

Laborator 4

I Problema: Să presupunem că avem un automat finit determinist cu două stări: "A" și "B", și două simboluri de intrare: "0" și "1". Regulile de tranziție sunt definite astfel:

1. Dacă automatul se află în starea "A" și primește simbolul "0", trece în starea "B".
2. Dacă automatul se află în starea "A" și primește simbolul "1", rămâne în starea "A".
3. Dacă automatul se află în starea "B" și primește simbolul "0", trece în starea "A".
4. Dacă automatul se află în starea "B" și primește simbolul "1", rămâne în starea "B".

Cerință: Să se determine starea finală în care se va afla automatul după ce primește următoarele secvențe de intrare: "010", "110", "1001".

Soluție:

Pentru secvența "010":

- Automatul începe în starea "A".
- După ce primește "0", trece în starea "B".
- După ce primește "1", rămâne în starea "B".
- După ce primește "0", trece înapoi în starea "A".
- Starea finală este "A".

Pentru secvența "110":

- Automatul începe în starea "A".
- După ce primește "1", rămâne în starea "A".
- După ce primește "1", rămâne în starea "A".
- După ce primește "0", trece în starea "B".
- Starea finală este "B".

Pentru secvența "1001":

- Automatul începe în starea "A".
- După ce primește "1", rămâne în starea "A".
- După ce primește "0", trece în starea "B".
- După ce primește "0", trece înapoi în starea "A".
- După ce primește "1", rămâne în starea "A".
- Starea finală este "A".

a) Implementați/simulați automatul într-un limbajul de programare (la alegere).

b) Creați diagrama și tabelul de tranziție pentru automat.

II Problema:

Să presupunem că avem o parcare automată cu un număr finit de locuri de parcare, numerotate de la 1 la N. Fiecare loc de parcare poate fi fie liber, fie ocupat. Un șofer poate să încerce să parcheze o mașină sau să plece cu mașina din parcare.

Cerințe:

1. Implementați un automat finit determinist care să reprezinte starea locurilor de parcare și să gestioneze cererile șoferilor pentru a parca sau a pleca cu mașina din parcare.
 2. Starea inițială a parcării este definită de numărul de locuri libere și ocupate.
 3. Automatul trebuie să poată face tranziții între stări în funcție de cererile șoferilor. De exemplu, atunci când un șofer dorește să parcheze, automatul va trebui să actualizeze starea locurilor de parcare și să schimbe starea locului de parcare corespunzător.
 4. Interfața automatului trebuie să permită șoferilor să verifice starea locurilor de parcare și să facă cereri de parcare sau de plecare.
 5. Automatul trebuie să fie capabil să gestioneze situații de eroare sau cereri invalide (de exemplu, cereri de parcare într-o parcare deja plină sau cereri de plecare dintr-o parcare goală).
- a) Implementați/simulați automatul într-un limbajul de programare (la alegere).
- b) Creați diagrama și tabelul de tranziție pentru automat.

III Pentru a crea un NFA (automat finit nedeterminist) cu o gramatică asociată, vom construi un NFA care să recunoască limbajul generat de următoarea gramatică regulată simplă:

- a) Implementați/simulați automatul într-un limbajul de programare (la alegere) incluzând gramatica.
- b) Creați diagrama și tabelul de tranziție pentru automat.

Gramatica:

$S \rightarrow aA \mid bB$

$A \rightarrow a \mid bA \mid \epsilon$

$B \rightarrow aB \mid \epsilon$

Această gramatică generează șiruri care încep cu 'a' și se termină cu 'b', sau șiruri care încep cu 'b' și se termină cu 'a'. De exemplu, "ab", "aab", "bb", "bba", etc.

Vom construi NFA-ul astfel:

Stările:

- Starea inițială: S0

- Starea de acceptare: S1

Alfabetul:

- $\Sigma = \{a, b\}$

Funcția de tranziție (δ):

	a	b
S0 S2	S3	
S1 -	-	
S2 S1	S2	
S3 S3	S1	

Explicație:

- Starea inițială (S0) are tranziții către starea (S2) pentru intrarea 'a' și către starea (S3) pentru intrarea 'b'.

- Starea (S1) este starea de acceptare și nu are nicio tranziție.

- Starea (S2) este starea care recunoaște că am început cu 'a'. Are o tranziție către starea de acceptare (S1) pentru intrarea 'b' și o tranziție către propria sa starea pentru intrarea 'a'.

- Starea (S3) este starea care recunoaște că am început cu 'b'. Are o tranziție către starea de acceptare (S1) pentru intrarea 'a' și o tranziție către propria sa starea pentru intrarea 'b'.

Acest NFA va recunoaște șirurile care sunt generate de gramatica dată, respectiv cele care încep cu 'a' și se termină cu 'b', sau cele care încep cu 'b' și se termină cu 'a'.

IV Problema:

a) Implementați/simulați automatul într-un limbajul de programare (la alegere).

b) Creați diagrama și tabelul de tranziție pentru automat.

c) Echivalența NFA în DFA (matematic)

Equivalence of NFA and DFA:

Practice problem-1:

Construct an equivalent DFA for the given NFA.

