

ALGORITMI ȘI SCHEME LOGICE

- **Caracteristicile algoritmilor**
- **Iterativitate și recursivitate**
- **Reprezentarea algoritmilor**
- **Descrierea structurilor fundamentale**
- **Structurarea algoritmilor**
- **Erorile în algoritmi**
- **Proiectarea algoritmilor**

Caracteristicile algoritmilor

- Generalitate
- Determinare (claritate)

Exemplul 1: ecuația de grad 2

Exemplul 2:

- *Suma elementelor impare dintr-un șir*
 - *Suma elementelor pare dintr-un șir*
- Finitudine

Clase de algoritmi:

♠ *Algoritmi cu număr finit de pași, a priori cunoscut*

Produs scalar între două mulțimi

♠ *Algoritmi cu număr finit de pași, a posteriori cunoscut*

- CMMDC între două numere
- Numerele prime până la o limită dată

♠ *Algoritmi cu număr infinit de pași*

- Rezolvarea unei ecuații transcendente
- Numărarea unor elemente care îndeplinesc o condiție dată

Iterativitate și recursivitate

Iterativitate

- ✗ Produs vectorial
- ✗ Pătratele elementelor unui șir
- ✗ Creare vectori

- formula iterativă

Recursivitate

- ✗ Suma elementelor unui șir
- ✗ Produsul elementelor unui șir
- ✗ Produs scalar
- ✗ Maxim (minim) dintr-un șir
- ✗ Cmmdc dintre două numere

- formula de start
- formula recursivă

Reprezentarea algoritmilor prin scheme logice

Blocul START

START

Blocul STOP

STOP

Blocul de citire

Citește
date_de_intrare

Blocul de scriere

Scrie
date_de_ieșire

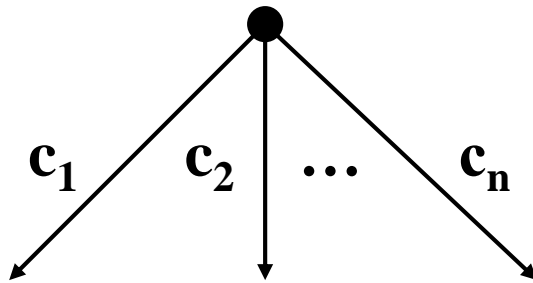
Blocul de atribuire

$v = e$

$v \leftarrow e$

$e \rightarrow v$

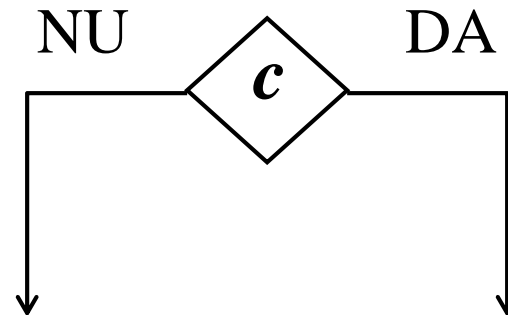
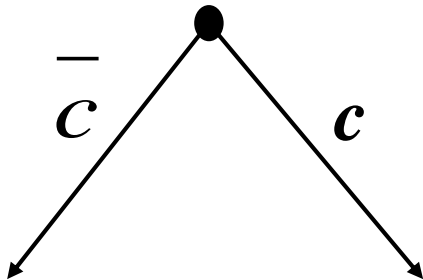
Blocul de ramificare



