

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*

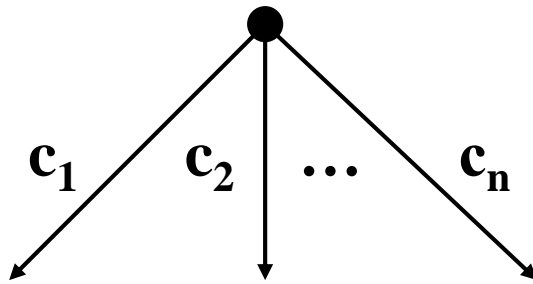
Blocul de atribuire

$v = e$

$v \leftarrow e$

$e \rightarrow v$

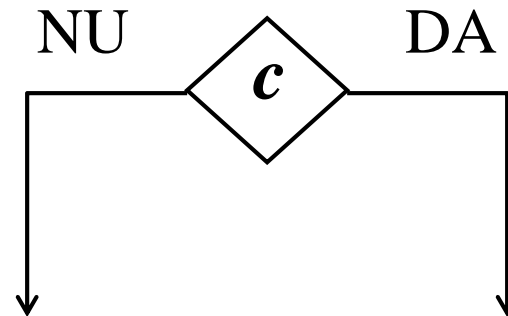
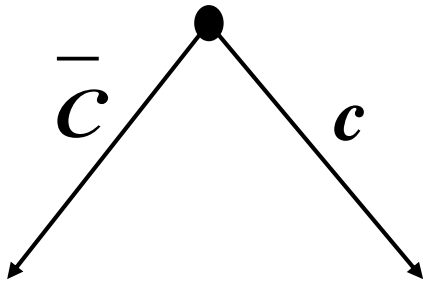
Blocul de ramificare



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

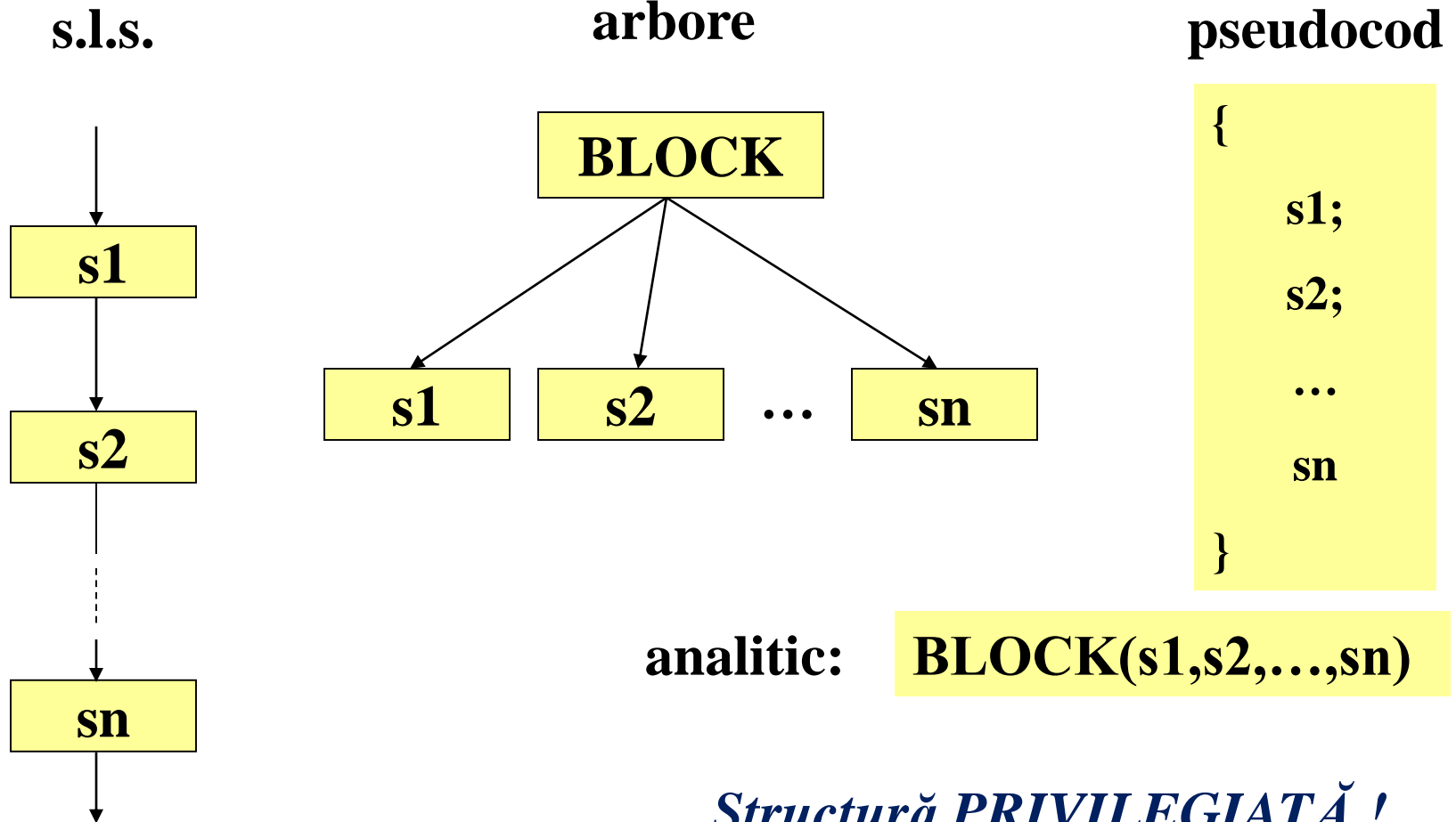
$$c_i \wedge c_j = 0, \forall i \neq j; i, j = 1, n$$

