu un graf de tranzitie obisnuit, numai ca pe langa simboluri si λ, arcele pot fi etichetate si cu alte expresii regulate.

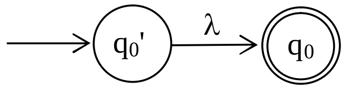
Intr-un graf de tranzitie generalizat, etichetele unui drum de la nodul initial la un nod final reprezinta o concatenare de expresii regulate, iar acest drum este la randul lui o alta expresie regulata. Aceasta expresie regulata genereaza un sublimbaj. Evident, reuniunea sublimbajelor generate de expresiile regulate corespunzatoare drumurilor de la starea initiala la o stare finala reprezinta limbajul acceptat de graful de tranzitie generalizat.

Exemplu: In figura de mai jos este prezentat un graf de tranzitie generalizat, care accepta limbajul *L(a* + a*(a+b)b*)*:



Lema 2: Pentru fiecare automat finit A exista un automat finit echivalent *A'* astfel incat starea initiala a lui A' nu este stare finala.

Demonstratie: Daca A = (Q, Σ, δ, q_0, F) are starea initiala q_0 stare finala ($q_0 \in$ F), atunci construim automatul A' = (Q', Σ, δ', q_0', F) in care se introduce suplimentar starea q_0' (Q' = Q ∪ {q_0'}), care devine stare initiala in A' si in A' se introduce λ-tranzitia de la q_0' la q_0. Este evident ca A' este echivalent cu A.



é .

Teorema 2: Pentru orice limbaj regulat *L* exista o expresie regulata *r* astfel incat $L = L(r)$.

Demonstratie: Fie *L* un limbaj regulat si A un automat finit nedeterminist astfel incat $L = L(A)$ si A are o singura stare finala, iar starea initiala nu este finala. Lemele 1 si 2 justifica faptul ca nu se reduce generalitatea problemei daca automatul A il consideram cu o singurŁ stare finalŁ diferitŁ de starea iniŞalŁ.

Fie q_0 starea initala si q_f starea finala ale automatului A, astfel încât $q_0 \neq q_f$.

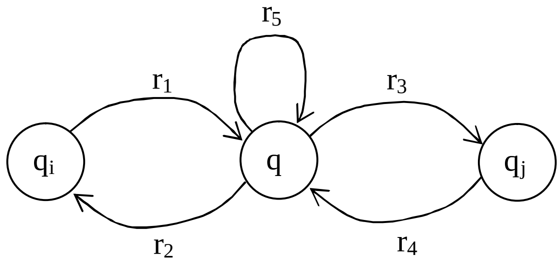
Consideram graful de tranzitii G_A atasat automatului finit A.

Fiecarui arc etichetat multiplu cu simbolurile a, b, c, d, é . si eventual λ in G_A ii inlocuim eticheta multipla cu eticheta a+b+c+d+é . si eventual $+\lambda$. Se obtine astfel un graf de tranzitii generalizat (in care toate etichetele sunt expresii regulate).

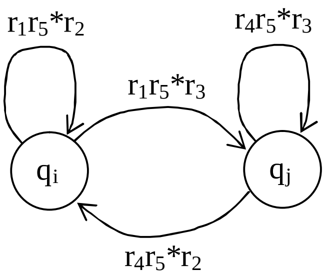
In continuare, daca graful generalizat are mai mult de doua stari, vom elimina succesiv stari ale grafului asa incat in urma eliminarii vom obtine grafuri generalizate echivalente cu mai putine noduri.

Daca graful de tramzitii generalizat are mai mult de doua stari, alegem o stare intermediara q, adica $q \neq q_0$ si $q \neq q_f$.

Consideram o pereche de stari q_i si q_j din graful de tranzitii generalizat astfel incat $q \neq q_i$ si $q \neq q_j$. Avem urmatoarea situatie (daca un arc nu exista, expresia regulata corespunzatoare este $\emptyset$):



Pentru fiecare pereche de stari q_i si q_j in graful de tranzitii generalizat, dupa eliminarea starii q, se introduc arcele si etichetele:

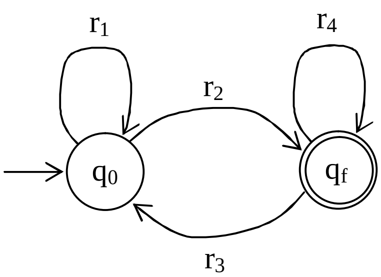


Daca in graful initial arcul etichetat cu r_i ($i \in \{1, 2, 3, 4\}$) nu exista (este etichetat cu $\emptyset$), atunci arcele care contin in eticheta expresia r_i nu se introduc in graful modificat (vezi proprietatea 4 de la propozitia 1). De asemenea, daca arcul etichetat cu r_5 nu exista (este etichetat cu $\emptyset$), atunci r_5* nu apare in etichetele grafului modificat, deoarece $\emptyset^* = \lambda$ (vezi proprietatea 6 de la propozitia 1).

