

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

1. Definitii

- gramatici și limbaje independente de context
- arbori de derivare
- proiectarea g.i.c.

2. Ambiguitate

3. Forme normale

- definitii
- exemple
- aducerea la forma normala Chomsky

4. Lema de pompare

5. Operatii de inchidere

- \mathcal{L}_2 este inchisa la reuniune, concatenare, operatia *, omomorfism
- \mathcal{L}_2 nu este inchisa la intersectie si complementara

6. Automatul pushdown;

echivalenta cu gramatica independenta de context.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

- ∃ limbaje suficient de simple pe care
un AFD / o expresie regulata / o gramatica regulata
nu le pot descrie (recunoaste):
ex.: limbajul $\{0^n 1^n \mid n \geq 0\}$
motivul: memoria **finita** a unui *AFD* nu poate memora
numere **n** foarte mari;
- ∃ mecanisme mai puternice:
 - ✓ gramaticile independente de context (*G/C*)
 - ✓ automatele pushdown (*APD*).

LF: C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Avantajul *G/C*: pot descrie structuri recursive =>
variate domenii de aplicabilitate a *G/C*:

1. studiul limbilor naturale:

- ✓ **relatiile dintre termeni** precum: substantiv, verb, adjectiv, prepozitie,
- ✓ **relatiile dintre expresiile substantivale, verbale etc.**

sunt – in mod natural – de tip recursiv: o expresie verbala poate contine o expresie substantivala (de ex.) care, la randul ei, poate contine o expresie verbala sau adjectivala etc.

2. specificarea si compilarea limbajelor de programare,

□ sintaxa unui limbaj de programare,

- ✓ **sintaxa unui lb.de programare poate fi invatata** si pornind de la gramatica sa,
- ✓ **proiectarea compilatoarelor si interpretoarelor de limbaje de programare**

incepe deseori cu construirea unei *G/C* pt acel limbaj,

□ parsere:

- ✓ **o posibila reprezentare a semnificatiei unui program extrasa de parser** chiar inainte de compilarea codului /executarea instructiunii interpretate, se poate realiza cu ajutorul arborelui de derivare al codului, obtinut cu *G/C* a limbajului de programare respectiv,
- ✓ exista numeroase metodologii care permit **construirea** – uneori automat – a **unui parser direct din *G/C* a limbajului de programare** respectiv.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Reamintim:

Definitia 1

Gramatica = (V, Σ, S, P) unde:

V = multime finita, nevida, ale carei elemente se numesc variabile sau simboluri neterminale (alta notatie V_N);

Σ = multime finita, nevida, numita alfabet de intrare, ale carei elemente se numesc [simboluri] terminale (alta notatie V_T); $V \cap \Sigma = \emptyset$;

$S \in V$ se numeste simbolul de start (axioma) gramaticii;

$P \subseteq (V \cup \Sigma)^* V (V \cup \Sigma)^* \times (V \cup \Sigma)^* =$ multime finita, nevida (productii) =
 $\{\alpha \rightarrow \beta \mid \alpha \in (V \cup \Sigma)^* V (V \cup \Sigma)^*; \beta \in (V \cup \Sigma)^*\}$

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

Definitii 2

Se numeste **substitutie = derivare directa** = aplicarea unei productii i.e.:

daca $\alpha \rightarrow \beta \in P$ și $\delta, \gamma \in (V \cup \Sigma)^*$, atunci $\delta \alpha \gamma \Rightarrow \delta \beta \gamma$

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

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

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Reamintim:

Definitia 3

Gramatica independenta de context = GIC = (V, Σ, S, P) unde:

$\forall p \in P$: p este de forma $A \rightarrow \alpha$ unde: $A \in V$, $\alpha \in (V \cup \Sigma)^*$

Definitia 4

Limbaj independent de context = LIC =

$L(G) = \{ w \in \Sigma^* \mid \exists S \xrightarrow{*}_G w \text{ si } G = \text{GIC} \}$

Notatii 5

Multimea gramaticilor independente de context: $\mathcal{G}_2 = \{ G \mid G = \text{GIC} \}$

Multimea limbajelor independente de context: $\mathcal{L}_2 = \{ L \mid L = \text{LIC} \}$.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Exemple 6

$$G = (\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \varepsilon\}) \in \mathcal{G}_2 \Rightarrow$$

$$L = \{a^n b^n \mid n \in \mathcal{N}\} \in \mathcal{L}_2$$

$$G = (\{S\}, \{a, b\}, S, \{S \rightarrow SS, S \rightarrow aSb, S \rightarrow \varepsilon\}) \in \mathcal{G}_2 \Rightarrow$$

$$L = \{\varepsilon, ab, a^n b^n, a^n b a b^n, a^n (ba)^k b^n, \dots\} \in \mathcal{L}_2.$$

$$G = (\{S, A, B\}, \{a, b\}, S, \{S \rightarrow aB, S \rightarrow bA, A \rightarrow aS, B \rightarrow bS, A \rightarrow bAA, B \rightarrow aBB, A \rightarrow a, B \rightarrow b\}) \in \mathcal{G}_2 \Rightarrow$$

$$L(G) = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}.$$

ℒFA: C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Definitie 7

Arbori de derivare = arbori de parsare =

= o metoda de reprezentare vizuala a derivarilor dintr-o gramatica independenta de context $G = (V, \Sigma, S, P)$

= un arbore in care

1. fiecare nod este etichetat cu un simbol din $V \cup \Sigma$;
2. radacina este etichetata cu S ;
3. daca un nod n , etichetat cu A , are cel putin un nod descendent, atunci A trebuie sa fie din V ;
4. daca nodul n și descendentii sai directi n_1, n_2, \dots, n_k sunt etichetati respectiv cu $A, A_1, A_2, \dots, A_k \in V \cup \Sigma$ atunci P trebuie sa contina productia

$$A \rightarrow A_1 A_2 \dots A_k ;$$

Observatie 8

Etichetele nodurilor terminale formeaza un cuvant din $L(G)$, numit și rezultat al derivarii.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Exemple 9

1. Fie $G_1 = (\{S, C\}, \{0, 1, \#\}, S, \{S \rightarrow 0S1, S \rightarrow C, C \rightarrow \#\})$ si $\alpha = 0^3\#1^3 \Rightarrow$

\Rightarrow Reprezentarea derivarilor:

