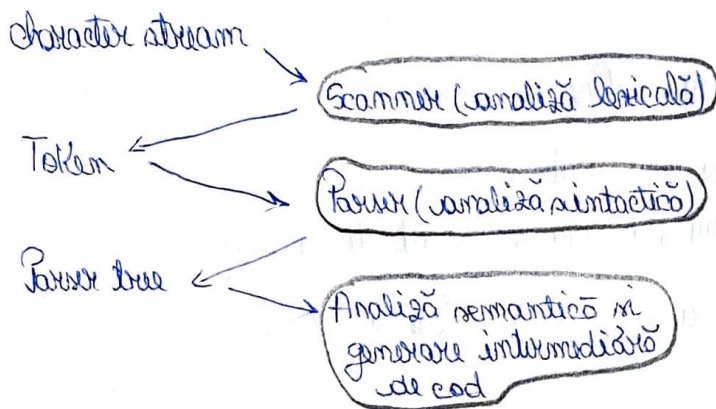


# TEORIE EXAMENE ANTERIOARE

- 9) Interpretorul face parte din execuția unui program: **DA**  
Compilatorul **NU** face parte din execuția unui program.

- 2) Etapa de analiză lexicală generează un șir de tokeni pentru: **analiza sintactică**



- 3) Care dintre algoritmi realizează parcurgere ascendentă:

LL  $\rightarrow$  **analiză sintactică descendentă**  
LR  $\rightarrow$  **analiză sintactică ascendentă**

- 4) Un arbore de parcurgere include în rădăcină: **SIMBOLUL DE START AL GRAMATICII**  
(derivare)  
RĂDĂCINA = program  
FRUNZELE = tokenii de la analizatorul lexical

- 5) Pentru orice automat stivă, există o gramatică regulată cu același limbaj.  
Pentru anumite automate stivă, există o gramatică regulată cu același limbaj.  
Pentru anumite automate stivă, există o gramatică independentă de context cu același limbaj.  
• Pentru orice automat stivă, există o gramatică independentă de context cu același limbaj.

- 6) La construirea automatului LL(K), orice stare e: **set de situații**  
**LR(K)  $\rightarrow$  o mulțime de stări**

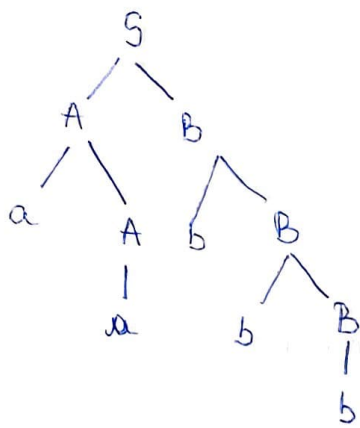
- 7) Orice gramatică regulată e și independentă de context și dependentă de context.

- 8) Algoritmul LR(K) are drept rezultat: **un automat stivă determinist pentru orice  $k$**

- 9) Parcurgere descendentă reconstruită: **derivarea ea mai din stânga**  
Parcurgere ascendentă reconstruită: **derivarea ea mai din dreapta**  
(LR)

- 10) Se dă un arbore de derivare.

Alegi răspunsurile corecte:



- a) a, A, b sunt terminale ale gramaticii  
 b) Stringul ab aparține limbajului gramaticii pe baza căruia s-a construit arborele de derivare.  
 c) Stringul aabbb aparține limbajului gramaticii  
 d) niciun răspuns  
 e) producția  $B \rightarrow bB$  aparține setului de producții
- $S \Rightarrow AB \Rightarrow ab$
- $P = \{ S \rightarrow AB, A \rightarrow aA \mid a, B \rightarrow bB \mid b \}$

11) În arborel parșării ascendente, stiva conține la fiecare moment:

(LR)

- a) stringul de la intrare care a fost deja recunoscut  
 b) stringul din care a fost derivat stringul deja recunoscut de la intrare  
 c) stringul din care se va începe derivarea stringului de la intrare  
 d) două simboluri nonterminale  
 e) două simboluri terminale  
 f) niciuna dintre variante

12) Următoarele procese fac parte din fazele unui compilator:

- a) analiză lexicală d) căutare  
 b) analiză comparativă e) generare cod  
 c) analiză sintactică f) reamare (reammer = analiză lexicală)  
 = parser

13) Care dintre algoritmii de parșare construiește la final rădăcina arborelui de parșare

- a) ascendentă (bottom-up) → construcția începe de la frunze  
 b) descendentă (top-down) → construcția începe de la rădăcină  
 c) ambele  
 d) niciunul

14) Pentru gramatici ambigue:

- a) pt. orice propoziție a limbajului, există mai mult de un arbore de parșare  
 b) pt. cel puțin o propoziție a limbajului, există mai mult de un arbore de parșare  
 c) există propoziții din limbaj pt. care nu există niciun arbore de parșare.