$$c_1 \vee c_2 \vee \dots \vee c_n = 1$$

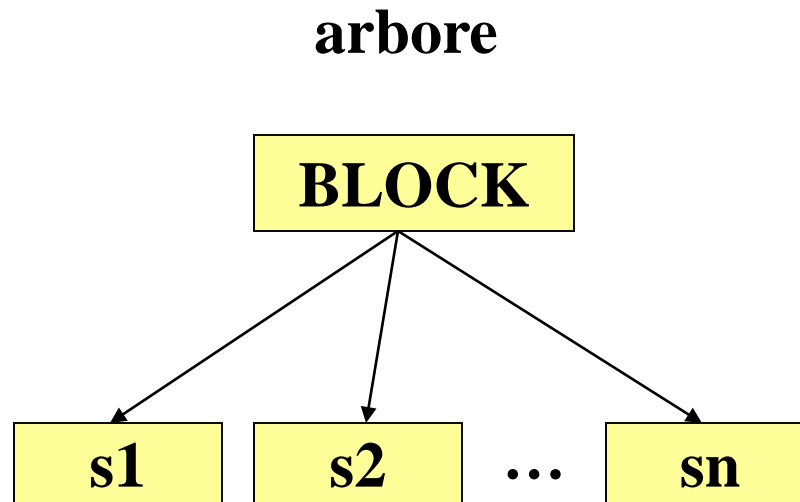
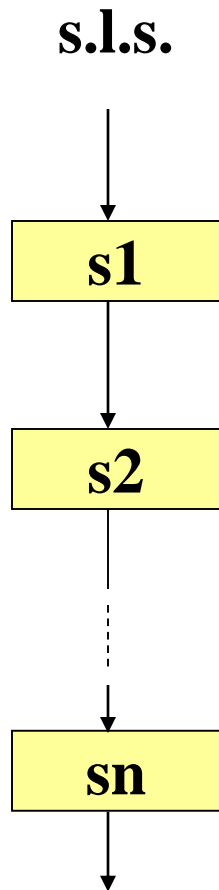
$$c_i \wedge c_j = 0, \forall i \neq j; i, j = 1, n$$

Pentru cazul $n = 2$



Structurile fundamentale din programarea structurată

Structura secvențială (liniară)



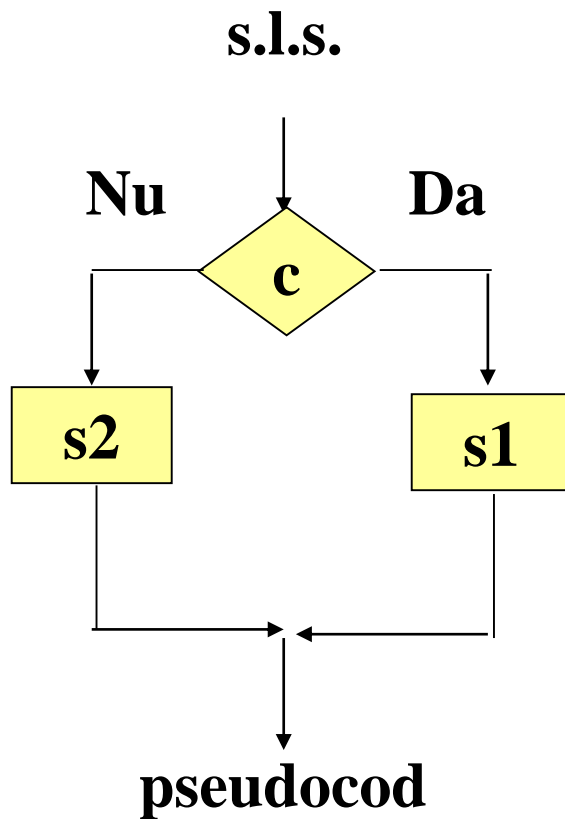
pseudocod

```
{  
    s1;  
    s2;  
    ...  
    sn  
}
```

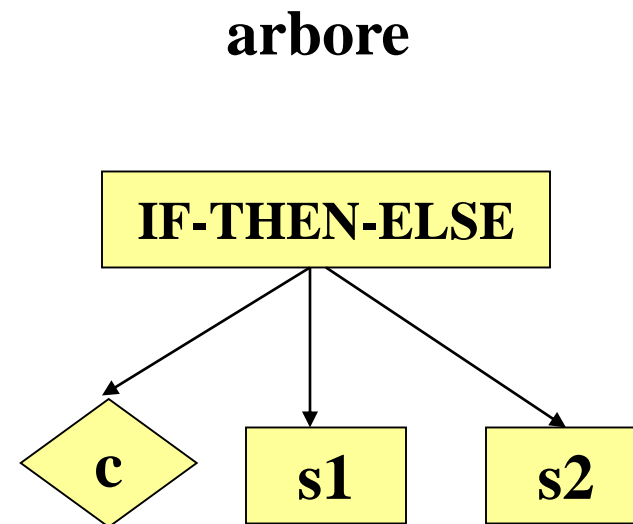
analitic: **BLOCK(s1,s2,...,sn)**

Structură PRIVILEGIATĂ !