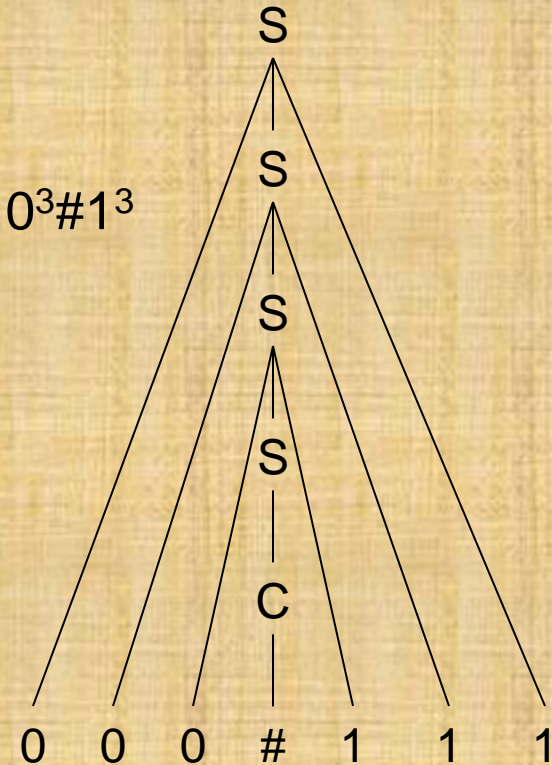
✓ linear:

$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000S111 \Rightarrow 0^3S1^3 \Rightarrow 0^3C1^3 \Rightarrow 0^3\#1^3$

✓ sintetic:

$S_G \Rightarrow^* 0^3\#1^3$

✓ arbore de derivare:



\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

2. Fie $G_2 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow aAS, S \rightarrow a, A \rightarrow SS, A \rightarrow SbA, A \rightarrow ba\})$
si $\alpha = a^2b^2a^2 \Rightarrow$

\Rightarrow Reprezentarea derivarilor:

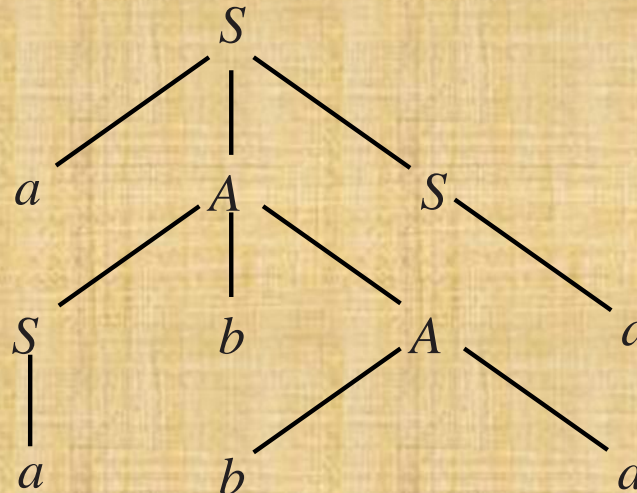
✓linear:

$S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbaa$

✓sintetic:

$S \Rightarrow^* a^2b^2a^2$

✓arbore de derivare:



\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

3. Fie $G_3 = (\{S\}, \{a,b\}, S, \{S \rightarrow SS, S \rightarrow aSb, S \rightarrow \varepsilon\})$

OBS.: $daca \ a \leftarrow (\text{ si } b \leftarrow) \Rightarrow$

L_3 = limbajul parantezelor corect imbricate

fie $\alpha = (())() \Rightarrow S \xRightarrow{*} (())()$

\Rightarrow Reprezentarea derivarilor:

✓linear:

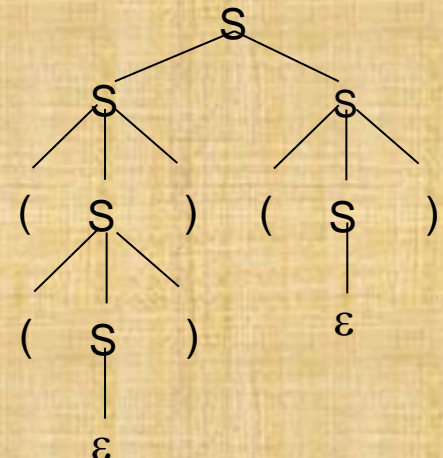
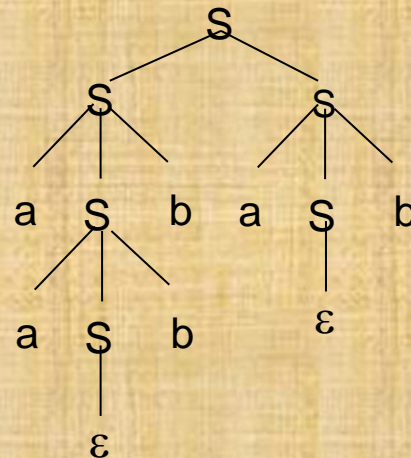
$S \Rightarrow SS \Rightarrow aSbS \Rightarrow aSbaSb \Rightarrow aaSbbaSb \Rightarrow aaSbbab \Rightarrow aabbab$

$S \Rightarrow SS \Rightarrow (S)S \Rightarrow (S)(S) \Rightarrow ((S))(S) \Rightarrow ((S))() \Rightarrow (())()$

✓sintetic:

$S \xRightarrow{*} a^2b^2ab$

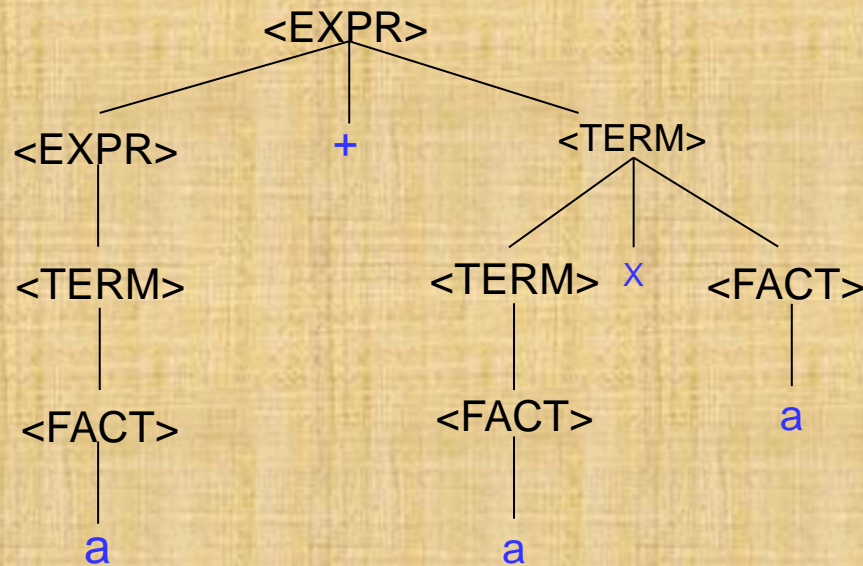
✓arborii de derivare.