Pentru cazul $n = 2$

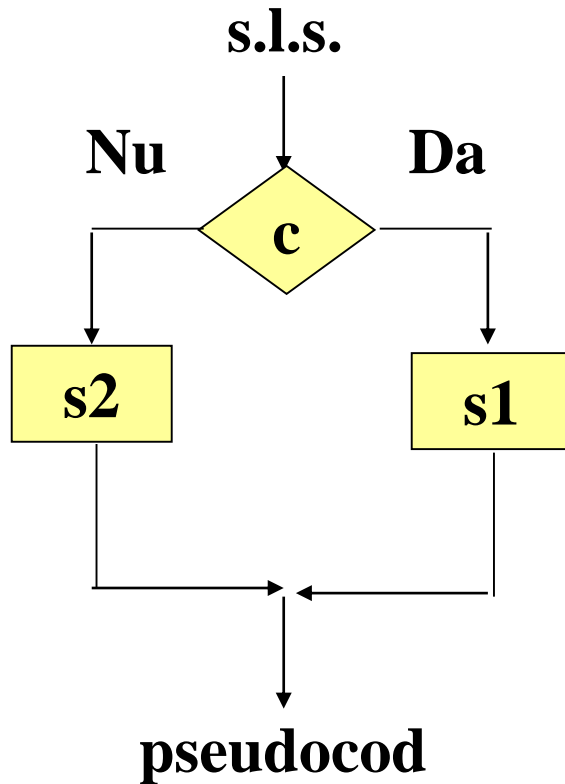


Structurile fundamentale din programarea structurată

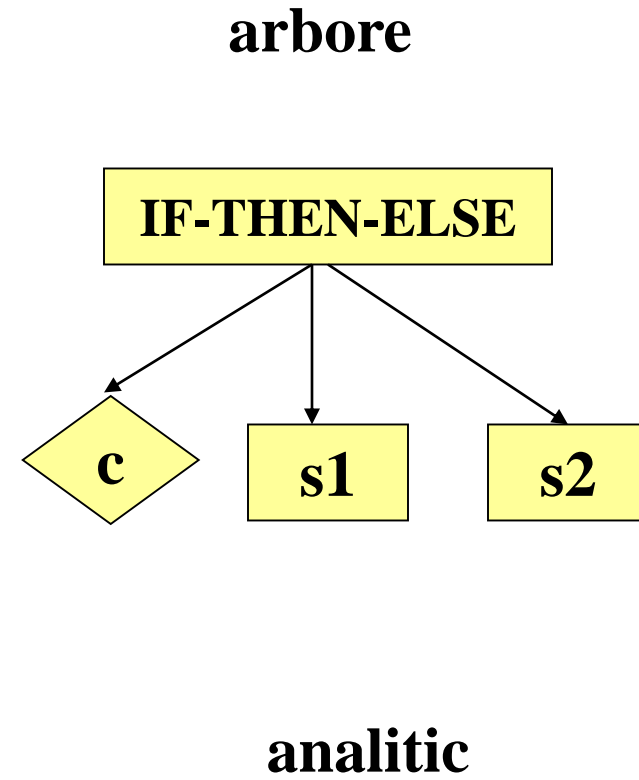
Structura secvențială (liniară)



