**Ministerul Educației al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică.**

**Raport**

Lucrarea de laborator nr.2

Disciplina: Limbaje Formale și Automate

Tema: Automate finite nedeterministe şi automate finite deterministe.

Varianta: 5.

**Efectuat**: st.gr. TI-207 Bunescu Gabriel

**Verificat**: lect. sup. Duca Ludmila

Chișinău 2021

Cuprins:

[I.Scopul lucrării: 2](#_Toc84092233)

[1. Datele initiale: 3](#_Toc84092234)

[2. gramatica regulată 4](#_Toc84092235)

[3. Este sau nu automatul dat determinist 4](#_Toc84092236)

[4. Construiţi automatul finit determinist echivalent (PRIN AMBELE METODE). 4](#_Toc84092237)

[5. Gramatica regulată echivalentă cu automatul finit determinist. 6](#_Toc84092238)

[6. Construirea șirul peste vocabularulm ∑ care nu va fi acceptat de către automatul finit determinist. 6](#_Toc84092239)

[7. Pentru AFD= (Q, ∑ , δ, q0, F) construiți 5 șiruri acceptate de către automatul finit determinist. 6](#_Toc84092240)

[8. Pentru fiecare șir x scrieți secvența de configurații pentru acceptarea șirului, adică (q0, x) |---- (qi1, x1) |---- (qi2, x2) |---- …. |---- (qif, e), unde qf ∈ F. 7](#_Toc84092241)

[9. Aplicăm leama de pompare. 7](#_Toc84092242)

[10. Concluzie: 8](#_Toc84092243)

# I.Scopul lucrării:

1. Construirea unei gramatici regulate;
2. De construit 14 producții și 5 cuvinte cu arborii lor de derivare pe baza gramatici de tip 3;
3. De construit 5 producții de tip 2 și 5 producții de tip 1;

**Mersul lucrării:**

1. Este dat automatul finit AF = (Q, ∑ , δ, q0, F). Reprezentați automatul finit sub formă de graf.
2. Construiți gramatica regulată echivalentă cu automatul dat.
3. Este sau nu automatul dat determinist? De ce?
4. Dacvă automatul este nedeterminist, construiți automatul finit determinist echivalent. Reprezentați AFD în formă de graf.
5. Construiți gramatica regulată echivalentă cu automatul finit determinist.
6. Inventați un șir peste vocabularulm ∑ care nu va fi acceptat de către automatul finit determinist. Arătați acest lucru scriind (Secvențele) de configurațuii respectivă.
7. Pentru AFD= (Q, ∑ , δ, q0, F) construiți 5 șiruri acceptate de către automatul finit determinist. Lungimea șirurilor să nu fie mai mică decât n+2, unde n este numărul de stări din Q.
8. Pentru fiecare șir x scrieți secvența de configurații pentru acceptarea șirului, adică (q0, x) |---- (qi1, x1) |---- (qi2, x2) |---- …. |---- (qif, e), unde qf ∈ F.
9. Pentru toate cele 5 șiruri obținute construiți descompunerea x=uvw aplicând lema de pompare.

# Datele initiale:

**Varianta 5**AF=(Q, ∑, δ, q0, F), Q = {q0, q1, q2 , q3}, ∑ = { a, b, c}, F = { q3}.  
δ (q0, a) = {q1},  
δ (q1, a) ={q2},  
δ (q2, c) = {q2, q3},  
δ (q2, b) ={q2},  
δ (q3, a) ={q3}

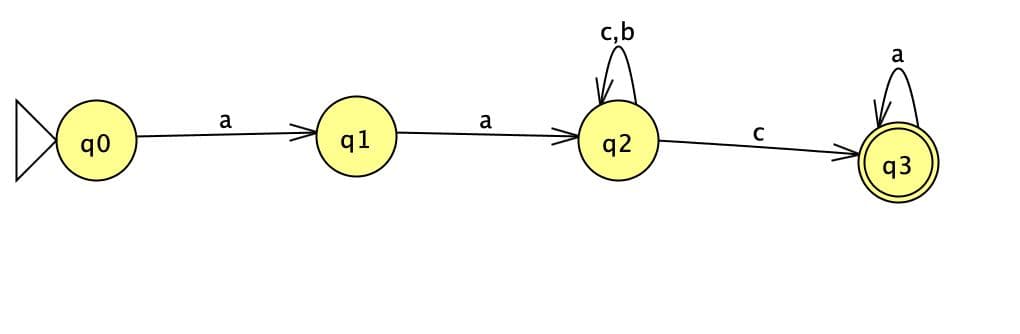


Fig.1.Graful automatului finit

**Metoda tabel**

Tabel 1.1 - Reprezentarea automatului finit prin tabel

|  |  |  |  |
| --- | --- | --- | --- |
|  | **a** | **b** | **c** |
| **q0** | q1 |  |  |
| **q1** | q2 |  |  |
| **q2** |  | q2 | q2q3 |
| **q3** | q3 |  |  |

