

# Limbaje formale și automate

Seriile 13 și 15

Săptămâna 1, 24 februarie 2025

Andrei Păun

# Cuprinsul cursului:

## 24 februarie 2025

- Să ne cunoaștem
- Generalități despre curs
- Reguli de comportament
- Generalități despre LFA
- Altele ...

# Să ne cunoaștem

- Cine predă? Andrei Păun
- [apaun@fmi.unibuc.ro](mailto:apaun@fmi.unibuc.ro)
- [andreipaun@gmail.com](mailto:andreipaun@gmail.com)

# Generalități despre curs

- Cursul: Luni 8-10 și 10-12

Fizic și online prin Zoom

dacă după 15 min nu se conectează nici un student  
în Zoom cel mai probabil închid sesiunea online

- Laborator: OBLIGATORIU
- Seminar: din două în două săptămâni

# Generalități despre curs

- Curs de “teorie”
- Oferă o bază de pornire pentru alte cursuri
- prima întâlnire cu limbajele formale
- în majoritate ne ocupăm de automate
- ... și gramatici

# Motivație

- Modelarea unui limbaj infinit (mulțime infinită de cuvinte) cu calculatorul
- De ce?
- Slujbe bune: Google, UIPath, Microsoft
- Corelații cu Algoritmi și structuri de date
- ChatGPT4

- Idee de bază în curs: dorim să definim exact/corect/matematic mulțimi de cuvinte în memoria calculatorului
- Memoria e finita deci și definiția trebuie să fie finită!!!
- Dacă mulțimea e finită: o listă
- $M = \{1, 2, 5, 6, 8, 9\}$
- Dacă mulțimea e infinită: ??
- $M = \{2k | k \text{ număr natural}\}$

- Pentru definirea unei mulțimi infinite  $A$  putem avea o definire mult mai “dificilă” decât mulțimea  $B$  chiar dacă  $A$  inclus în  $B$
- Sau în alte cuvinte:  $B$  e mulțime “mai mare” decât  $A$  și totuși de multe ori ne trebuie mai mult efort să definim mulțimea  $A$  decât efortul pentru a defini mulțimea  $B$
- Ex:  $A$ =numere prime,  $B=\mathbb{N}$



# Reguli și sugestii pentru curs

- Web-cam pornit, întrebările le răspund cei care nu au camera pornită
- Ești întrebat și nu răspunzi: waiting room
- Cu cât mai multe întrebări cu atât mai bine
- Moodle
- E-mail

# Important

- Prezența la curs/seminar: nu e obligatorie
- Nota la examen de multe ori corelată cu prezența
- Laboratoarele: obligatorii!

# Organizatorice

- Examenul
  - **19 iunie 2025 la ora 9:00** (nu e încă sigur)
  - **Examenul se susține fizic în facultate**
  - după definativarea orarului definitivăm și programarea examenului
  - POO: 12 iunie 2025 la ora 9:00 (nu e încă sigur)

Împreună toate seriile de Informatică (13, 14, 15)

# Programa cursului

1. Automate finite deterministe și automate finite nedeterministe
2. Automate finite cu lambda-mișcări, proprietăți de închidere (reuniune, concatenare, stelare)
3. Echivalența automatelor finite, proprietăți de închidere
4. Automatul deterministic minimal, Expresii regulate, probleme de decizie
5. Echivalența expresiilor regulate cu automatele, lema de pompare
6. Gramatici; gramatici regulate, reuniune, concatenare, stelare de gramatici
7. Transformările dintre gramatici automate finite și expresii regulate
8. Gramaticile independente de context, ierarhia Chomski, forma normală
9. proprietăți de închidere pentru gramatici independente de context, teorema  $uvwxxy$ , arbori de derivare
10. Automate push-down, echivalența modurilor de acceptare, echivalența gramatici independente de context cu automate push-down
11. Proprietăți de închidere și probleme de decizie
12. Mașini Turing, gramatici dependente de context, automate liniar mărginite
13. Automatele cover deterministe, minimizare, proprietăți
14. Exemple și recapitulare

# Bibliografie

1. J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison -Wesley, 1979.
2. A. Aho, R. Sethi, J. Ullman, Compilers, Principles, Techniques and Tools, Addison Wesley Pub., 1986
3. M.D. Davis, E.J. Weyuker, Computability, Complexity and Languages, Academic Press 1984.
4. A. Salomaa, G. Rozenberg (eds.), Handbook of Formal Languages, 3 vol., Springer Verlag, 1997.

# Regulament de desfășurare și notare

- Disciplina **Limbaje formale și automate** pentru anul I Informatică este organizată în seminar și laborator, fiecare dintre aceste activități având alocată o oră pe săptămână, precum și curs cu 2 ore pe săptămână.
- Disciplina este programată în semestrul II, având o durată de desfășurare de 14 săptămâni.
- Materia este de nivel elementar mediu
- Limbajul de programare folosit la laborator este la alegere C/C++/Java/Python/...
- Programa disciplinei este împărțită în 14 cursuri.
- Evaluarea studenților se face cumulativ prin:
  - Nota de laborator
  - Nota de seminar
  - Test scris
- Nota de trecere la examen este obligatorie pentru promovarea acestei discipline.

