$\mathcal{LFA}$: C2 – Ierarhia Chomsky

1. Alfabet, cuvant, operatii cu cuvinte
2. Limbaj, operatii cu limbaje
3. Gramatica; exemple
4. Clasificarea gramaticilor generative; ierarhia lui Chomsky

LFA: C2 – Ierarhia Chomsky

Notiunea de alfabet: aceeasi semnificatie pentru

- limbile naturale
- informatica, in general
- teoria algoritmilor (prelucrarea algoritmica a informatiei), in particular**:**

  <u>*mijloc de comunicare*</u>:

  - *intre oameni,*
  - *intre om si calculator*
  - *intre oameni prin intermediul calculatorului,*
  - *intre calculatoare.*

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitie 1

Alfabet = Σ = orice multime finita, nevida

Elementele = **simboluri**

Exemple 2

Σ_{bool} = {0, 1},

Σ_{num} = {I,V,X,L,C,D,M}

Σ_{latin} = {a,b,c,…,z},

Σ_{ADN}= {A,C,G,T} --> lanturile de nucleotide ale ADN-ului

Σ_{logic} = {0,1,(,), $\neg,\wedge,\vee$,p, q, r, …} sau

{ 0,1,(,), $\neg,\wedge,\vee,\chi$ }!?!?

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

<u>Definitie 3</u>

Cuvant peste un alfabet Σ = orice secventa finita de simboluri din Σ

Cuvantul vid = ε = singurul cuvant care consta din 0 simboluri

$\boldsymbol{\Sigma^*}$ = multimea tuturor cuvintelor peste alfabetul $\boldsymbol{\Sigma}$

$\boldsymbol{\Sigma^+} = \boldsymbol{\Sigma^*} - \{\varepsilon\}$

<u>Observatie 4</u>

Nu orice cuvant peste un alfabet reprezinta o notiune:

$\Sigma_{latin}{}^{\boldsymbol{*}}$={$\varepsilon$,a,…,aa,ab,ac,…,aaa,aab,…,caa,cab,cal,..,cla,…, lac,…,lca,…, horse,…, cheval,…}.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitie 5

Lungimea unui cuvant *w* peste un alfabet Σ = |*w*| =

= numarul de simboluri din ***w***

$|\varepsilon| = 0$

Notatie 6

$\#_s(w)$ = numarul de aparitii ale simb. $s \in \Sigma$ in cuvantul $w \in \Sigma^*$

Exemplu 7

$\#_t$(complexitate)= 2

Observatie 8

$\forall \Sigma, \forall w \in \Sigma^*$: $$|w| = \sum_{s \in \Sigma} \#_s(w).$$

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitie 9

Fie un alfabet $\Sigma = \{s_1, s_2, \ldots, s_k\}$, $k \geq 2$;

se numeste **functia lui Parikh,** functia

$\psi_\Sigma : \Sigma^* \rightarrow \mathcal{N}^K$, $\psi_\Sigma(\omega) = (|\omega|_{s1}, |\omega|_{s2}, \ldots, |\omega|_{sk}) = (\#_{s1}|\omega|, \#_{s2}|\omega|, \ldots, \#_{sk}|\omega|)$

Exemplu 10

Fie $\Sigma = \{a, b, \ldots, z\}$; atunci $\psi_\Sigma : \Sigma^* \rightarrow \mathcal{N}^{27}$,

ψ_Σ (Constantinopol) = (1,0,1,0,0,0,0,0,1,0,0,1,0,3,3,1,0,1,2,0,0,0,0,0)

Definitie 11

Fie un alfabet Σ; atunci, pentru orice $n \in N$:

$\Sigma^n = \{ \omega \in \Sigma^* \mid |\omega| = n \}$ şi $|\Sigma^n| = |\Sigma|^n$

Exemplu 12

$\{0,1\}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$ => 8 cuvinte

$\{0,1\}^5 = \{00000, 00001, 00010, 00011, 00100, 00101, 00110, \ldots, 11111\}$.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitii 13: Operatii cu cuvinte

(i) Fie un cuvant w peste un alfabet Σ; notam prin $mi(w)$ sau $w^{\mathcal{R}}$ **reversul** cuvantului w, adica un cuvant din Σ^* obtinut din w prin scrierea simbolurilor acestuia in ordine inversa,

(ii) Fie doua cuvinte v si w peste un alfabet Σ; notam prin vw sau $v.w$ cuvantul din Σ^* obtinut prin **concatenarea** lui v cu w;

Exemple 14

w = 856 => $w^{\mathcal{R}}$ = 658; w = capac => $w^{\mathcal{R}}$ = capac,

v = ori, w = cand => vw = oricand, wv = candori;

Observatie 15

1. In general, $vw \neq wv$ dar intotdeauna: $|vw| = |wv| = |v|+|w|$
2. $\forall\Sigma$: (Σ^*,**.**) este un monoid (ε=elementul neutru).
3. $\forall w \in \Sigma^*$, definim: $\begin{cases} w^0 = \varepsilon, si \\ w^{n+1} = w \cdot w^n = w^n \cdot w, \forall n \in N. \end{cases}$

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitie 16

Prefix al unui cuvant $w \in \Sigma^*$ **:** $\forall v \in \Sigma^*$: $\exists x \in \Sigma^*$ a.i. $w = vx$.

Sufix al unui cuvant $w \in \Sigma^*$ **:** $\forall v \in \Sigma^*$: $\exists y \in \Sigma^*$ a.i. $w = yv$.

Subcuvant al unui cuvant $w \in \Sigma^*$ **:** $\forall v \in \Sigma^*$: $\exists x, y \in \Sigma^*$ a.i. $w = xvy$.

Observatie 17

x si/sau y pot fi si ε .

LFA: C2 – Ierarhia Chomsky

<u>Exemple 18</u>

w = întrucâtva =>

$\text{prefix}_w \in$ {ε, î, în, înt, într, întru, întruc, întrucâ, întrucâtv, întrucâtva}

$\text{sufix}_w \in$ {ε, a, va, tva, âtva, câtva, ucâtva, rucâtva, trucâtva, ntrucâtva, întrucâtva}

$\text{subcuvant}_w \in$ {ε, î, n, t, r, u, c, â, v, în, nt, tr, …, întru, …, câtva, …, întrucâtva}

<u>Observatie 19</u>

Fie w_1 = cucuta => cu = prefix_{w1} si cu = subcuvant_{w1}

Fie w_2 = curara => ra = sufix_{w2} si ra = subcuvant_{w2} .

