Ministerul Educației și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Departamentul Ingineria Software și Automatică

**Laborator nr.2**

Disciplina: Limbaje formale si automate finite

Tema: Automate Finite Nedeterministe (AFND) şi Automate Finite Deterministe (AFD)

A efectuat: st. Popa Cătălin, TI-211

A verificat: asist. univ. Duca Ludmila

Chișinău – 2023

**Scopul lucrarii :**

**1.** Creaţi un AFND unde:

Q = |4|

S = |3|

12 funcţii de tranziţie

F =|1|

Reprezentaţi AFND prin toate metodele.

**2.** Construiţi Gramatica Regulată pentru AFND.

**3.** Construiţi Expresia regulată pentru AFND.

**4.** Minimizaţi AFND cu descrierea fiecărui pas prin funcţii de tranziţii.

**5.** Creaţi 2 cuvinte acceptate de AFND arătând secvenţele de configuraţii.

**6.** Creaţi 2 cuvinte neacceptate de AFND arătând secvenţele de configuraţii

**7.** Transformaţi AFND în AFD prin ambele metode. Desenaţi graful AFD.

**8.** Demonstraţi prin secvenţe de configuraţii că cuvintele create la punctul **5** sunt acceptate şi de AFD.

**9.** Demostraţi că L(AFND) este echivalent cu L(AFD).

**1. Creaţi un AFND.**

AF = (Q, ∑, ∂, q0, F),

Q = {q0, q1, q3, q2},

∑ = {0, 1, 2},

q0 = {q0},

F = {q2},

∂(q0, 0) = {q0, q2}

∂(q0, 1) = {q1, q3}

∂(q1, 1) = {q2}

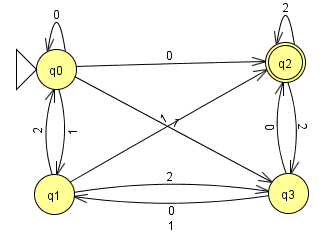
∂(q1, 2) = {q0, q3}

∂(q2, 2) = {q2, q3}

∂(q3, 0) = {q1, q2}

∂(q3, 1) = {q1 }

Reprezentarea automatului finit nedeterminist prin graf este afișată în figura 1.



**Figura 1** – Reprezentarea AFND prin graf.

Reprezentarea automatul finit nedeterminist prin tablel este reprezentă în tabelul 1.

**Tabel 1** - Reprezentarea AFND prin tabel.

|  |  |  |  |
| --- | --- | --- | --- |
| Q S | 0 | 1 | 2 |
| q0 | q0, q2 | q1, q3 | - |
| q1 | - | q2 | q0,q3 |
| q2 | - | - | q2, q3 |
| q3 | q1, q2 | q1 | - |

**2. Construiţi Gramatica Regulată pentru AFND.**

G = { VN, VT, P, S},

VN = { q0, q1, q3, q2},

VT = {0, 1, 2},

S = { q0 },

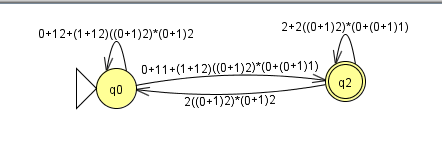
P = {1. q0 → 0q0

1. q0 → 0q2
2. q0 → 1q1
3. q0 → 1q3
4. q1 → 1q2
5. q1 → 2q0
6. q1 → 2q3
7. q2 → 2q2
8. q2 → 2q3
9. q3 → 0q1
10. q3 → 0q2
11. q3 → 1q1
12. q0 → 0
13. q1 → 1
14. q3 → 0
15. q2 → 2}

**3.** **Construiţi Expresia regulată pentru AFND.**

Expresia regulată pentru automatul finit nedeterminist este reprezentată în figura 2.