Structurile alternative - selecția simplă



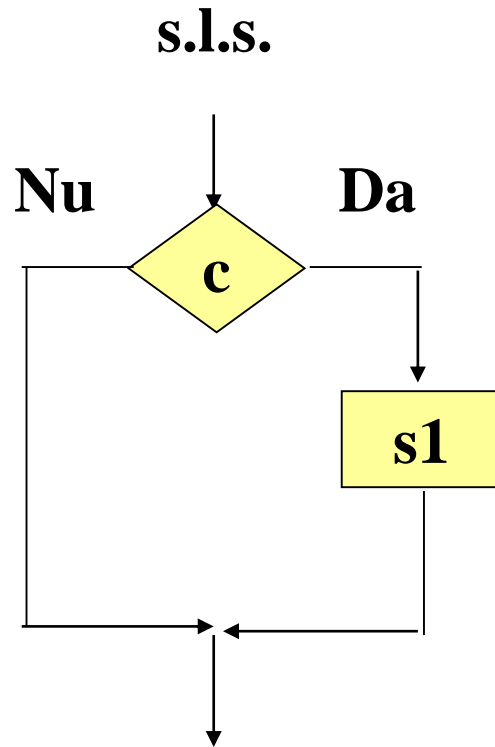
```
IF c THEN
    s1
ELSE
    s2
ENDIF
```



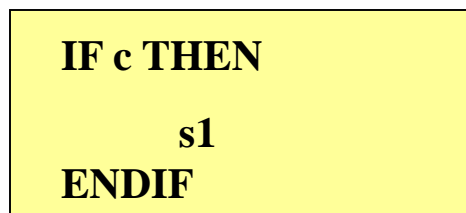
```
IF-THEN-ELSE(c,s1,s2)
```

Structură PRIVILEGIATĂ !

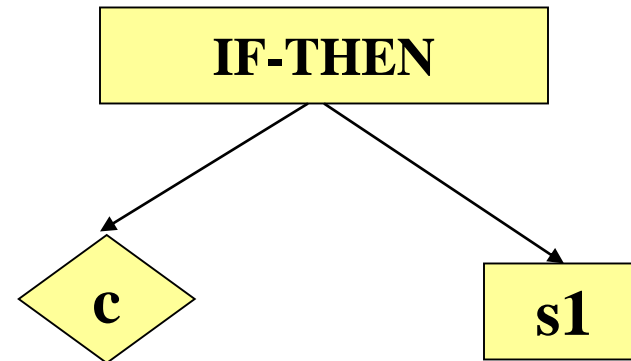
Structurile alternative - pseudoalternativa



pseudocod



arbore

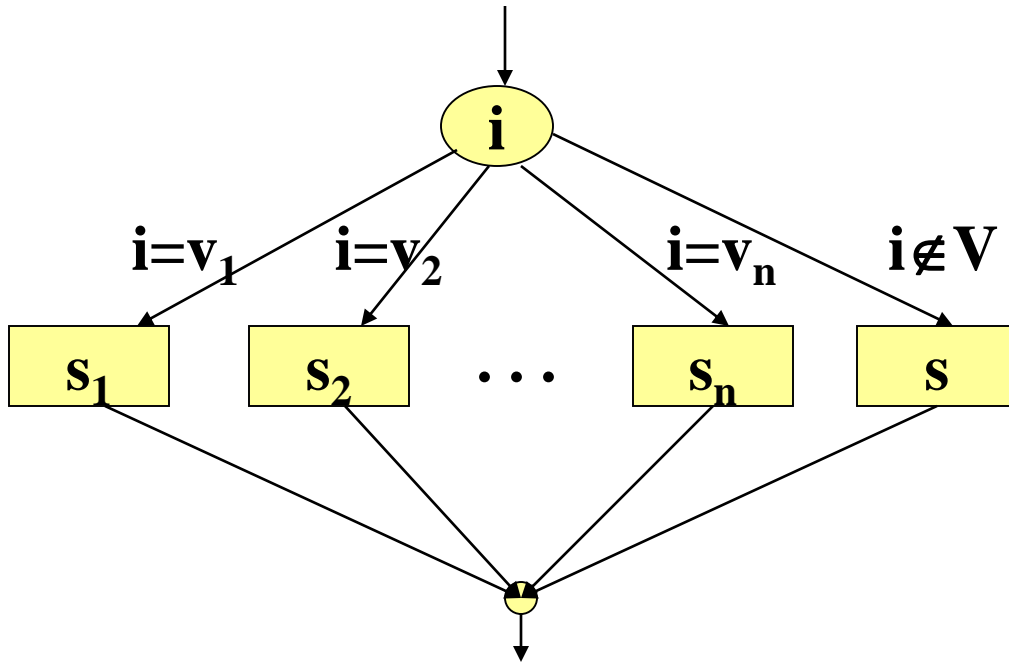


analitic



Structura alternativă multiplă

s.l.s.