Pentru fiecare arc etichetat multiplu cu expresiile regulate R_1, R_2, é ., acestea se inlocuiesc cu eticheta $R_1 + R_2 +$ é .

Asadar, in urma eliminarii starii q in maniera de mai sus se obtine un nou graf de tranzitie generalizat echivalent cu cel initial, de asemenea avand ca unica stare initiala pe q_0 (care nu este finala) si ca unica stare finala pe q_f.

Eliminam succesiv starile intermediare ale grafului de tranzitii generalizat in maniera de mai sus, pana cand se ajunge la un graf de tranzitii generalizat care are ca stari numai pe q_0 si pe q_f. Adica obtinem un graf de tranzitii generalizat de forma:



Evident, in graful de mai sus unele arce pot lipsi, situatie echivalenta cu a introduce arcul respectiv etichetat cu $\emptyset$.

Pentru graful de tranzitii generalizat de mai sus se obtine expresia regulata echivalenta:

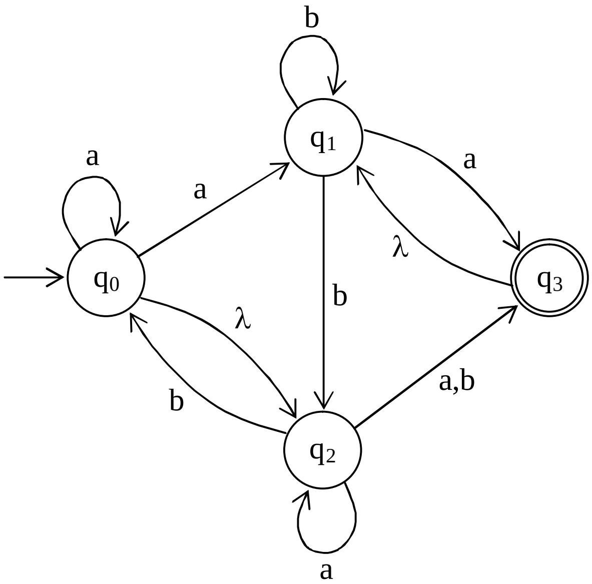
$$r = r_1^* r_2 (r_4 + r_3 r_1^* r_2)^*.$$

Asadar, pornind de la automatul A am gasit expresia regulata echivalenta, adica avem: $L = L(A) = L(r)$ (q.e.d.).

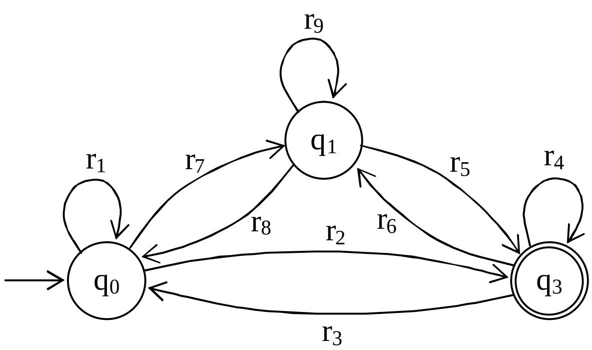
Teoremele 1 si 2 arata faptul ca multimea limbajelor regulate (acceptate de automate finite) coincide cu multimea limbajelor generate de expresii regulate. Cu alte cuvinte, limbajele generate de automate finite sau de expresii regulate le putem numi limbaje regulate.

In continuare luam un exemplu prin care ilustram modul in care se construieste expresia regulata echivalenta cu un automat finit.

Fie automatul finit nedeterminist al carui graf de tranzitie este prezentat in figura urmatoare:



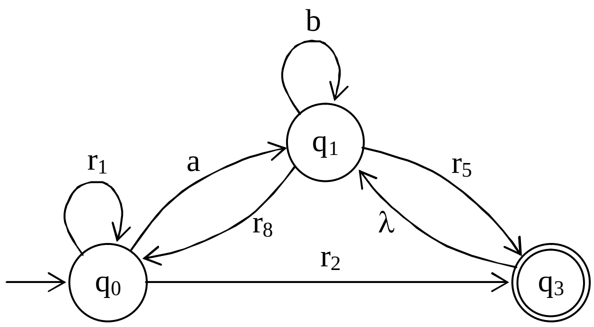
Multieticheta a,b a arcului (q_2, q_3) o inlocuim cu a+b.
Sa eliminam intr-o prima faza starea q_2.