((0+12+(1+12)((0+1)2)\*(0+1)2)\*(0+11+(1+12)((0+1)2)\*(0+(0+1)1))(2+2((0+1)2)\*(0+(0+1)1))\*2((0+1)2)\*(0+1)2)\*(0+12+(1+12)((0+1)2)\*(0+1)2)\*(0+11+(1+12)((0+1)2)\*(0+(0+1)1))(2+2((0+1)2)\*(0+(0+1)1))\*



**Figura 2**. Expresia regulata.

**4.** **Minimizaţi AFND cu descrierea fiecărui pas prin funcţii de tranziţii.**

Deoarece pe AFND nu îl pot minimiza, mai întâi al convertim în AFD. După care cu ajutorul la JFLAP, al minimizăm, iar rezultatul este în figura 2.1 de mai jos.

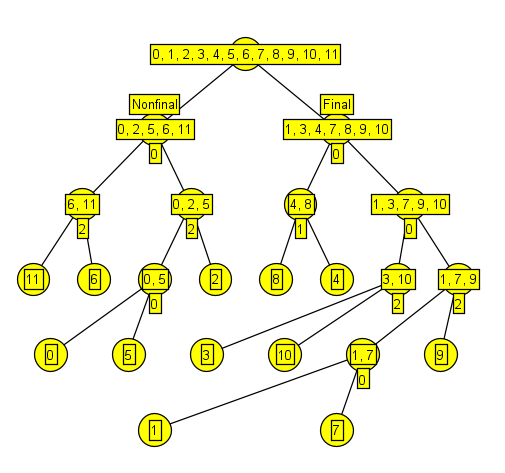
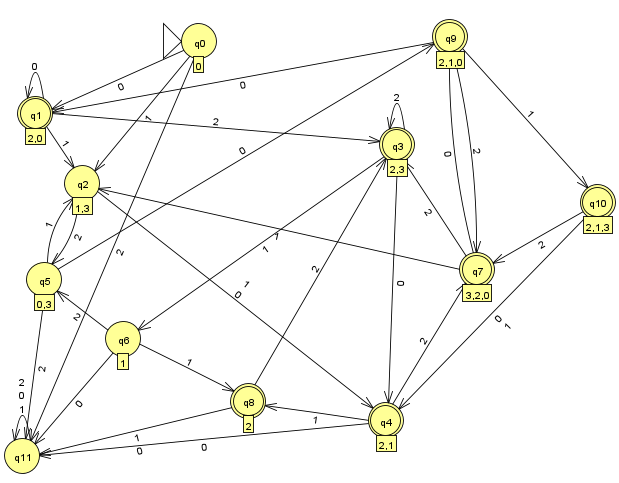


Figura 2.1 – Minimizarea AFD

Pentru această metodă, vom avea nevoie de graful AFD, prezent în figura 2.2, după care însemnăm toate tranzițiile.



**Figura 2.2** – Graful AFD.

**Tabelul 1.2** – Tabelul de minimizare.

|  |
| --- |
| Q11 |
| \* | Q10\* |
| \* | \* | Q9\* |
| \* | \* | \* | Q8\* |
| \* | \* | \* | \* | Q7\* |
| \* | \* | \* | \* | \* | Q6 |
| \* | \* | \* | \* | \* | \* | Q5 |
| \* | \* | \* | \* | \* | \* | \* | Q4\* |
| \* | \* | \* | \* | \* | \* | \* | \* | Q3\* |
| \* | \* | \* | \* | \* | \* | \* | \* | \* | Q2 |
| \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | Q1\* |
| \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | Q0 |

{(q0, q2), 0} → (q1, q4)\* {(q2, q6), 0} → (q4, q11)\*

{(q0, q2), 1} → (q2, q4)\* {(q2, q6), 1} → (q4, q8)\*

{(q0, q2), 2} → (q11, q5) {(q2, q6), 2} → (q5)