Exercitiu: $S \xRightarrow{*} abab$ $S \xRightarrow{*} a^3b^3$ $S \xRightarrow{*} a^2bab^2$

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

4. Fie $G_4 = (\{<EXPR>, <TERM>, <FACT>\}, \{a, +, x, (,)\}, <EXPR>, \{<EXPR> \rightarrow <EXPR> + <TERM>, <EXPR> \rightarrow <TERM>, <TERM> \rightarrow <TERM> x <FACT>, <TERM> \rightarrow <FACT>, <FACT> \rightarrow (<EXPR>), <FACT> \rightarrow a\})$
 \Rightarrow arbore de derivare pentru cuvintul $a+axa$.



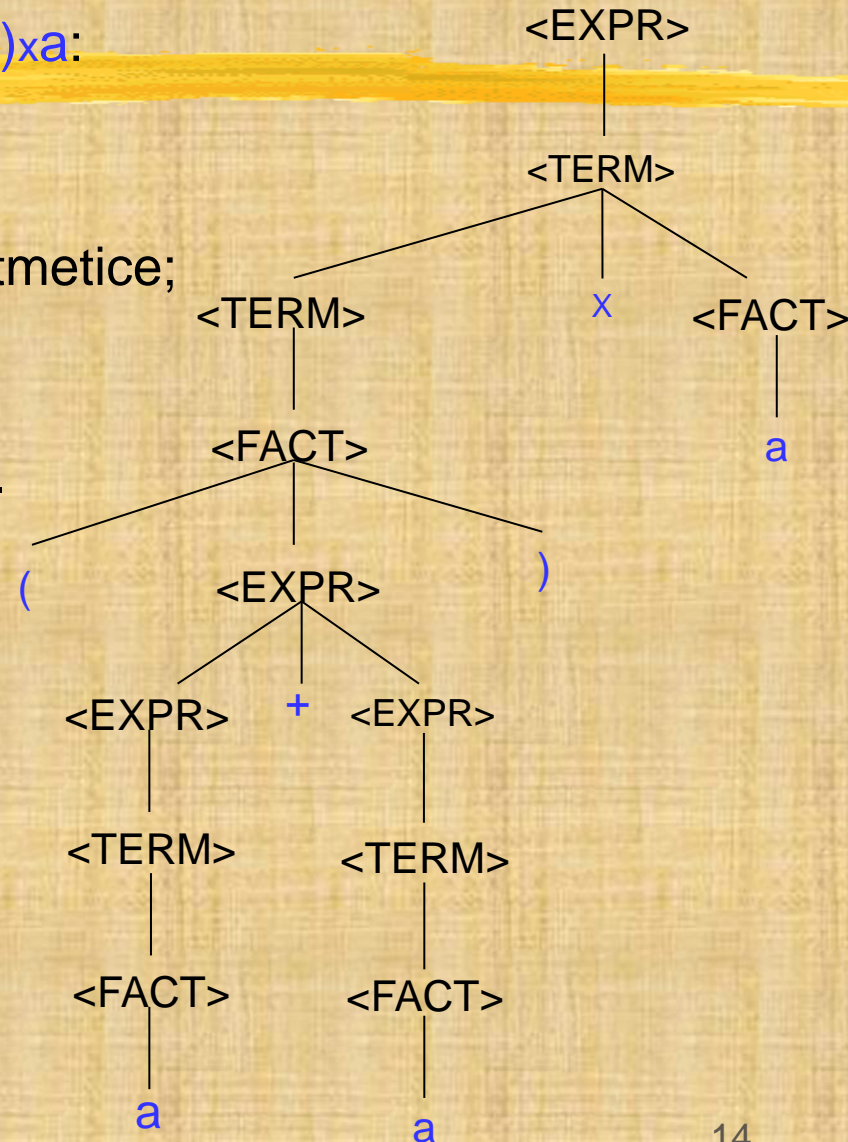
\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

=> arbore de derivare pentru cuvintul $(a+a)xa$:

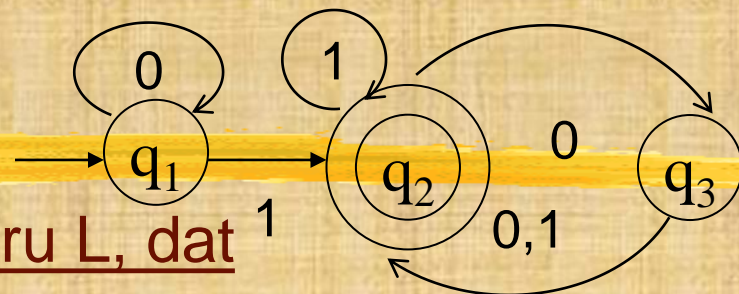
Observatie:

G_4 descrie acel fragment din limbajele de programare care trateaza expresiile aritmetice;

arborele de parsare grupeaza operatorii in conformitate cu regulile de precedenta (respectand inclusiv rolul parantezelor).



\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT



Observatie 10: Proiectarea G.I.C. pentru L, dat

Fie $L = L(G)$; $G \in \mathcal{G}_2$; Distingem cel puțin 4 situatii:

1. $L \in \mathcal{L}_3$; \Rightarrow

(i) construim AFD $A=(Q, \Sigma, \delta, s, F)$: $L=L(A)$;

(ii) convertim $A=(Q, \Sigma, \delta, s, F)$ in $G=(V, \Sigma, S, P) \in \mathcal{G}_2$ astfel:

$$Q \leadsto V$$

$$s \leadsto S$$

$$\delta(q_i, a) = q_k \leadsto B_i \rightarrow aB_k$$

$$\forall q_k \in F \leadsto \text{se adauga } B_k \rightarrow \varepsilon \in P$$

(iii) verificam direct ca $L(A)=L(G)$

ex.: $L = \{w \in \{0,1\}^* \mid w = \alpha 1 (00)^n, \alpha \in \{0,1\}^*\}$

(i) $A = (\{q_1, q_2, q_3\}, \{0,1\}, \delta, q_1, \{q_2\})$

(ii) $G = (\{S, A, B\}, \{0,1\}, S,$

$$\{S \rightarrow 0S, S \rightarrow 1A, A \rightarrow 0B, A \rightarrow 1A, B \rightarrow 0B, B \rightarrow 1B, A \rightarrow \varepsilon\})$$

(iii) 1, 01, 11, 0101, 010100, 01001,

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Observatie 10: Proiectarea G.I.C. pentru L , dat (cont.)