# gramatica regulată

G=(VN, VT, P, S) VN={ q0, q1, q2 , q3} VT={ a, b, c }

* + - 1. q0→aq1
      2. q1→aq2
      3. q2→bq2
      4. q2→cq2
      5. q2→cq3
      6. q3→aq3
      7. q2→c
      8. q3→a

# Este sau nu automatul dat determinist

Automatul dat este nedeterminist, deoarece din starea q2, prin elementul din alfabetul de intrare „c”, se poate ajunge în două stari: q2 și q3.

# Construiţi automatul finit determinist echivalent (PRIN AMBELE METODE).

Metoda analitica:

AFD = {Q’, Σ, σ, q0’, F’}

Q’ = Ø, q0’ = [q0]

Q’ = {[q0]}

ẟ(q0, a)= q1

ẟ(q0, b)= Ø

ẟ(q0, c)= Ø

Q’ = {[q0], [q1]}

ẟ(q1, a)= q2

ẟ(q1, b)= Ø

ẟ(q1, c)= Ø

Q’ = {[q0], [q1], [q2]}

ẟ(q2, a)= Ø

ẟ(q2, b)= q2

ẟ(q2, c)= q2q3

Q’ = {[q0], [q1], [q2], [q2q3]}

ẟ(q2q3, a)= q3

ẟ(q2q3, b)= q2

ẟ(q2q3, c)= q2q3

Q’ = {[q0], [q1], [q2], [q2q3], [q3]}

ẟ(q3, a)= q3

ẟ(q3, b)= Ø

ẟ(q3, c)= Ø

F’=Q’ ∩F

Q’ = {[q0], [q1], [q2], [q2q3], [q3]}

F’={[ q0], [q1], [q2], [q2q3],[ q3] ∩ [ q3]}={ [q2q3],[ q3] }

**Metoda tabel**

Tabel 4.1– Metoda tabelara de construire a automatului finit determinist

|  |  |  |  |
| --- | --- | --- | --- |
|  | **a** | **b** | **c** |
| **q0** | q1 |  |  |
| **q1** | q2 |  |  |
| **q2** |  | q2 | q2q3 |
| **q2q3** | q3 | q2 | q2q3 |
| **q3** | q3 |  |  |

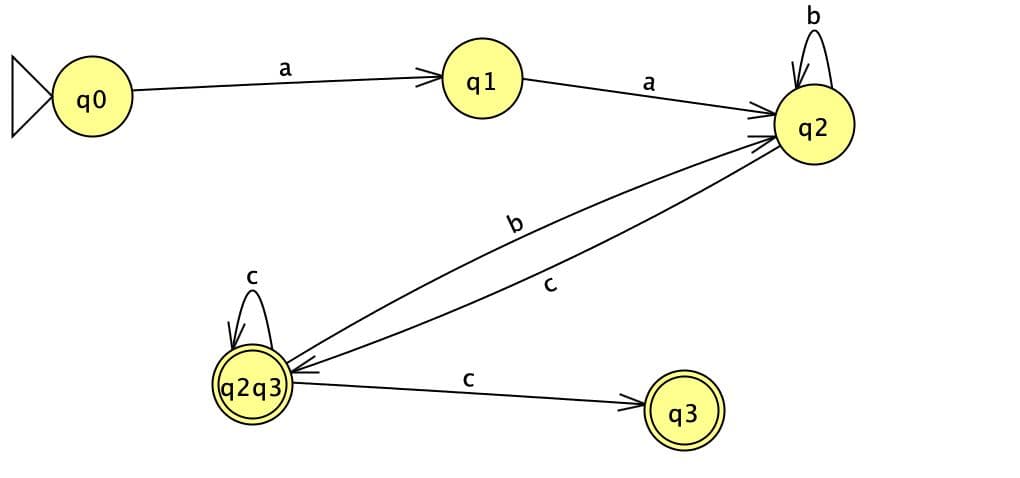


Fig.4.1. Graful automatului finit determinist

# Gramatica regulată echivalentă cu automatul finit determinist.

G=(VN, VT, P, S) VN={ q0, q1, q2 ,q2q3, q3} VT={ a, b, c }

* + - 1. q0→aq1
      2. q1→aq2
      3. q2→bq2
      4. q2→cq2q3
      5. q2q3→bq2
      6. q2q3→cq2q3
      7. q2q3→cq3
      8. q3→aq3
      9. q2→c
      10. q2q3→c
      11. q2q3→c
      12. q3→a

