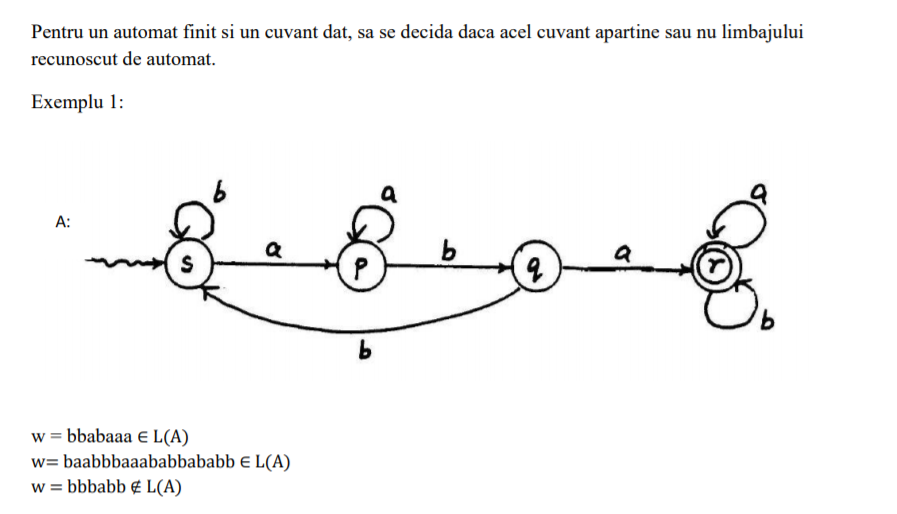
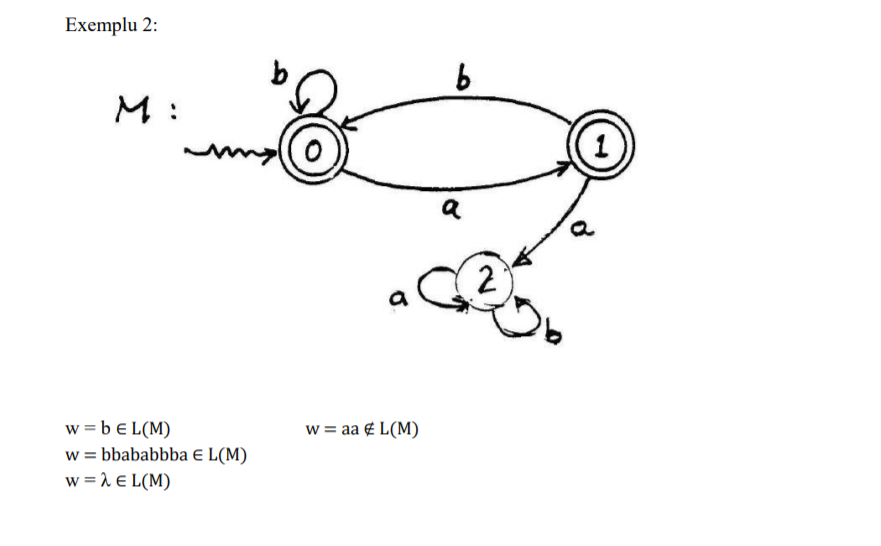
**Proiect 1 LFA – Implementare NFA**



Sursa main.cpp prezintă următoarea structură:

* Funcția AFN
* Funcția solve
* Funcția main
* O structura declarata global

Funcția AFN:

* Primește ca parametrii în urmatoarea ordine

- cuvântul de verificat

- graful propriu zis pe care îl voi parcurge

- starea curentă

- vectorul cu stări finale

- Declar o variabilă care mă va ajuta să determin validitatea cuvântului dorit

- Verific dacă am terminat de parcurs cuvântul într-o stare finală și dacă da, atunci acesta este valid, deci voi returna 1

- Verific dacă am terminat de parcurs cuvântul într-o stare nefinală, și dacă da, atunci acesta nu este valid, deci voi returna 0

- Pentru fiecare nod din graf voi lua pe rând toate muchiile pentru a verifica dacă găsesc una corectă, care are drept cheie caracterul căutat

- Dacă am găsit o tranzitie pentru care litera căutată este gasită atunci “valid” va primii rezultatul apelului recursiv al funcției AFN care va verifica următoarea literă din cuvânt

- Dacă valid este 1 atunci înseamnă că am găsit un drum care acceptă cuvîntul, respectiv dacă nu am găsit niciun drum bun după ce am încercat toate posibilitătile returnez 0

Funcția SOLVE:

* Citește în următoarea ordine: cuvântul de verificat, starea inițială, numărul de stări finale
* Citește starile finale și le adaugă în vectorul “final”
* Inițializează fiecare nod din vectoul tranziții cu NULL
* Preia fiecare tranziție. Astfel x reprezintă starea din care plec, y reprezintă starea în care ajung și c reprezintă caracterul din cadrul respectivei tranziții. Se creează un nod nou în graf pentru fiecare tranziție și este adăugat în lista cu noduri care pleacă din respectiva stare ( nodul este pus la începutul listei de fiecare dată)
* La final se apelează funcția AFN și este testat cuvântul

Funcția main:

* Apelează funcția SOLVE