2. putem descompune L in limbaje mai simple: $L_1, L_2, \dots \Rightarrow$

(i) construim gramaticile $G_1, G_2, \dots: L_i = L(G_i)$;

(ii) le asamblam și adaugam:

- un nou simbol de start S și
- productiile $S \rightarrow S_i$

ex.: $L = L_1 \cup L_2$ (ex.: $L = \{a^n b^n | n \in \mathcal{N}\} \cup \{b^n a^n | n \in \mathcal{N}\}$)

$\Rightarrow L_1 = \{a^n b^n | n \in \mathcal{N}\}$ și $L_2 = \{b^n a^n | n \in \mathcal{N}\}$

\Rightarrow construim $G_i = (V_i, \Sigma, S_i, P_i)$ $i=1,2$ unde

$V_i = \{S_i\}$, $\Sigma = \{a, b\}$ și $P_1 = \{S_1 \rightarrow aS_1b \mid \varepsilon\}$, $P_2 = \{S_2 \rightarrow bS_2a \mid \varepsilon\}$

adaugam S și productiile $S \rightarrow S_1$ și $S \rightarrow S_2$

$\Rightarrow G = (\{S, S_1, S_2\}, \{a, b\}, S, \{S \rightarrow S_1 \mid S_2, S_1 \rightarrow aS_1b \mid \varepsilon, S_2 \rightarrow bS_2a \mid \varepsilon\})$.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Observatie 10: Proiectarea G.I.C. pentru L, dat (cont.)

3. descoperim simetriile / dependentele dintre subcuvintele care formeaza cuvintele din L =>

le transpunem in simetrii / dependente intre [ne]terminalele care apar in m.dr. al productiilor din P

ex.: $L = \{a^n b^{3n} | n \in \mathcal{N}\}$

=> construim $G = (V, \Sigma, S, P)$ unde

$V = \{S, A\}$, $\Sigma = \{a, b\}$ și $P = \{S \rightarrow aSbbb \mid \varepsilon\}$

Exercitiu

Fie L = limbajul parantezelor corect imbricate; sa se defineasca G a.i. $L = L(G)$, aplicand observatiile metodologice (2) și apoi (3).

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Observatie 10: Proiectarea G.I.C. pentru L, dat (cont.)

4. descoperim structurile recursive care apar in cuvintele din L =>

- (i) fie *rec* acea structura recursiva și *R* variabila care o genereaza prin productia $R \rightarrow \text{rec}$,
- (ii) plasam variabila *R* in m.dr. al productiilor care genereaza *rec* in pozitia in care apare repetitia ;

ex.: limbajul expresiilor aritmetice: intr-o expresie aritmetica, orice aparitie a unei constante poate fi inlocuita cu o noua expresie

⇒ a se vedea ultimele 2 productii din

$$G_4 = (\{ \langle \text{EXPR} \rangle, \langle \text{TERM} \rangle, \langle \text{FACT} \rangle \}, \{ a, +, x, (,) \}, \langle \text{EXPR} \rangle, \\ \{ \langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle, \langle \text{EXPR} \rangle \rightarrow \langle \text{TERM} \rangle, \\ \langle \text{TERM} \rangle \rightarrow \langle \text{TERM} \rangle x \langle \text{FACT} \rangle, \langle \text{TERM} \rangle \rightarrow \langle \text{FACT} \rangle, \\ \langle \text{FACT} \rangle \rightarrow (\langle \text{EXPR} \rangle, \langle \text{FACT} \rangle \rightarrow a) \}) .$$

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

1. Definitii

- gramatici și limbaje independente de context
- arbori de derivare
- proiectarea g.i.c.

2. Ambiguitate

3. Forme normale

- definitii
- exemple
- aducerea la forma normala Chomsky

4. Lema de pompare

5. Operatii de inchidere

- \mathcal{L}_2 este inchisa la reuniune, concatenare, operatia *, omomorfism
- \mathcal{L}_2 nu este inchisa la intersectie si complementara

6. Automatul pushdown;

echivalenta cu gramatica independenta de context.

LF: C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Observatie 11

$\exists G \in \mathcal{G}_2$ și $\exists w \in L(G)$ pe care G le poate genera in mai multe moduri \Rightarrow
cuvantul w va admite mai multi arbori de derivare (parsare) \Rightarrow
cuvantul w va avea mai multe intelesuri:

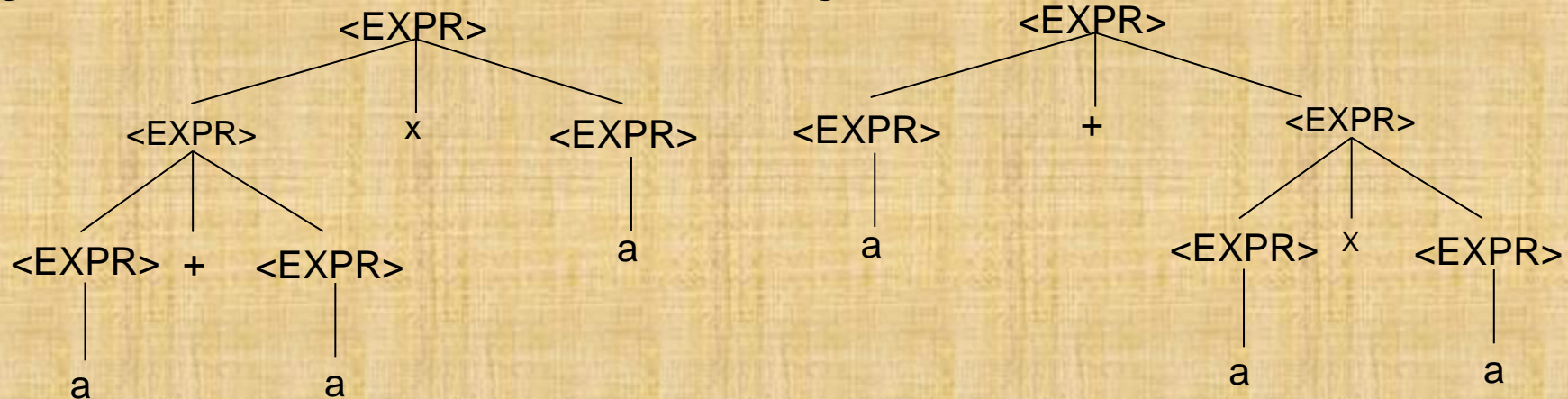
\Rightarrow in limbile naturale: avantaj,

in limbajele de programare: **mare** dezavantaj;

Exemplu 12

Fie $G_5 = (\{<EXPR>\}, \{a, +, x, (,)\}, <EXPR>, \{<EXPR> \rightarrow (<EXPR>), <EXPR> \rightarrow a, <EXPR> \rightarrow <EXPR> + <EXPR>, <EXPR> \rightarrow <EXPR> x <EXPR>\})$

G_5 genereaza cuvintul $a+axa$ in mod ambiguu:



motivul: G_4 modeleaza și regulile de precedenta a operatorilor, G_5 : NU!

