LUCRAREA 4

Expresii regulate

Cuprinsul lucrării:

- Definiţia unei expresii regulate
- Proprietăţile expresiilor regulate. Exemple de expresii regulate
- Conversia unei expresii regulate într-un AFD
- Conversia unui AFD într-o expresie regulată

Unele exemple şi figuri din acest capitol au fost preluate şi adaptate din [8] şi [2].

4.1 Definiţie. Proprietăţi. Exemple.

Conform ierarhiei lui Chomsky, gramaticile se pot clasifica după cum urmează:

- Gramatici de tip 0 (generale sau recursiv enumerabile). Reguli de forma $\alpha \to \beta$ (Nu există restricţii asupra regulilor).
- Gramatici de tip 1 (dependente de context). Reguli de forma $\alpha A \beta \to \alpha \gamma \beta$ unde $A \in N$, $\gamma \neq \varepsilon$, $\alpha, \beta \in (N \cup T)^*$, $S \to \varepsilon$, caz în care S nu apare în dreapta niciunei reguli.

- Gramatici de tip 2 (independente de context). Reguli de forma $A \to \gamma$ unde $A \in N$ şi $\gamma \in (N \cup T)^*$.
- Gramatici de tip 3 (regulate). Reguli de forma $A \to a$ sau $A \to aB$ unde $A, B \in N$ şi $a \in T$.

Expresiile regulate constituie reprezentarea limbajelor de tip 3 prin expresii algebrice. Acestea sunt foarte întâlnite în sistemele de calcul, în special în cazul sistemelor de operare Linux. De exemplu, expresii regulate sunt folosite când se lucrează în consolă (sau terminal) şi se doreşte manipularea fişierelor sau specificarea unor opţiuni pentru comenzi Linux.

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

- $A \to u$ sau $A \to uB$ unde $A, B \in N$ şi $u \in T^\star$.

Exemplu:
$G = (\{D\}, \{0, 1, ..., 9\}, D, P)$
Unde P este mulţimea producţiilor:
$D \to 0D|1D|2D|...|9D$
$D \to 0|1|...|9$

Reprezentarea limbajelor de tip 3 prin expresii algebrice

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

- a, unde a este orice caracter
- ε, cuvântul vid
- ϕ, mulţimea vidă
- $R|S$, alternarea, R sau S (notaţie echivalentă: $R \cup S$)
- RS, concatenarea, R urmat de S (notaţie echivalentă $R \circ S$)
- $R^\star$, 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 $L(R) \cdot L(S)$
- $(R^\star)$ este expresie regulată ce descrie limbajul $L(R)^\star$

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

Extensii ale expresiilor regulate

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

- $R? = \varepsilon|R$ (zero or un singur R)

- $R+ = RR^\star$ (unul sau mai muli R)
- $(R) = R$ (parantezele au rol de grupare)
- $[abc] = a|b|c$ (oricare caracter din paranteze)
- $[a - e] = a|b|...|e$ (interval de caractere)
- $[\hat{}ab] = c|d|...$ (orice mai puin caracterele enumerate)
- $abc =$ şirul de caractere abc
- $\backslash($= caracterul (

Proprietăţi ale expresiilor regulate

- $(p|q)|r = p|(q|r)$ Asociativitatea la reuniunie.
- $(pq)r = p(qr)$ Asociativitatea la concatenare.
- $p|q = q|p$ Comutativitatea la reuniune.
- $p \cdot \varepsilon = \varepsilon \cdot p = p$ Simbolul vid este element neutru la concatenare.
- $p|\phi = p|p = p$ Mulţimea vidă este element neutru la reuniune.
- $\phi \cdot p = p \cdot \phi = \phi$ Concatenarea cu mulţimea vidă.
- $p(q|r) = pq|pr$ Distributivitatea operaţiilor de concatenare şi reuniune.
- $(p|q)r = pr|qr$ Distributivitatea operaţiilor de concatenare şi reuniune.
- $\varepsilon|pp^\star = p^\star$ Proprietăţi ale produsului Kleene.
- $\varepsilon|p^\star p = p^\star$ Proprietăţi ale produsului Kleene.