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

similara ordinii cuvintelor in dictionar; exceptie: cuvintele scurte preced cuvintele lungi

Definitia 20

Fie un alfabet $\Sigma=\{s_1,s_2,\ldots,s_m\}$, $m\geq 1$ si

fie $s_1<s_2<\ldots<s_m$ o ordine pe Σ;

ordinea canonica (lexicografica) pe Σ^* se defineste astfel:

$\forall\ v,w \in \Sigma^*$: **$v<w$** daca $|v|<|w|$ sau

$|v|=|w|$ si $\exists\ i,j\in N, 1\leq i<j\leq m$ si $\exists\ x,v',w'\in\Sigma^*$

astfel incat $v=xs_iv'$ si $w=xs_jw'$.

Simbolul cu care incep v', respectiv w' nu conteaza.

LFA: C2 – Ierarhia Chomsky

Observatie 21

Fie Σ = alfabetul latin; atunci:

capac < capsa < capsat < captat =captat < cazuta (a<s; |capsa| < |capsat| ; p<z etc.)

Observatie 22

Ordinea canonica permite enumerarea “tuturor” cuvintelor peste orice alfabet:

fie Σ={a,b} →

Σ* ={ε,a,b,aa,ab,ba,bb,aaa,aab,aba,abb,..., abbaababbabb, ...}.

LFA: C2 – Ierarhia Chomsky

1. Alfabet, cuvant, operatii cu cuvinte
2. Limbaj, operatii cu limbaje
3. Gramatica; exemple
4. Clasificarea gramaticilor generative; ierarhia lui Chomsky

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitie 23

Fie un alfabet Σ;

- ✓ se numeste **limbaj** peste Σ, orice submultime $L \subseteq \Sigma^*$
- ✓ se numeste **limbaj ε-liber** peste Σ, orice submultime $L \subseteq \Sigma^+$

Exemple 24

1. Fie $\Sigma = \{a,b\}$ =>

   $\emptyset$, $\{\varepsilon\}$, $\{a,b\}$, $\{a,b\}^*$, $\{ab, bba, b^{10}a^{20}, abbaba\}$, $\{a^n b^{2n} | n \in \mathcal{N}\}$, $\{aw | w \in \{a,b\}^*\}$, $\{aw | w \in \{b\}^*\}$

Notatie 25

Multimea tuturor limbajelor peste alfabetul Σ:

$\mathcal{L}_{\Sigma} = \{ L \subseteq \Sigma^* \mid L = \text{limbaj} \}$.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Observatie 26

Fie multimile A, M si $A \subseteq M$;

Se numeste functie caracteristica a multimii A in multimea M

functia λ_A: $M \rightarrow \{0,1\}$ astfel: $\lambda_M(x)=\begin{cases} 1, & x \in A \\ 0, & x \in M - A \end{cases}$

Definitie 27

Fie un alfabet $\Sigma=\{s_1,s_2,\ldots,s_m\}$, $m\geq 1$ si

$\Sigma^*=\{x_1,x_2,\ldots,x_n,\ldots\}$ enumerarea cuvintelor peste Σ, indusa de ordinea canonica;

atunci, $\forall L\subseteq\Sigma^* \Rightarrow \exists!$ o secventa binara infinita, notata λ_L, definita astfel:

cel de-al i-lea bit din λ_L este: 1, daca $x_i\in L$,

0, daca $x_i\notin L$.

λ_L se numeste **secventa caracteristica a limbajului L** peste $\Sigma=\{s_1,s_2,\ldots,s_m\}$.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

λ_L = 0101011011011100010100100101110….
∈ $\mathcal{B}$ = multimea secventelor binare infinite

Exemple 28

1. Fie ∑ = {a,b} şi L = {a, ab, abb}

=> ∑*= {ε,a,b,aa,ab,ba,bb,aaa,aab,aba,abb,baa,..., aaaa, aaab, aaba,...}

L = { , a, , , ab, , , , , , abb }

=> λ_L= 0 1 0 0 1 0 0 0 0 0 1 0 0 0 ,...= 0100100000100....

2. Fie L = {w ∈ ∑*| ∃y∈∑*: w=aay}

L = { , , , aa, , , , aaa,aab, , ,..., aaaa, aaab, aaba,.. }

=> λ_L= 0 0 0 1 0 0 0 1 1 0 0 0 ,..., 1 1 1…..=

= 000100011000….111....

3. Fie L = {w ∈ ∑*| ∃x∈∑*:w=bx}

L = { , ,b, , ,ba,bb, , , , ,baa,bab,bba,bbb,...}

=> λ_L= 0 010 0 1 1 0000 1 1 1 1 ….. = 00100110000 1111....

4. Fie L = {w ∈ ∑*| w= palindrom} =

= {ε,a,b,aa,bb,aaa,aba,bab,bbb,aaaa,….}

=> λ_L= 111100110100 …

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Teorema 29

Multimea $\mathcal{L} = \{ L \subseteq \Sigma^* \mid L = \text{limbaj} \}$ este nenumarabila

Demonstratie

(i) multimea $\mathcal{B}$ a secventelor binare infinite este nenumarabila

Folosim metoda diagonalizarii şi a p.p.a.

ppa $\exists$ f:N → $\mathcal{B}$, bijectiva a.i. $f(n)=b_n \in \mathcal{B}$ →

n	$f(n)=b_n$
1	100…
2	010…
3	110…
4	001…
…	…

putem construi o secventa binara ***b*** astfel:

a n^a cifra binara din b este:

0, daca a n-a cifra binara din f(n) este 1,

1, daca a n-a cifra binara din f(n) este 0

=> b=00111......

