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: mijloc de comunicare:
  - intre oameni,
  - intre om si calculator,
  - intre calculatoare.

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

Definitie 1

Alfabet = Σ = orice multime finita, nevida

Elementele = **simboluri**

Exemple 2

Σ_{bool} = {0, 1},

Σ_{latin} = {a,b,c,…,z},

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

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

LFA: C2 – Ierarhia Chomsky

Definitie 3

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

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

Σ^* = multimea tuturor cuvintelor peste alfabetul Σ

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

Observatie 4

Nu orice cuvant peste un alfabet reprezinta o notiune:

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

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

<u>Definitie 5</u>

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

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

$|\varepsilon| = 0$

<u>Notatie 6</u>

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

<u>Exemplu 7</u>

$\#_t$(complexitate)= 2

<u>Observatie 8</u>

$\forall \Sigma, \forall w \in \Sigma^*$:

$$|w| = \sum_{s \in \Sigma} \#_s(w).$$

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

Definitie 9

Fie un alfabet $\Sigma = \{s_1, s_2, \ldots, s_k\}$; se numeste **functia lui Parikh,** functia

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

Exemplu 10

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

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 $n \in N$:

$\Sigma^n = \{ \omega \in \Sigma^* \, / \, |\omega| = n \}$ 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 cuvint 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 $v \cdot w$ 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. ∀Σ: (Σ*,**.**) este un monoid (ε=elementul neutru).
3. ∀w∈Σ*, 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 .

Exemple 18

$w =$ intrucatva =>

prefix$_w\in\{\varepsilon$, I, in, int, intr, intru, intruc, intruca, intrucatv, intrucatva}

sufix$_w\in\{\varepsilon$, a, va, tva, atva, catva, ucatva, rucatva, trucatva, ntrucatva, intrucatva,}

subcuvant$_w\in\{\varepsilon$, i, n, t, r, u, c, a, v, in, nt, tr, …, intru, …, catva, …, intrucatva}

LFA: C2 – Ierarhia Chomsky

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

Definitia 19

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

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

$|v|=|w|$ 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 $w=xs_jw'$

Observatie 20

capac < capsa < capsat < captat =captat < cazuta (c<a; at=at; t<u)

Observatie 21

Ordinea canonica permite enumerarea “tuturor” cuvintelor peste orice alfabet: fie $\Sigma=\{a,b\}$ →

Σ^* $=\{\varepsilon$,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

<u>Definitie 22</u>

Fie un alfabet Σ;

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

<u>Exemple 23</u>

1. Fie V = {a,b} =>

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

<u>Notatie 24</u>

Multimea tuturor limbajelor peste alfabetul Σ:

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

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

<u>Definitie 25</u>

Fie un alfabet $\Sigma=\{s_1,s_2,\ldots,s_m\}$, $m\geq1$ 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….
$\in \mathcal{B}$ = multimea secventelor binare infinite

Exemple 26

1. Fie $\sum$ = {a,b} şi L = {a, ab, abb}

=> $\sum$*= {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 ∈ $\sum$*| ∃y∈$\sum$*: 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 ∈ $\sum$*| ∃x∈$\sum$*: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 ∈ $\sum$*| w= palindrom} =

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

=> λ_L= 111100110100 …

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

Teorema 27

Multimea $\mathcal{L}$ ={ L$\subseteq\Sigma^*$ | L =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

n	f(n)=b_n
1	$\underline{1}$0000…
2	0$\underline{1}$000…
3	11$\underline{0}$00…
4	001$\underline{0}$0…
5	1010$\underline{0}$…
…	…

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

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

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

(ii) multimea $\mathcal{L} = \{L \subseteq \Sigma^* \mid L = \text{limbaj}\}$ 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.

LFA: C2 – Ierarhia Chomsky

Definitii 28: 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

Definitii 28: 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 :

$L_1L_2 = L_1 \circ L_2 = \{vw \in (\Sigma_1 \cup \Sigma_2)^* \mid v \in L_1 \text{ si } w \in L_2\}$

limbajul obtinut prin **concatenarea** (**produsul**) acestora;

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

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 28: Operatii cu limbaje (cont.)

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

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

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 $s:\{a,b\} \to \{0,1,x\}^*$ $s(a)= 0x$, $s(b)=x11$

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

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

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

Definitii 28: 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 28: Operatii cu limbaje (cont.)

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

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

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

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

Observatie 29

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

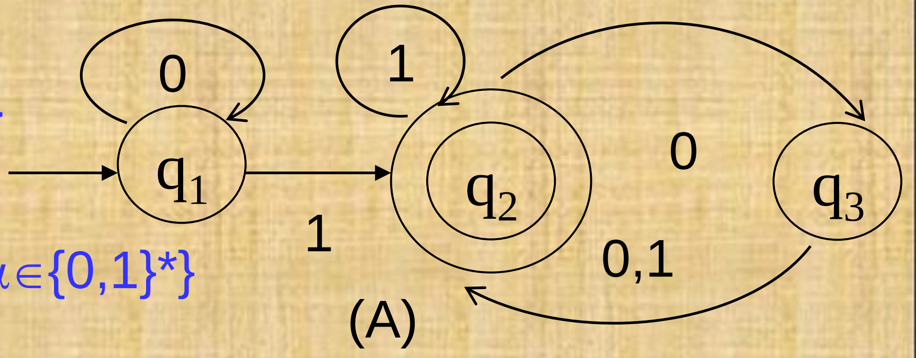
G=({S}, {a,b}, {S →aSb | SS | ε}, S) ⇒

L = {ε, ab, a^nb^n, a^nbab^n, $a^n(ba)^kb^n$, … }

- ✓definirea unui automat A

L(A) = {w∈{0,1}* | w=α1 sau w=α00, α∈{0,1}*}

- ✓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.