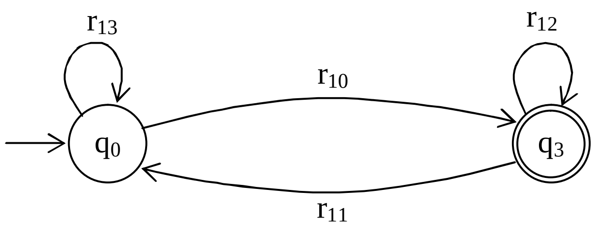
Expresiile regulate r_i ($i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$) sunt:

r_1 = a+a*b
r_2 = a*(a+b)
$r_3 = \emptyset$
$r_4 = \emptyset$
r_5 = a+ba*(a+b)
$r_6 = \lambda$
r_7 = a
r_8 = ba*b
r_9 = b

Adica avem:



Sa eliminam acum starea q_1:



Expresiile regulate r_i ($i \in \{9, 10, 11, 12\}$) sunt:

$r_{10} = \quad r_2+ab^*r_5 = \quad a^*(a+b)+ab^*(a+ba^*(a+b))$
$r_{11} = \quad b^*r_8 = \quad b^*ba^*b$
$r_{12} = \quad b^*r_5 = \quad b^*(a+ba^*(a+b))$
$r_{13} = \quad r_1+ab^*r_8 = \quad a+a^*b+ab^*ba^*b$

Obtinem urmatoarea expresie echivalenta automatului finit nedeterminist dat:

$r = r_{13}^*r_{10}(r_{12} + r_{11}r_{13}^*r_{10})^*$

Teme:
1. Sa se scrie expresia regulata pentru numere intregi in limbajul C.
2. Aceeasi cerinta pentru numere reale.
3. Denumire de variabila in limbajul C.
4. Denumire de functie in limbajul C.

Gramatici regulate

Un al treilea mod de a descrie limbajele regulate il reprezinta gramaticile regulate.

Definitia 5: G = (V, Σ, S, R) se numeste **gramatica**, unde:

V = multimea **simbolurilor neterminale** sau **variabile**
Σ = multimea **simbolurilor terminale**, $V \cap \Sigma = \emptyset$
$S \in V$ este **simbolul initial**
R = o **multime de reguli**, fiecare regula este de forma:

$$(V \cup \Sigma)^*V(V \cup \Sigma)^* \rightarrow (V \cup \Sigma)^*$$

Definiatia 6: O gramatica G se numeste **gramatica liniara**, daca toate regulile ei sunt de forma:

$A \rightarrow xB$
$A \rightarrow Bx$
$A \rightarrow x$,

unde $A, B \in V$ si $x \in \Sigma^*$.

Definiatia 7: O gramatica G se numeste **gramatica liniara la dreapta** (**gramatica LR**), daca toate regulile ei sunt de forma:

$A \rightarrow xB$
$A \rightarrow x$,

unde $A, B \in V$ si $x \in \Sigma^*$.

Definiatia 8: O gramatica G se numeste **gramatica liniara la stanga** (**gramatica LL**), daca toate regulile ei sunt de forma:

$A \rightarrow Bx$
$A \rightarrow x$,

unde $A, B \in V$ si $x \in \Sigma^*$.

Definiatia 9: O gramatica G se numeste **gramatica regulata**, daca este lineara la dreapta sau lineara la stanga. Gramaticile LL se mai numesc si **gramatici regulate la stanga**, iar gramaticile LR se mai numesc si **gramatici regulate la dreapta**.

Teorema 3: Fie G = (V, Σ, S, R) o gramatica LR. Limbajul L(G) generat de G este limbaj regulat.

Demonstratie:

Orice regula a gramaticii LR contine reguli de forma:

1. $V_i \rightarrow a_1a_2$ é .a_kV_j, unde a_1, a_2, é ., $a_k \in \Sigma$ ($k \geq 1$) si V_i, $V_j \in V$.

sau

2. $V_i \rightarrow a_1a_2$ é .a_k, unde a_1, a_2, é ., $a_k \in \Sigma$ ($k \geq 1$) si $V_i \in V$

sau

3. $V_i \rightarrow V_j$

sau

4. $V_i \rightarrow \lambda$.