=> $b \neq f(n), \forall n \in N$:

=> $\mathcal{B}$ este nenumarabila.

n	$f(n)=b_n$
1	<u>1</u>0000…
2	0<u>1</u>000…
3	11<u>0</u>00…
4	001<u>0</u>0…
5	1010<u>0</u>…
…	…

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

(ii) multimea $\mathcal{L}$ = {L$\subseteq\Sigma$ | L = limbaj} este nenumarabila*

E suficient sa gasim f: $\mathcal{L} \rightarrow \mathcal{B}$, bijectiva

ori, $\exists$ f: $\mathcal{L} \rightarrow \mathcal{B}$: f(L)= λ_L

şi, evident, f = bijectiva;

cf. (i) $\mathcal{B}$ =nenumarabila => $\mathcal{L}$ nenumarabila.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitii 30: Operatii cu limbaje

Fie limbajul L $\subseteq\Sigma$; definim şi notam prin:

$mi(L)=L^R=\{mi(v) \mid v\in L\}$

reversul limbajului L in raport cu Σ;

$L^C=\{v\in\Sigma^* \mid v\notin L\}$

complementul limbajului L in raport cu Σ;

Fie limbajele $L_1\subseteq\Sigma_1^*$ şi $L_2\subseteq\Sigma_2^*$ (Σ_1 şi Σ_2 oarecare); definim şi notam prin:

$L_1\cup L_2=\{v\in(\Sigma_1\cup\Sigma_2)^* \mid v\in L_1 \text{ sau } v\in L_2\}$

reuniunea limbajelor L_1 şi L_2;

$L_1\cap L_2=\{v\in(\Sigma_1\cap\Sigma_2)^* \mid v\in L_1 \text{ şi } v\in L_2\}$

intersectia limbajelor L_1 şi L_2;

$L_1 - L_2=\{v\in\Sigma_1^* \mid v\in L_1 \text{ şi } v\notin L_2\}$

diferenta limbajelor L_1 şi L_2.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitii30: Operatii cu limbaje (cont.)

Fie doua alfabete Σ_1 si Σ_2 si doua limbaje $L_1 \subseteq \Sigma_1^*$ si $L_2 \subseteq \Sigma_2^*$; definim si notam :

$L_1 L_2 = L_1 \circ L_2 = \{vw \in (\Sigma_1 \cup \Sigma_2)^* \mid v \in L_1 \text{ si } w \in L_2\}$

limbajul obtinut prin **concatenarea** (**produsul**) limbajelor L_1 şi L_2,

$$L^* = \bigcup_{n \geq 0} L^n, \quad unde \quad L^0 = \{\varepsilon\} \quad si \quad L^{n+1} = L \cdot L^n, \quad \forall n \in N$$

inchiderea reflexiva şi tranzitiva (Kleene) a limbajului L,

$$L^+ = \bigcup_{n \geq 1} L^n, \quad obs.: \quad L^* = L^+ \cup \{\varepsilon\}$$

inchiderea tranzitiva a limbajului L;

Fie alfabetul Σ si doua limbaje $L_1, L_2 \subseteq \Sigma^*$; definim si notam :

$L_1 / L_2 = \{w \in \Sigma^* \mid \exists\, v \in L_2 : wv \in L_1\}$

câtul la dreapta al limbajului L_1 prin limbajul L_2 ,

$L_1 \setminus L_2 = \{w \in \Sigma^* \mid \exists\, v \in L_2 : vw \in L_1\}$

câtul la stanga al limbajului L_1 prin limbajul L_2 .

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitii 30: Operatii cu limbaje (cont.)

(i) Fie doua alfabete Σ si Ψ; se numeste **substitutie** o functie

$s : \Sigma \rightarrow \mathcal{P}(\Psi^*)$

Extindem aceasta aplicatie la Σ^* prin

$s(\varepsilon) = \{\varepsilon\}$,

$s(a\beta) = s(a)s(\beta), \quad \forall a \in \Sigma, \quad \forall \beta \in \Sigma^*$

Obs. aceasta extensie este canonica :

daca $w = \alpha\beta \in \Sigma^*$, atunci $s(w) = s(\alpha)s(\beta)$, $s(\alpha), s(\beta) \subseteq \Psi^*$

(ii) Fie un limbaj $L \subseteq \Sigma^*$; atunci definim prin:

$$s(L) = \bigcup_{\alpha \in L} s(\alpha)$$

limbajul obtinut din L prin **substitutie canonica**

Ex.: fie $s:\{a,b\} \rightarrow \mathcal{P}(\{0,1,x\}^*)$ $s(a)= \{0x\}$, $s(b)=\{x11\}$

daca $L=\{a,b, aa, ab, ba, bb\}$ =>

$s(L) = \{0x, x11, 0x0x, 0xx11, x110x, x11x11\}$.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitii 30: Operatii cu limbaje (cont.)

O substitutie $s : \Sigma^* \to \mathcal{P}(\Psi^*)$ se numeste

finita:

$card\ (s(a)) < \infty,\ \forall a \in \Sigma$

(orice simbol din Σ este substituit de un limbaj peste Ψ, finit)

[**omo**]**morfism**:

$card\ (s(a)) = 1,\ \forall a \in \Sigma$

(multimea s(a) este singleton)

substitutie / morfism ε-free:

$\varepsilon \notin s(a)\ ,\ \forall a \in \Sigma\ .$

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitii 30: Operatii cu limbaje (cont.)

Ex.: fie $s^{-1} : \{0,1,x\}^* \rightarrow \mathcal{P}(\{a,b\}^*)$ $s^{-1}(0x)=\{a\}$, $s^{-1}(x11)=\{b\}$

unde $s : \{a,b\} \rightarrow \mathcal{P}(\{0,1,x\}^*)$ $s(a)= \{0x\}$, $s(b)=\{x11\}$

daca $L = \{0x, x11, 0x0x, 0xx11, x110x, 0x0x0x\}$

=> $s^{-1}(L) = \{a, b, aa, ab, ba, aaa\}$;

Observatie 31: $\varnothing$ şi $\{\varepsilon\}$

1. $L \circ \varnothing = \varnothing \circ L = L, \forall L$
2. $L \circ \{\varepsilon\} = \{\varepsilon\} \circ L = L, \forall L$
3. $L \cup \varnothing = \{\varepsilon\} \cup L = L, \forall L$
4. $L \cap \varnothing = \varnothing, \forall L$
5. $L \cap \{\varepsilon\} = \{\varepsilon\}, \forall L.$

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Cum caracterizam formal un limbaj L?

printr-o reprezentare **finita** a tuturor secventelor sale

Mai multe metode:

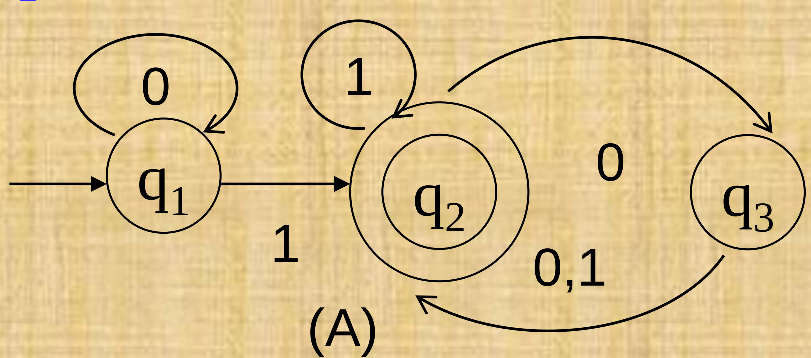
- ✓enumerarea tuturor elementelor limbajului
- ✓enuntarea proprietatilor distinctive ale elementelor sale
- ✓definirea unei gramatici generative G (∈ clasa procedeelor generative)

  $L_1 = \{\varepsilon, 01, 0^n1^n, 0^n101^n, 0^n(10)^k1^n, \ldots\} \Rightarrow$

  $G=(\{S\}, \{0,1\}, \{S \rightarrow 0S1 \mid SS \mid \varepsilon\}, S)$

- ✓definirea unui automat A (∈ clasa procedeelor analitice)

  $L_2 = \{w \in \{0,1\}^* \mid w=\alpha 1 \text{ sau } w=\alpha 00, \alpha \in \{0,1\}^*\} \Rightarrow$



- ✓etc.

LFA: C2 – Ierarhia Chomsky

1. Alfabet, cuvant, operatii cu cuvinte
2. Limbaj, operatii cu limbaje
3. Gramatica; exemple
4. Clasificarea gramaticilor generative; ierarhia lui Chomsky

LFA: C2 – Ierarhia Chomsky

Gramaticile

- ✓ initial: notiune introdusa de lingvisti pentru studierea limbajelor naturale (Noam CHOMSKY, ‘1950):
  - caracterizarea frazelor corecte dintr-un limbaj,
  - o definitie structurala a frazelor corecte dintr-un limbaj;
- ✓ ulterior: un instrument de reprezentare finita, generativa a oricarui limbaj, constand din:
  - o multime finita de elemente de baza,
  - un set finit de reguli de producere a frazelor corecte (sintactic) din limbaj.

LFA: C2 – Ierarhia Chomsky

Gramaticile: puterea generativă =

= un atribut al gramaticii care îi permite să producă elemente de limbaj.

N. CHOMSKY,("*Three models for the description of language*," vol. 2, no. 3, pp. 113-124, 1956) :

> “Copiii, după ce învață un număr relativ mic de elemente ale unui limbaj natural (*linguist’s corpus*), dobândesc aptitudinea de a înţelege şi produce un număr infinit de propoziţii diferite și corecte. ‘’

Ex.: (aproape) orice vorbitor nativ al limbii române, va înțelege “la prima vedere” propoziția

> Romania este a sasea tara din CE, in functie de numarul de locuitori.

și va respinge ca incorectă propoziția:

> Televizorul aleargă în parc cu zmeură.

Ex.: un “vorbitor nativ” al unui limbaj de programare (programator/compilator!!) va face imediat diferența între propozițiile:

> for (i = 15; i < 175433299; i++)   for (i = 0; i < ; i++) .

LFA: C2 – Ierarhia Chomsky

Gramaticile: puterea generativă’

= creativitatea G =expresivitatea G =
un atribut al G care îi permite să producă elemente de limbaj

- capacitatea (puterea) generativă **slabă** a gramaticii =
  = limbajul formal generat
- capacitatea generativă **tare** =
  = mulţimea combinărilor structurale utilizate pentru a produce sau recunoaşte propozițiile limbajului;

Observatii

1. Din punct de vedere lingvistic, descrierile structurale ale gramaticilor (capacitatea generativă tare): mult mai interesante decât simpla enumerarea a propozițiilor bine formate (capacitatea generativă slabă).

LFA: C2 – Ierarhia Chomsky

Definitie 32

Se numeste **gramatica** un sistem **G= (V_T,V_N,S,P)** unde:

- V_T = multime finita, nevida (*simboluri terminale*),
- V_N = multime finita, nevida (*simboluri neterminale=variabile*):

  $V_T \cap V_N = \emptyset$; $V_T \cup V_N = V$

  (**vocabularul** terminalelor şi neterminalelor gramaticii!);
- $S \in V_N$;= *simbolul de start* (*axioma* gramaticii),
- P = multime finita, nevida (*productii*):

  $P \subseteq (V_N \cup V_T)^* V_N (V_N \cup V_T)^* \times (V_N \cup V_T)^*$

  OBS. : ($(\alpha,\beta) \in P \equiv \alpha \rightarrow \beta$: α se inlocuieste cu β) =>

  $P = \{\alpha \rightarrow \beta \mid \alpha \in V^* \cdot V_N \cdot V^*;\ \beta \in V^*\}$.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

$P \subseteq (V_N \cup V_T)^* V_N (V_N \cup V_T)^* \times (V_N \cup V_T)^*$

$((\alpha,\beta) \in P \equiv \alpha \rightarrow \beta$: α se inlocuieste cu β) =>P = $\{\alpha \rightarrow \beta \mid \alpha \in V^* \cdot V_N \cdot V^*;\ \beta \in V^*\}$.

<u>Exemple 33</u>: G= (V_T, V_N, S, P)

1. G_1=({0,1,2,…,9}, {S,C}, S, {S→CC, C→0, C→1, C→2, C→3, C→4, C→5, C→6, C→7, C→8, C→9, C→ ε}) =>

   S→CC→0C→01

   S→CC→7C→70

   S→CC→5C→52 etc.

   Obs.: $L_1 \neq \{n \in \mathcal{N} \mid n<100\}$;

   => $L_1=\{xy \mid x,y \in\{0,1,2,\ldots,9\}\}$;

=> G_1'= ({0,1,2,…,9}, {S,A,C}, S, {S→AC, A→1, …, A→9, A→ ε, C→0, C→1, C→2, C→3, C→4, C→5, C→6, C→7, C→8, C→9}) =>

   S→AC→C→0

   S→AC→C→5

   S→AC→7C→70

   S→AC→5C→52 etc.

