

# ORGANIZAREA INTERNĂ A DATELOR

- ❖ **Informația, data și cunoștința**
- ❖ **Clasificarea datelor**
- ❖ **Structuri statice de date**
- ❖ **Structuri dinamice de date**

# INFORMAȚIA, DATA ȘI CUNOȘTINȚA

## Abordare teoretică

### Direcții de abordare

- **în general:** semne care circulă pe diferite canale între elementele lumii reale, cu forme specifice de receptare la nivelul materiei vii;
- **în particular:** când receptorul este omul, în cadrul procesului de cunoaștere

### Caracteristici definitorii

- **semn cu semnificație,** prin existența unui limbaj cunoscut de către receptor;
- **noutate,** prin completarea tezaurului de cunoștințe al receptorului;
- **utilitate,** prin receptarea în timp util, în contextul dorit.

## Abordare informațională

**Cunoștința:** informația caracterizată de **utilitate și importanță** pentru receptorul uman

◆ Tezaur de cunoștințe

◆ Proces de informare

◆ Redundanță

## Abordare matematică

- **Cantitate de informație:** **formula lui Shannon**

## Abordare informațională

**Data:** mod concret de reprezentare a informației pentru un anumit procesor (om, calculator, algoritm, program etc.)

**Parametrii de definire a unei date:**

- **identificator**
- **valoare**
- **attribute**

# CLASIFICAREA DATELOR

## Abordare la nivel logic

- **După natură:**
  - **numerice:** naturale, întregi, reale, complexe;
  - **alfabetice**
  - **alfanumerice**
  - **logice**
- **După numărul de valori în timpul execuției programului:**
  - **variabile**
  - **constante propriu-zise (literali)**
  - **constante simbolice**
- **După numărul de valori memorate concomitent:**
  - **elementare (scalare)**
  - **structurate (structuri de date)**

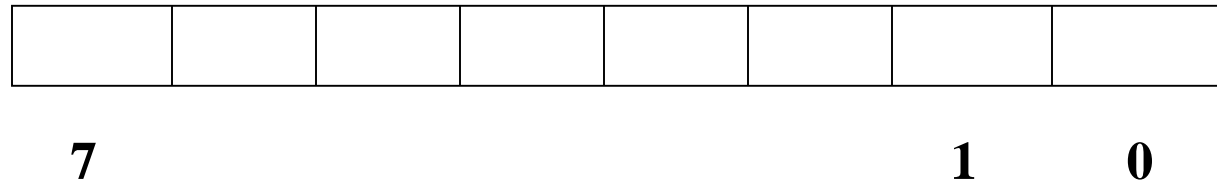
## Clasificarea structurilor de date

- **După adresa fizică de memorie în timpul execuției:**
  - structuri statice
  - structuri dinamice
- **După modul de referire a elementelor:**
  - cu acces secvențial
  - cu acces direct
- **După natura elementelor:**
  - omogene
  - eterogene
- **După tipul elementelor:**
  - cu elemente date scalare
  - cu elemente date structurate (structură recursivă)
- **După mediul de memorare:**
  - interne (în memoria principală)
  - externe (pe medii magnetice: fișiere, baze de date)

## Abordare la nivel fizic

### ◆ Reprezentarea internă a datelor numerice naturale (întregi fără semn)

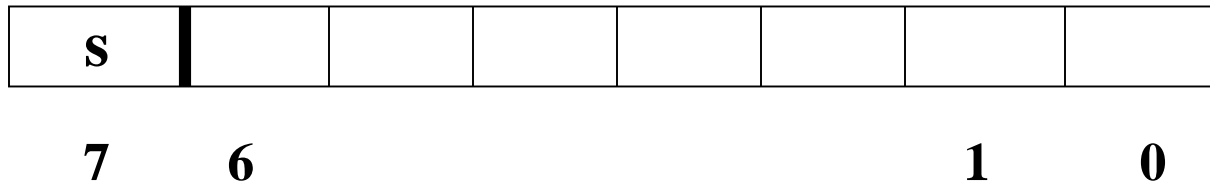
Virgulă fixă aritmetică (1 sau 2 octeți)