\Rightarrow in G_4 orice cuvânt are 1! arbore de parsare, in G_5 : NU!

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Observatie 13

Notiunea de ambiguitate este legata de notiunea de arbore de derivare :
daca $w \in L(G)$, $G \in \mathcal{G}_2$, admit mai multe derivari dar 1! arbore de parsare \Rightarrow
 G **NU** este ambigua

motivul: 2 derivari ale w in G pot diferi prin ordinea in care sunt alese
neterminalele care se substituie, ceea ce nu altereaza structura derivarii
 \Rightarrow s-a introdus notiunea de derivare extrem stanga pt a evidentia structura
derivarii (independenta de alegerea neterminalelor);

Definitie 14

Fie $S \xRightarrow{*}_G w$ o derivare a cuvintului $w \in L(G)$, $G \in \mathcal{G}_2$;

aceasta derivare se numeste **derivare extrem stanga** daca la fiecare pas
de derivare neterminalul substituit este cel mai din stanga neterminal

Ex.: $G_2 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow aAS, S \rightarrow a, A \rightarrow SS, A \rightarrow SbA, A \rightarrow ba\})$

$S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbbaa$

Cex.:

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Definitie 15

Fie $S_G \Rightarrow^* w$ o derivare a cuvântului $w \in L(G)$, $G \in \mathcal{G}_2$;

Spunem ca **w este ambiguu derivat în G** ddaca w admite cel puțin 2 derivări extrem stângi diferite.

Gramatica **G se numeste ambiguu** ddaca generează ambiguu cel puțin un cuvânt din $L(G)$

Observatie 16

$\exists G \in \mathcal{G}_2$ ambiguu a.i. $\exists G' \in \mathcal{G}_2$, neambiguu, $L(G') = L(G)$.

Definitie 17

$L \in \mathcal{L}_2$ se numeste **inerent ambiguu** ddaca

$\forall G \in \mathcal{G}_2$, $L = L(G)$: $G = \text{ambiguu}$.

Exemplu 18

$L = \{a^k b^i c^n \mid i=k \text{ sau } i=n\}$ este inerent ambiguu.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

1. Definitii

- gramatici și limbaje independente de context
- arbori de derivare
- proiectarea g.i.c.

2. Ambiguitate

3. Forme normale

- definitii
- exemple
- aducerea la forma normala Chomsky

4. Lema de pompare

5. Operatii de inchidere

- \mathcal{L}_2 este inchisa la reuniune, concatenare, operatia *, omomorfism
- \mathcal{L}_2 nu este inchisa la intersectie si complementara

6. Automatul pushdown;

echivalenta cu gramatica independenta de context.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Definitia 19 (Sheila Adele Greibach, 1965)

$G \in \mathcal{G}_2$ se afla in **forma normala GREIBACH (FNG)** \Leftrightarrow

$\forall p \in P, p : A \rightarrow aB,$
unde: $A \in V,$
 $a \in \Sigma,$
 $B \in (V \cup \Sigma)^*;$

Teorema 20

$\forall L=L(G) \in \mathcal{L}_2., \varepsilon \notin L \Rightarrow \exists G' \text{ in FNG a.i. } L=L(G').$

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Definitia 21

$G \in \mathcal{G}_2$ se afla in **forma normala CHOMSKY (FNC)** \Leftrightarrow

$\forall p \in P, \quad p : \quad A \rightarrow BC \quad \text{sau} \quad A \rightarrow a,$
unde: $A, B, C \in V, \quad B \neq S \neq C,$
 $a \in \Sigma,$

in plus, $S \rightarrow \varepsilon \in P;$

Lema 22 (demonstratia: EXERCITIU)

Fie $G \in \mathcal{G}_2$ aflata in FNC si $L = L(G) \Rightarrow$

$\forall w \in L, |w|=n: \quad \forall S \xRightarrow{*} w : \quad |S \xRightarrow{*} w| = 2n-1, n \in \mathbb{N}.$

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Teorema 23

Fie $L \in \mathcal{L}_2$ și $G \in \mathcal{G}_2$ a.i. $L=L(G) \Rightarrow \exists G'$ in FNC a.i. $L(G')=L$

Ideea demonstratiei

Convertim pe rand $p \in P$ (in ordinea crescatoare a nr de neterminale din m.dr.) aplicand un fel de tranzitivitate a productiilor;

Demonstratie

Fie $G = (V, \Sigma, S, P) \in \mathcal{G}_2$, $L = L(G)$;

construim $G' = (V', \Sigma', S', P')$ a.i. $L(G') = L(G)$ și G' in FNC;

Evident: $\Sigma' = \Sigma$, dar V', S', P' se vor modifica astfel:

(1) adaugam un nou simbol de start S' si

o noua productie: $S' \rightarrow S$

$\Rightarrow S'$ nu va apare in m.dr. vreunei $p' \in P'$.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

(2) **eliminam** ε -productiile $A \rightarrow \varepsilon, A \neq S$;

apoi, pt fiecare aparitie a simbolul A in m.dr. al unei productii

adaugam o productie similara, in care simbolul A respectiv a disparut

exemplul 1:

pt. productia $R \rightarrow uAvAw$, $u,v,w \in (V \cup \Sigma)^*$, adaugam:

$R \rightarrow uvAw$,

$R \rightarrow uAvw$,

$R \rightarrow uvw$,

exemplul 2:

pt. productia $R \rightarrow uAv$, $u,v \in (V \cup \Sigma)^*$, adaugam

$R \rightarrow uv$

exemplul 3:

pt productia $R \rightarrow A$ adaugam

$R \rightarrow \varepsilon$ (doar daca $R \rightarrow \varepsilon$ NU a fost deja eliminata);

acest pas se repeta pana cand toate ε -productiile care nu il implica pe S au fost eliminate.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