Aşa cum s-a menţionat mai sus, expresiile regulate sunt folosite intesiv în cadrul sistemelor de operare, a compilatoarelor şi a altor aplicaţii software. În Tabelul 4-1 sunt ilustrate câteva exemple de expresii regulate pentru identificarea anumitor ,,cuvinte“ dintr-un program C.

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

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

1. Expresia regulată $\Sigma^\star$ descrie limbajul tuturor şirurilor de caractere peste alfabetul Σ. Cu alte cuvinte, $L(\Sigma^\star) = \Sigma^\star$.

2. Limbajul tuturor şirurilor de caractere care încep cu 1 este descris de expresia regulată $1\Sigma^\star$.

3. Limbajul tuturor şirurilor de caractere care se termină cu 1 este descris de expresia regulată $\Sigma^\star 1$.

4. Limbajul tuturor şirurilor de caractere de lungime cel puţin 2 care încep şi se termină cu acelaşi simbol este descris de expresia regulată $0\Sigma^\star 0 \cup 1\Sigma^\star 1$.

5. Limbajul şirurilor de caractere care conţin subşirul 001 este descris de expresia regulată $\Sigma^\star 001\Sigma^\star$.

6. Limbajul şirurilor de caractere care conţin un număr par de 1 este descris de expresia regulată $(0^\star 10^\star 1)^\star 0^\star$.

7. Limbajul şirurilor de caractere care conţin un număr de 1 care este multiplu de k este descris de expresia regulată $(0^\star 1)^k)^\star 0^\star$.

4.2 Conversia unei expresii regulate într-un AFD

O expresie regulată poate fi privită ca o structură destinată validării unui şir de simboluri de intrare, după următoarele principii:

- Validarea elementului nul, reprezentat de caracterul ε.

- Validarea unui simbol α aparţinând alfabetului limbajului reprezentat de expresia respectivă.
- Validarea uneia din două expresii regulate r_1 şi r_2, scrisă sub forma $r_1|r_2$.
- Validarea unei secvenţe de două expresii regulate r_1 şi r_2, scrisă sub forma r_1r_2.
- Validarea a zero sau mai multe instanţe ale unei expresii regulate r, scrisă sub forma $r^\star$.

O expresie regulată se poate converti în primă fază într-un automat finit nedeterminist (AFN) prin rezolvarea fiecărui caz din cele de mai sus. Prin definiţie, fiecare automat (AFN sau AFD) au o singură stare iniţială şi una sau mai multe stări finale (acceptoare). Datorită particularităţilor conversiei, AFN-ul rezultat va avea totuşi o singură stare acceptoare.

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

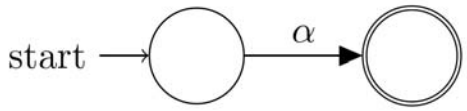

Figura 4-1: Automatul corespunzător expresiei regulate α.

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

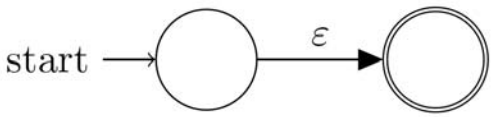

Figura 4-2: Automatul corespunzător expresiei regulate ε.

În ce priveşte cazurile expresiilor regulate compuse din mai multe structuri, procedura de conversie acţionează recursiv. Astfel, rezultatul conversiei unei expresii regulate oarecare r într-un AFN este cel din Figura 4-3. Starea

din partea stângă reprezintă starea iniţială a automatului corespunzător expresiei r, iar starea din partea dreaptă reprezintă stările acceptoare ale aceluiaşi automat.

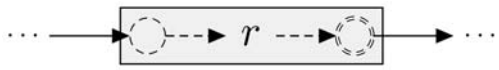

Figura 4-3: Automatul corespunzător expresiei regulate r.