# Construirea șirul peste vocabularulm ∑ care nu va fi acceptat de către automatul finit determinist.

1. x = aabbba

(q0, aabbba) |— (q1, abbba) |— (q2, bbba) |— (q2, bba) |— (q2,ba) |— (q2, a) - impas

1. x = aaccba

(q0, aaccba) |— (q1, accba) |— (q2, ccbab) |— (q2q3, cbab) |— (q3,bab) - impas

# Pentru AFD= (Q, ∑ , δ, q0, F) construiți 5 șiruri acceptate de către automatul finit determinist.

1. x = aabbcca

(q0, aabbcca) |— (q1, abbcca) |— (q2, bbcca) |— (q2, bcca) |— (q2,cca) |— (q2q3, ca) |—

|— (q3, a) |— (q3, ε)

1. x = aacbccc

(q0, aacbccc) |— (q1, acbccc) |— (q2, cbccc) |— (q2q3, bccc) |— (q2, ccc) |— (q2q3, cc) |—

|— (q2q3, c) |— (q3, ε)

1. x = aacbbcbc

(q0, aacbbcbc) |— (q1, acbbcbc) |— (q2, cbbcbc) |— (q2q3, bbcbc) |— (q2,bcbc) |—

|— (q2,cbc) |— (q2q3, bc) |— (q2,c) |— (q2q3, ε)

1. x = aaccaaa

(q0, aaccaaa) |— (q1, accaaa) |— (q2, ccaaa) |— (q2q3, caaa) |— (q3,aaa) |— (q3,aa) |—

|— (q3, a) |— (q3, ε)

1. x = aabbcca

(q0, aabbcca) |— (q1, abbcca) |— (q2, bbcca) |— (q2, bcca) |— (q2, cca) |— (q2 q3, ca) |— |— (q3, a) |— (q3, ε )

# Pentru fiecare șir x scrieți secvența de configurații pentru acceptarea șirului, adică (q0, x) |---- (qi1, x1) |---- (qi2, x2) |---- …. |---- (qif, e), unde qf ∈ F.

1. x = aabbcca

(q0, aabbcca) |— (q1, abbcca) |— (q2, bbcca) |— (q2, bcca) |— (q2,cca) |— (q2q3, ca) |—

|— (q3, a) |— (q3, ε)

1. x = aacbccc

(q0, aacbccc) |— (q1, acbccc) |— (q2, cbccc) |— (q2q3, bccc) |— (q2, ccc) |— (q2q3, cc) |—

|— (q2q3, c) |— (q3, ε)

1. x = aacbbcbc

(q0, aacbbcbc) |— (q1, acbbcbc) |— (q2, cbbcbc) |— (q2q3, bbcbc) |— (q2,bcbc) |—

|— (q2,cbc) |— (q2q3, bc) |— (q2,c) |— (q2q3, ε)

1. x = aaccaaa

(q0, aaccaaa) |— (q1, accaaa) |— (q2, ccaaa) |— (q2q3, caaa) |— (q3,aaa) |— (q3,aa) |—

|— (q3, a) |— (q3, ε)

1. x = aabbcca

(q0, aabbcca) |— (q1, abbcca) |— (q2, bbcca) |— (q2, bcca) |— (q2, cca) |— (q2 q3, ca) |— |— (q3, a) |— (q3, ε )