- 15) a) Pt. orice gramatică regulată, există un automat finit de stare cu același limbaj  
 b) Pt. anumite gramatici regulată, există un automat finit de stare cu același limbaj  
 c) Pt. orice gramatică regulată, nu există un automat finit de stare cu același limbaj  
 d) Pt. orice gramatică regulată, cel mai simplu automat este el stivă



17) Pentru orice produs din gramatică, există :  
 a) o singură situație ce o conține  
 b) mai multe situații ce o conține  
 c) nicio situație ce o conține

18) Automatul rezultat în urma aplicării LL(K):

- a) automat finit de stări
- b) automat activ
- c) automat stivă determinist pt orice K
- d) niciuna dintre variante

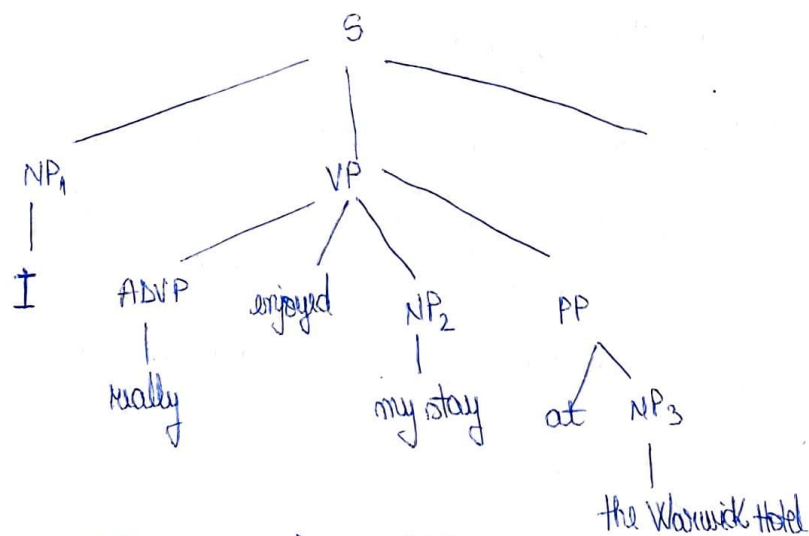
252

19) Acțiunile de shift reduce sunt specifice:

- a) parcurgere ascendentă
- b) derivare în gramatică regulată
- c) parcurgere descendentă
- d) analiză lexicală

LL(K)

20) Se da un arbore de parcurgere pentru care descendentul din dreapta nu se cuminte.



21) Simbolul de start al gramaticii e sigur S

b) producția  $S \rightarrow NPVP$  sigur face parte din gramatică

c) producția  $S \rightarrow NP VP$  sigur nu face parte din gramatică

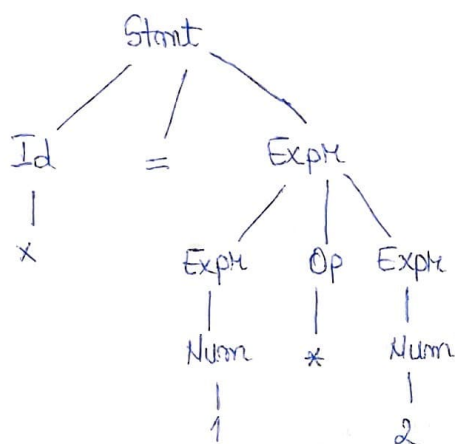
d) dacă știm că  $ADVP \rightarrow$  never aparține producțiilor gramaticii, propoziția I never enjoyed my stay aparține cu siguranță limbajului gramaticii

e) niciuna dintre variante

21) În codbul parcurut descendente, stiva conține la fiecare moment:

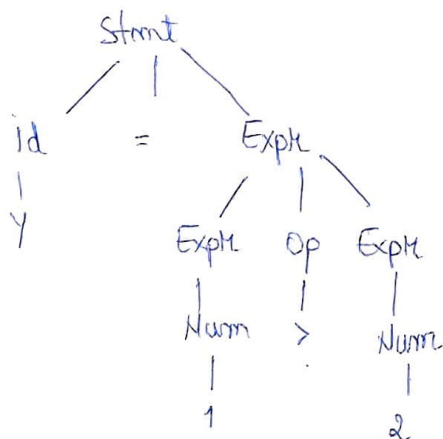
- stringul de la intrare care a fost deja recunoscut
- stringul din care a fost derivat stringul deja recunoscut la intrare
- stringul din care se va deriva derivarea stringului de la intrare
- doar simboluri nonterminale
- doar simboluri terminale
- mecanismul de intrare

22) Se dă arborele de parcurere construit pe baza unei gramatici  $G_1$ .