Structurile alternative - selecția simplă



```
if c then
    s1
else
    s2
endif
```



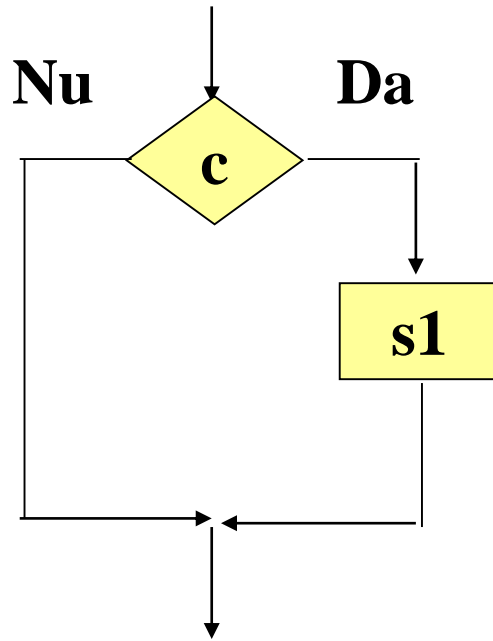
analitic

```
IF-THEN-ELSE(c,s1,s2)
```

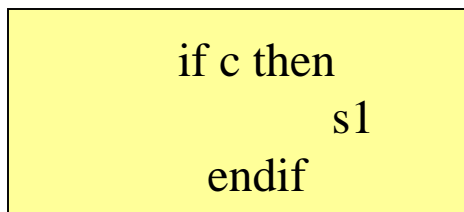
Structură PRIVILEGIATĂ !

Structurile alternative - pseudoalternativa

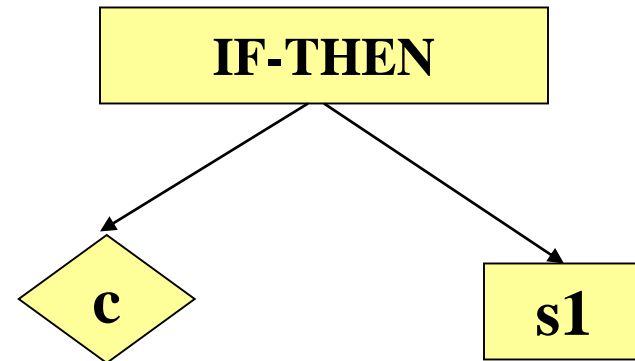
s.l.s.



pseudocod



arbore

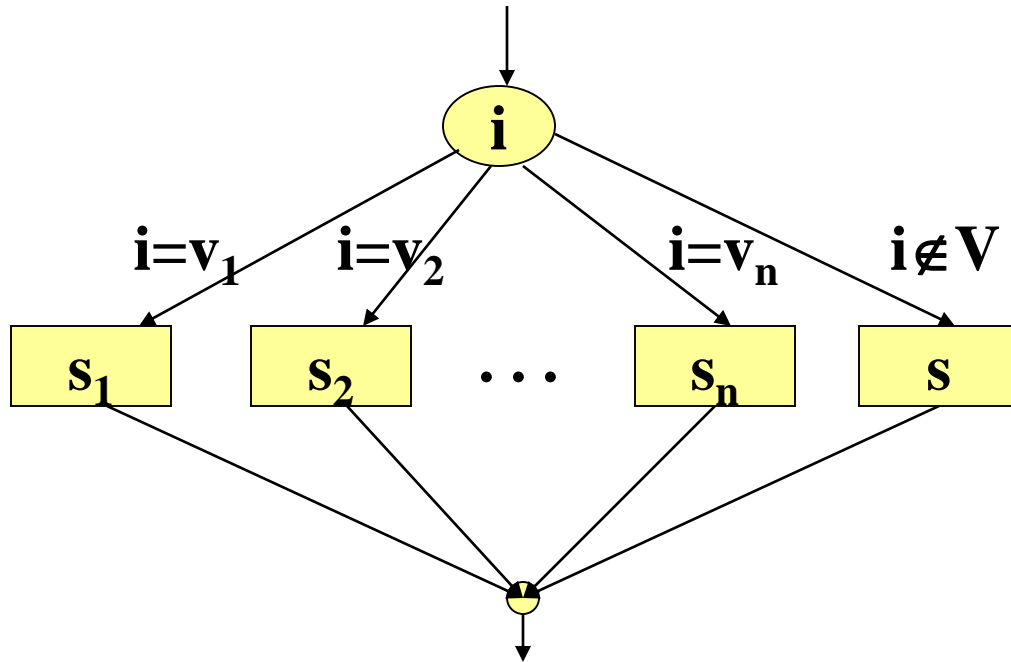


analitic



Structura alternativă multiplă

s.l.s.