# Regulament de desfășurare și notare (2)

- Consultațiile de laborator se desfășoară pe baza întrebărilor studenților.
- Conținutul lucrărilor practice va urmări materia predată la curs. Lucrările practice se realizează individual. Notarea fiecărei lucrări practice se va face cu note de la 1 la 10.
- Predarea lucrărilor practice se face conform indicațiilor primite de la tutorele de laborator (poate fi Git, email cat și pe serverul MOODLE), înainte de termenele limită de predare, indicate de tutorele de laborator pentru fiecare temă în parte. După expirarea termenelor respective, lucrarea practică se mai poate trimite prin email pentru o perioadă de grație de 2 zile (48 de ore). Pentru fiecare zi parțială de întârziere se vor scădea 2 puncte din nota atribuită pe lucrare. După expirarea termenului de grație, lucrarea nu va mai fi acceptată și va fi notată cu 1.
-

# Regulament de desfășurare și notare (3)

- Testul scris se va susține în sesiunea de examene. **Studentii nu pot promova la acest curs decât dacă obțin cel puțin nota 5 la testul scris.**



# Notarea la curs

- nota finală a fiecărui student se calculează după cum urmează:
  - La laborator 15% din nota finală
  - La seminar se vor obține cel mult 0.5 puncte din nota finală (max 20% din fiecare grupă)
  - Prin prezența/participarea la curs se pot obține cel mult 0.5 puncte din nota finală pentru primii 20% clasati la scorul total de Kahoot-uri
  - Examenul din sesiune are o pondere de 85% din nota finală

# Kahoot

- Se va defini un nume unic de forma 133ionescu (unde Ionescu este numele de familie și 133 este grupa)
- mai mulți studenți cu același nume în grupa respectivă se trece la prenume/inițiala prenume:

131lucaGEORGE și 132lucaMIHAI

133popescuMIHAI și 133popescuMARTA

134serbanLUCA și 134serbanSTELIAN

151popaBOGDAN și 151popaSTEFAN

151soltanCRISTIAN și 151soltanMARIAN

152avadaneiRARES și 152avadaneiRADU

- Dacă sunt mai mulți studenți cu același nume în grupa respectivă se trece la prenume/inițiala prenume:

131lucaGEORGE și 132lucaMIHAI

133popescuMIHAI și 133popescuMARTA

134serbanLUCA și 134serbanSTELIAN

151popaBOGDAN și 151popaSTEFAN

151soltanCRISTIAN și 151soltanMARIAN

152avadaneiRARES și 152avadaneiRADU

- Nume greșit la Kahoot
- Ex: 153Danut
- Popescu153
- Danut Popescu
- Danut\_P
- Etc.
- Mă contactați dar pierdeți 20% din scorul de Kahoot în ziua respectivă

# Limbajele formale

- ce este un limbaj
- ce este un limbaj formal
- privire de ansamblu pentru curs:
  - acceptoare
  - generatoare
  - calculabilitate

# Privire de ansamblu

- DFA, NFA
- RE
- gramatici, CFG
- PDA
- TM

# Deterministic finite automata (DFA)

- Modelează calculatoarele cu memorie limitată
  - este un acceptor de limbaje
- Idee de bază
  - Tine minte **starea** curentă
  - **Evenimentele** pot să ne schimbe dintr-o stare în alta
- Astăzi învățăm să
  - descriem formal DFA-urile
  - interpretăm DFA-urile

# Exemplu

- Modelarea unei mingii într-o camera (fără frecare)
- Mișcările: stânga, dreapta, sau deloc
  - Trei stări: stânga, dreapta stop
  - Start în starea stop
- Starea se schimbă sub influența următoarelor condiții
  - Mingea lovește un zid (își schimbă direcția)
  - Paleta o lovește în stânga (mingea merge în stânga)
  - Paleta o lovește în dreapta (mingea merge în dreapta)
  - Mana oprește mingea (o punem în starea stop)







