

# **Limbaje formale si tehnici de compilare**

2020-2021

# Compiler

- Translator, compiler
- Compiler vs. interpreter

Translator: program

care traduce textul unui program, numit program sursa, intr-un alt program, numit program obiect.

Cele doua programe sunt echivalente din punct de vedere semantic.



Compiler: translator

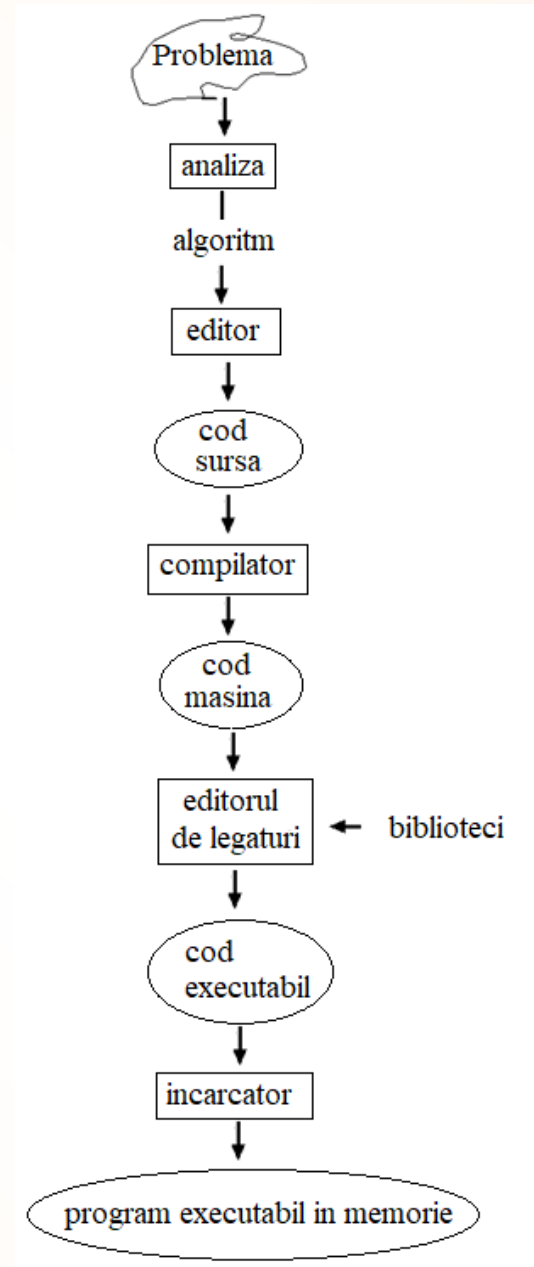
programul obiect este in cod (limbaj) masina  
(sau cod al unei masini virtuale)

- Ex.: tpc, bcc, gcc

**Translator. Compiler.**

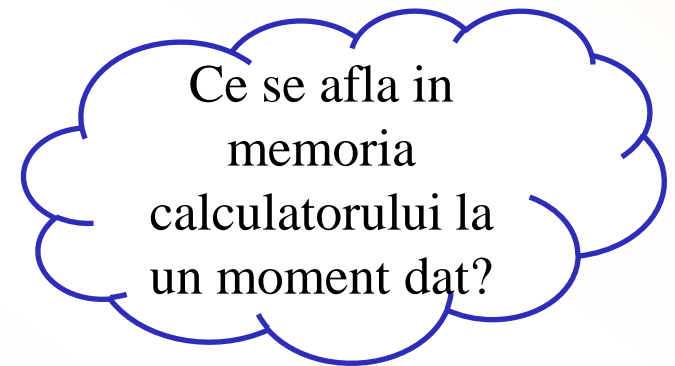
# Compiler

- Locul  
compilerului



# Compiler vs. interpreter

- Compiler
  - parcurește întregul program și îl traduce în cod de mașină
- Interpreter: (program)
  - la un moment dat
    - citește o singură declarație de program,
    - o traduce în cod mașină
    - o execută



Un compiler este mai rapid decât un interpreter.

# Limbaj de programare

- De ce e importanta specificarea limbajului de programare ?

Comunicare între:

- proiectantul limbajului,
- programatorii care scriu compilatoarele, ...
- programatorii care utilizeaza compilatoarele



- Ce specificăm?
  - Reguli referitoare la corectitudinea textului ca limbaj de programare (semantică statică)
  - Semanticii programelor (numită și semantică de rulare)
- Ce mecanisme se folosesc pentru specificare?
  - ...

# Limbaaj de programare

- limbaaj de programare  $\leftrightarrow$  compiler  
evolutia

specificare

limbaje: BNF si extinderi, ...