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

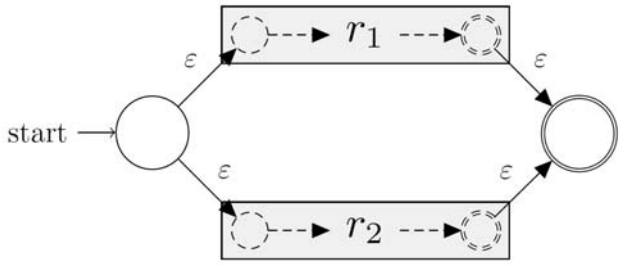

Figura 4-4: AFN-ul corespunzător expresiei $r_1|r_2$.

O secvenţă de două expresii regulate concatenate r_1 şi r_2, scrisă sub forma r_1r_2 se converteşte într-un AFN prin simpla adăugare de tranziţii între stările acceptoare ale AFN-ului lui r_1 şi starea iniţială a AFN-ului lui r_2, ca în Figura 4-5.

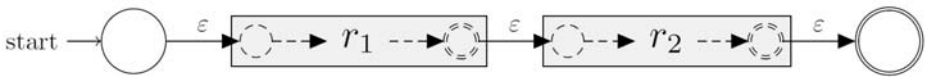

Figura 4-5: AFN-ul corespunzător expresiei r_1r_2.

AFN-ul expresiei $r^\star$ are o ε-tranziţie de la starea sa iniţială la starea iniţială a lui r, şi o ε-tranziţie de la starea acceptoare a lui r la propria stare acceptoare. In plus, există ε-tranziţii de la starea iniţială la cea acceptoare şi invers. Această construcţie permite ca r să fie trecută cu vederea sau repetată de câte ori este necesar. AFN-ul corespunzător expresiei regulate $r^\star$ este ilustrat în Figura 4-6.

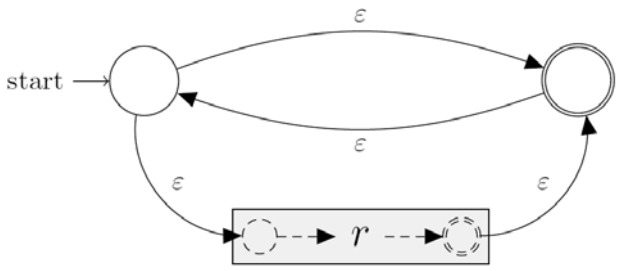

Figura 4-6: AFN-ul corespunzător expresiei $r^\star$.

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

Exemplu

Să se scrie AFD-ul corespunzător expresiei regulate $ba^\star b$.

Rezolvare:

Primul pas îl constituie scrierea automatelor care corespund simbolurilor a şi b (Figura 4-7):

Figura 4-7: Automatele corespunzătoare simbolurilor a şi b.

În continuare se construieşte automatul corespunzător expresiei $a^\star$ (Figura 4-8):

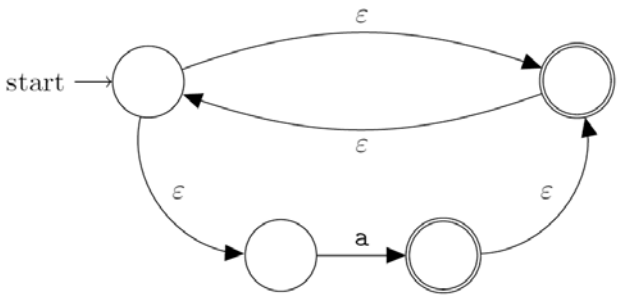

Figura 4-8: Automatul corespunzător expresiei $a^\star$.

În final se ataşează automatul corespunzător expresiei b la fiecare capăt al acestuia. De menţionat este faptul că dispar cele două stări acceptoare şi că ataşarea se realizează introducând două ε-tranziţii (Figura 4-9).