- Codul utilizat: **cod direct**
- Plaja de valori:  **$[0, 2^8-1]$**

### ◆ Reprezentarea internă a datelor numerice întregi (întregi cu semn)

Virgulă fixă algebrică (1, 2 sau 4 octeți)



- Codul utilizat:
  - pentru numere pozitive: **cod direct**
  - pentru numere negative: **cod complementar**
- Plaja de valori:  **$[-2^7, 2^7-1]$**

## ◆ Reprezentarea internă a datelor numerice reale

### Virgulă mobilă

S	Caracteristică (8/11 biți)	Fracție (23/52 biți)
---	----------------------------	----------------------

**Normalizare:**  $n = (-1)^s * 1,fracție * 2^{exponent}$

Caracteristica = exponent + 127  $\Rightarrow$  simplă precizie

Caracteristica = exponent + 1023  $\Rightarrow$  dublă precizie

- Codul utilizat: **cod direct**
- Plaja de valori:
  - simplă precizie:  $[-10^{38}, 10^{38}]$
  - dublă precizie:  $[-10^{307}, 10^{307}]$



## ◆ Reprezentarea internă a datelor alfabetice și alfanumerice

**Codul ASCII ⇒ un caracter pe octet ⇒ 256 de caractere distincte**

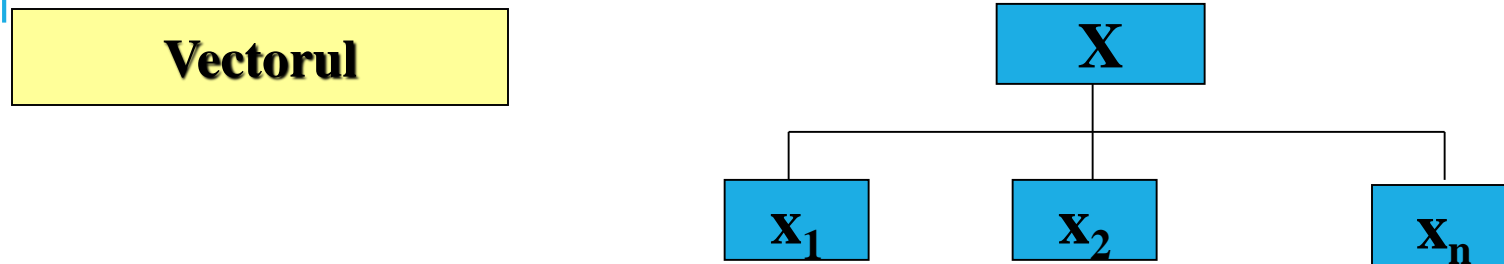
Codurile ASCII	Caracterele
0 – 31	Coduri de control
32 – 47	Caractere speciale de pe tastatură
48 – 63	0 1 2 3 4 5 6 7 8 9
64 – 79	! " # \$ % & ' ( ) * + , - . /
80 – 95	: ; < = > ? @ [ \ ] ^ _
96 – 111	` {   } ~
112 – 127	! " # \$ % & ' ( ) * + , - . /
128 – 143	0 1 2 3 4 5 6 7 8 9
144 – 159	! " # \$ % & ' ( ) * + , - . /
160 – 175	0 1 2 3 4 5 6 7 8 9
176 – 191	! " # \$ % & ' ( ) * + , - . /
192 – 207	0 1 2 3 4 5 6 7 8 9
208 – 223	! " # \$ % & ' ( ) * + , - . /
224 – 239	0 1 2 3 4 5 6 7 8 9
240 – 255	! " # \$ % & ' ( ) * + , - . /