! limbaje formale

# Backus Naur Form

- primitive (terminale)
- variabile metalingvistice : intre paranteze unghiulare
- conective metalingvistice:  $::=$  , |

$\langle \text{intreg\_fara\_semn} \rangle ::= \langle \text{cifra} \rangle |$   
 $\langle \text{cifra} \rangle \langle \text{intreg\_fara\_semn} \rangle$

$\langle \text{cifra} \rangle ::= 0 | 1 | \dots | 9$

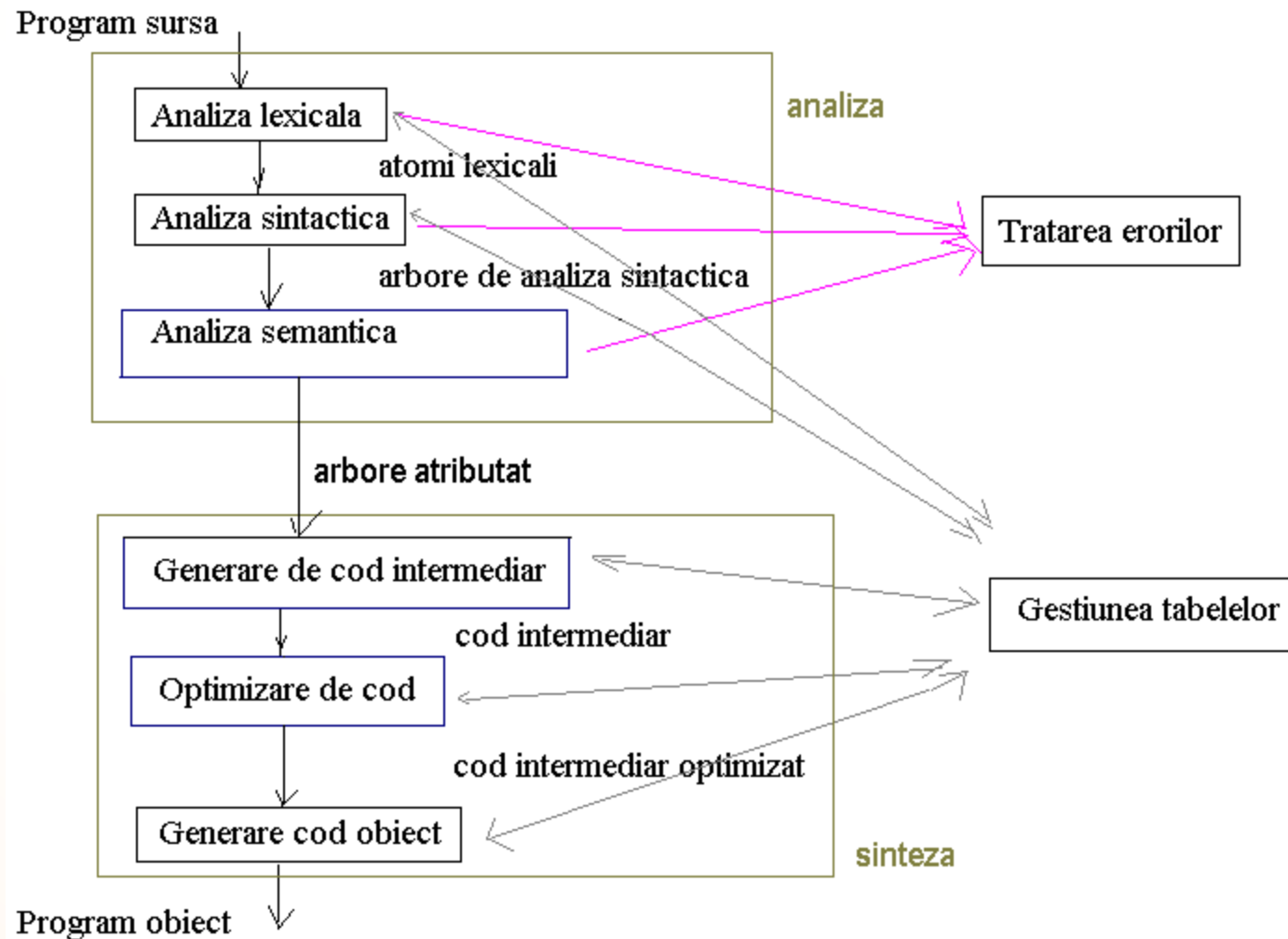


# Extended Backus Naur Form

- primitive (terminale): între “ghilimele”
- variabile metalingvistice
- conective metalingvistice:  $=$ ,  $|$ ,  $[ ]$ ,  $\{ \}$
- punctul este folosit ca sfarsit de regula

$\text{intreg\_fara\_semn} = \text{cifra} \{ \text{cifra} \} .$

$\text{cifra} = \text{“0”} | \text{“1”} | \dots | \text{“9”} .$



# Structura unui compilator

# Limbaje formale si tehnici de compilare. Continut

## Limbaj formal

Limbaj regulare:

- gramatici regulate,
- automate finite,
- expresii regulate.

Limbaje independente de context:

- Gramatici independente de context, automate push-down
- Gramatici speciale: LL(k), LR(k)

Gramatici de atribut

## Structura unui compilator

