

# Temă 2: Conversia unei expresii regulate în automat finit determinist (DFA)

Disciplina: Limbaje Formale și Automate

## Obiectiv

Implementați un program care primește o expresie regulată și construiește un automat finit determinist (DFA) echivalent. Nu este permisă utilizarea bibliotecilor externe pentru manipularea automatelor sau a expresiilor regulate — totul trebuie construit de la zero.

## Cerințe

- Parser pentru expresii regulate, inclusiv operatori: repetare odată sau mai multe ori (+), alternare (|), repetare de zero sau mai multe ori (\*), prezență opțională (?).
- Opțional: Transformare expresie regulată în forma postfixată (folosind algoritmul Shunting-Yard).
- Construcția unui NFA folosind algoritmul lui Thompson.
- Conversia NFA în DFA folosind construcția prin mulțimi (subset construction).
- Opțional: minimizarea DFA rezultat.

## Subpuncte (Sarcini)

1. Scrieți un convertor de expresii regulate în notație postfixată.
2. Implementați construcția NFA folosind postfixul și algoritmul lui Thompson.
3. Implementați conversia NFA  $\rightarrow$  DFA prin subset construction.
4. Implementați un simulat de DFA (verifică dacă un cuvânt este acceptat de DFA).
5. Testați programul folosind un fișier JSON cu cel puțin 20 de cazuri.

## Format fișier de test (JSON)

```
[
  {
    "name": "R1",
    "regex": "a*b",
    "test_strings": [
      { "input": "b", "expected": true },
      { "input": "aaab", "expected": true },
      { "input": "ab", "expected": true },
      { "input": "a", "expected": false }
    ]
  }
]
```

## Livrabile și notare

- (1p) se acordă dacă folosiți GitHub/GitLab/BitBucket și vă încărcați tema pe un repository (SVC).
- (8p) Cod sursă în Python sau alt limbaj de programare ales de voi (fără librării externe de automatizare).
- (1p) Un document `README.md` care explică:
  - Structura proiectului
  - Cum se rulează codul
  - Deciziile luate în implementare

## Deadline

Tema se predă în maxim **14 zile** de la data primirii. Pentru grupa 151 termenul limita setat este de 21 aprilie 2025.

## Sugestii

- Testați incremental fiecare pas: postfix  $\rightarrow$  NFA, NFA  $\rightarrow$  DFA etc.
- Desenați automatul intermediar pentru înțelegere.
- Pentru debugging puteți printa tranzațiile sau scrie un vizualizator simplu (i.e. Graphviz).