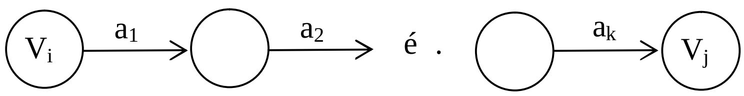
In continuare construim un automat finit nedeterminist A = (Q, $\sum$, δ, S, F) astfel incat *L(*A*)* = *L(*G*)*. Automatul A are acelasi alfabet, iar starea initiala este simbolul initial S al gramaticii G.

Vom arata acum cum se contruiesc Q, δ si F. Pentru inceput in Q consideram ca se afla toate simbolurile neterminale ale gramaticii G, adica $V \subseteq Q$.

Pentru fiecare regula de forma:

(1) $V_i \rightarrow a_1a_2$ é .a_kV_j

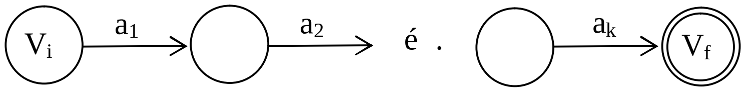
in automatul A vom defini tranzitia extinsa $\delta^*(V_i, a_1a_2$ é .$a_k) = V_j$. Practic, vom introduce k-1 noi stari in Q si tranzitiile:



Pentru fiecare regula de forma:

(2) $V_i \rightarrow a_1a_2$ é .a_k

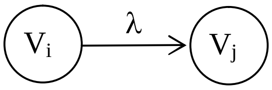
in automatul A vom defini tranzitia extinsa $\delta^*(V_i, a_1a_2$ é .$a_k) = V_f$. Practic vom introduce k noi stari in Q, dintre care ultima (notata V_f) este si stare finala (V_f se introduce si in F). In A se introduc tranzitiile:



Pentru fiecare regula de forma:

(3) $V_i \rightarrow V_j$

in automatul A vom defini λ-tranzitia $\delta(V_i, \lambda) = V_j$:



Pentru fiecare regula de forma:

(4) $V_i \rightarrow \lambda$

starea V_i o facem stare finala, adica o introducem in F ($V_i \in F$):

Sa aratam acum ca $L(G) = L(A)$.

Fiecare cuvant $w \in L(G)$. Fiind generat de gramatica G inseamna ca w a fost obtinut in urma aplicarii unor reguli de forma:

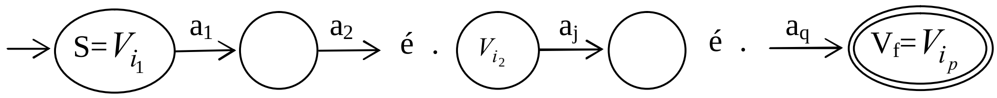
$S \rightarrow v_1V_1$
$V_1 \rightarrow v_2V_2$
$V_2 \rightarrow v_3V_3$
é .
$V_{t-1} \rightarrow v_t$,

unde $V_1, V_2,$ é ., $V_{t-1} \in V$ si $v_1, v_2,$ é ., $v_t \in \Sigma^*$. Evident, $w = v_1v_2$é .$v_t \in \Sigma^*$.

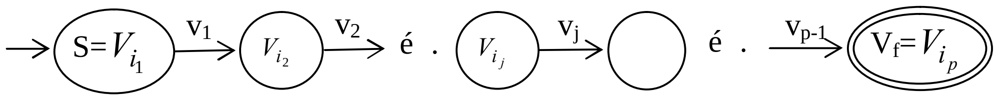
Primele t-1 reguli corespund situatiilor de tip (1) si/sau (3) descrise mai sus. Ultima regula (a t-a) este forma (2) sau (4), prin care se ajunge intr-o stare finala din F. Se obtine astfel un drum etichetat w in graful de tranzitii atasat automatului A, drum care pleaca din starea initiala S si ajunge intr-o stare finala din F. Cu alte cuvinte, cuvantul w este acceptat de catre A.

Pentru incluziunea inversa consideram acum un cuvant w $\in$ *L(*A*)*. El corespunde unui drum etichetat w in G_A de la S la o stare finala V_f din F.