{(q0, q5), 0} → (q1, q9)\* {(q2, q11), 0} → (q4, q11)\*

{(q0, q5), 1} → (q2) {(q2, q11), 1} → (q4, q11)\*

{(q0, q5), 2} → (q11) {(q2, q11), 2} → (q4, q11)\*

{(q0, q6), 0} → (q1, q11)\* {(q5, q6), 0} → (q9, q11)\*

{(q0, q6), 1} → (q2, q8)\* {(q5, q6), 1} → (q2, q8)\*

{(q0, q6), 2} → (q11, q5) {(q5, q6), 2} → (q11, q5)

{(q0, q11), 0} → (q1, q11)\* {(q5, q11), 0} → (q9, q11)\*

{(q0, q11), 1} → (q2, q11) {(q5, q11), 1} → (q2, q11)

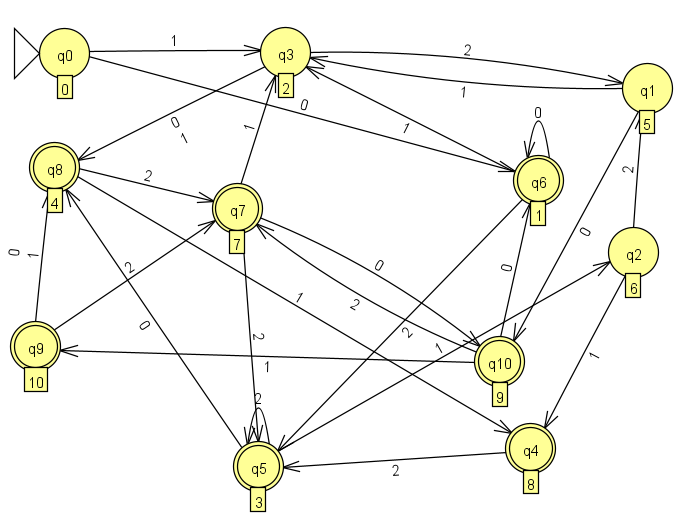
{(q0, q11), 2} → (q11) {(q5, q11), 2} → (q11)

{(q2, q5), 0} → (q4, q9)\* {(q6, q11), 0} → (q11)

{(q2, q5),1} → (q4, q2)\* {(q6, q11), 1} → (q8, q11)\*

{(q2, q5), 2} → (q5, q11) {(q6, q11), 2} → (q5, q11)

În figura 2.3, am reprezentat graful după minimizare.

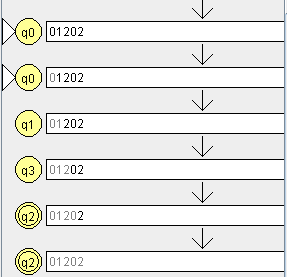


**Figura 2.3** – Graf minimizat.

**5.** **Creaţi 2 cuvinte acceptate de AFND arătând secvenţele de configuraţii.**

1. x = 01202, |x| = 5,

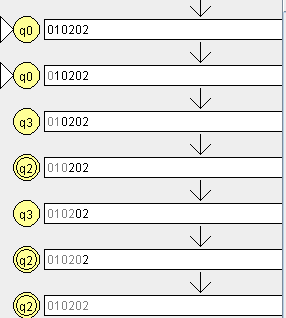
(q0, 01202) ├ (q0, 1202) ├ (q1, 202) ├ (q3, 02) ├ (q2, 2)├ (q2, Ɛ)



**Figura 3** – Primul cuvant acceptat.

2. x = 010202, |x| = 6,

(q0, 010202) ├ (q0, 10202) ├ (q3, 0202) ├ (q2, 202) ├ (q3, 02) ├ (q2, 2) ├ (q2, Ɛ)



**Figura 4** – Al doilea cuvant acceptat

**6.** **Creaţi 2 cuvinte neacceptate de AFND arătând secvenţele de configuraţii**

1. x = 0120221, |x| = 5,

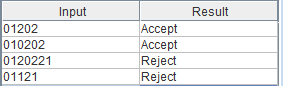
(q0, 0120221) ├ (q0, 120221) ├ (q1, 20221) ├ (q3, 0221) ├ (q2, 221) ├ (q2, 21) ├ (q3, 1)├ (q1, Ɛ)

Nu este acceptat

2. x = 01121, |x| = 5,

(q0, 01121) ├ (q0, 1121) ├ (q3, 121) ├ (q1, 21) ├ (q0, 1) ├ (q3, Ɛ) - Nu este acceptat

Cuvintele au fost verificate în JFLAP, acestea fiind reprezentate în figura 5.



