

Limbaje Formale și Compilatoare (LFC) - Curs -

Ș.L.dr.ing. Octavian MACHIDON

octavian.machidon@unitbv.ro



Universitatea
Transilvania
din Brașov



Astăzi



- Expresii regulate
 - Definiție
 - Proprietățile expresiilor regulate. Exemple de expresii regulate
 - Conversia unei expresii regulate într-un AFD
 - Conversia unui AFD într-o expresie regulată

Gramatici de tip 3 (regulate)

- O gramatică $G = (N, T, S, P)$ este de tip 3 dacă regulile sale au forma:

$$A \rightarrow u \text{ sau } A \rightarrow uB \text{ unde } A, B \in N \text{ și } u \in T^*.$$

- Exemplu:

$$G = (\{D\}, \{0, 1, \dots, 9\}, D, P)$$

Unde P este:

$$D \rightarrow 0D \mid 1D \mid 2D \mid \dots \mid 9D$$

$$D \rightarrow 0 \mid 1 \mid \dots \mid 9$$

Ce sunt expresiile regulate?

Reprezentarea limbajelor de tip 3 prin expresii algebrice

Dacă R și S sunt expresii regulate atunci sunt expresii regulate și următoarele:

- a , unde a este orice caracter
- ε , cuvântul vid
- ϕ , mulțimea vidă
- $R|S$, alternarea, R sau S (notație echivalentă: $R \cup S$)
- RS , concatenarea, R urmat de S (notație echivalentă $R \circ S$)
- R^* , închiderea/produsul Kleene, R de zero sau mai multe ori
- $(R|S)$ este o expresie regulată ce descrie limbajul $L(R) \cup L(S)$
- (RS) este expresie regulată ce descrie limbajul $L(R) \cdot L(S)$
- (R^*) este expresie regulată ce descrie limbajul $L(R)^*$

Ordinea de prioritate a operatorilor este $\star, \cdot, |$

Extensii ale expresiilor regulate

Dacă R este o expresie regulată, la fel sunt și următoarele:

- $R? = \varepsilon | R$ (zero or un singur R)
- $R+ = RR^*$ (unul sau mai mulți R)
- $(R) = R$ (parantezele au rol de grupare)
- $[abc] = a|b|c$ (oricare caracter din paranteze)
- $[a - e] = a|b|...|e$ (interval de caractere)
- $[\wedge ab] = c|d|... ($ (orice mai puțin caracterele enumerate)
- $abc =$ șirul de caractere abc
- $\backslash(=$ caracterul $($

Proprietăți ale expresiilor regulate (1)

- $(p|q)|r = p|(q|r)$ Asociativitatea la reuniune.
- $(pq)r = p(qr)$ Asociativitatea la concatenare.
- $p|q = q|p$ Comutativitatea la reuniune.
- $p \cdot \varepsilon = \varepsilon \cdot p = p$ Simbolul vid este element neutru la concatenare.
- $p|\phi = p|p = p$ Mulțimea vidă este element neutru la reuniune.
- $\phi \cdot p = p \cdot \phi = \phi$ Concatenarea cu mulțimea vidă.

Proprietăți ale expresiilor regulate (2)

- $p(q|r) = pq|pr$ Distributivitatea operațiilor de concatenare și reuniune.
- $(p|q)r = pr|qr$ Distributivitatea operațiilor de concatenare și reuniune.
- $\varepsilon|pp^* = p^*$ Proprietăți ale produsului Kleene.
- $\varepsilon|p^*p = p^*$ Proprietăți ale produsului Kleene.

Example (1)

Tabelul 4-1: Exemple de expresii regulate pentru identificarea elementelor de cod sursă C.