- Stringul  $x=1$  aparține rigur gramaticii  $G_1$ .
- Stringul  $x=1$  rigur nu aparține gramaticii  $G_1$ .
- $1 * 1$  este frază pentru Expr.
- $1 * 1$  este frază pt. Stmt
- gramatica  $G_1$  este rigur regulată

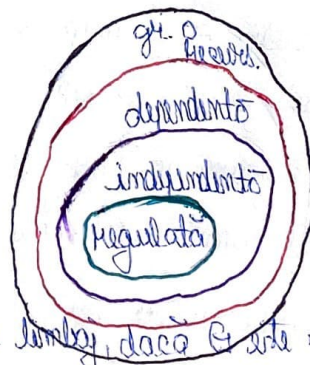
23) Se dă arborele de mai jos pe baza gramaticii  $G_2$  în el ambiguit.



- E posibil ca  $G_1$  și  $G_2$  să fie aceeași gramatică
- E posibil ca  $G_1$  și  $G_2$  să fie gramatici distincte
- Gramatica  $G_2$  este ambiguit
- E rigur că gramatica  $G_2$  nu e ambiguit.

24) Orice gramatică independentă de context e și:

- gramatică regulată
- gramatică dependentă de context
- gramatică recursivă



25) Pentru orice automat stivă există o gramatică  $G$  cu același limbaj, dacă  $G$  este:

- gramatică regulată
- gramatică dependentă de context
- gramatică recursivă

26) Dacă  $q_1, q_2$  sunt stări,  $t$  este un simbol, producția  $q_1 t \rightarrow q_2$  este o producție a:

- a) unui automat finit de stare
- b) unui automat stivă
- c) a unei gramatici regulate
- d) a unei gramatici independente de context

27) a) Orice gramatică strong  $LL(K)$  este și  $LL(K)$  pt. același  $K$ .

b) Orice gramatică  $LL(K)$  este și  $LR(K)$  pt. același  $K$ .

c) Orice gramatică  $LL(K)$  este și strong  $LL(K)$  pentru același  $K$ .

28) Pentru orice automat stivă se poate construi un automat finit de stare cu același limbaj:

a) adevărat

b) fals

? c) adevărat pt. anumite automate stivă, dar nu toate

d) fals pentru orice automat stivă

29) Pentru orice expresie regulată există o gramatică  $G$  cu același limbaj dacă:

a)  $G$  este regulată

b)  $G$  este independentă de context

c)  $G$  dependentă de context

30) Gramatica  $G = \{ \{0,1\}, A, A, (A \rightarrow 0A, A \rightarrow 1A, A \rightarrow 0) \}$  are limbajul egal cu limbajul expresiei regulate (scrieți în minternele lui):

a)  $01^+0$

b)  $[01]^*0$

c)  $[01]0$

d)  $(01)^+0$

e)  $[01]^+1$

31) Expresia regulată  $C[0-9]^+[a-z]^+$  include în limbajul ei stringul:

a) C120 ABC

b) C[1]A

c) C11000ZK

d) 0-9A

- nu e niciun răspuns



② Dați exemple de o producție dintr-o gramatică regulată.

$\mathcal{P} = \{c \rightarrow m\}$ , wobei  $c \rightarrow$  nonterminal  
 $m \rightarrow$  terminal

(33) Coste e limboajul descris de expresia regulată :  $1(1+0)^*(\epsilon+11)$

74) Cum este limbajul  $\{(), ()(), ()()\}$  definit de o gramatică regulată, independentă de context, dependentă de context, de tip 0 (conform ierarhiei lui Chomsky)?

1.  $S \rightarrow (S)$

2.  $S \rightarrow T$

3.  $T \rightarrow ()$

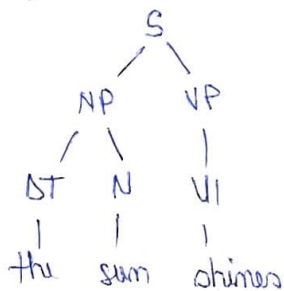
4.  $T \rightarrow T, T$

5.  $T \rightarrow TT$

6.  $T \rightarrow (T)$

$$\begin{aligned} 5 &\xRightarrow{1} (5) \xRightarrow{2} (T) \xRightarrow{4} (T, T) \xRightarrow{4} (T, T, T) \xRightarrow{3} \\ &\xRightarrow{3} ((), T, T) \xRightarrow{5} ((), TT, T) \Rightarrow^* ((), ()(), T) \xRightarrow{6} \\ &\xRightarrow{6} ((), ()(), (T)) \xRightarrow{3} ((), ()(), (())) \end{aligned}$$

35) Menționați două fraze din textul de derivat, menționând în paranteză cînd sunt frazele aliv



"the sum"  $\rightarrow$  NP

"the sum chimera"  $\rightarrow S$

36) Două automate finite  $A, A'$  sunt echivalente dacă  $L(A) = L(A')$  (A) F

It. orice gramatică regulată  $G$  există un automat finit determinist a. î.  $L(A) = L(G)$  A F

Există gramatici independente de context care nu sunt gramatici regulate. (A) F

Robotul poate să uite recombinația dintr-unu primu casu o CFG poate genera un string de intrare. (A) F

Un automat stivă este determinist dacă parțile dreapta ale tuturor produselor sunt distincte. A (F)