## ◆ Reprezentarea internă a datelor logice

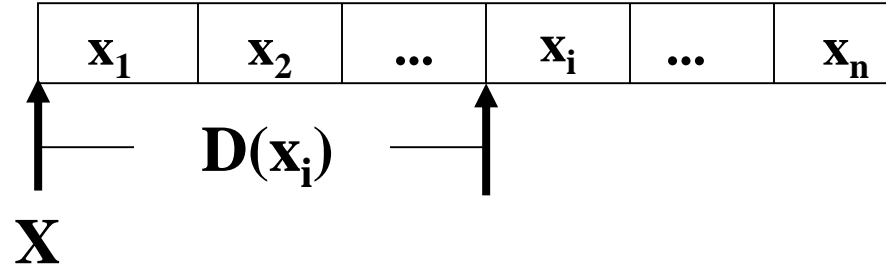
- **adevărat - 1** reprezentat în virgulă fixă, pe un octet
- **fals - 0** reprezentat în virgulă fixă, pe un octet

# STRUCTURI STATICE DE DATE

**Masivul:** structură de date omogenă, cu acces direct, între elementele căreia există o relație ierarhică, pe mai multe niveluri



## • Reprezentare internă

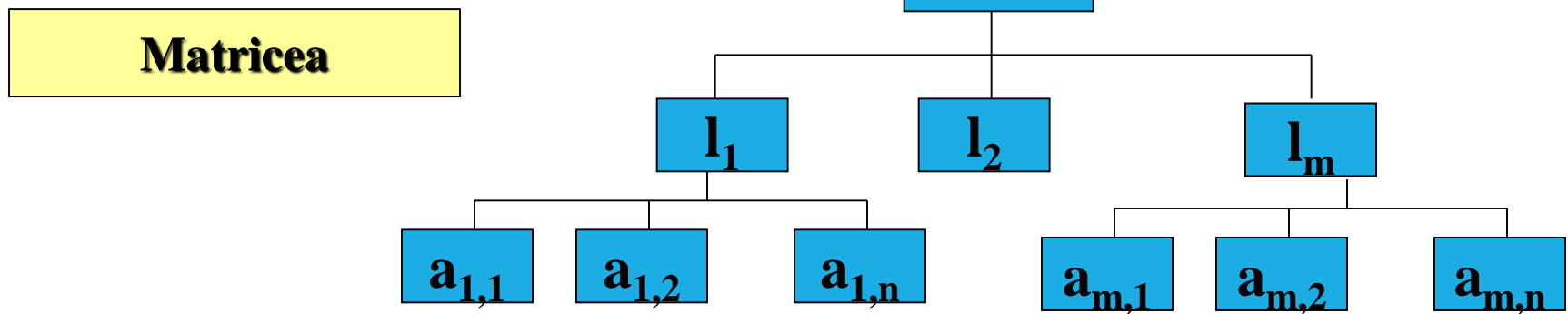


## • Referire element

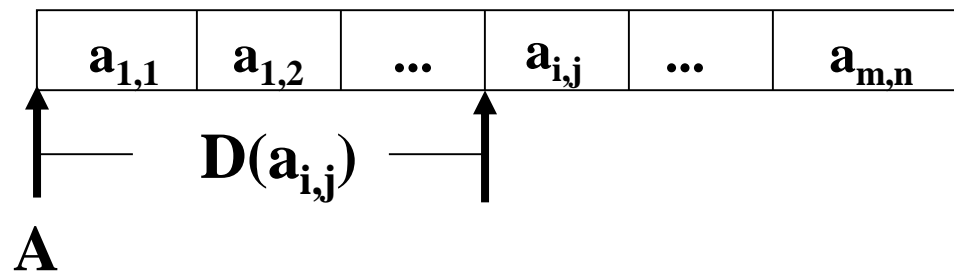
$$\text{Adr}(x_i) = \text{Adr}(X) + D(x_i)$$