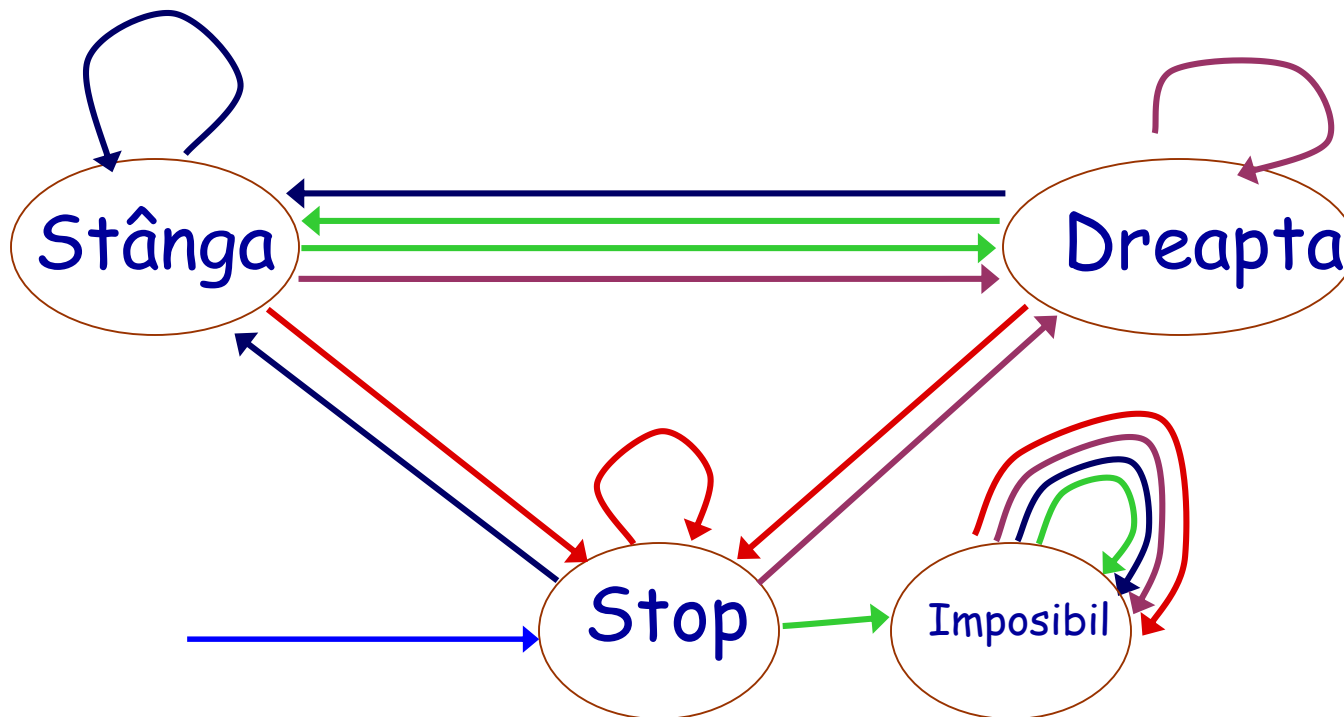
# Exemplu

- Tabelul stărilor (State table)

Eveniment Stare	Loveste Zid	Paleta Stânga	Paleta Dreaptă	Oprește
Stânga	Dreapta	Stânga	Dreapta	Stop
Dreapta	Stânga	Stânga	Dreapta	Stop
Stop	Imposibil	Stânga	Dreapta	Stop
Imposibil	Imposibil	Imposibil	Imposibil	Imposibil

# Exemplu

- Mingea lovește zid (schimbă direcția 180 gr) 
- Paleta spre stânga (mingea spre stânga) 
- Paleta spre dreapta (mingea spre dreapta) 
- Oprim mingea (mingea nu se mai mișcă) 