Expresii regulate	Cuvinte din L(R)
digit = [0-9]	0 , 1, 2, ...
posint = digit+	8, 412, ...
int = -? posint	-42, 105, ...
real = int ((.posint)?)	-1.56, 12, 25.0, ...
[a-zA-Z_][a-zA-Z0-9_]	Identificatori C
else	Cuvântul cheie else

Example (2)

Alte exemple de expresii regulate (pentru toate alfabetul este $\Sigma = 0, 1$):

1. Expresia regulată Σ^* descrie limbajul tuturor șirurilor de caractere peste alfabetul Σ . Cu alte cuvinte, $L(\Sigma^*) = \Sigma^*$.
2. Limbajul tuturor șirurilor de caractere care încep cu 1 este descris de expresia regulată $1\Sigma^*$.
3. Limbajul tuturor șirurilor de caractere care se termină cu 1 este descris de expresia regulată Σ^*1 .
4. Limbajul tuturor șirurilor de caractere de lungime cel puțin 2 care încep și se termină cu același simbol este descris de expresia regulată $0\Sigma^*0 \cup 1\Sigma^*1$.
5. Limbajul șirurilor de caractere care conțin subșirul 001 este descris de expresia regulată $\Sigma^*001\Sigma^*$.
6. Limbajul șirurilor de caractere care conțin un număr par de 1 este descris de expresia regulată $(0^*10^*1)^*0^*$.
7. Limbajul șirurilor de caractere care conțin un număr de 1 care este multiplu de k este descris de expresia regulată $(0^*1)^k)^*0^*$.

Cursul viitor:

- Analiza Lexicală

Conversia unei expresii regulate într-un AFD

- O expresie regulată poate fi privită ca o structură destinată validării unui șir de simboluri de intrare
- O expresie regulată poate fi convertită în primă fază într-un AFN, iar folosind algoritmul de transformare se poate obține AFD-ul echivalent
- Pentru a simplifica procedura conversiei E.R. – AFN, o putem împărți în sub-probleme gestionabile individual

Conversia unei expresii regulate într-un AFD

Vom începe cu primele două cazuri, cele elementare: expresiile regulate α și ε . Simbolul α va fi validat de un automat care are o stare inițială, una acceptoare și o tranziție cu simbolul α de la starea inițială la cea acceptoare (Figura 4-1).

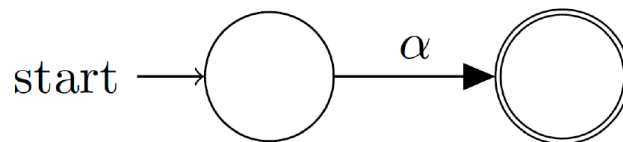


Figura 4-1: Automatul corespunzător expresiei regulate α .

Aceeași structură este folosită pentru a valida simbolul nul ε , și anume o ε -tranziție către starea acceptoare în locul uneia cu un simbol al alfabetului. Cu alte cuvinte, expresia regulată ε corespunde automatului din Figura 4-2.

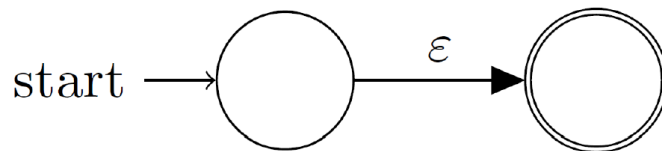


Figura 4-2: Automatul corespunzător expresiei regulate ε .

Conversia unei expresii regulate într-un AFD

În ce privește cazurile expresiilor regulate compuse din mai multe structuri, procedura de conversie acționează recursiv. Astfel, rezultatul conversiei unei expresii regulate oarecare r într-un AFN este cel din Figura 4-3. Starea din partea stângă reprezintă starea inițială a automatului corespunzător expresiei r , iar starea din partea dreaptă reprezintă stările acceptoare ale aceluiași automat.

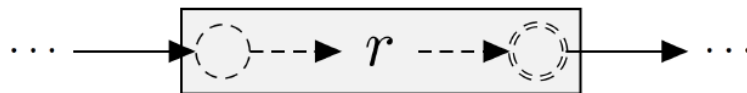


Figura 4-3: Automatul corespunzător expresiei regulate r .

Conversia unei expresii regulate într-un AFD

În cazul unei structuri de alegere între două expresii regulate r_1 și r_2 , scrisă sub forma $r_1|r_2$, automatul este compus dintr-o stare inițială nouă cu ε -tranziții către stările inițiale ale automatelor lui r_1 și r_2 , și o nouă stare acceptoare cu ε -tranziții de la stările acceptoare ale celor două automate, după cum este ilustrat în Figura 4-4.