   => $L_1' = \{n \in \mathcal{N} \mid n<100\}$.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Exemple 34: $G= (V_T,V_N,S,P)$

2. $G_2=(\{0,1,2,\dots,9\}, \{S,B,C\}, S, \{S\rightarrow CB, B\rightarrow CB|C, C\rightarrow 0|1|\dots|9|\varepsilon\}) \Rightarrow$

$S\rightarrow CB \rightarrow CCB \rightarrow \dots \rightarrow C^nB \rightarrow C^{n+1}\rightarrow 0C^n \rightarrow 04C^{n-1} \rightarrow \dots\ 0409194\dots 7$

Obs.: $L_2 \neq \mathcal{N}$;

$\Rightarrow G_2'=(\{0,1,2,\dots,9\}, \{S,A,B,C\}, S, \{S\rightarrow AC|ACB, B\rightarrow CB|C, A\rightarrow 1, \dots, A\rightarrow 9,$

$C\rightarrow 0|1|\dots|9|\varepsilon,\}) \Rightarrow$

$S\rightarrow AC\rightarrow A\rightarrow 1, S\rightarrow AC\rightarrow A\rightarrow 2,\dots, S\rightarrow AC\rightarrow A \rightarrow 9, S\rightarrow AC\rightarrow 1C\rightarrow 10,$

$S\rightarrow AC\rightarrow 1C\rightarrow 12,\dots, S\rightarrow AC\rightarrow 1C\rightarrow 19,\dots,$

$S\rightarrow AC\rightarrow 2C\rightarrow 20,\dots, S\rightarrow AC\rightarrow 9C\rightarrow 99,$

$S\rightarrow ACB\rightarrow ACCB\rightarrow \dots ACB^nB\rightarrow AC^{n+1}\rightarrow 1C^{n+1}\rightarrow 12C^n\rightarrow 124C^{n-1} \rightarrow \dots$

$12409194\dots 7$

$\Rightarrow L_2=\mathcal{N}$;

LFA: C2 – Ierarhia Chomsky

<u>Exemple 35</u>: G= (V_T,V_N,S,P)

3. G_3=({I,V,X,L,C,D,M}, {S,A,B}, S, {S→AB, B→AB, B→A, A→I|V|X|L|C|D|M|ε})

S→AB →AAB →AAAB →AAAA→MMXV

=> L_3 = multimea numerelor naturale in grafia latina (fara respectarea regulilor de tipul: 4 se fomeaza ca IV şi nu ca IIII; 100 este C şi nu LL).

LFA: C2 – Ierarhia Chomsky

G_2=({0,1,2,…,9}, {S,C,B}, S,
{S→CB, B→CB, B→C, C→0|1|...|9|ε})

=> Cum procedam pentru a descrie un limbaj cu ajutorul unei gramatici generative?

Generam fiecare cuvant din limbaj dupa urmatorul algoritm:

1. Scriem simbolul de start (apare in m. stg. al primei productii din P şi este notat cu S (de obicei)),
2. Alegem una dintre productiile care au acest simbol in m. stg. şi inlocuim simbolul ales cu m. dr. al respectivei productii,

$S \Rightarrow^1 CB \Rightarrow^2 CCB$ sau $S \Rightarrow^1 CB \Rightarrow^3 CC$ sau $S \Rightarrow^1 CB \Rightarrow^4 7B$

3. Repetam Pasul 2 pana cand in m.dr. nu mai exista neterminale care pot fi inlocuite

$S \Rightarrow^1 CB \Rightarrow^2 CCB \Rightarrow^2 CCCB \Rightarrow^3 CCCC \Rightarrow^4 CCC9 \Rightarrow^4 CC09 \Rightarrow^4 0C09 \Rightarrow^4 0509;$

Observatie 36

La fiecare pas de calcul, se aplica o singura productie, unui singur neterminal.

Fie doua alfabete Σ si Ψ; se numeste **substitutie** o functie $s : \Sigma \rightarrow \mathcal{P}(\Psi^*)$
Extindem aceasta aplicatie la Σ^* prin
$s(\varepsilon) = \{\varepsilon\}$,
$s(a\beta) = s(a)s(\beta)$, $\forall a \in \Sigma$, $\forall \beta \in \Sigma^*$

LFA: C2 – Ierarhia Chomsky

Notatie: $\Rightarrow$

Substitutie = Derivare directa

Observatie 37

$$\underbrace{S \Rightarrow CB \Rightarrow CCB \Rightarrow CCCB \Rightarrow CCCC \Rightarrow CCC9 \Rightarrow CC09 \Rightarrow 1C09 \Rightarrow 1109}_{S \Rightarrow^* 1109}$$

Notatie: $\Rightarrow^*$

Inchiderea tranzitiva a derivarii directe = Derivare

Definitii 38

Se numeste **substitutie = derivare directa** = aplicarea unei productii =

daca $\mathbf{A \rightarrow \beta} \in P$ şi $\delta, \gamma \in V^*$, atunci $\delta \mathbf{A} \gamma \Rightarrow \delta \boldsymbol{\beta} \gamma$

Se numeste **derivare** = aplicarea consecutiva a mai multor productii =

daca $\alpha_1 \Rightarrow \alpha_2, \alpha_2 \Rightarrow \alpha_3, \ldots, \alpha_{n-1} \Rightarrow \alpha_n$ atunci $\boldsymbol{\alpha_1 \Rightarrow^* \alpha_n}$.

LFA: C2 – Ierarhia Chomsky

<u>Definitie 39</u>

Se numeste **limbaj generat de o gramatica $G=(V_T,V_N,S,P)$** multimea

$L(G) = \{ \omega \in V_T^* \mid S \Rightarrow^* \omega \}$;

<u>Observatie 40</u>

Pentru ca o secventa de simboluri ω sa faca parte limbajul L generat de gramatica G, ea trebuie sa indeplineasca 2 conditii:

1. sa fie formata numai din simboluri terminale şi care sa provina din vocabularul de terminale V_T al gramaticii G,
2. sa se obtina printr-o derivare care "pleaca" din simbolul de start S al G;

<u>Definitie 41</u>

Doua gramatici G_1 şi G_2 se numesc **echivalente** daca genereaza acelasi limbaj:

$G_1 \equiv G_2 \Leftrightarrow L(G_1) = L(G_2)$.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Observatie 42: Gramatica unui limbaj finit / infinit !!