analitic

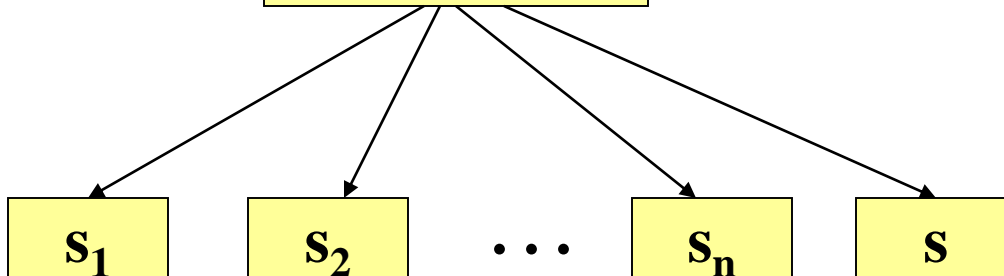
CASE-OF ($i, s_1, s_2, \dots, s_n, s$)

pseudocod

```
case of i
  i=v1: s1;
  i=v2: s2;
  ...
  i=vn: sn
  else s
endcase
```

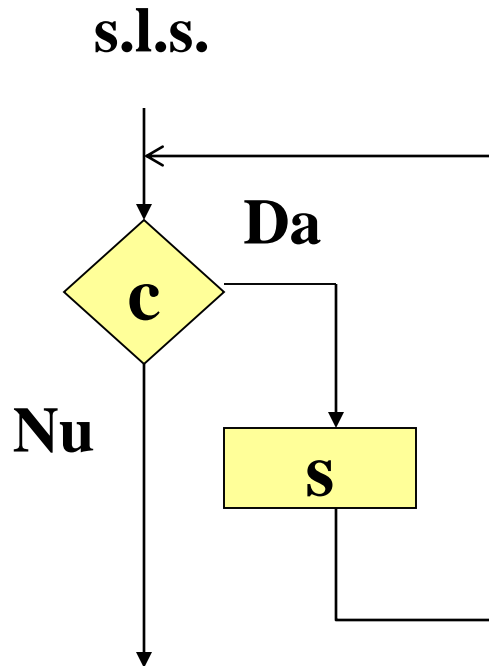
arbore

CASE-OF i



Structurile repetitive

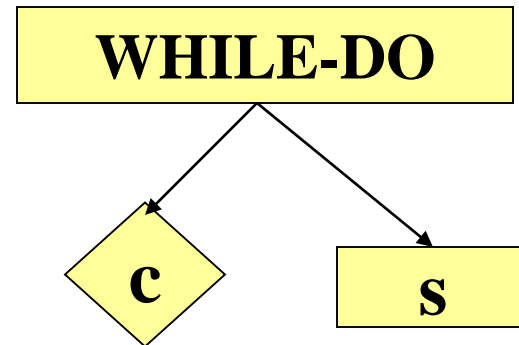
Structura repetitivă condiționată anterior



pseudocod

```
while c do
    S
endwhile
```

arbore



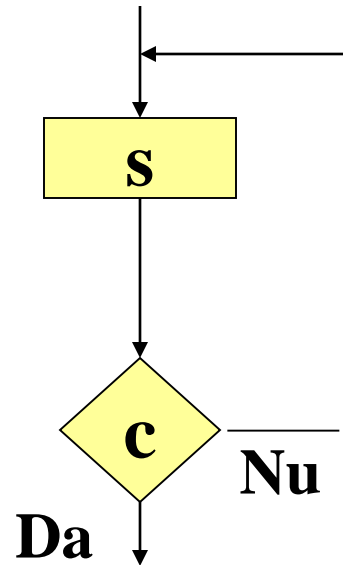
analitic

```
WHILE-DO(c,s)
```

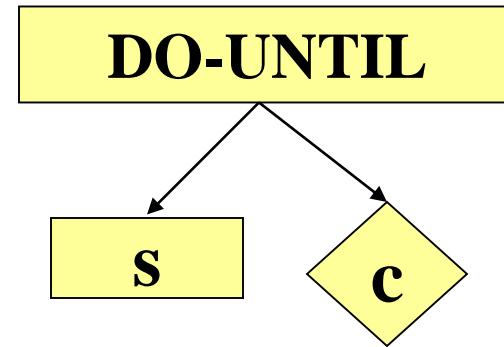
Structură PRIVILEGIATĂ !