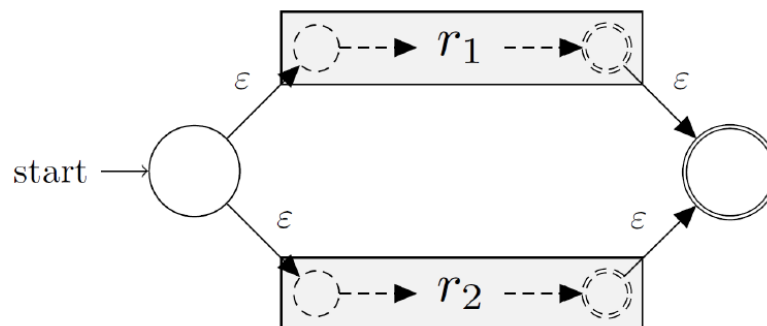


Figura 4-4: AFN-ul corespunzător expresiei $r_1|r_2$.

O secvență de două expresii regulate concatenate r_1 și r_2 , scrisă sub forma r_1r_2 se convertește într-un AFN prin simpla adăugare de tranziții între stările acceptoare ale AFN-ului lui r_1 și starea inițială a AFN-ului lui r_2 , ca în Figura 4-5.

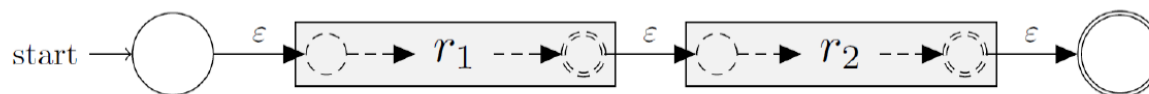


Figura 4-5: AFN-ul corespunzător expresiei r_1r_2 .

Conversia unei expresii regulate într-un AFD

AFN-ul expresiei r^* are o ε -tranziție de la starea sa inițială la starea inițială a lui r , și o ε -tranziție de la starea acceptoare a lui r la propria stare acceptoare. În plus, există ε -tranziții de la starea inițială la cea acceptoare și invers. Această construcție permite ca r să fie trecută cu vederea sau repetată de câte ori este necesar. AFN-ul corespunzător expresiei regulate r^* este ilustrat în Figura 4-6.

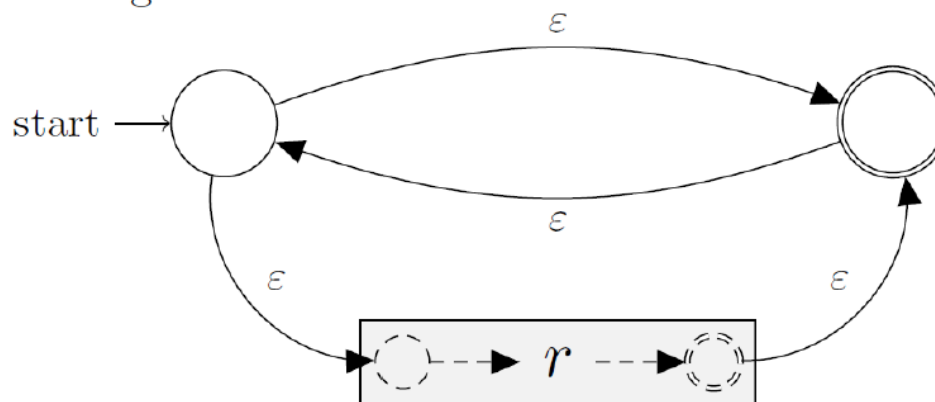


Figura 4-6: AFN-ul corespunzător expresiei r^* .

Pe baza regulilor de mai sus se obține AFN-ul corespunzător unei expresii regulate. Prin aplicarea algoritmului de transformare prezentat în capitolul precedent, se obține AFD-ul echivalent.