Datorita faptului ca automatul A a fost construit pe baza unor reguli de tip (1), (2), (3) si (4), inseamna ca drumul de la S la V_f este de forma:



unde a_1, a_2, é ., a_q $\in \Sigma$ (w = a_1a_2é .a_q) si V_{i_1}, V_{i_2}, é ., $V_{i_{p-1}} \in$ V, iar celelalte stari (cu exceptia V_{i_p}) nu sunt din V, sunt stari ce au fost introduse cu reguli de forma (1) si/sau (2). Rezulta din (1), (2) si eventual (3) ca avem:



unde v_1, v_2, é ., $v_{p-1} \in \Sigma^*$ (w = v_1v_2é .v_{p-1}), ceea ce corespunde regulilor:

$V_{i_1} \to v_1 V_{i_2}$
$V_{i_2} \to v_2 V_{i_3}$
é .
$V_{i_{p-1}} \to v_{p-1}$.

De aici rezulta cuvantul w este generat cu reguli ale gramaticii G, adica w $\in$ *L(*G*)*.
In concluzie, *L(*G*)* = *L(*A*)*, *L(*G*)* este limbaj regulat (q.e.d.).

Teorema 4: Daca L este un limbaj regulat, atunci exista o gramatica LR notata G astfel incat L = *L(*G*)*.

Demonstratie: Fie A = (Q, Σ, δ, q_0, F) un automat finit determinist care accepta limbajul L, adica L = *L(*A*)*. Ne propunem sa construim o gramatica G = (Q, Σ, q_0, R) astfel incat *L(*G*)* = *L(*A*)*.

Pentru fiecare tranzitie din A:

δ(q, c) = p, unde q, p $\in$ Q si c $\in \Sigma$,

in multimea R a gramaticii G adaugam regula:

(1) q $\to$ cp,

daca p nu este stare finala sau

(2) $q \to c$,

daca p este stare finala.

Sa aratam acum ca $L(A) = L(G)$.

Fie $w = a_1a_2\ldots a_k \in L(A)$. Acest lucru inseamna ca avem:

$\delta^*(q_0, w) = q_f \in F$, adica exista starile $q_{i_j} \in Q$ ($j \in \{1, 2, \ldots, k+1\}$) astfel incat:

$$\begin{cases} \delta(q_0 = q_{i_1}, a_1) = q_{i_2} \\ \delta(q_{i_2}, a_2) = q_{i_3} \\ \ldots. \\ \delta(q_{i_{k-1}}, a_{k-1}) = q_{i_k} \\ \delta(q_{i_k}, a_k) = q_{i_{k+1}} = q_f \in F \end{cases}$$

Din (1) si (2) rezulta ca tranzitiile de mai sus corespund regulilor:

$$\begin{cases} q_0 = q_{i_1} \to a_1 q_{i_2} \\ q_{i_2} \to a_2 q_{i_3} \\ \ldots. \\ q_{i_{k-1}} \to a_{k-1} q_{i_k} \\ q_{i_k} \to a_k \end{cases}$$

ceea ce inseamna ca $w = a_1a_2\ldots a_k \in L(G)$.

Pentru incluziunea inversa consideram un cavant $w = a_1a_2\ldots a_k \in L(G)$. Deoarece gramatica G contine numai reguli de formele (1) si (2), rezulta ca exista $q_{i_j} \in Q$ ($j \in \{1,2,\ldots,k\}$) astfel incat:

$$\begin{cases} q_0 = q_{i_1} \to a_1 q_{i_2} \\ q_{i_2} \to a_2 q_{i_3} \\ \ldots. \\ q_{i_{k-1}} \to a_{k-1} q_{i_k} \\ q_{i_k} \to a_k \end{cases}$$

care corespund in automatul A tranzitiilor:

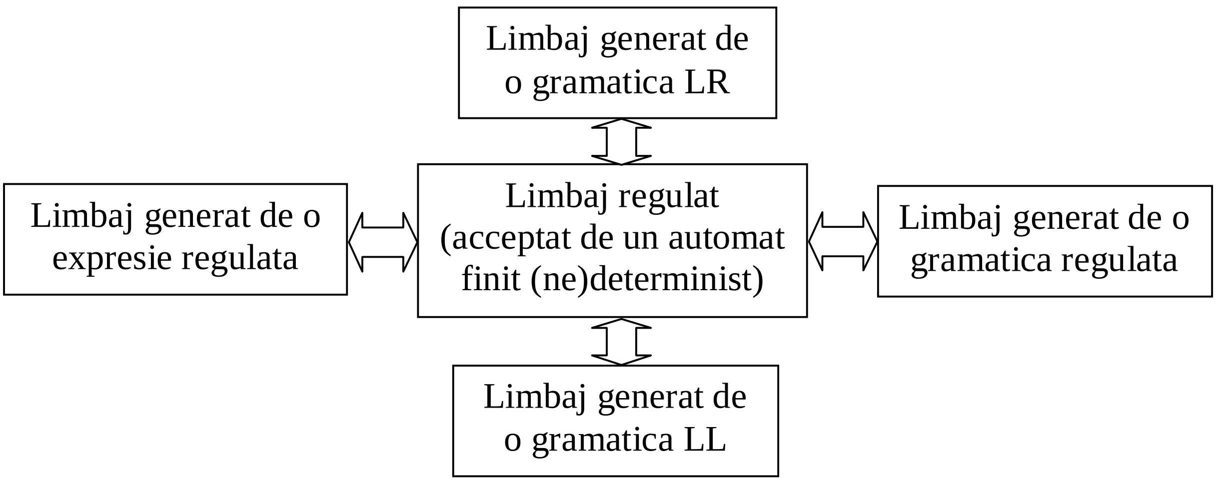
$$\begin{cases} \delta(q_0 = q_{i_1}, a_1) = q_{i_2} \\ \delta(q_{i_2}, a_2) = q_{i_3} \\ .... \\ \delta(q_{i_{k-1}}, a_{k-1}) = q_{i_k} \\ \delta(q_{i_k}, a_k) = q_f \in F \end{cases}$$

ceea ce inseamna ca w = a_1a_2é .$a_k \in$ *L(*A*)*.
Asadar, *L(*A*)* = *L(*G*)* (q.e.d.).

In mod similar se arata ca si limbajele LL sunt regulate si ca orice limbaj regulat se poate descrie cu o gramatica LL.

In concluzie, **limbajul generat de o gramatica regulata este limbaj regulat si orice limbaj regulat poate fi exprimat printr-o gramatica regulata (gramatica LR sau LL)**.

De asemenea, am demonstrat la inceputul capitolului faptul ca **limbajul generat de o expresie regulata este limbaj regulat si orice limbaj regulat poate fi exprimat printr-o expresie regulata** (rezulta din teoremele 1 si 2).



Vom studia acum cateva proprietati ale limbajelor regulate.

Teorema 5: Daca L_1 si L_2 sunt limbaje regulate, atunci si $L_1 \cup L_2$, $L_1 \cap L_2$, L_1L_2, $\overline{L_1}$, L_1^* sunt limbaje regulate.

Demostratie: L_1 si L_2 fiind limbaje regulate inseamna ca exista expresiile regulate r_1 si r_2 astfel incat $L_1 = L(r_1)$ si $L_2 = L(r_2)$.

Cum r_1+r_2, r_1r_2 si r_1* sunt expresii regulate (vezi definitia 1 a expresiilor regulate), rezulta imediat ca limbajele $L_1 \cup L_2$, L_1L_2, L_1^* sunt regulate (vezi definitia 2 a limbajelor generate de expresii regulate).

Sa aratam acum faptul ca $\overline{L_1} = \Sigma^* \setminus L_1$ este limbaj regulat.

Fie automatul finit determinist A = (Q, Σ, δ, q_0, F) care accepta L_1. Construim automatul A' = (Q, Σ, δ, q_0, Q\F). Fie un string w din multimea Σ^*. Avem una si numai una dintre urmatoarele doua situatii:

$\delta^*(q_0,w) \in F$

sau

$\delta^*(q_0,w) \in Q\setminus F$

ceea ce inseamna ca fie $w \in L(A)$, fie $w \in L(A')$. In concluzie, $\overline{L_1} = L(A')$, adica $\overline{L_1}$ este limbaj regulat.

Vom arata in final ca $L_1 \cap L_2$ este de asemenea limbaj regulat. Pentru aceasta consideram automatele finite deterministe A_1 = (Q, Σ, δ_1, q_0, F_1) care accepta L_1 si, respectiv, A_2 = (P, Σ, δ_2, p_0, F_2) care accepta L_2.

Pornind de la A_1 si A_2 construim automatul finit determinist A = (R, Σ, δ, r_0, F), unde R = Q x P (produsul cartezian), $r_0 = (q_0, p_0)$, $F = \{(q,p) \mid q \in F_1 \text{ si } p \in F_2\}$ (1). Tranzitiile automatului A se definesc pentru fiecare $q \in Q$, $p \in P$ si $c \in \Sigma$ astfel:

(2) $\delta((q,p), c) = (s,t)$,

unde $\delta_1(q, c) = s$ si $\delta_2(p, c) = t$.

Sa aratam ca $L(A) = L_1 \cap L_2$.