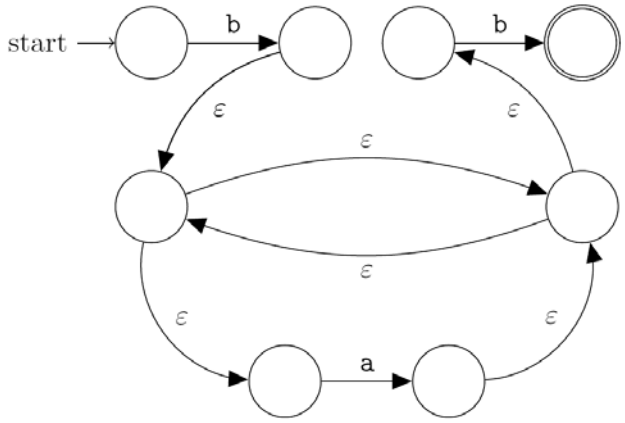

Figura 4-9: AFN-ul obţinut.

Pornind de la acest AFN, se poate obţine relativ uşor AFD-ul echivalent aplicând algoritmul prezentat in secţiunea 3.4. Înainte de aceasta, stările AFN-ului trebuie denumite (Figura 4-10).

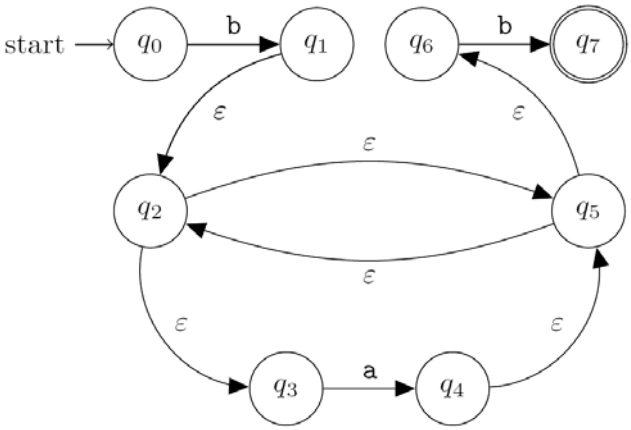

Figura 4-10: AFN-ul din Figura 4-9 după denumirea tuturor stărilor.

În urma transformării se obţine AFD-ul din (Figura 4-11), cu corespondenţa stărilor explicitată în Tabelul 4-2.

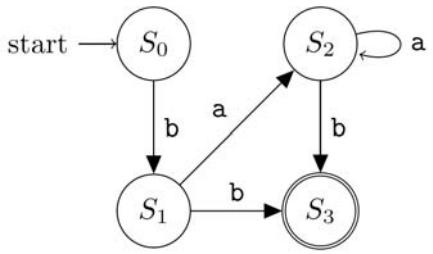

Figura 4-11: AFD-ul echivalent AFN-ului din Figura 4-10.

Tabelul 4-2: Corespondenţa stărilor AFD - AFN

Stare AFD	**Corespondenta AFN**
S_0	$\{q_0\}$
S_1	$\{q_1, q_2, q_3, q_5, q_6\}$
S_2	$\{q_2, q_3, q_4, q_5, q_6\}$
S_3	$\{q_7\}$

4.3 Conversia unui AFD într-o expresie regulată

În practică există cazuri de limbaje pentru care este mai uşor să se construiască AFD-ul direct decât să se scrie expresia regulată corespunzătoare. Daca este necesară şi 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 $\Sigma^\star$ şi ϕ. Vom trece în continuare la un AFD cu două stări. (Figura 4-13)

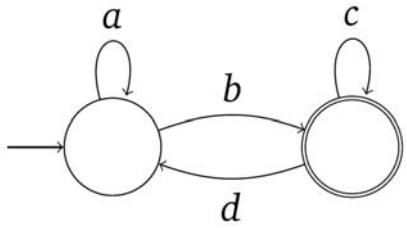

Figura 4-13: Un AFD cu două stări pentru alfabetul de intrare $\Sigma = \{a, b, c, d\}$.

Pentru a construi expresia regulată echivalentă acestui automat, considerăm că un şir de caractere de intrare poate tranzita cele două stări, de la starea iniţială la cea acceptoare şi invers, de un număr de ori oarecare şi apoi poate trece în starea acceptoare şi rămâne acolo. Aplicând acest raţionament, obţinem expresia regulată: $(a \cup bc^\star d)^\star bc^\star$.