Conversia unui AFD într-o expresie regulată

În practică există cazuri de limbaje pentru care este mai ușor să se construiască AFD-ul direct decât să se scrie expresia regulată corespunzătoare. Dacă este necesară și expresia regulată pentru acel limbaj, atunci după construirea AFD-ului acesta se poate converti la expresia regulată echivalentă. Pentru a elabora un algoritm generic de conversie a unui AFD într-o expresie regulată, vom începe de la câteva cazuri particulare simple, crescând complexitatea pe parcurs.

Cele mai simple AFD-uri care se pot transforma în expresii regulate au o singură stare. Pentru alfabetul de intrare $\Sigma = \{0, 1\}$ acestea sunt ilustrate în Figura 4-12.



Figura 4-12: Cele mai simple AFD-uri cu o singură stare pentru alfabetul de intrare $\Sigma = \{0, 1\}$.

Aceste automate pot fi convertite direct în expresiile regulate Σ^* și ϕ . Vom trece în continuare la un AFD cu două stări. (Figura 4-13)

Conversia unui AFD într-o expresie regulată

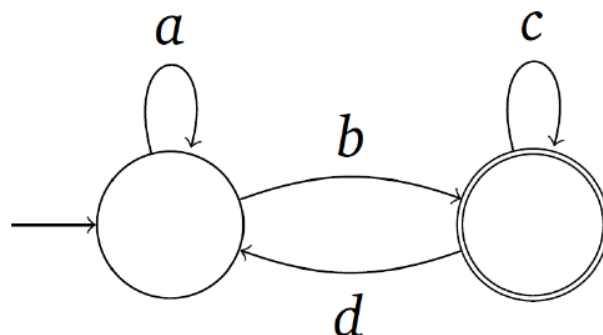


Figura 4-13: Un AFD cu două stări pentru alfabetul de intrare $\Sigma = \{a, b, c, d\}$.

Pentru a construi expresia regulată echivalentă acestui automat, considerăm că un șir de caractere de intrare poate tranzita cele două stări, de la starea inițială la cea acceptoare și invers, de un număr de ori oarecare și apoi poate trece în starea acceptoare și rămâne acolo. Aplicând acest raționament, obținem expresia regulată: $(a \cup bc^*d)^*bc^*$.

Conversia unui AFD într-o expresie regulată

Un alt exemplu de AFD cu două stări este cel din Figura 4-14, pentru care expresia regulată echivalentă este: $(a \cup bc^*d)^*$.

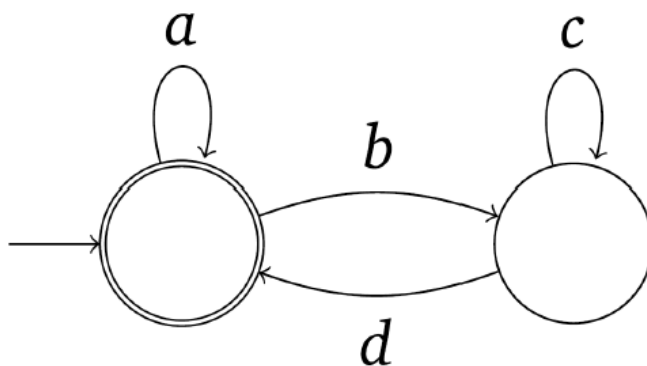


Figura 4-14: Exemplu de AFD corespunzător expresiei regulate $(a \cup bc^*d)^*$.

Conversia unui AFD într-o expresie regulată

Există AFD-uri în care tranzițiile sunt etichetate cu mai mult de un simbol, caz în care putem spune că fiecare astfel de tranziție constituie de fapt mai multe tranziții de un simbol. Pentru a gestiona astfel de cazuri, se notează fiecare astfel de tranziție cu o expresie regulată care reprezintă reuniunea acelor simboluri, ca în exemplul din Figura 4-15.

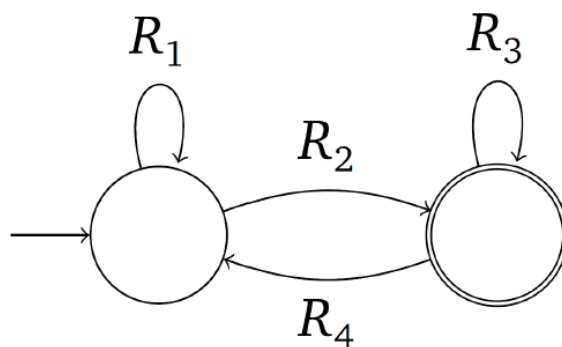


Figura 4-15: Exemplu de AFD cu notarea echivalentă a tranzițiilor.