# Aplicăm leama de pompare.

1. x = aabbcca

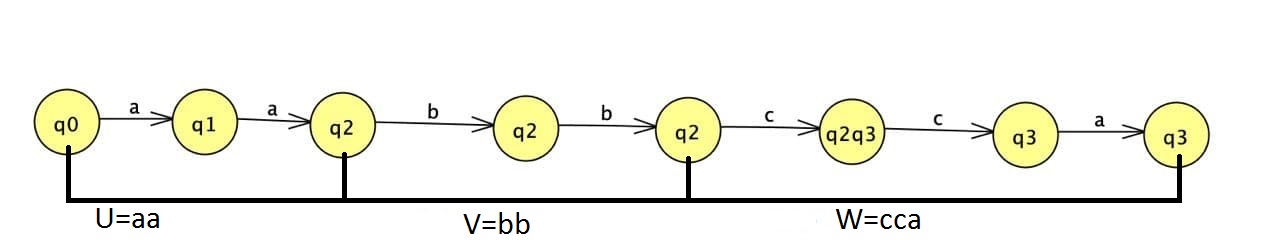


Fig.9.1 leama de pompare x1

1. x = aacbccc

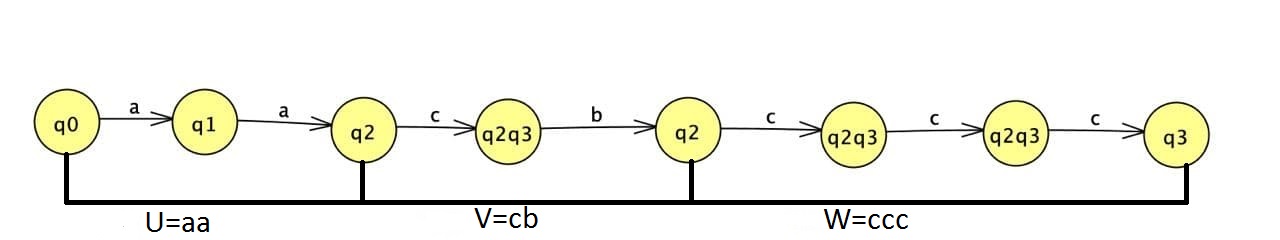


Fig.9.1 leama de pompare x2

1. x = aacbbcbc

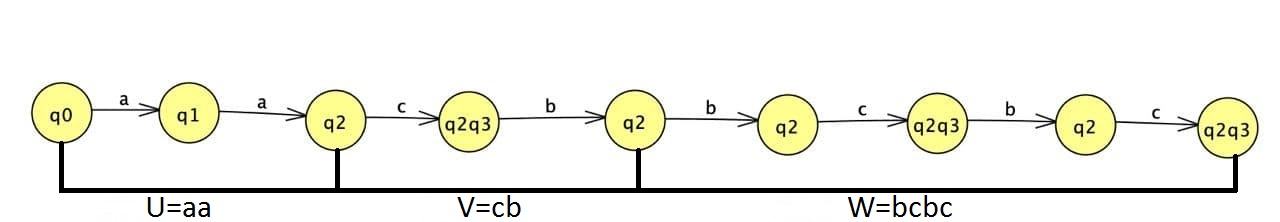


Fig.9.1 leama de pompare x3

1. x = aaccaaa

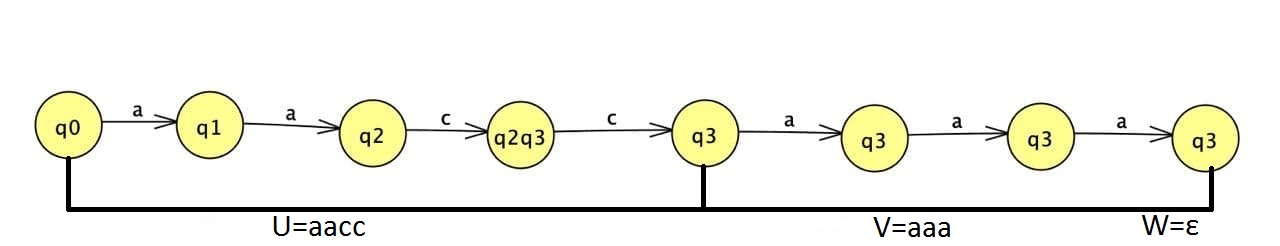


Fig.9.1 leama de pompare x4

1. x = aabbcca

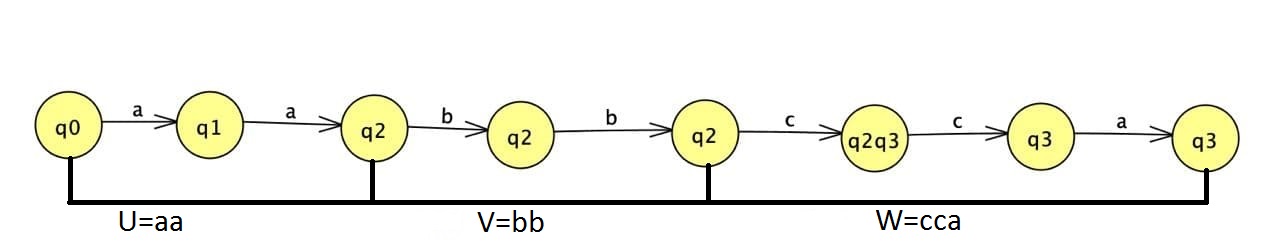


Fig.9.1 leama de pompare x5

# Concluzie:

Pe parcursul elaborării acestei lucrări de laborator eu am studiat despre automatul finit, reprezentarea lui sub formă de graf și tabel. Am construit gramatica regulată echivalentă pentru automatul finit nedeterminist și determinist, am alcătuit cuvinte neacceptate și acceptate pentru automatul finit determinist , am construit secventa de configurare pentru acceptarea lor și am construit descompunerea x=uvw aplicând lema de pompare.