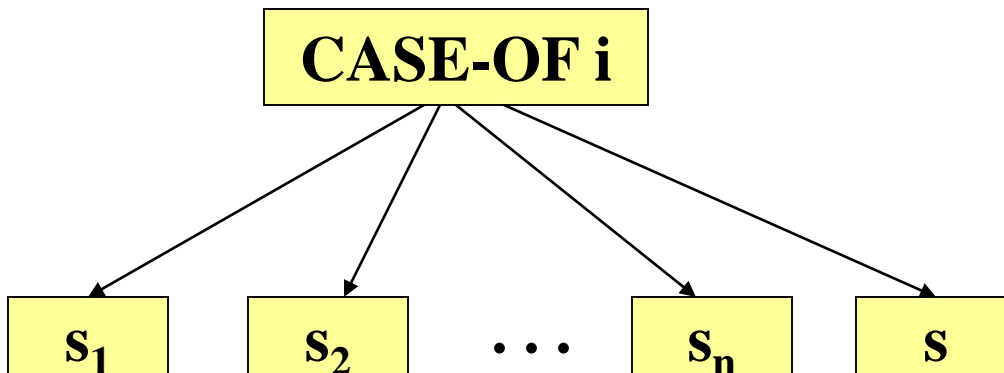
analitic

CASE-OF (i,s1,s2,...,sn,s)

pseudocod

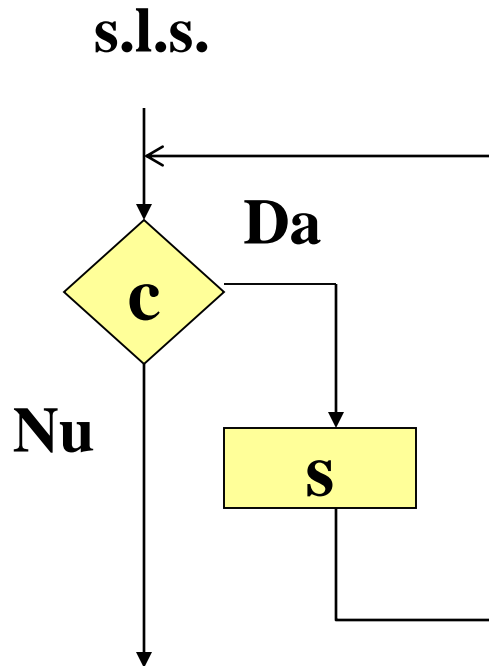
```
CASE-OF i  
    i=v1: s1 ;  
    i=v2: s2 ;  
    ...  
    i=vn: sn  
    ELSE s  
ENDCASE
```

arbore



Structurile repetitive

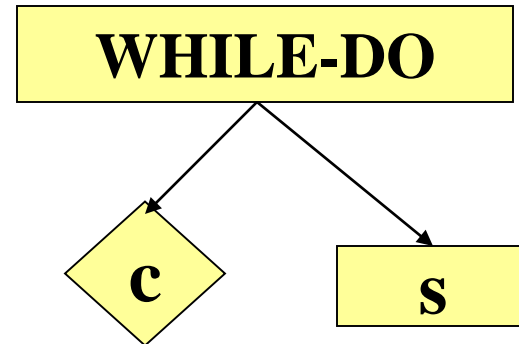
Structura repetitivă condiționată anterior



pseudocod

```
WHILE c DO  
  S  
ENDWHILE
```

arbore



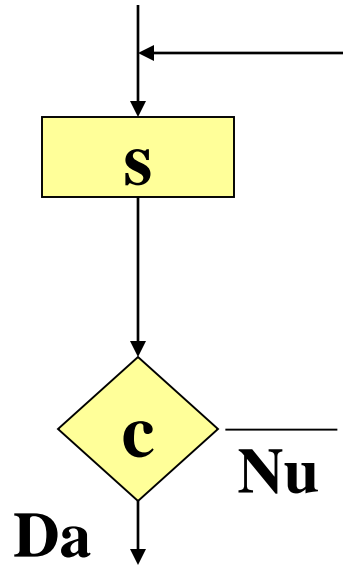
analitic

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

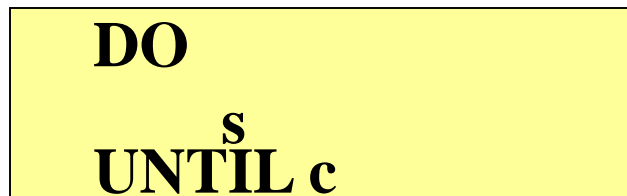
Structură PRIVILEGIATĂ !