Dacă o tranziție este etichetată $R = a_1 \cup a_2 \cup \dots \cup a_k$, atunci acea tranziție poate fi executată la citirea oricăruia din simbolurile a_1, a_2, \dots, a_k . Expresia regulată echivalentă unui astfel de automat poate fi obținută aplicând același raționament ca mai sus, de exemplu pentru automatul din cazul nostru aceasta este: $(R_1 \cup (R_2)(R_3)^*(R_4))^*(R_2)(R_3)^*$.

Conversia unui AFD într-o expresie regulată

În cazul automatelor cu trei stări lucrurile se complică, dar în acest caz putem încerca să *eliminăm* starea din mijloc a acestuia. Spre exemplu, în cazul automatului din Figura 4-16, pentru a elimina q_1 trebuie luate în considerare toate modalitățile prin care q_1 este implicată în tranzitarea stărilor automatului.

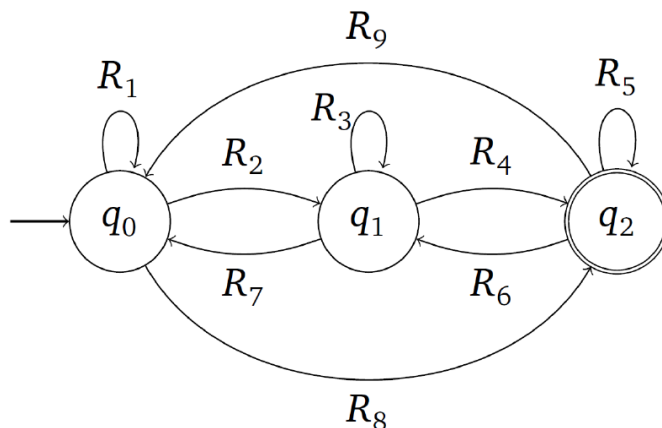


Figura 4-16: Exemplu de AFD cu trei stări.

Starea q_1 este tranzitată în drumul din q_0 spre q_2 folosind șirul de caractere $w \in L((R_2)(R_3)^*(R_4))$, deci pentru a elimina starea q_1 trebuie adăugată aceasta expresie regulată pe tranziția care merge direct de la starea q_0 la starea q_2 (Figura 4-17).

Conversia unui AFD într-o expresie regulată

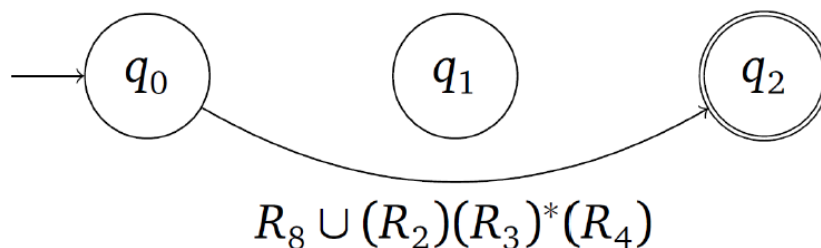


Figura 4-17: Eliminarea stării din mijlocul AFD-ului din Figura 4-16.

Un alt caz care trebuie luat în calcul este utilizarea stării q_1 pentru a tranzita din starea q_0 în starea q_0 . AFD-ul realizează acest lucru pentru un șir de caractere din $L((R_2)(R_3)^*(R_7))$, deci pentru a elimina starea q_1 trebuie adăugată această expresie regulată la tranziția din starea q_0 în starea q_0 (Figura 4-18).

$$R_1 \cup (R_2)(R_3)^*(R_7)$$

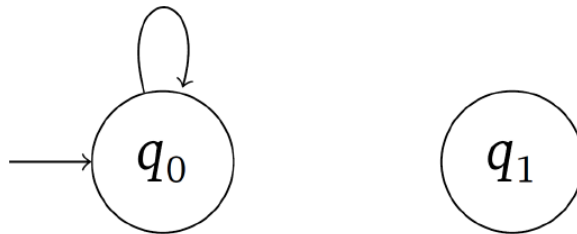


Figura 4-18: Adăugarea expresiei regulate necesare tranzitării din starea q_0 în starea q_0 prin starea q_1 în urma eliminării celei din urmă.