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

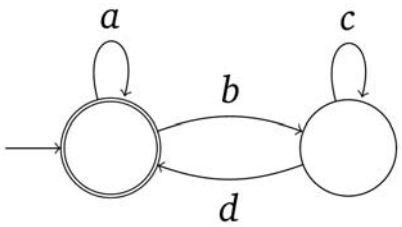

Figura 4-14: Exemplu de AFD corespunzător expresiei regulate $(a \cup bc^\star d)^\star$.

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

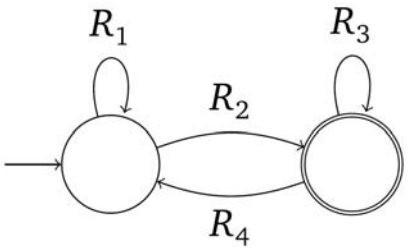

Figura 4-15: Exemplu de AFD cu notarea echivalentă a tranziţiilor.

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

Pot exista şi tranziţii lipsă, caz în care acestea pot fi considerate ca fiind etichetate cu expresia regulată ϕ. De exemplu, dacă tranziţia în buclă la a doua stare a automatului de mai sus ar fi lipsit, am fi putut considera $R_3 = \phi$ şi am fi făcut această înlocuire în expresia regulată obţinută pentru cazul general: $(R_1 \cup (R_2)\phi^\star(R_4))^\star(R_2)\phi^\star = (R_1 \cup (R_2)(R_4))^\star(R_2)$.

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

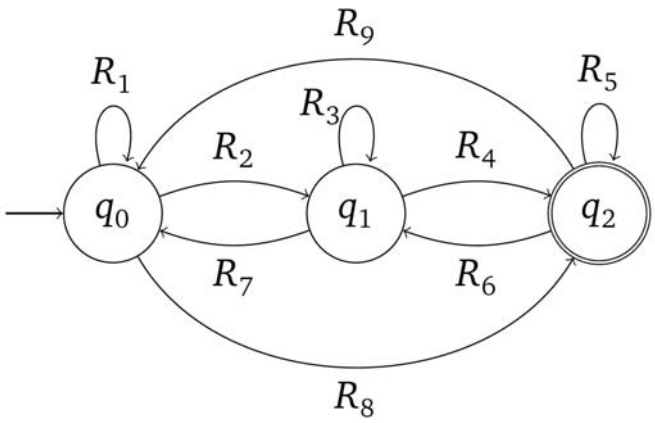

Figura 4-16: Exemplu de AFD cu trei stări.

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

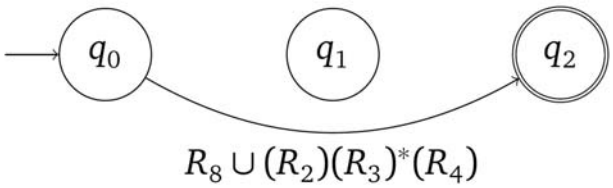

Figura 4-17: Eliminărea stării din mijlocul AFD-ului din Figura 4-16.

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 şir de caractere din $L((R_2)(R_3)^\star(R_7))$, deci pentru a elimina starea q_1 trebuie adăugată această expresie regulată la tranziţia din starea q_0 în starea q_0 (Figura 4-18).

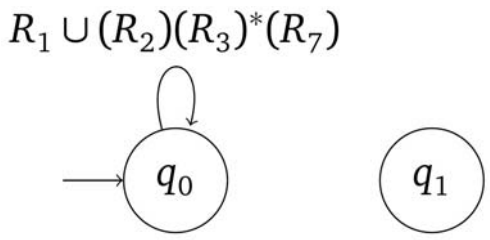

Figura 4-18: Adăugarea expresiei regulate necesare tranzitării din starea q_0 în starea q_0 prin starea q_1 în urma eliminării celei din urmă.

Eliminarea corectă a stării q_1 din AFD are loc după ce metodad 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 conţine o tranziţie etichetată cu o expresie regulată diferită de ϕ sau de o reuniune de simboluri. Cu toate acestea, am obţinut un automat cu două stări (Figura 4-19) care poate fi transformat în expresia regulată echivalentă folosind abordarea prezentată mai sus.