**Figura 5** – Verificare cuvinte.

**7.** **Transformaţi AFND în AFD prin ambele metode. Desenaţi graful AFD.**

Metoda tabel

**Tabelul 2** – Transformarea prin matrice de tranziție a AFND în AFD.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **0** | **1** | **2** |
| **q0** | q0 q2 | q0 q2 | - |
| **q0 q2** | q0 q2 | q1 q3 | q2 q3 |
| **q1 q3** | q1 q2 | q1 q2 | q0 q3 |
| **q2 q3** | q1 q2 | q1 | q2 q3 |
| **q1 q2** | - | q2 | q0 q2 q3 |
| **q0 q3** | q0 q2 q1 | q1 q3 | - |
| **q1** | - | q2 | q0 q3 |
| **q2** | - | - | q2 q3 |
| **q0 q2 q3** | q0 q2 q1 | q1 q3 | q2 q3 |
| **q0 q1 q2** | q0 q2 | q1 q3 q2 | q0 q3 q2 |
| **q1 q2 q3** | q1 q2 | q1 q2 | q2 q3 |

Metoda Analitica

Q’ = { [q0], [q0 q2], [q1 q3], [q2 q3], [q1 q2], [q0 q3], [q1], [q2], [q0 q2 q3], [q0 q1 q2], [q1 q2 q3]}

∂([q0], 0) = q0 q2 ∂([q1], 0) = ᴓ

∂([q0], 1) = q0 q2 ∂([q1], 1) = q2

∂([q0], 2) = ᴓ ∂([q1], 2) = q0 q3

∂([q0 q2], 0) = q0 q2 ∂([q2], 0) = ᴓ

∂([q0 q2], 1) = q1 q3 ∂([q2], 1) = ᴓ

∂([q0 q2], 2) = q2 q3 ∂([q2], 2) = q2 q3

∂([q1 q3], 0) = q1 q2 ∂([q0 q2 q3], 0) = q0 q2 q1

∂([q1 q3], 1) = q1 q2 ∂([q0 q2 q3], 1) = q1 q3

∂([q1 q3], 2) = q0 q3 ∂([q0 q2 q3], 2) = q2 q3

∂([q2 q3], 0) = q1 q2 ∂([q0 q1 q2], 0) = q0 q2

∂([q2 q3], 1) = q1 ∂([q0 q1 q2], 1) = q1 q3 q2

∂([q2 q3], 2) = q2 q3 ∂([q0 q1 q2], 2) = q0 q3 q2

∂([q1 q2], 0) = ᴓ ∂([q1 q2 q3], 0) = q1 q2

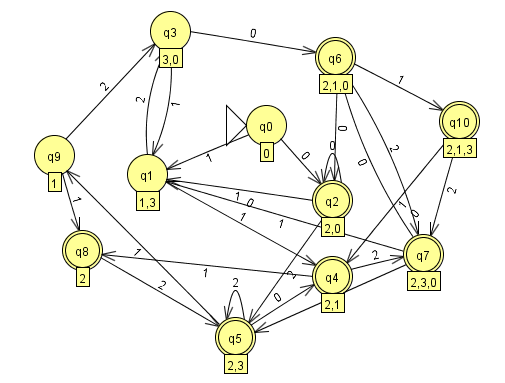
∂([q1 q2], 1) = q2 ∂([q1 q2 q3], 1) = q1 q2