# Automatele finite (definiție formală)

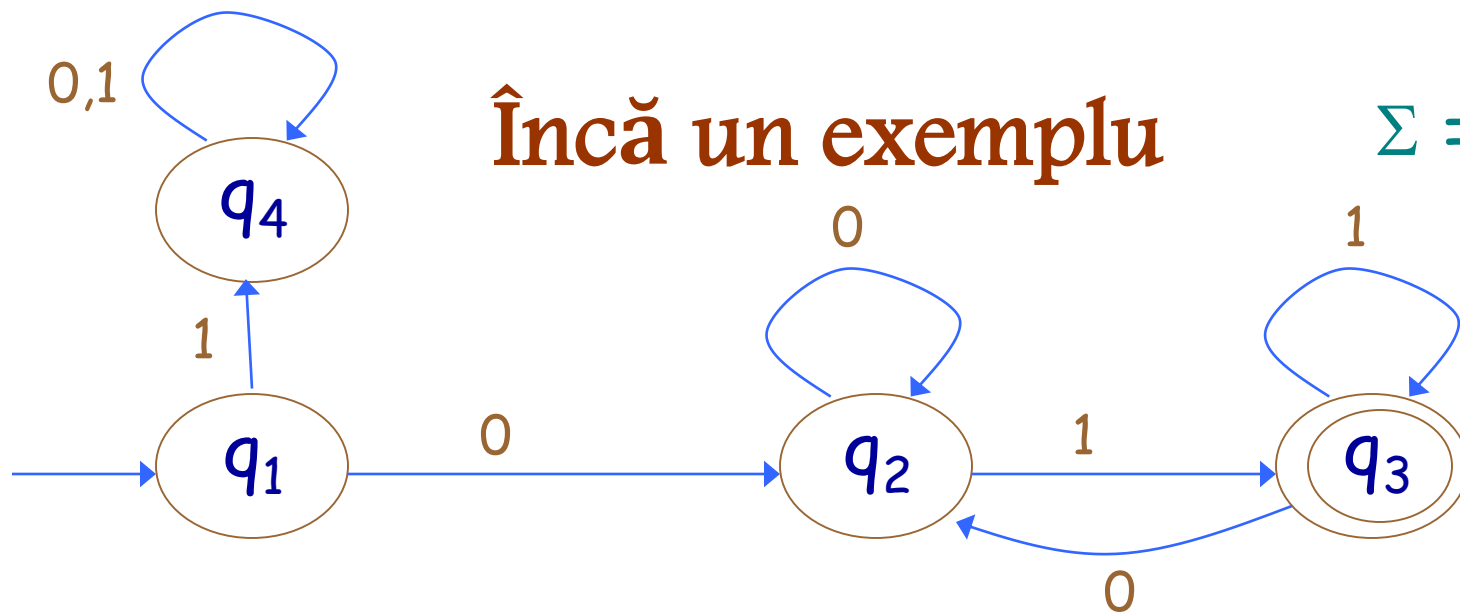
- Un automat finit este un 5-tuplu  $(Q, \Sigma, \delta, q_0, F)$ , unde
  1.  $Q$  e o mulțime finită nevidă care conține **stările**
  2.  $\Sigma$  e o mulțime finită numită **alfabet**
  3.  $\delta : Q \times \Sigma \rightarrow Q$  este **funcția de tranziție**
    - $\delta$  corespunde funcției evenimentelor din exemplul anterior
  4.  $q_0$  este **starea inițială**, și
  5.  $F \subseteq Q$  este mulțimea **stărilor finale** (numite si **stări acceptoare**).

# Exemplu

- Din exemplul precedent
  - $Q = \{\text{Stânga}, \text{Dreapta}, \text{Stop}, \text{Imposibil}\}$
  - $\Sigma = \{\text{Lovește perete}, \text{Paleta stânga}, \text{Paleta dreapta}, \text{Oprește}\}$
  - $\delta =$  tabelul de stări construit
  - $q_0 = \text{Stop}$
  - $F = \{\text{Stânga}, \text{Dreapta}, \text{Stop}\}$ 
    - Dar dacă vrem să acceptăm doar când mingea este în mișcare?
      - $F = \{\text{Stânga}, \text{Dreapta}\}$

## Încă un exemplu

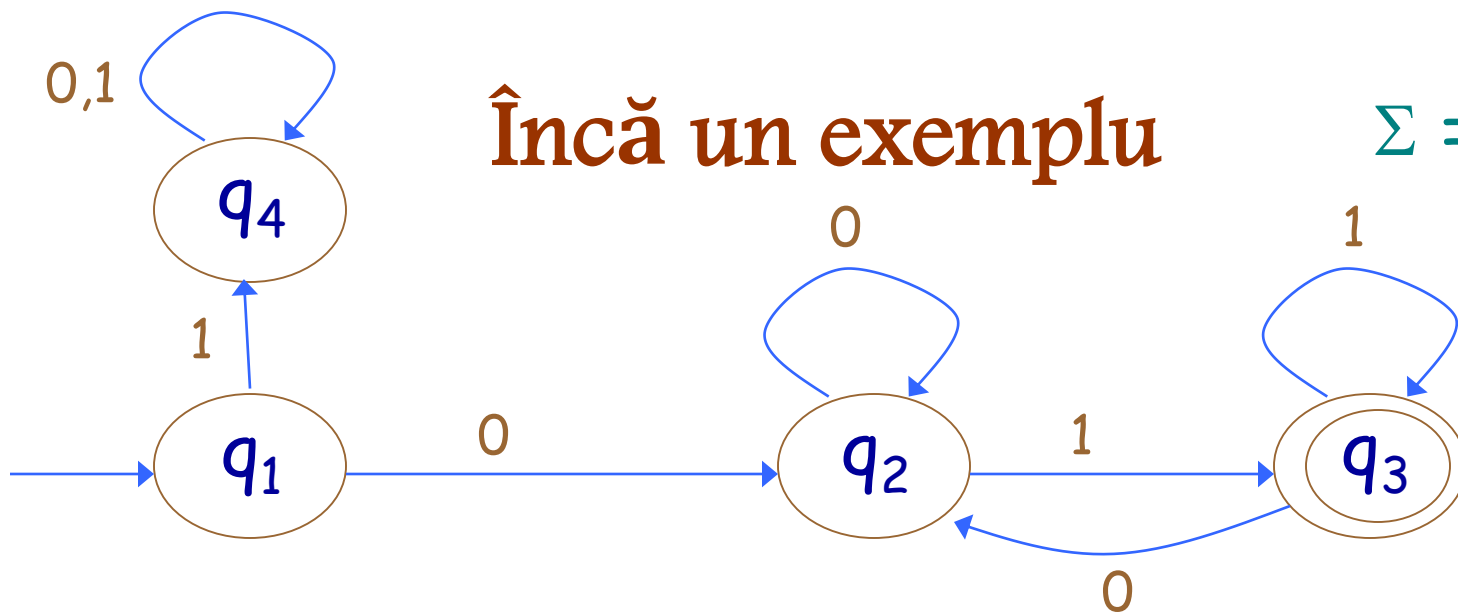
$\Sigma = \{0,1\}$



- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$
- $\delta$  (slide-ul urmator)
- $q_0 = q_1$
- $F = \{q_3\}$