Structura repetitivă condiționată posterior

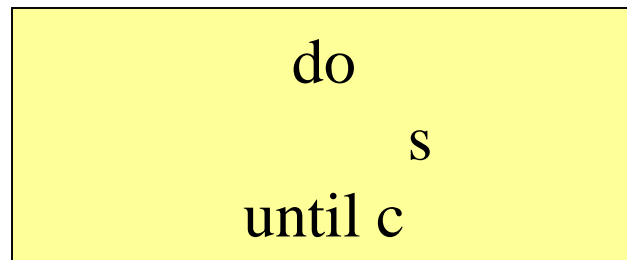
s.l.s.



arbore



pseudocod

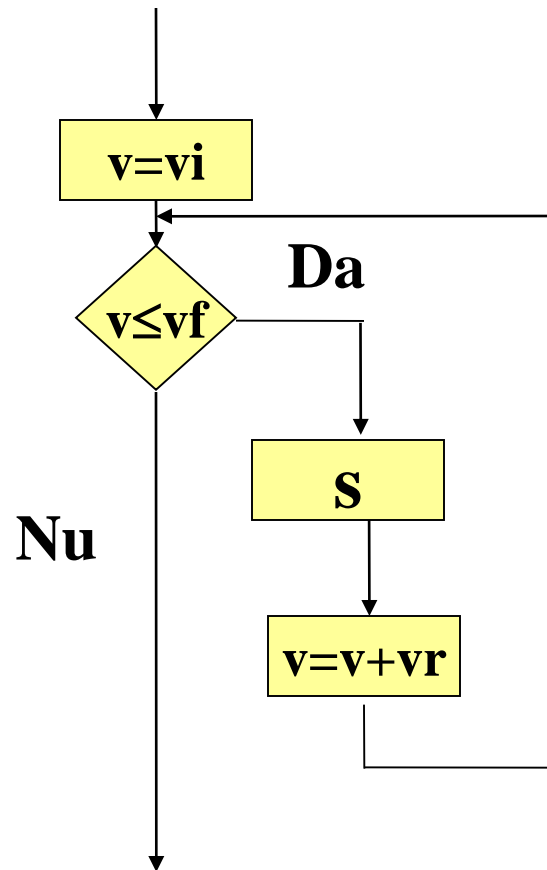


analitic



Structura repetitivă cu numărător

s.l.s.



$$N = [(v_f - v_i) / v_r] + 1$$

arbore

DO-FOR(v_i, v_f, v_r)

S

pseudocod

```
do for v=vi,vf,vr
    S
enddo
```

analitic

DO-FOR(v, v_i, v_f, v_r, S)

Structurarea algoritmilor

Mulțimea structurilor privilegiate

S = (BLOCK, IF-THEN-ELSE, WHILE-DO)

Mulțimea structurilor fundamentale

**S' = (BLOCK, IF-THEN-ELSE, IF-THEN, CASE-OF, WHILE-DO,
DO-UNTIL, DO-FOR)**

- **Un algoritm este S structurat (sau S' structurat) dacă este format numai din elemente din mulțimea S (respectiv S').**

Teorema fundamentală de structură (Boem-Jacopini)

- Fie P un algoritm nestructurat, format dintr-o mulțime A de acțiuni (operații) și o mulțime C de condiții. Dacă se adaugă un număr finit de acțiuni și/sau de condiții, se obține un algoritm structurat, echivalent cu P .

Corolarul top-down

- Un algoritm P structurat este echivalent cu un algoritm pus sub una din următoarele forme:
 - $P = \text{BLOCK}(s_1, s_2, \dots, s_n)$
 - $P = \text{IF-THEN-ELSE}(c, s_1, s_2)$
 - $P = \text{WHILE-DO}(c, s)$

Metode de structurare a algoritmilor

❖ *Metoda dublării codurilor*

- structurarea secvențelor alternative
- structurarea secvențelor repetitive

❖ *Metoda folosirii de variabile booleene*

- structurarea secvențelor repetitive

Erorile în algoritmi

☐ Erori în datele inițiale:

- erori de observare
- erori datorate numerelor iraționale

☐ Erori de rotunjire

☐ Erori de metodă

☐ Erori reziduale

Proiectarea algoritmilor

- Proiectarea, codificarea și testarea top-down
- Proiectarea modularizată
- Proiectarea structurată