

### Tema 3 - Automate finite deterministe

#### Cerințe:

Să se implementeze un AFD (automat finit determinist) astfel: se citesc din fișier elementele componente ale automatului **Q,  $\Sigma$ ,  $\delta$ ,  $q_0$ , F**. Se citește de la tastatură un cuvânt și se verifică, dacă este acceptat de către automat.

#### Barem

1. Se cere crearea unei clase AFD (alta decât clasa principală). În funcția principală main se declară un obiect de tip AFD. Apoi se citesc de la tastatură cuvinte (do-while) și pentru fiecare se afișează dacă este acceptat sau nu.

Membrii clasei vor fi: Stari, Sigma, Delta, StareInit, Finale

Printre metodele clasei obligatoriu:

(1) afisare () - afișarea frumoasă a automatului **0.5p**

(2) **accepta(cuvant)** - verifică dacă cuvântul dat ca parametru este acceptat de către automat și afișează: "accepta" - dacă este cuvânt acceptat, "neacceptat" - dacă nu este acceptat, "blocaj" - dacă automatul se blochează pe parcurs. **5p**

(3) **verifică()** – verifică dacă automatul este ok (dacă starea inițială / stările finale se găsesc în mulțimea de stări, dacă tranzițiile conțin doar elemente ale automatului) **1p**

Construcția corectă a clasei **1p**

Citirea elementelor automatului se face din fișier. Funcția de citire poate fi membră a clasei sau nu.

2. Citirea din fișier a elementelor AFD-ului - **1p**

3. Posibilitatea de a verifica mai multe cuvinte, fără a reporni algoritmul – **0.5p**

#### Un punct din oficiu

Un algoritm funcțional care doar citește și afișează elementele automatului - se punctează cu nota 3.