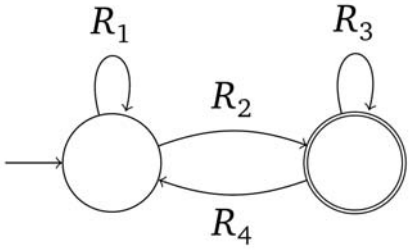

Figura 4-19: AFD-ul cu două stari rezultat.

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

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 conţină o singură stare acceptoare. Acest lucru se realizează adăugând o nouă stare acceptoare, noi ε-tranziţii de la vechea stare acceptoare la cea nouă, şi în final vechea stare acceptoare işi pierde acest statut în cadrul automatului.

Pentru automate mai mari de trei stări, metoda descrisă mai sus se aplică în acelaşi 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-tranziţii de la vechea stare acceptoare la cea nouă. Vechea stare acceptoare devine o stare obişnuită, ne-acceptoare.

3. Se elimină fiecare stare, una câte una, cu excepţia stării iniţiale şi a celei acceptoare. Se va obţine un automat de forma celui din Figura 4-20.

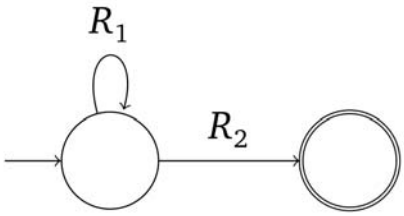

Figura 4-20: AFD-ul obţinut în urma etapei 3.

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

Exemplu

Să se scrie expresia regulată echivalentă pentru AFN-ul din Figura 4-21.

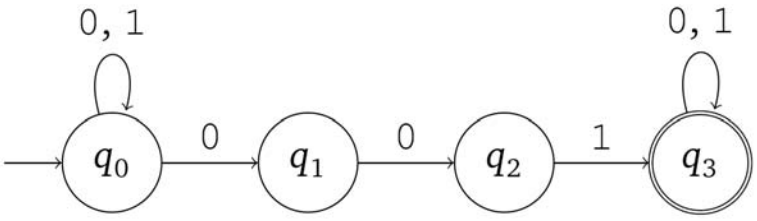

Figura 4-21: Exemplu de AFN pentru care se cere determinarea expresiei regulate echivalente.

Rezolvare:

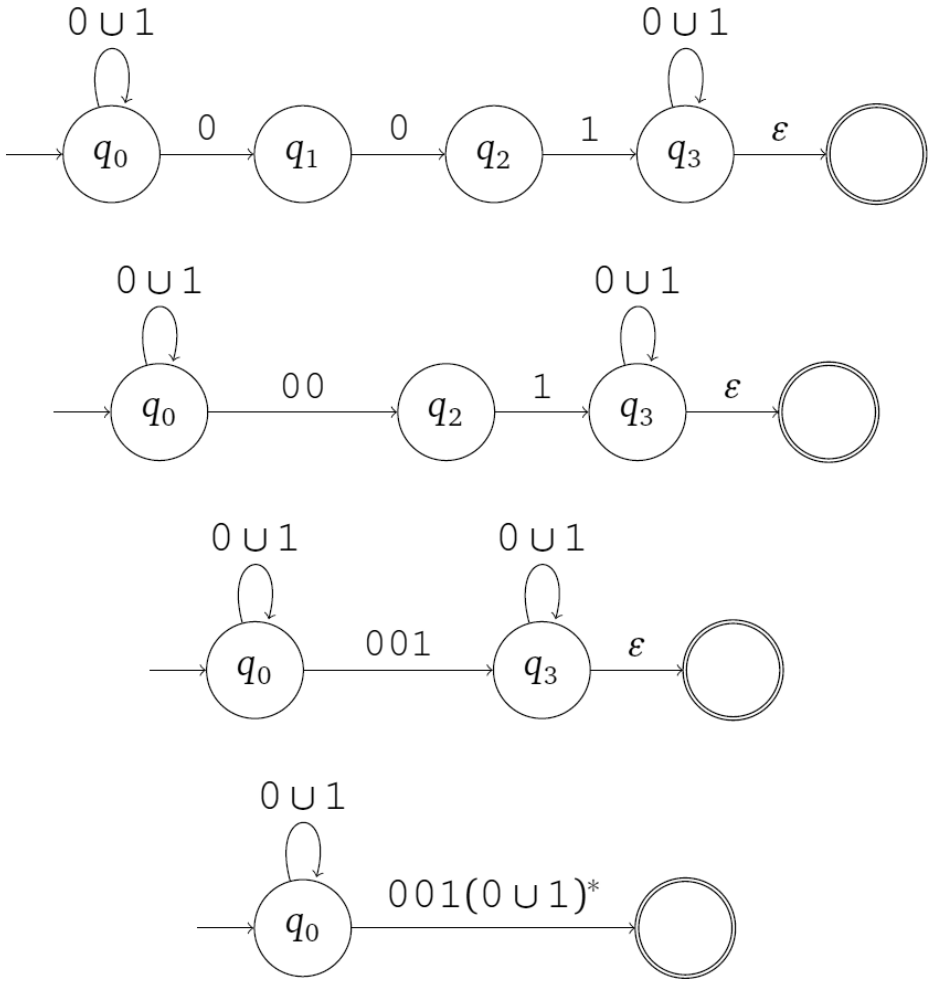

Figura 4-22: Obţinerea expresiei regulate echivalente pentru AFN-ul din Figura 4-21 prin aplicarea algoritmului în 4 etape prezentat mai sus.

4.4 Probleme propuse

1. Folosind algoritmul prezentat în secţiunea 4.2, să se convertească următoarele expresii regulate în AFN-uri. Pentru toate cazurile, alfabetul este $\Sigma = \{0, 1\}$.

   (a) $0^\star \cup 1^\star$

   (b) $0\Sigma^\star 0$

   (c) $0^\star 10^\star$

(d) $(11)^\star$

2. Folosind algoritmul prezentat în sectiunea 4.3, să se convertească următoarele AFN-uri în expresii regulate:

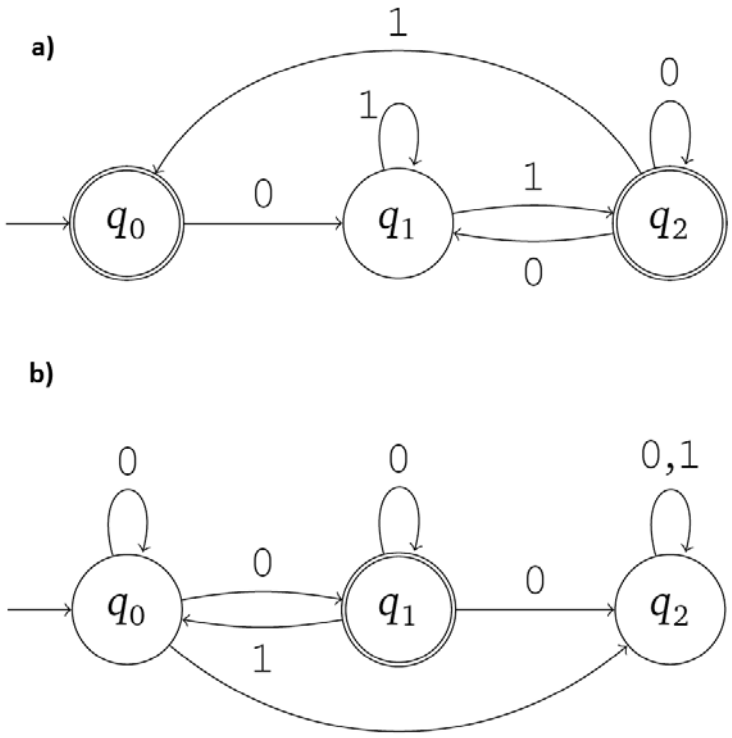

Figura 4-23: AFN-uri ce trebuie convertite în expresii regulate.