$$D(x_i) = (r(x_i) - 1) * l$$

$$r(x_i) = i$$



- **Reprezentare internă:** lexicografică sau invers lexicografică



- **Referire element**

$$\text{Adr}(a_{i,j}) = \text{Adr}(A) + D(a_{i,j})$$

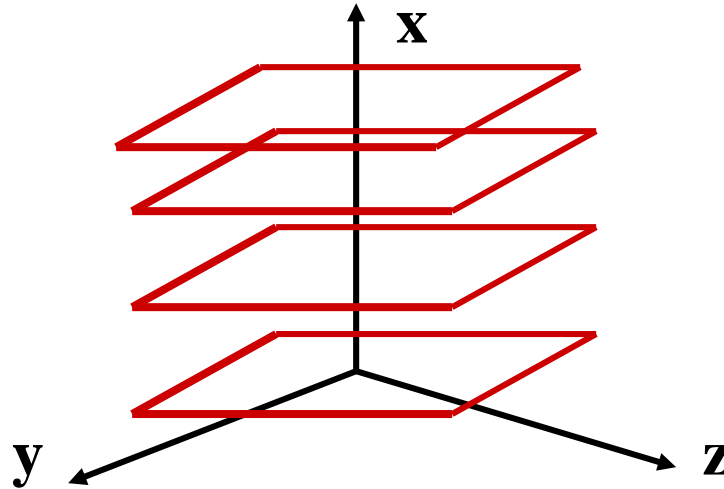
$$D(a_{i,j}) = (r(a_{i,j}) - 1) * l$$

$$r(a_{i,j}) = (i - 1) * n + j \Rightarrow \text{memorare lexicografică}$$

$$r(a_{i,j}) = (j - 1) * m + i \Rightarrow \text{memorare invers lexicografică}$$

## Masivul tridimensional

$T(m,n,p)$  = vector  $m$  dimensional, de matrice  $n * p$



### • Referire element

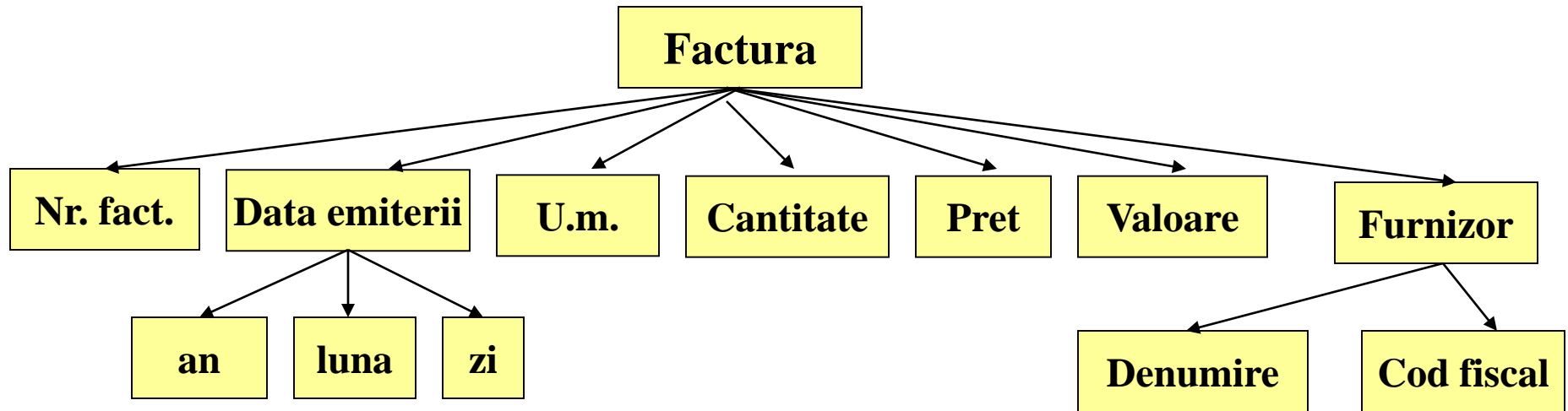
$$\text{Adr}(t_{i,j,k}) = \text{Adr}(T) + D(t_{i,j,k})$$

$$D(t_{i,j,k}) = (r(t_{i,j,k}) - 1) * l$$

$$r(t_{i,j,k}) = ((i - 1) * p + j - 1) * n + k \quad \Rightarrow \text{memorare lexicografică}$$

$$r(t_{i,j,k}) = ((k - 1) * n + j - 1) * m + i \quad \Rightarrow \text{memorare invers lexicografică}$$