Fie un cuvant $w \in L(A)$. Inseamna ca exista un drum $D = ((q_0,p_0), (q_1,p_1), é .., (q_k,p_k))$ etichetat w in G_A de la $r_0 = (q_0,p_0)$ la starea $(q_k,p_k) \in F$. Din (1) si (2) rezulta ca drumul $D_1 = (q_0, q_1, é .., q_k)$ este etichetat w in G_{A1} de la q_0 la starea $q_k \in F_1$ si drumul $D_2 = (p_0, p_1, é .., p_k)$ este etichetat w in G_{A2} de la p_0 la starea $p_k \in F_2$. Cu alte cuvinte, cuvantul $w \in L_1$ si $w \in L_2$, adica $w \in L_1 \cap L_2$.

Pentru incluziunea inversa consideram cuvantul $w \in L_1 \cap L_2$. Inseamna ca exista drumul $D_1 = (q_0, q_1, é .., q_k)$ este etichetat w in G_{A1} de la q_0 la starea $q_k \in F_1$ si drumul

$D_2 = (p_0, p_1,$ é ., $p_k)$ este etichetat w in G_{A2} de la p_0 la starea $p_k \in F_2$. Construim in G_A drumul $D = (r_0=(q_0,p_0), (q_1,p_1),$ é ., $(q_k,p_k))$. Din (2) rezulta ca drumul D este etichetat w si, din (1), rezulta ca $(q_k,p_k) \in F$. In consecinta, $w \in L(A)$.

Asadar, $L(A) = L_1 \cap L_2$ (q.e.d.).

Teorema 6: Orice limbaj finit L (cu numar finit de cuvinte) este regulat.

Demonstratie: Daca L este finit, atunci inseamna ca $L = \{w_1, w_2,$ é ., $w_n\}$, $w_i \in \Sigma^*$, $\forall$ i $\in \{1, 2,$ é ., $n\}$. Construim gramatica regulata $G = (V=\{S\}, \Sigma, S, R)$ formata cu regulile:

$S \rightarrow w_i$, $\forall$ i $\in \{1, 2,$ é ., $n\}$,

care evident ca genereaza L (q.e.d.).

Asadar, daca un limbaj nu este regulat, atunci el este cu siguranta infinit (este format cu un numar infinit de cuvinte). Sa aratam acum faptul ca nu toate limbajele sunt regulate:

Propozitia 2: Limbajul $L = \{ a^n b^n \mid n \geq 0 \}$ nu este limbaj regulat.

Demonstratie: Presupunem prin absurd ca exista un automat finit determinist $A = (Q,\{a, b\},\delta,q_0,F)$ astfel incat $L(A) = L$.

Cum multimea starilor Q este finita rezulta ca exista doua numere naturale diferite m si n ($m \neq n$) astfel incat $\delta^*(q_0, a^m) = \delta^*(q_0, a^n)$, in caz contrar, ar insemna ca pentru fiecare numar natural $n > 0$ $\delta^*(q_0, a^n)$ ar fi o stare diferita de starile $\delta^*(q_0, a^k)$ $(k < n)$, ceea ce ar rezulta ca multimea Q ar fi infinita (fals). Deci, exista $n, m \geq 0$, $m \neq n$ si o stare $q \in Q$ astfel incat:

(1) $\delta^*(q_0, a^m) = \delta^*(q_0, a^n) = q$

Cum cuvantul $a^n b^n$ este acceptat de automatul A, rezulta din (1) ca:

(2) $\delta^*(q_0, a^n b^n) = \delta^*(\delta^*(q_0, a^n), b^n) = \delta^*(q, b^n) = q_f \in F$.

Pentru cuvantul $w = a^m b^n$ avem din (1) si (2) ca:

(3) $\delta^*(q_0, a^m b^n) = \delta^*(\delta^*(q_0, a^m), b^n) = \delta^*(q, b^n) = q_f \in F$,

ceea ce inseamna ca w este acceptat de A (contradictie cu faptul ca $m \neq n$ si cu presupunerea initiala ca $L(A) = L =\{ a^n b^n \mid n \geq 0 \}$).

Asadar, limbajul L nu este regulat.

In continuare vom da o proprietate pe care o au limbajele formale inifinite (cu numar infinit de cuvinte). Aceasta proprietate ajuta la justificarea faptului ca un limbaj nu este regulat. Cu alte cuvinte, un limbaj care nu respecta urmatoarea proprietate nu este regulat.