## Încă un exemplu

$\Sigma = \{0,1\}$



Tab. stări

0

1

$q_1$

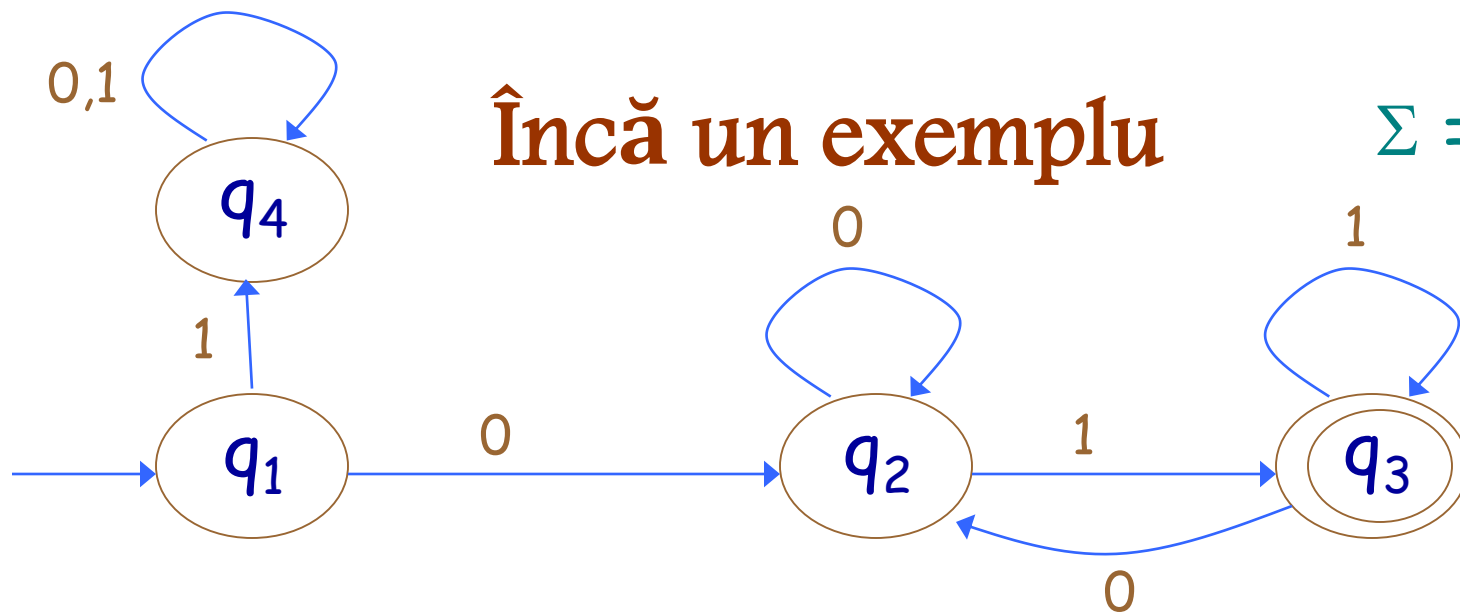
$q_2$

$q_3$

$q_4$

## Încă un exemplu

$\Sigma = \{0,1\}$



- descriere informală a șirurilor acceptate de acest DFA
  - Toate șirurile de 0 și 1 care încep cu 0 și se termină cu un 1