**Articolul:** structură de date neomogenă, cu acces direct, între elementele căreia există o relație de ordine ierarhică, pe mai multe niveluri de arborescență



- **Date (câmpuri) elementare** ⇒ date fără descendenți
- **Date de grup** ⇒ date care au descendenți
- **Articolul** ⇒ data de grup de cel mai înalt nivel

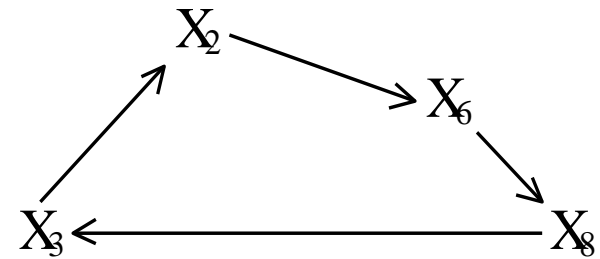
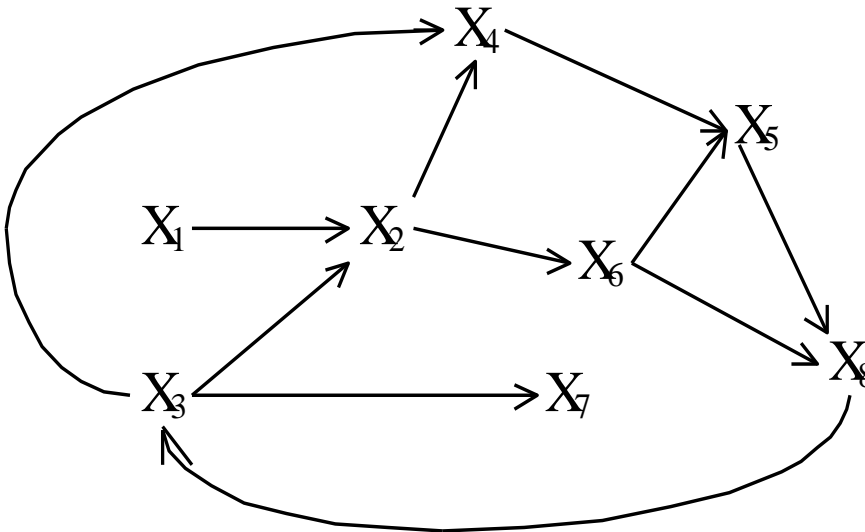
- **Reprezentare internă:** juxtapunerea datelor elementare
- **Referire elemente:** prin nume = deplasare față de adresa de început