- (3) **eliminam** “productiile unitare” $A \rightarrow B$,
apoi, pt fiecare productie $B \rightarrow u$, $u \in (V \cup \Sigma)^*$,
adaugam productia $A \rightarrow u$,

doar daca $A \rightarrow u$ NU este o “productie unitara” care a fost deja eliminata;
acest pas se repeta pana cand toate “productiile unitare” au fost eliminate;

- (4) pentru productiile “duble” de tipul $A \rightarrow u_1 u_2$, unde fie $u_1 \in \Sigma$ fie $u_2 \in \Sigma$
se inlocuieste terminalul cu neterminalul U ,
se adauga neterminalul U la V' și

se adauga productia $U \rightarrow u_1$, $u_1 \in \Sigma$ (respectiv $U \rightarrow u_2$, $u_2 \in \Sigma$) la P ;
acest pas se repeta pana cand toate “productiile duble” au fost prelucrate;
exemplul 4:

productia $R \rightarrow Ab$ este inlocuita cu $R \rightarrow AU$ și se adauga $U \rightarrow b$.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

(5) restul productiilor, adica cele de tipul:

$A \rightarrow u_1 u_2 u_3 \dots u_n$, $n \geq 3$ si $\forall 1 \leq i \leq n$: $u_i \in V$ sau $u_i \in \Sigma$,

se inlocuiesc, fiecare, cu setul de productii

$A \rightarrow u_1 A_1$,

$A_1 \rightarrow u_2 A_2$,

$A_2 \rightarrow u_3 A_3$,

.....

$A_{n-2} \rightarrow u_{n-1} u_n$,

unde A_1, A_2, \dots, A_{n-2} sunt noi variabile, adaugate lui V'

q.e.d.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Exemplu 24

Fie $G \in \mathcal{G}_2$ cu productiile:

$S \rightarrow ASA \mid aB, \quad A \rightarrow B \mid S, \quad B \rightarrow b \mid \varepsilon \Rightarrow$

? $G' \in \mathcal{G}_2$, FNC echivalenta

(1) $V' = V \cup \{S'\}$ si $P' = P \cup \{S' \rightarrow S\} \Rightarrow$

$P' = \{S' \rightarrow S, S \rightarrow ASA, S \rightarrow aB, S \rightarrow a, A \rightarrow B, A \rightarrow S, B \rightarrow b, B \rightarrow \varepsilon\}$

(2) eliminam $B \rightarrow \varepsilon \Rightarrow$ tb. adaugate $S \rightarrow a$ si $A \rightarrow \varepsilon \Rightarrow$

$P' = \{S' \rightarrow S, S \rightarrow ASA, S \rightarrow aB, S \rightarrow a, A \rightarrow B, A \rightarrow \varepsilon, A \rightarrow S, B \rightarrow b\}$

obs. ca a aparut o noua ε -productie: $A \rightarrow \varepsilon$,
deci trebuie eliminata

dupa care aduagam $S \rightarrow AS \mid SA \mid S \Rightarrow$

$P' = \{S' \rightarrow S, S \rightarrow ASA, S \rightarrow AS, S \rightarrow SA, S \rightarrow S, S \rightarrow aB, S \rightarrow a, A \rightarrow B, A \rightarrow S, B \rightarrow b\}$.

1. $V' = V \cup \{S'\}$ si $P' = P \cup \{S' \rightarrow S\}$

2. eliminam ε -productiile $A \rightarrow \varepsilon, A \neq S$; apoi, pt \forall aparitie a smb. A in m.dr. al unei productii **adaugam o productie** similara, in care simbolul A respectiv a disparut;

3. eliminam “productiile unitare” $A \rightarrow B$; apoi, pt. \forall productie $B \rightarrow u, u \in (V \cup \Sigma)^*$, adaugam productia $A \rightarrow u$

4. pentru productiile “duble”:

$A \rightarrow u_1 u_2$, unde $u_1 \in \Sigma$ sau $u_2 \in \Sigma$:

se inlocuieste $u \in \Sigma$ cu $U \in V'$ si se adauga productia $U \rightarrow u$ la P;

5. restul productiilor, adica :

$A \rightarrow u_1 u_2 u_3 \dots u_n, n \geq 3, \forall 1 \leq i \leq n: u_i \in V$ sau $u_i \in \Sigma$, **se inlocuiesc**,

fiecare, cu setul de productii

$A \rightarrow u_1 A_1, A_1 \rightarrow u_2 A_2, A_2 \rightarrow u_3 A_3, \dots, A_{n-2} \rightarrow u_{n-1} u_n$, unde A_1, A_2, \dots, A_{n-2} sunt noi variabile, adaugate lui V' .

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Exemplu (cont.)

(3) tb. eliminate $S \rightarrow S$ și $S' \rightarrow S$ cu adaugările impuse de $S' \Rightarrow$

$$P' = \{S' \rightarrow ASA, S' \rightarrow AS, S' \rightarrow SA, S' \rightarrow aB, S' \rightarrow |a, S \rightarrow ASA, S \rightarrow AS, S \rightarrow SA, S \rightarrow aB, S \rightarrow a, A \rightarrow B, A \rightarrow S, B \rightarrow b\}$$

(3) tb. eliminate $A \rightarrow B$ și $A \rightarrow S \Rightarrow$

$$P' = \{S' \rightarrow ASA|AS|SA|aB|a, S \rightarrow ASA|AS|SA|aB|a, B \rightarrow b\}$$

și apoi aduagarile impuse:

$$P' = \{S' \rightarrow ASA|AS|SA|aB|a, S \rightarrow ASA|AS|SA|aB|a, A \rightarrow S|b|ASA|AS|SA|aB|a, B \rightarrow b\}$$

obs. ca avem o noua productie unitara $A \rightarrow S$ care tb. eliminata (și care nu aduce noi productii):

$$P' = \{S' \rightarrow ASA|AS|SA|aB|a, S \rightarrow ASA|AS|SA|aB|a, A \rightarrow b|ASA|AS|SA|aB|a, B \rightarrow b\}.$$

1. $V' = V \cup \{S'\}$ și $P' = P \cup \{S' \rightarrow S\}$

2. eliminam ϵ -productiile $A \rightarrow \epsilon$, $A \neq S$; apoi, pt \forall aparitie a smb. A in m.dr. al unei productii **adaugam o productie** similara, in care simbolul A respectiv a disparut;

3. eliminam “productiile unitare $A \rightarrow B$ ”; apoi, pt. \forall productie $B \rightarrow u$, $u \in (V \cup \Sigma)^*$, adaugam productia $A \rightarrow u$

4. pentru productiile “duble”: $A \rightarrow u_1 u_2$, unde $u_1 \in \Sigma$ sau $u_2 \in \Sigma$:

se inlocuieste $u \in \Sigma$ cu $U \in V'$ și se adauga productia $U \rightarrow u$ la P;

5. restul productiilor, adica : $A \rightarrow u_1 u_2 u_3 \dots u_n$, $n \geq 3$, $\forall 1 \leq i \leq n$: $u_i \in V$ sau $u_i \in \Sigma$, **se inlocuiesc**, fiecare, cu setul de productii $A \rightarrow u_1 A_1, A_1 \rightarrow u_2 A_2, A_2 \rightarrow u_3 A_3, \dots, A_{n-2} \rightarrow u_{n-1} u_n$, unde A_1, A_2, \dots, A_{n-2} sunt noi variabile, adaugate lui V' .