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

Definitie 30

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

Exemple 31: 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.

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

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

   S→CB → CCB →..... → C^nB → C^{n+1}→ $2C^n$ → $24C^{n-1}$ →.... 2409194...7

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

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).

$\mathcal{LFA}$: 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,

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

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

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

Observatie 32

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

LFA: C2 – Ierarhia Cho

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

Notatie: $\Rightarrow$

Substitutie = Derivare directa

Observatie 33

$S \Rightarrow CB \Rightarrow CCB \Rightarrow CCCB \Rightarrow CCCC \Rightarrow CCC9 \Rightarrow CC09 \Rightarrow 0C09 \Rightarrow 0509$

Notatie: $\Rightarrow^*$

Inchiderea tranzitiva a derivarii directe = Derivare

Definitii 34

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

daca $\mathbf{a} \to \boldsymbol{\beta} \in P$ 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 $\alpha_1 \Rightarrow^* \alpha_n$.

LFA: C2 – Ierarhia Chomsky

Definitie 35

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

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

Observatie 36

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;

Definitie 37

Fie $G_1=(V_T,V^1_N,S,P^1)$, $G_2=(V_T,V^2_N,S,P^2)$ si $L_1=L(G_1)$, $L_2=L(G_2)$,

Limbajele L_1 şi L_2 se numesc **echivalente** ddaca

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

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

Observatie 38: 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$\in\mathcal{N}^*$} .

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

Exemple 39

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

   $V_T = \{a,b\}$

   $V_N = \{S,C\}$

   S

   p_1: $S \to aSb$

   p_2: $S \to \varepsilon$

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

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

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

   $V_T = \{0,1\}$

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

   S

   p_1: $S \to AB$

   p_2: $A \to 0A$

   p_3: $A \to \varepsilon$

   p_4: $B \to 1B$

   p_5: $B \to \varepsilon$

$S \Rightarrow^1 AB \Rightarrow^2 0AB \Rightarrow^4$
$0A1B \Rightarrow^2 00A1B \Rightarrow^2$
$000A1B \Rightarrow^3 0001B \Rightarrow^4$
$00011B \Rightarrow^5 00011$

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

$\mathcal{LFA}$: 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: $L \rightarrow \varepsilon$

$S \Rightarrow^1 LZL \Rightarrow^2 LRL \Rightarrow^4$
$LZZL \Rightarrow^2 LRZL \Rightarrow^3$
$LZZRL \Rightarrow^4 LZZZZL \Rightarrow^{5*}$
$L0000L \Rightarrow^{6*} 0000$

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

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: $A \rightarrow \varepsilon$

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

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

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

Clasificare a gramaticilor generative

determinata 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

<u>Definitie 40</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

Definitie 41

Gramatica de tip 0: productiile nu suporta nicio restrictie

Tipul 0: $\alpha \rightarrow \beta$

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:

$P = \{S \rightarrow LZL,\ LZ \rightarrow LR,\ RZ \rightarrow ZZR,\ RL \rightarrow ZZL,\ Z \rightarrow 0\ L \rightarrow \varepsilon\}$

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

$\Rightarrow^6 0000L^6 \Rightarrow 0000$

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

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

Definitie 42

Gramatica de tip 1 (dependenta de context):

Tipul 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,

α, β formeaza contextul in care A poate fi inlocuit cu ν ;

Ex. : $G_8 = (\{0\}, \{S,B\}, S, P)$, unde:

$P = \{S \to aSBc,\ S \to abc,\ cB \to Bc,\ bB \to bb\}$

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

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

$\mathbf{L_5 = \{ a^nb^nc^n \mid n \geq 0 \}}$.

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

Teorema 39

Fie G = gramatica dependenta de context

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

Demonstratie

Fie $w \in \Sigma^*$: $|w|=n$

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

Evident: $\forall\ 1 \le i<j \le k$: $\alpha_i \neq \alpha_j$;

Ip: $|w|=n$ şi G dependenta de context => $\forall\ 1 \le i \le k$: $|\alpha_i| \le n$;

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.

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

Definitie 44

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;

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 secvbenta data - sa determine daca şi cum poate fi generata de gramatica respectiva;

Ex. ant.: G_3= ({a,b}, {S,C},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 ex.:

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>;<instruc¸tie> | <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 45

Gramatica de tip 3 (regulata):

Tipul 2: $A \rightarrow aB$ sau $A \rightarrow Ba$

$A \rightarrow a$

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

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

productiile de tipul $A \rightarrow aB$ ($A \rightarrow Ba$) 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:

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

$L_6 = \{1\} \cdot \{0,1\}^*$ este limbajul reprezentarilor binare ale nr naturale

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

Ierarhia lui Chomsky

<u>Teorema 46</u>

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 47

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 48

- ✓ 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

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