

## Tema 4 - Automat finit nedeterminist (AFN)

Să se implementeze un AFN astfel: se citesc din fișier elementele componente ale AFN-ului  $Q$ ,  $\Sigma$ ,  $\delta$ ,  $q_0$ ,  $F$ . Se testează succesiv cuvinte și pentru fiecare cuvânt automatul verifică dacă e acceptat sau nu.

### Barem

1. Se cere crearea unei clase AFN (alta decât clasa principală – aviz celor cu Java, C#). În funcția principală main se declară un obiect de tip AFN, apoi se citesc pe rând cuvinte de la tastatură (într-un `do – while`) și pentru fiecare se verifică dacă e sau nu acceptat.

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

Printre metodele clasei trebuie să existe:

- (1) `afisare()` - afișarea frumoasă a automatului (tabel sau grafic) **0.5p**
- (2) `verifica()` – verificări legate de corectitudinea automatului (de exemplu: starea inițială și stările finale să existe în mulțimea de stări, Delta să fie definită pentru stări existente și caractere existente) **1p**
- (3) `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**

Construcția corectă a clasei (membrii publici/privati):- **1p**

2. Citirea elementelor automatului se poate face printr-o funcție membră sau printr-o funcție externă, care să preia ca parametru un automat. Citirea elementelor AFN-ului se face dintr-un fișier - **1p**

3. Posibilitatea de a verifica mai multe cuvinte, fără a reporni algoritmul (`do-while`) – **0.5p**

### Un punct din oficiu

Un algoritm funcțional care doar citește și afișează elementele automatului - nota 3