Conversia unui AFD într-o expresie regulată

Eliminarea corectă a stării q_1 din AFD are loc după ce metoda de compensare de mai sus a fost aplicată pentru toate perechile de stări din automat, altele decât q_1 . Automatul rezultat nu mai este un AFD propriu-zis deoarece conține o tranziție etichetată cu o expresie regulată diferită de ϕ sau de o reuniune de simboluri. Cu toate acestea, am obținut un automat cu două stări (Figura 4-19) care poate fi transformat în expresia regulată echivalentă folosind abordarea prezentată mai sus.

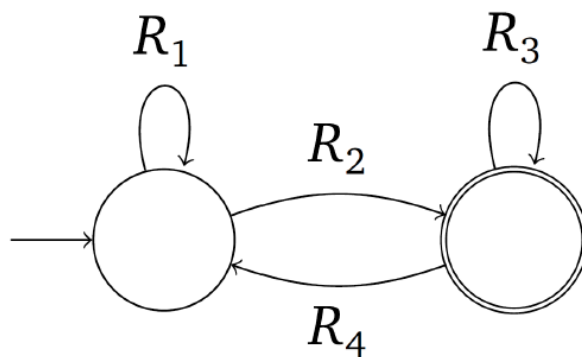


Figura 4-19: AFD-ul cu două stări rezultat.

Astfel, expresia regulată echivalentă este $(R_1 \cup (R_2)(R_3)^*(R_4))^*(R_2)(R_3)^*$.

Conversia unui AFD într-o expresie regulată

Pentru a aplica această metodă de eliminare a stărilor pentru orice automat, trebuie rezolvat cazul particular în care starea ce trebuie eliminată este una acceptoare. Pentru aceasta, AFD-ul se poate modifica astfel încât să conțină o singură stare acceptoare. Acest lucru se realizează adăugând o nouă stare acceptoare, noi ε -tranziții de la vechea stare acceptoare la cea nouă, și în final vechea stare acceptoare își pierde acest statut în cadrul automatului.

Pentru automate mai mari de trei stări, metoda descrisă mai sus se aplică în același fel, procedându-se la eliminarea stărilor una câte una până rămân doar două. Această metodă se poate formaliza ca un algoritm în 4 etape:

1. Dacă automatul nu are stări acceptoare, expresia regulată este ϕ .
2. Se adaugă o nouă stare acceptoare împreună cu noi ε -tranziții de la vechea stare acceptoare la cea nouă. Vechea stare acceptoare devine o stare obișnuită, ne-acceptoare.
3. Se elimină fiecare stare, una câte una, cu excepția stării inițiale și a celei acceptoare. Se va obține un automat de forma celui din Figura 4-20.

Conversia unui AFD într-o expresie regulată

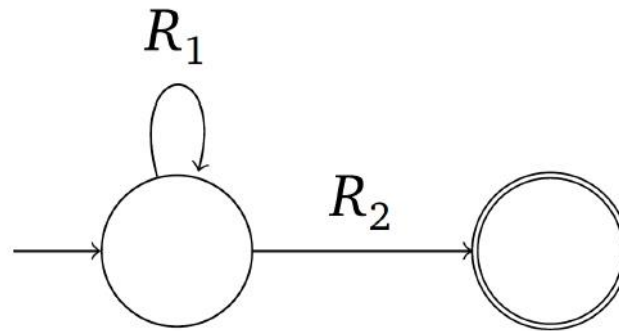


Figura 4-20: AFD-ul obținut în urma etapei 3.

4. Se scrie expresia regulată echivalentă ($R_1^*(R_2)$).

Cursul viitor:

- Analiza lexicală

Întrebări?