Structura repetitivă condiționată posterior

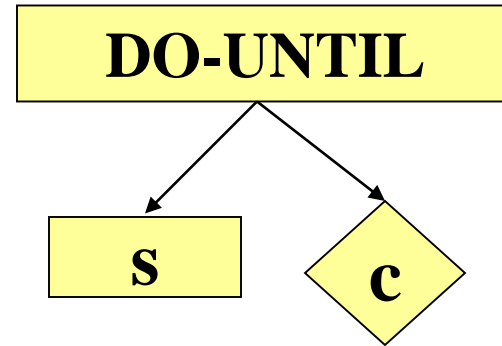
s.l.s.



pseudocod



arbore

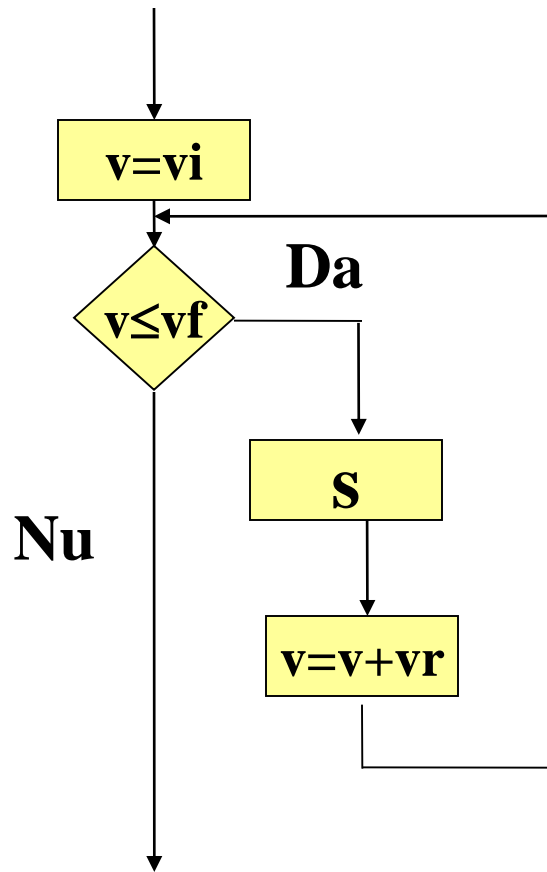


analitic

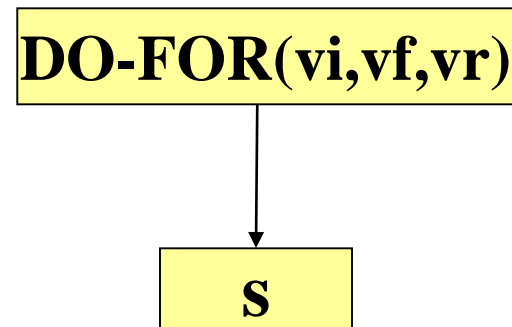


Structura repetitivă cu numărător

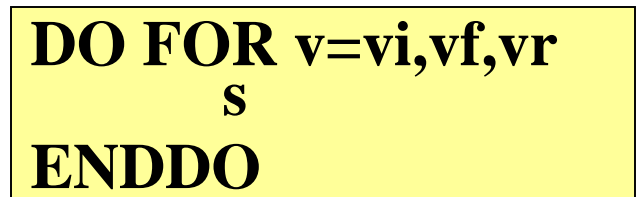
s.l.s.



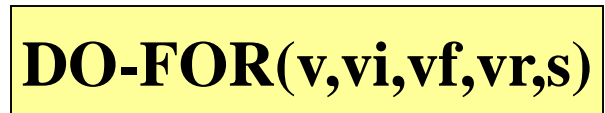
arbore



pseudocod



analitic



$$N = [(vf - vi) / vr] + 1$$

STRUCTURAREA ALGORITMILOR

Mulțimea structurilor privilegiate

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

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-Jacoppini)

- 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