ℒFA: C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Exemplu (cont.)

(4) tratăm productiile duble $S' \rightarrow aB$, $S \rightarrow aB$ și $A \rightarrow aB \Rightarrow$

$V' = V' \cup \{U\}$ și

$P' = \{S' \rightarrow ASA | AS | SA | UB | a,$
 $S \rightarrow ASA | AS | SA | UB | a,$
 $A \rightarrow b | ASA | AS | SA | UB | a, B \rightarrow b, U \rightarrow a\}$

(5) înlocuim productiile $S' \rightarrow ASA$, $S \rightarrow ASA$, $A \rightarrow ASA$ respectiv cu

$S' \rightarrow ASA$: $S' \rightarrow AX$, $X \rightarrow SA$

$S \rightarrow ASA$: $S \rightarrow AX$, $X \rightarrow SA$

$A \rightarrow ASA$: $A \rightarrow AX$, $X \rightarrow SA$ și $V' = V' \cup \{X\} \Rightarrow$

$P' = \{S' \rightarrow AX | AS | SA | UB | a,$
 $S \rightarrow AX | AS | SA | UB | a,$
 $A \rightarrow AX | AS | SA | UB | a | b, B \rightarrow b, U \rightarrow a,$
 $X \rightarrow SA\} \Rightarrow$

$\Rightarrow G' = (\{S', S, A, B, U, X\}, \{a, b\}, S', \{B \rightarrow b, U \rightarrow a,$
 $X \rightarrow SA, S' \rightarrow AX | AS | SA | UB | a,$
 $S \rightarrow AX | AS | SA | UB | a, A \rightarrow AX | AS | SA | UB | a | b\})$

1. $V' = V \cup \{S'\}$ și $P' = P \cup \{S' \rightarrow S\}$

2. eliminăm ε -productiile $A \rightarrow \varepsilon$, $A \neq S$; apoi, pt \forall apariție a smb. A în m.dr. al unei productii **adaugăm o productie** similară, în care simbolul A respectiv a dispărut;

3. eliminăm “productiile unitare” $A \rightarrow B$; apoi, pt. \forall productie $B \rightarrow u$, $u \in (V \cup \Sigma)^*$, adăugăm productia $A \rightarrow u$

4. pentru productiile “duble”:
 $A \rightarrow u_1 u_2$, unde $u_1 \in \Sigma$ sau $u_2 \in \Sigma$:

se înlocuiește $u \in \Sigma$ cu $U \in V'$ și se adăuga productia $U \rightarrow u$ la P;

5. restul productiilor, adică :

$A \rightarrow u_1 u_2 u_3 \dots u_n$, $n \geq 3$, $\forall 1 \leq i \leq n$: $u_i \in V$ sau $u_i \in \Sigma$, se înlocuiesc, fiecare, cu setul de productii
 $A \rightarrow u_1 A_1$, $A_1 \rightarrow u_2 A_2$, $A_2 \rightarrow u_3 A_3$, ..., $A_{n-2} \rightarrow u_{n-1} u_n$, unde A_1, A_2, \dots, A_{n-2} sunt noi variabile, adăugate lui V' .

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

1. Definitii

- gramatici și limbaje independente de context
- arbori de derivare
- proiectarea g.i.c.

2. Ambiguitate

3. Forme normale

- definitii
- exemple
- aducerea la forma normala Chomsky

4. Lema de pompare

5. Operatii de inchidere

- \mathcal{L}_2 este inchisa la reuniune, concatenare, operatia *, omomorfism
- \mathcal{L}_2 nu este inchisa la intersectie si complementara

6. Automatul pushdown;

echivalenta cu gramatica independenta de context.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Lema de pompare pentru L.I.C.

Fie $L \subseteq \Sigma^*$, $L \in \mathcal{L}_2 \Rightarrow$

$\exists p \in \mathbb{N}$ (constanta=lungimea de pompare) a.i. $\forall s \in L, |s| \geq p \rightarrow$

$\exists u, v, x, y, z \in \Sigma^*$ cu proprietatile:

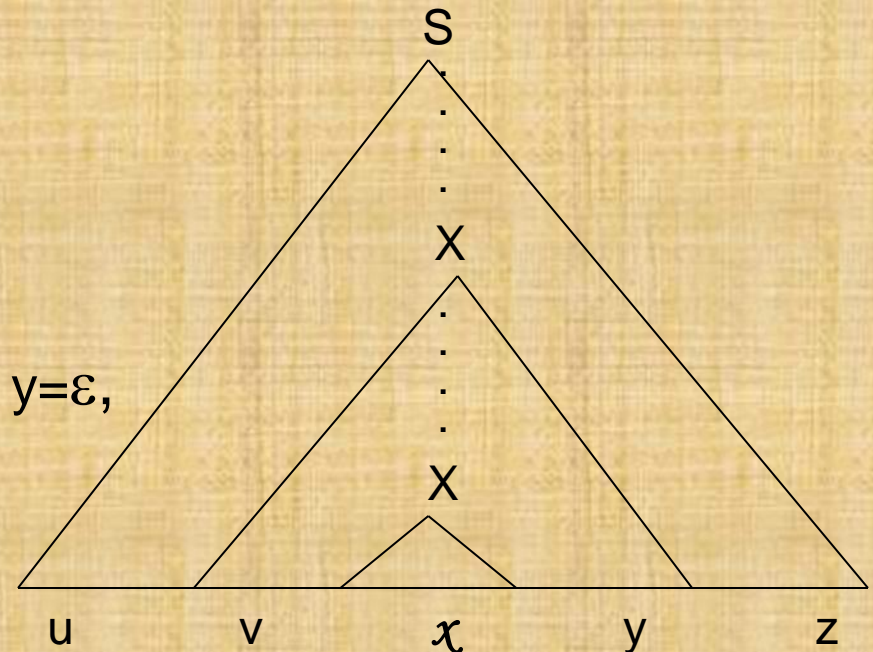
- (i) $s = uvxyz$,
- (ii) $\forall i \geq 0: uv^i xy^i z \in L$,
- (iii) $|vy| > 0$,
- (iv) $|vxy| \leq p$