∂([q1 q2], 2) = q0 q2 q3 ∂([q1 q2 q3], 2) = q2 q3

∂([q0 q3], 0) = q0 q2 q1

∂([q0 q3], 1) = q1 q3

∂([q0 q3], 2) = ᴓ

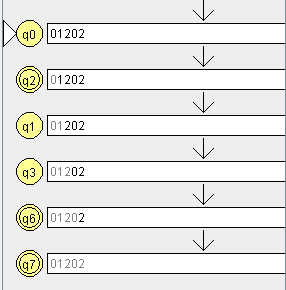


**Figura 6** – Reprezentarea AFD prin graf.

**8.** **Demonstraţi prin secvenţe de configuraţii că cuvintele create la punctul 5 sunt acceptate şi de AFD.**

1. x = 01202, |x| = 5

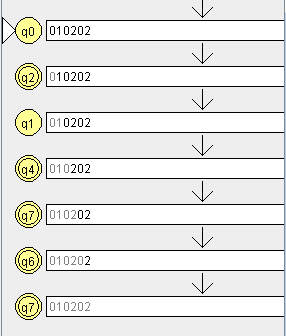
(q0, 01202) ├ (q2, 1202) ├ (q1, 202) ├ (q3, 02) ├ (q6, 2)├ (q7, Ɛ)



**Figura 7** – Primul cuvant acceptat.

2. x = 010202, |x| = 6

(q0, 010202) ├ (q2, 10202) ├ (q1, 0202) ├ (q4, 202) ├ (q7, 02) ├ (q6, 2) ├ (q7, Ɛ)



**Figura 8** – Al doilea cuvant acceptat.

**9.** **Demostraţi că L(AFND) este echivalent cu L(AFD).**

Demonstrarea se va efectua prin crearea un cuvânt nou care va fi verificat prin AFND, apoi prin AFD fiind afișate secvențele de configurații.

x = 010202, |x| = 6,

Prin AFND:

(q0, 010202) ├ (q0, 10202) ├ (q3, 0202) ├ (q2, 202) ├ (q3, 02) ├ (q2, 2) ├ (q2, Ɛ) – acceptat

Prin AFD:

(q0, 010202) ├ (q2, 10202) ├ (q1, 0202) ├ (q4, 202) ├ (q7, 02) ├ (q6, 2) ├ (q7, Ɛ) - acceptat

**Concluzie:**

În concluzie, lucrarea realizată a fost una complexă și a inclus mai multe etape importante în analiza și transformarea unui automat finit nedeterminist (AFND) într-un automat finit determinist (AFD).

În prima etapă, s-a creat un AFND cu 4 stări, 3 simboluri de intrare și 12 funcții de tranziție, unde o singură stare era finală. Acest AFND a fost reprezentat prin toate metodele posibile, cum ar fi tabelul de tranziție, diagrama de stare și simbol și graful. În etapele următoare, s-au construit o gramatică regulată și o expresie regulată pentru AFND și s-a minimizat AFND-ul prin descrierea fiecărui pas prin funcțiile de tranziție. De asemenea, s-au creat două cuvinte acceptate și două cuvinte neacceptate de AFND, arătând secvențele de configurații corespunzătoare. În continuare, s-a realizat transformarea AFND-ului în AFD prin cele două metode, respectiv metoda tabelei de tranziție și metoda submulțimilor. Graful AFD-ului a fost desenat și s-a demonstrat prin secvențe de configurații că cuvintele create la punctul 5 sunt acceptate și de AFD. În final, s-a demonstrat că limbajul acceptat de AFND este echivalent cu cel acceptat de AFD. Această demostrare a implicat verificarea faptului că orice cuvânt acceptat de unul dintre automate este și acceptat de celălalt și că orice cuvânt respins de unul dintre automate este și respins de celălalt.

În ansamblu, lucrarea a presupus o întelegere solidă a teoriei limbajelor formale și automatelor și o aplicare practică a acestora, folosind diverse instrumente și tehnici pentru analiza și transformarea unui AFND într-un AFD echivalent.