Teorema 7: Pentru fiecare limbaj regulat infinit L exista un numar natural $m > 0$ astfel incat orice cuvant $w \in L$ cu $|w| \geq m$ poate fi descompus sub forma $w = xyz$, unde $|xy| \leq m$, $|y| \geq 1$ si pentru orice $i \geq 0$ avem ca $w_i = xy^iz \in L$.

Demonstratie: Fie $A = (Q, \Sigma, \delta, q_0, F)$ un automat finit determinist care accepta limbajul L, adica $L(A) = L$. Notam cu $m = |Q|$. Fiind un limbaj infinit, inseamna ca in L exista cuvinte de lungime mai mare sau egala m. Fie $w = c_1c_2$é .c_k un astfel de cuvant ($w \in L(A)$ si $k \geq m$). In graful de tranzitii G_A asociat automatului A exista un drum D = (q_0,q_1,é .,q_k) etichetat w si $q_k \in F$. Cum drumul D trece prin $k+1 > m$ stari din Q si $|Q| = m$, rezulta ca drumul D nu este elementar, el continand cel putin un circuit. Consideram $i, j \in \{0, 1,$ é $., m\}$, $i < j$ astfel incat $q_i = q_j$.

Este clar ca daca eliminam din drumul circuitul ce incepe de pe pozitia i (nodul q_i), se obtine drumul D_0 = (q_0,q_1,é .,q_i,q_{j+1},é .,q_k) etichetat $w_0 = xz$, unde $x = c_1c_2$é .c_i si $z = c_{j+1}$é .c_k, adica w_0 este acceptat de automatul A.

Notam $y = c_{i+1}c_{i+2}$é .c_j. Daca adaugam de $i > 0$ ori circuitul dat de cuvantul y (pe care l-am eliminat in constructia lui w_0) intre x si y, obtinem drumul etichetat $w_i = xy^iz$ de la q_0 la $q_k \in F$, ceea ce inseamna ca w_i este acceptat de A. Mai mult, $|xy| \leq m$ (deoarece $j \leq m$) si $|y| \geq 1$ (q.e.d.).

Teorema 7': Pentru fiecare limbaj regulat infinit L exista un numar natural $m > 0$ astfel incat orice cuvant $w \in L$ cu $|w| \geq m$ poate fi descompus sub forma $w = xyz$, unde $|yz| \leq m$, $|y| \geq 1$ si pentru orice $i \geq 0$ avem ca $w_i = xy^iz \in L$.

Demonstratie: Este similara cu cea de la teorema 7, singura deosebire este ca se considera i si j la sfarsitul drumului D, adica $i, j \in \{k-m, k-m+1,$ é $., k-1, k\}$, $i < j$ astfel incat $q_i = q_j$.

Sa demonstram acum propozitia 2 folosind teorema 7.

Este evident ca limbajul $L = \{ a^nb^n \mid n \geq 0 \}$ este infinit.

Presupunem prin absurd ca L este limbaj regulat. Fie $m > 0$ valoarea data de teorema 7 astfel incat orice cuvant $w \in L$ cu $|w| \geq m$ poate fi descompus sub forma $w = xyz$, unde $|xy| \leq m$, $|y| \geq 1$ si pentru orice $i \geq 0$ avem ca $w_i = xy^iz \in L$.

Consideram cuvantul $w = a^mb^m$. Evident, $|w| = 2m > m$. L fiind limbaj regulat, inseamna ca exista descompunerea $w = xyz$, unde $|xy| \leq m$, $|y| \geq 1$ si pentru orice $i \geq 0$ avem ca $w_i = xy^iz \in L$.

Din $|xy| \leq m \Rightarrow xy = a^t$.

$y = a^k$ $(m \geq t \geq k \geq 1) \Rightarrow w_0 = a^{m-k}b^m \notin L$

Am obtinut contradictie. Asadar, limbajul L nu este regulat.

Tema: Sa se demonstreze ca urmatoarele limbaje nu sunt regulate:

1. $L_1 = \{ a^n \mid n = k^2, k \geq 0 \}$
2. $L_2 = \{ a^n \mid n = 2^k, k \geq 0 \}$
3. $L_3 = \{ a^{n!} \mid n \geq 0 \}$
4. $L_4 = \{ (ab)^p b^n \mid p > n \geq 0 \}$
5. $L_5 = \{ a^p b^n \mid p, n \geq 0, p \neq n \}$

Indicatii:

4) $w = (ab)^{m+1}b^{m}$ (teorema 7')
5) $w = a^{m!}b^{(m+1)!}$