# Example

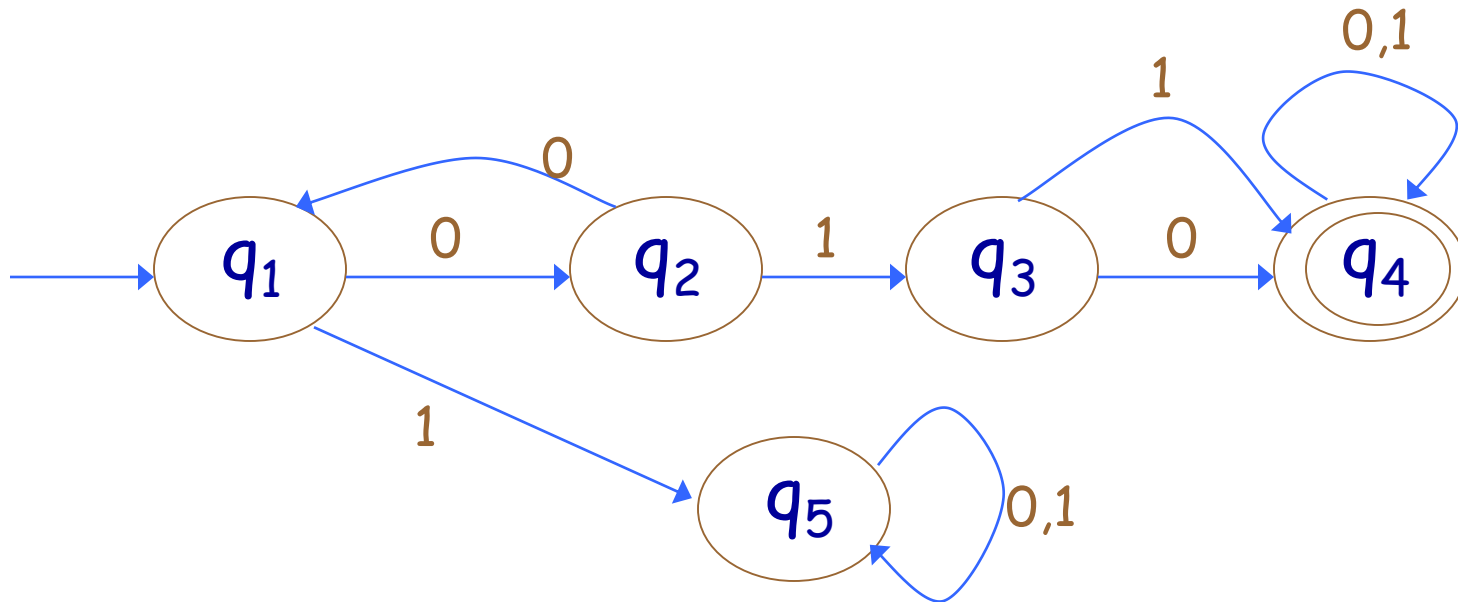
$\Sigma = \{0, 1\}$  pentru toate exemplele următoare

1.  $Q$  e o mulțime finită care conține **stările**
2.  $\Sigma$  e o mulțime finită numită **alfabet**
3.  $\delta : Q \times \Sigma \rightarrow Q$  este **funcția de tranziție**
  - $\delta$  corespunde funcției evenimentelor din exemplul anterior
4.  $q_0$  este **starea inițială**, și
5.  $F \subseteq Q$  este mulțimea **stărilor finale** (numite și **stări acceptoare**).