# Structuri dinamice de date

## Graf orientat

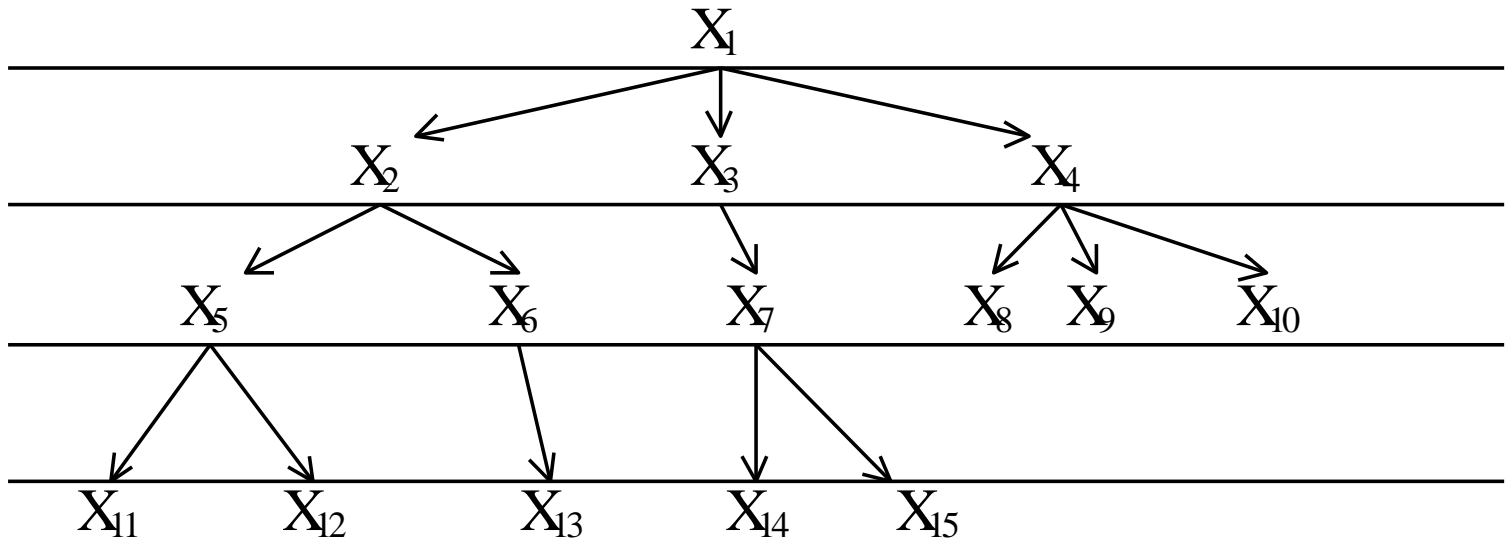
$G = (X, U); X = \{x_1, x_2, \dots, x_n\}; (x_i, x_j) \in U$