1. G= (V_T,V_N,S,P) , unde:

   V_T = {a,b,c}

   V_N = {S,F,H,J}

   S

   p_1: S→FHJ

   p_2: F→ a

   p_3: H→ b

   p_4: J → c

$S \Rightarrow^1 FHJ \Rightarrow^2 aHJ \Rightarrow^3$

$abJ \Rightarrow^4 abc$

L = { abc} finit !!! DE CE ?

2. G_4= (V_T,V_N,S,P) , unde:

   V_T = {a,b,c}

   V_N = {S,F,H,J}

   S

   p_1: S→FHJ

   p_2: F→ aF | a

   p_3: H→ bH | b

   p_4: J → cJ | c

$S \Rightarrow^1 FHJ \Rightarrow^2 aFHJ \Rightarrow^2 \ldots \Rightarrow^2 a^nFHJ \Rightarrow^{2'}$
$a^{n+1}HJ \Rightarrow^3 a^{n+1}bHJ \Rightarrow^3 \ldots \Rightarrow^3 a^{n+1}b^mHJ$
$\Rightarrow^{3'} a^{n+1}b^{m+1}J \Rightarrow^4 a^{n+1}b^{m+1}J \Rightarrow^4 \ldots \Rightarrow^{4'}$
$a^{n+1}b^{m+1}c^{k+1}$

L = { $a^nb^mc^k$ | n,m,k∈$\mathcal{N}$*} .

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Exemple 43

1. $G_3 = (V_T, V_N, S, P)$, unde:

   $V_T = \{a,b\}$

   $V_N = \{S,C\}$

   S

   p_1: $S \rightarrow aSb$

   p_2: $S \rightarrow \varepsilon$

$S \Rightarrow^1 aSb \Rightarrow^1 aaSbb \Rightarrow^1$

$aaaSbbb \Rightarrow^2 aaabbb$

$L_3 = \{ a^n b^n \mid n \in \mathcal{N} \}$;

2. $G_4 = (V_T, V_N, S, P)$, unde:

   $V_T = \{0,1\}$

   $V_N = \{S,A,B\}$

   S

   p_1: $S \rightarrow AB$

   p_2: $A \rightarrow 0A$

   p_3: $A \rightarrow \varepsilon$

   p_4: $B \rightarrow 1B$

   p_5: $B \rightarrow \varepsilon$

$S \Rightarrow^1 AB \Rightarrow^2 0AB \Rightarrow^4$

$0A1B \Rightarrow^2 00A1B \Rightarrow^2$

$000A1B \Rightarrow^3 0001B \Rightarrow^4$

$00011B \Rightarrow^5 00011$

$L_4 = \{ 0^j 1^k \mid j,k \in \mathcal{N} \}$.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

3. $G_5 = (V_T, V_N, S, P)$, unde:

   $V_T = \{0\}$

   $V_N = \{S,L,Z,R\}$

   p_1: $S \rightarrow LZL$

   p_2: $LZ \rightarrow LR$

   p_3: $RZ \rightarrow ZZR$

   p_4: $RL \rightarrow ZZL$

   p_5: $Z \rightarrow 0$

   p_6: $L \rightarrow \varepsilon$

$S \Rightarrow^1 LZL \Rightarrow^2 LRL \Rightarrow^4$

$LZZL \Rightarrow^2 LRZL \Rightarrow^3$

$LZZRL \Rightarrow^4 LZZZZL \Rightarrow^{5*}$

$L0000L \Rightarrow^{6*} 0000$

$L_5 = \{ 0^{(2\text{^}n)} \mid n \in \mathcal{N} \}$;

4. $G_6 = (V_T, V_N, S, P)$, unde:

   $V_T = \{0,1\}$

   $V_N = \{S,A\}$

   p_1: $S \rightarrow 1A$

   p_2: $A \rightarrow 0A$

   p_3: $A \rightarrow 1A$

   p_4: $A \rightarrow \varepsilon$

$S \Rightarrow^1 1A \Rightarrow^2 10A \Rightarrow^2$

$100A \Rightarrow^3 1001A \Rightarrow^2$

$10010A \Rightarrow^3 100101A \Rightarrow^3$

$1001011A \Rightarrow^4 1001011$

L_6 = limbajul reprezentarilor binare ale numerelor naturale nenule: $L_6 = \{1\} \cdot \{0,1\}^*$.

LFA: C2 – Ierarhia Chomsky

5. G_7= (V_T,V_N,S,P) , unde:

V_T consta dintr-o multime finita de cuvinte din limba româna

V_N = {<propozitie>, <subiect>, <predicat>, <substantiv>, <prenume>, <verb>}

S = <propozitie>

p_1: <propozitie> $\rightarrow$ <subiect> <predicat>

p_2: <subiect> $\rightarrow$ <substantiv>

p_3: <subiect> $\rightarrow$ <pronume>

p_4: <predicat> $\rightarrow$ <verb>

p_5: <substantiv> $\rightarrow$ piersic | vapor | functie| ….

p_6: <pronume> $\rightarrow$ eu | tu | el …

p_7: <verb> $\rightarrow$ scrie | pluteste| creste

<propozitie> $\Rightarrow$ <subiect> <predicat> $\Rightarrow$ < substantiv > <predicat> $\Rightarrow$ < substantiv > <verb> $\Rightarrow$ vapor <verb> $\Rightarrow$ vapor pluteste

<propozitie> $\Rightarrow$ <subiect> <predicat> $\Rightarrow$ <substantiv> <predicat> $\Rightarrow$ <substantiv> <verb> $\Rightarrow$ functie <verb> $\Rightarrow$ functie pluteste

L consta din propozitii fomate din substantivele, pronumele (personale) şi verbele limbii romane, corecte gramatical (semantic?).