Încercați să descrieți informal cuvintele acceptate

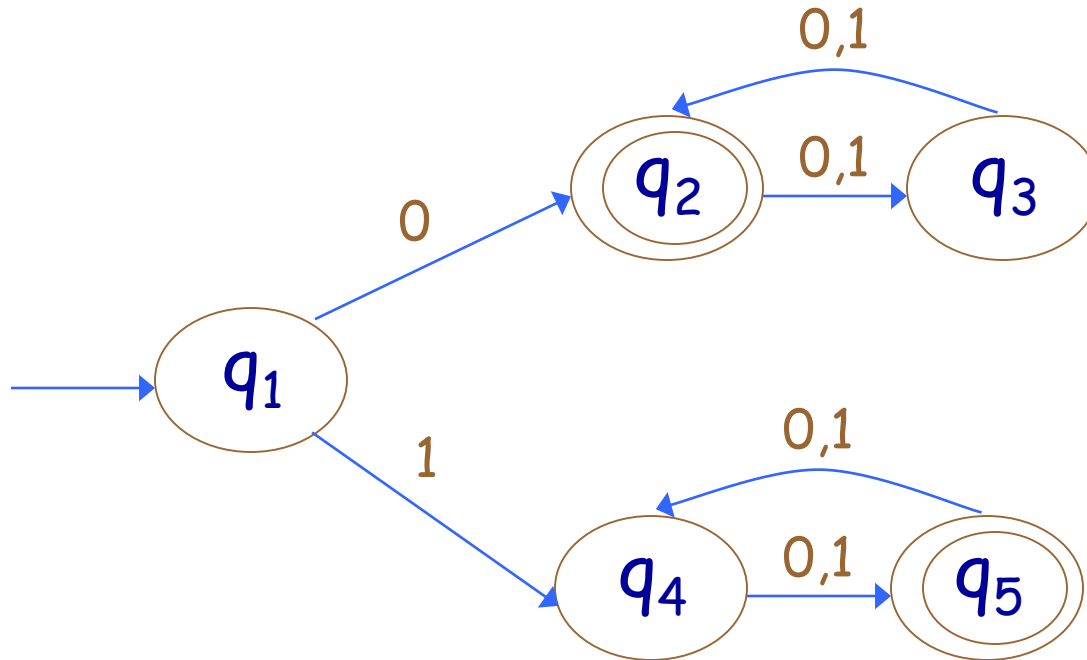


# Exemplul 1



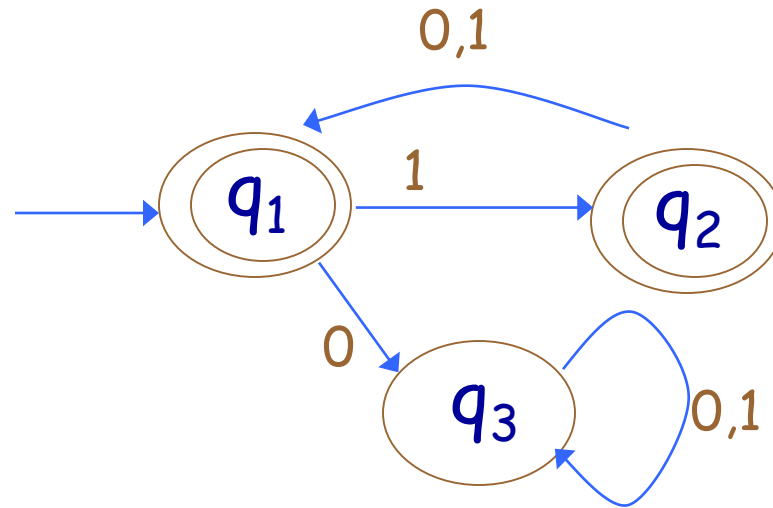
Sugestie ajutătoare: care sunt cuvintele care nu sunt acceptate?

## Exemplul 2



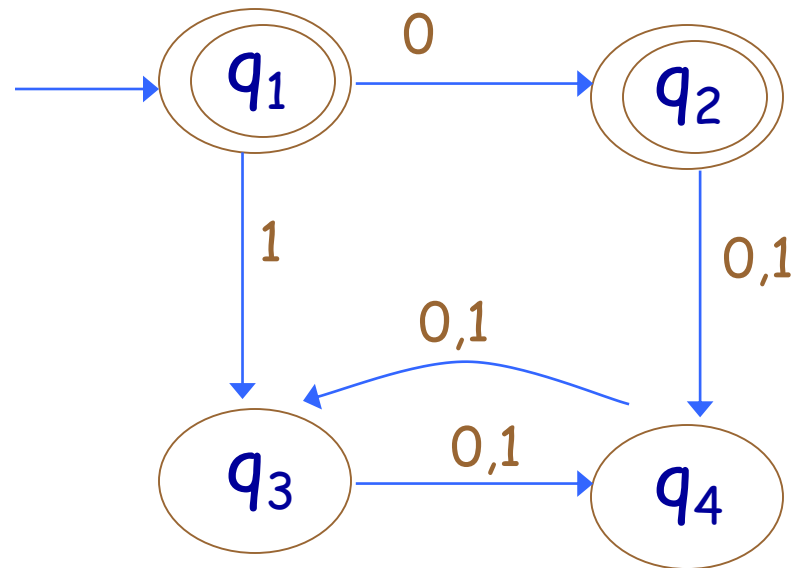
Sugestie ajutătoare: Lungimea șirurilor contează.