$H = (Y, V); Y \subseteq X$

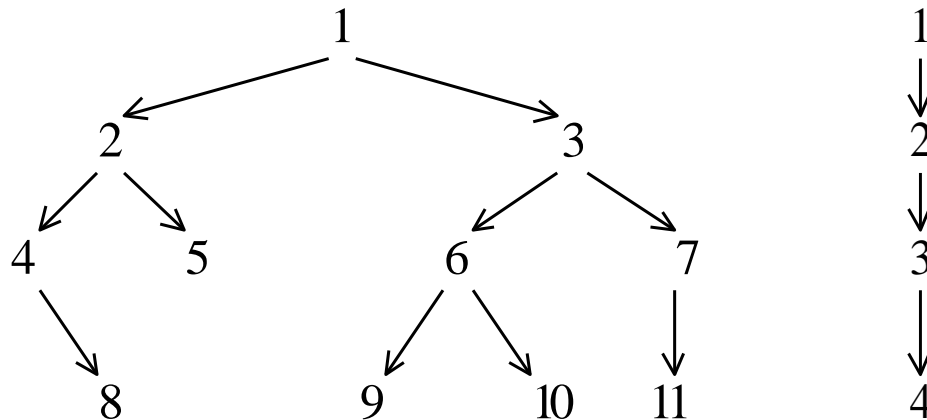


- **Drum de lungime  $n$  ( $n \leq 1$ )**
- **Drum elementar**
- **Circuit**

# Arbore

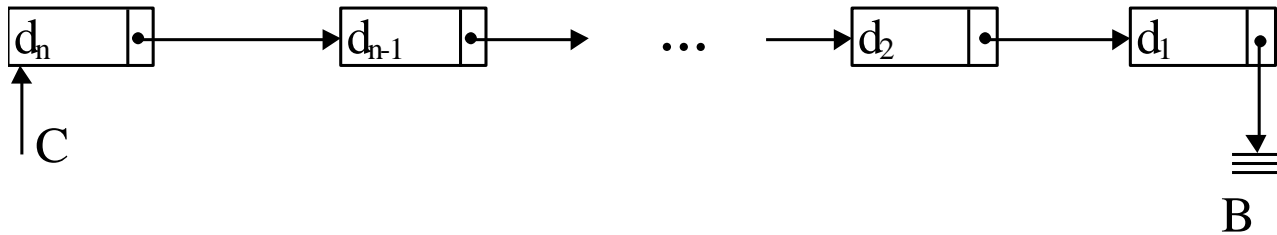


## Nod de ordin $n$



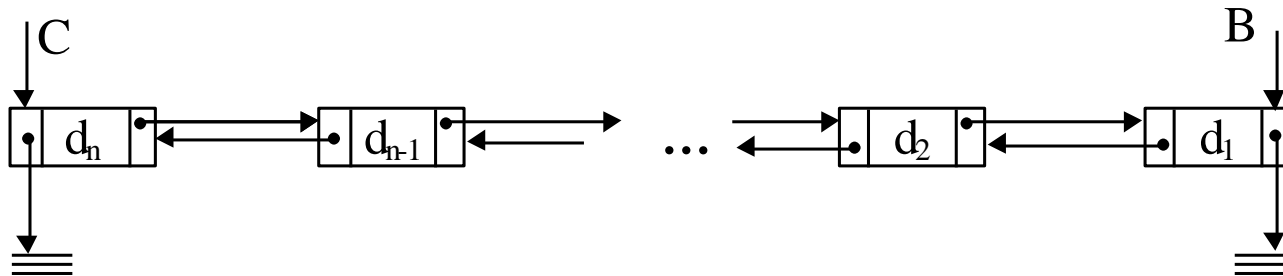
## Lista simplu înlănțuită

$$L_a = \{(d_i, s_i) \mid d_i \in D, s_i \in P\}$$



## Lista dublu înlănțuită

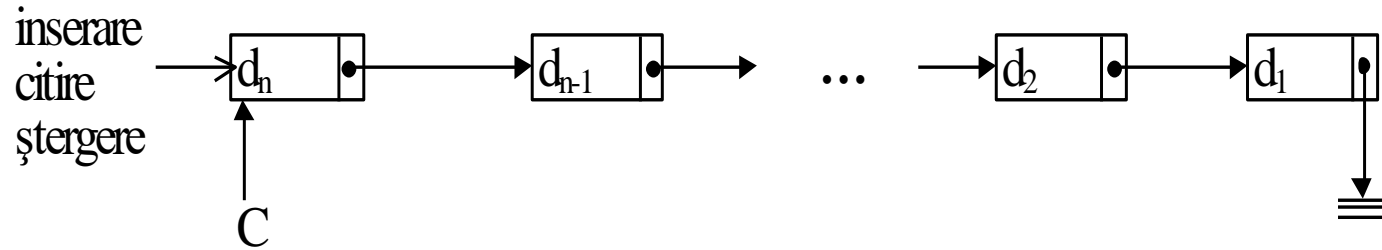
$$L_s = \{(p_i, d_i, s_i) \mid d_i \in D, p_i, s_i \in P\}$$



**Operații pe liste: traversare, inserare, ștergere etc.**

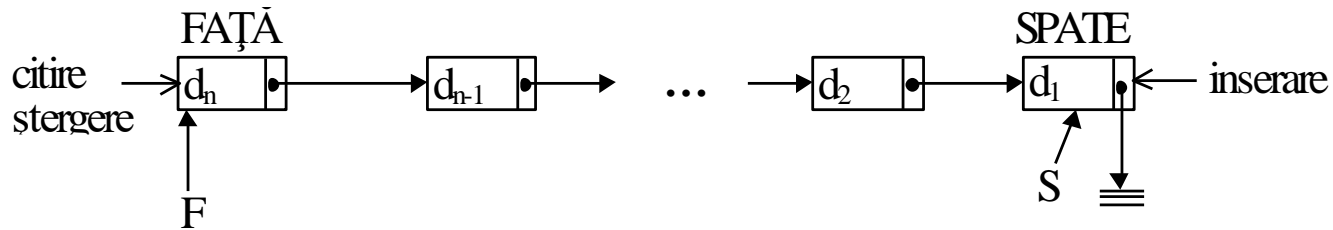


## Stiva (lista LIFO)



**Operații pe stivă: inserare în capul stivei, ștergere din capul stivei, citirea din capul stivei.**

## Coadă (lista FIFO)



**Operații specifice: inserare în spate, ștergere din față cozii, citirea din față cozii.**