LFA: C2 – Ierarhia Chomsky

1. Alfabet, cuvant, operatii cu cuvinte
2. Limbaj, operatii cu limbaje
3. Gramatica; exemple
4. Clasificarea gramaticilor generative; ierarhia lui Chomsky

LFA: C2 – Ierarhia Chomsky

<u>Observatie 44</u>

Clasificari ale gramaticilor generative:

determinate de restrictiile impuse productiilor:

Gramatici de tip 0 (fara restrictii)	Gramatici fara restrictii
Gramatici de tip 1 (dependente de context)	Gramatici monotone
	Gramatici dependente de context
Gramatici de tip 2 (independente de context)	Gramatici independente de context
Gramatici de tip 3 (regulate)	Gramatici lineare
	Gramatici lineare la dreapta/stanga
	Gramatici regulate

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitie 45

Gramatica de tip 0: productiile nu suporta nicio restrictie

Tipul 0: $\alpha \longrightarrow \beta$

unde: $\alpha \in (V_N \cup V_T)^* \cdot V_N \cdot (V_N \cup V_T)^*$, $\beta \in (V_N \cup V_T)^*$

Ex. ant.: G_5= ({0}, {S,L,Z,R},S,P), unde:

P={S→LZL , LZ→ LR, RZ→ ZZR, RL→ ZZL, Z→ 0 L→ ε}

$S \Rightarrow^1 LZL \Rightarrow^2 LRL \Rightarrow^4 LZZL \Rightarrow^2 LRZL \Rightarrow^3 LZZRL \Rightarrow^4 LZZZZL$

$\Rightarrow^{5555} L0000L^{66} \Rightarrow 0000$

$\mathbf{L_5 = \{ 0^{(2\wedge n)} \mid n \in \mathcal{N} \}}$.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitie 46

Gramatica de tip 1 (dependenta de context):

Tipul 1: $\alpha A\beta \rightarrow \alpha \nu \beta$

unde: $\alpha, \beta, \nu \in (V_N \cup V_T)^*$, $A \in V_N$, $\nu \neq \varepsilon$

obs.: - daca $S \rightarrow \varepsilon \in P$ atunci S nu poate aparea în m. dr. al nici unei productii din P,

- α, β formeaza contextul in care A poate fi inlocuit cu ν
- $\nu \neq \varepsilon$ → $|\alpha A\beta| \leq |\alpha \nu \beta|$

Ex. : G_8 = ({a,b,c}, {S,B},S,P), unde:

P = {S→aSBc, S→abc, cB→Bc, bB→bb}

$S \Rightarrow^1 aSBc \Rightarrow^1 aaSBcBc \Rightarrow^1 a^3SBcBcBc \Rightarrow^2 a^4bcBcBcBc \Rightarrow^3$
$a^4bBccBcBc \Rightarrow^3 a^4bBcBccBc \Rightarrow^3 a^4bBcBcBcc \Rightarrow^3 a^4bBBccBcc \Rightarrow^3$
$a^4bBBcBccc \Rightarrow^3 a^4bBBBcccc \Rightarrow^4 a^4bbBBc^4 \Rightarrow^4 a^4bbbBc^4 \Rightarrow^4$
a^4bbbbc^4

$S \Rightarrow^1 \ldots a^nS(Bc)^n \Rightarrow^2 a^{n+1}bc(Bc)^n \Rightarrow^3 \ldots a^{n+1}bB^nc^{n+1} \Rightarrow^4 \ldots a^{n+1}b^{n+1}c^{n+1}$

$L_5 = \{ a^nb^nc^n \mid n \geq 1 \}$.

LFA: C2 – Ierarhia Chomsky

Teorema 47

Fie G = gramatica dependenta de context

=> G este recursiva (i.e. exista un algoritm care decide, pentru orice secventa w, daca *w* $\in$ *L* (G) sau *w* $\notin$ *L* (G)) .

Demonstratie

Fie $w \in \Sigma_T^*$: $|w|=n$; Notam cu $(\alpha_i)_{i<k}$ o derivare oarecare pentru w: $S \Rightarrow^* w$;

Evident: $\forall$ $1 \le i<j \le k$: $\alpha_i \neq \alpha_j$; (i.e. *oricare 2 substitutii sunt distincte*)

Ip: $|w|=n$ şi G dependenta de context => $\forall\ 1 \le i \le k$: $|\alpha_i| \le n$ (i.e. *orice pas de substitutie (al derivarii) are cel mult n simboluri*);

Cf. def. G: numarul tuturor derivarilor posibile este finit => ele pot fi generate imediat (primitiv recursiv) =>

verificarea faptului ca cel putin una dintre aceste derivari genereaza w revine la cautarea intr-o multime finita;

Evident, timpul necesar pentru verificare creste exponential =>

metoda este efectiva dar nu este eficienta.

Explicitam: **G_1:** $\alpha A\beta \to \alpha\nu\beta$ unde: $\alpha,\beta,\nu \in (V_N \cup V_T)^*$, $A \in V_N$, $\nu \neq \varepsilon$

obs.: daca $S \to \varepsilon \in P$ atunci S nu poate aparea în m. dr. al nici unei productii din P; daca $\nu \neq \varepsilon$ → $|\alpha A\beta| \le |\alpha\nu\beta|$

Ex. : $G_8 = (\{a,b,c\}, \{S,B\}, S, P)$, unde: $P = \{S \to aSBc,\ S \to abc,\ cB \to Bc,\ bB \to bb\}$

$S \Rightarrow^1 aSBc \Rightarrow^1 aaSBcBc \Rightarrow^1 a^3SBcBcBc \Rightarrow^2 a^4bcBcBcBc \Rightarrow^3 a^4bBccBcBc \Rightarrow^3 a^4bBcBccBc \Rightarrow^3 a^4bBcBcBcc \Rightarrow^3 a^4bBBccBcc \Rightarrow^3$

$a^4bBBcBccc \Rightarrow^3 a^4bBBBcccc \Rightarrow^4 a^4bbBBc^4 \Rightarrow^4 a^4bbbBc^4 \Rightarrow^4 a^4bbbbc^4$

$S \Rightarrow^1 \ldots\ a^nS(Bc)^n \Rightarrow^2 a^{n+1}bc(Bc)^n \Rightarrow^3 \ldots\ a^{n+1}bB^nc^{n+1} \Rightarrow^4 \ldots\ a^{n+1}b^{n+1}c^{n+1}$ **$L_5 = \{ a^nb^nc^n \mid n \ge 1 \}$.**