## Exemplul 3



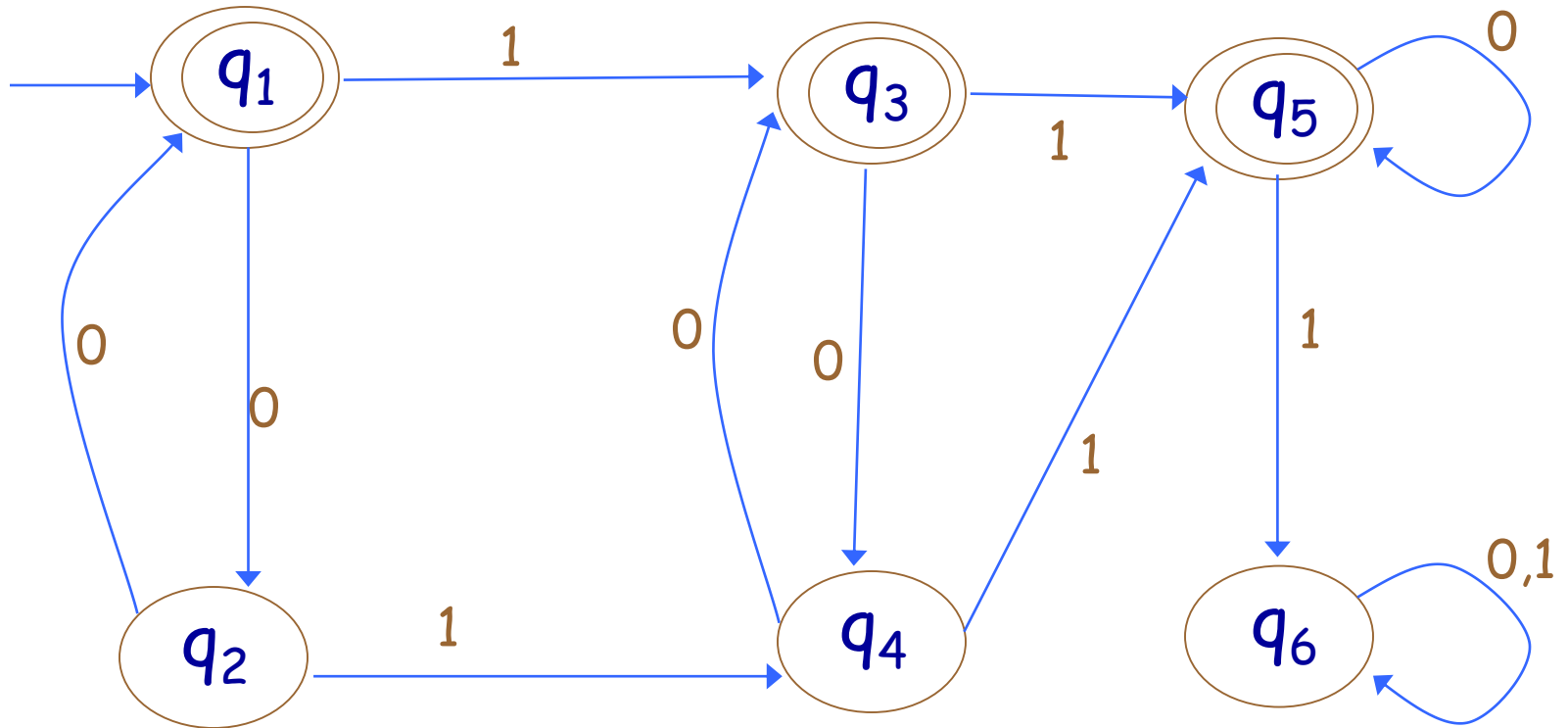
Sugestie ajutătoare: Poziția simbolurilor contează.

## Exemplul 4



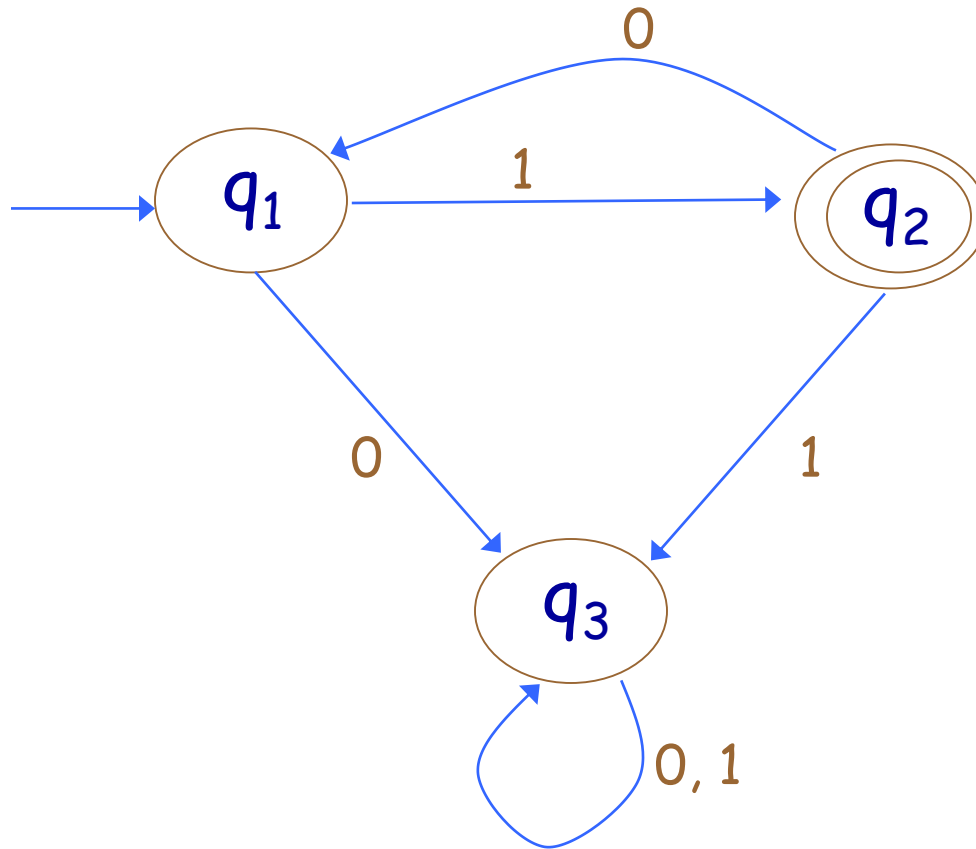
Sugestie ajutătoare: puteți simplifica acest DFA?

## Exemplul 5



Sugestie: de câte ori apare fiecare simbol?

## Exemplul 6



Sugestie ajutătoare: Ce se întâmplă când ajungem în  $q_3$ ?