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitie 48

Gramatica de tip 2 (independenta de context):

Tipul 2: $A \rightarrow \alpha$

unde: $A \in V_N$, $\alpha \in (V_N \cup V_T)^*$

obs. neterminalul A poate fi inlocuit cu secventa α in orice context ar aparea acesta;

GIC sunt f importante:

- ✓ putere generativa suficienta:
  pot **descrie sintaxa oricarui limbaj de programare,**
- ✓ destul de simple:
  permit **proiectarea unor algoritmi de parsare** eficienti care – pentru orice secventa data - sa determine daca şi cum poate fi generata de gramatica respectiva;

Ex. ant.: G_3= ({a,b}, {S},S,{S→aSb , S→ ε}):

$S \Rightarrow^1 aSb \Rightarrow^1 aaSbb \Rightarrow^1 aaaSbbb \Rightarrow^2 aaabbb$

$\mathbf{L_3 = \{ a^n b^n \mid n \in \mathcal{N} \}}$.

LFA: C2 – Ierarhia Chomsky

Ex.: G_8= (V_T,V_N,S,P) descrie instructiunea de atribuire intr-un limbaj de programare oarecare

V_N = {<atribuire>, <expr>, <op>},

V_T= {nume_ct , nume_var, + , - , * , / }

P = { <atribuire> → nume_var = <expr>

<expr> → nume_ct | nume_var | <expr> <op> <expr> | (<expr>)

<op> → + | - | * | / }

Alt exemplu:

Sintaxa unui limbaj de programare simplu: doar trei tipuri de instructiuni: atribuiri, if-then, stop

- ✓ Pentru constante şi identificatori: un singur element, notat i
- ✓ Variabilele: simple sau indexate
- ✓ Operatorii aritmetici: + si *
- ✓ Notatia folosita: notatia Backus Naur (a se vedea definirea sintaxei limbajului ALGOL60). ->

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Ex.: G_9 = (V_N , V_T , <program>, P), unde

V_N = {<program>, <instructiune>, <atribuire>, <if>, <expresie>, <termen>, <factor>, <variabila˘>, <index>},

V_T = { begin, end, if, then, stop, t, i, +, *, (,), =, ,, ; },

P = {<program>→begin <linie> end

<linie>→<linie>;< instructiune > | <instructiune>

<instructiune>→<atribuire> | <if> | stop

<atribuire>→<variabila˘>=<expresie>

<if>→if(<expresie>) then <atribuire>

<expresie>→<expresie> + <termen> | <termen>

<termen>→<termen> ∗ <factor> | <factor>

<factor>→(<expresie>)| <variabila˘>

<variabila˘>→t(<index>)|i

<index>→<index>, <expresie> | <expresie> } .

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitie 49

Gramatica de tip 3 (regulata):

Tipul 3: $A \to aB$ sau $A \to Ba$

$A \to a$

unde: $A,B \in V_N$, $a \in V_T$

obs.: daca $S \to \varepsilon \in P$ atunci S nu poate aparea în m. dr. al nici unei productii din P,

productiile de tipul $A \to aB$ ($A \to Ba$) definesc o gramatica regulata la dreapta (stanga); ele sunt echivalente,

✓ GR sunt f importante:

descriu **structura lexicala a limbajelor de programare**

Ex. ant.: G_6= ({0,1}, {S,A} ,S,{S→1A, A→0A, A→1A, A→ ε }) , unde:

$S \Rightarrow^1 1A \Rightarrow^2 10A \Rightarrow^2 100A \Rightarrow^3 1001A \Rightarrow^2 10010A \Rightarrow^3 100101A \Rightarrow^3 1001011A \Rightarrow^4 1001011$

L_6 = {1}{0,1}* limbajul reprezentarilor binare ale nr. nat. nenule.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Ierarhia lui Chomsky

Teorema 50

Notam cu:

$\mathcal{L}_0$ - multimea limbajelor generate de gramatici de tip 0

$\mathcal{L}_1$ - multimea limbajelor generate de gramatici de tip 1

$\mathcal{L}_2$ - multimea limbajelor generate de gramatici de tip 2

$\mathcal{L}_3$ - multimea limbajalor generate de gramatici de tip 3

Atunci:

$$\mathcal{L}_0 \supset \mathcal{L}_1 \supset \mathcal{L}_2 \supset \mathcal{L}_3$$

Incluziunile nestricte: forma productiilor;

Incluziunile stricte: contraexemple.

$\mathcal{LFA}$: C2 – Ierarhia Chomsky

Definitii 51

Gramatica monotona: $\alpha \to \beta$, $|\alpha| \leq |\beta|$,

unde: $\alpha, \beta \in (V_N \cup V_T)^*$

Gramatica lineara: $A \to wBv$,

unde: $A \in V_N, B \in V_N \cup \{\varepsilon\}$, $w, v \in V_T$

Gramatica ε-libera: o gramatica in care nu exista reguli de stergere (productii de forma $A \to \varepsilon$)

Observatii 52

- ✓ In gramaticile de tip 0 şi 1 se admit productii de forma $A \to \varepsilon$ cu conditia ca A sa nu apara in m.dr. al niciunei productii;
- ✓ Existenta/inexistenta regulilor de stergere poate modifica in mod semnificativ puterea generativa a gramaticii.

LFA: C2 – Ierarhia Chomsky

Observatie 53

O primă problemă relevantă pentru teoria gramaticilor formale :

locul gramaticilor care generează limbajele naturale în ierarhia Chomsky ?

limbajele naturale sunt limbaje dependente de context ;

Observatie 54

construirea unor parsere eficiente pt:

- ✓ gramaticile regulate / independente de context: suficient de simpla
  - ✓ gramaticile dependente de context: dificilă =>

    o posibila soluție: gramaticile slab dependente de context (“mildly context-sensitive”)

    i.e.: gramatici independente de context dar

    utilizate și dependent de context prin înzestrarea cu restricții suplimentare (ex.: gramaticile contextuale).

LFA: C2 – Ierarhia Chomsky

1. Alfabet, cuvant, operatii cu cuvinte
2. Limbaj, operatii cu limbaje
3. Gramatica; exemple
4. Clasificarea gramaticilor generative; ierarhia lui Chomsky