Obs.:

(iii) elimina cazurile in care fie $v = \varepsilon$ fie $y = \varepsilon$,
in care lema e trivial adevarata;

(iv) e utila in a demonstra ca $L \notin \mathcal{L}_2$

Ideea demonstratiei



\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Demonstratie

Fie $G \in \mathcal{G}_2$ și $L = L(G)$;

fie $b = \max \{|w| \mid A \rightarrow w \in P\}$

(nr. max. de simb., terminale și neterm., inclusiv repetitii, care pot aparea în m.stg al p , $\forall p \in P$);

$\Rightarrow \forall w \in L(G)$: orice nod din arborele de parsare al w va avea max. b descendenți directi;

$\Rightarrow \exists$ max. b frunze care se afla la distanța de 1 arc de radacina arborelui (etichetata cu S),

\exists max. b^2 frunze care se afla la distanța de 2 arce de radacina,

\exists max. b^k frunze care se afla la distanța de k arce de radacina,

\Rightarrow dacă $H =$ înălțimea arborelui și $H \leq k$, atunci $|w| \leq b^k$;

reciproc: dacă $|w| \geq b^{k+1}$, atunci \forall dintre arborii sai de parsare va avea o înălțime $H \geq k+1$;

\Rightarrow definim ct. de pompare: $p = b^{\text{card}(V)+1}$, unde $b = \max\{|w| \mid A \rightarrow w \in P\} \Rightarrow$

$\Rightarrow \forall w \in L(G)$: dacă $|w| \geq k$ atunci înălțimea arborelui sau de parsare va fi: $H \geq \text{card}(V)+1$
deorece $b^{\text{card}(V)+1} \geq b^{\text{card}(V)+1}$; (1)

Obs. ca, dacă $n = \text{card}(V)$, putem pp. $n \geq 2$:

$n=2$, dacă G e în FNC

$n>2$, altfel.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Demonstratie (cont.)

Cum putem “pompa” w ?

fie τ arborele de parsare al w , care are cel mai mic nr. de noduri,

cf. (1). $H_\tau \geq |V|+1$; atunci

$\Rightarrow \exists$ un drum in H_τ de la radacina la o frunza de lg. $\leq |V|+1$

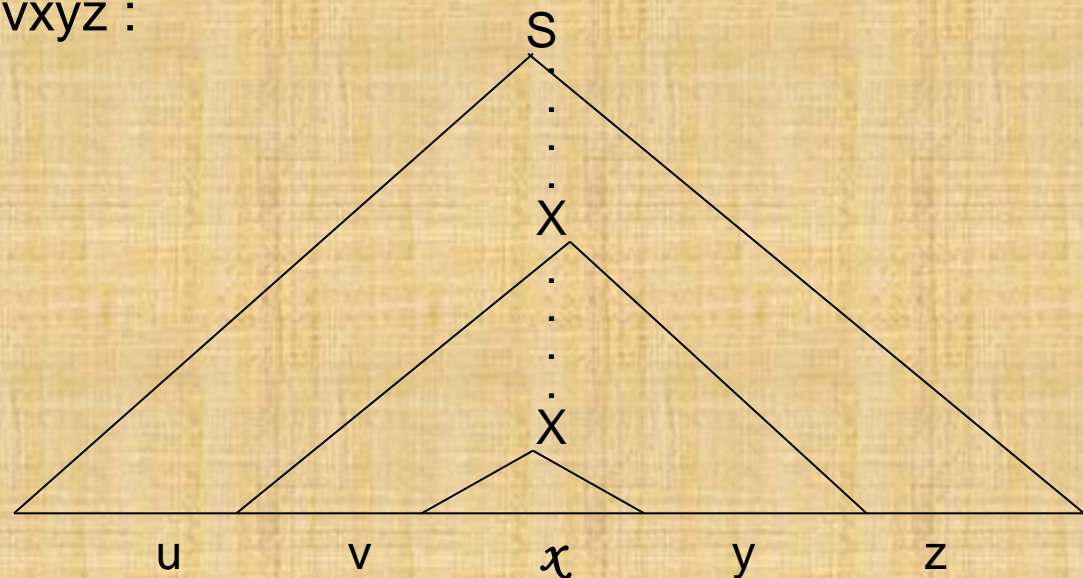
\Rightarrow acest drum are cel putin $|V|+2$ noduri, dintre care 1! nod e etichetat cu un terminal

\Rightarrow acest drum are cel putin $|V|+1$ noduri etichetate cu neterm.

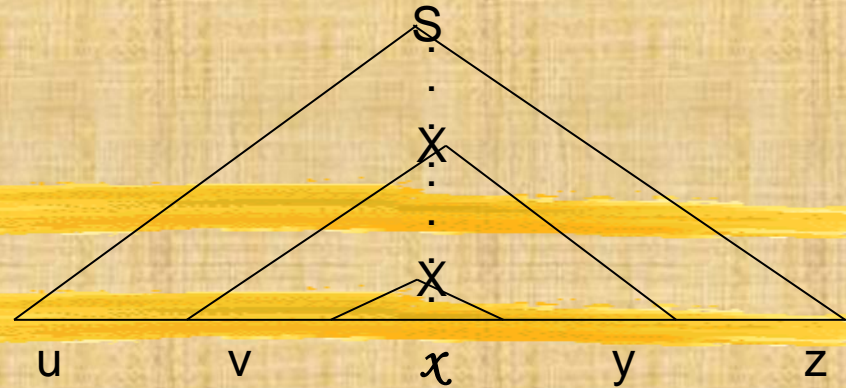
\Rightarrow cel putin un neterminal se repeta de-a lungul acestui drum

alegem neterminalul care se repeta cel mai aproape de frunza;

Descompunem $w=uvxyz$:



\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT



Demonstratie (cont.)

verificam cond.(ii) : $\forall i \geq 0: uv^i xy^i z \in L,$

fiecare aparitie a lui X "anunta" un subarbore care genereaza un subcuv. al cuv. w :

- ❑ “prima” aparitie reprezinta radacina subarborelui celui mai mare, cel care genereaza vxy ;
- ❑ “ultima” aparitie reprezinta radacina subarborelui celui mai mic, cel care genereaza x ;

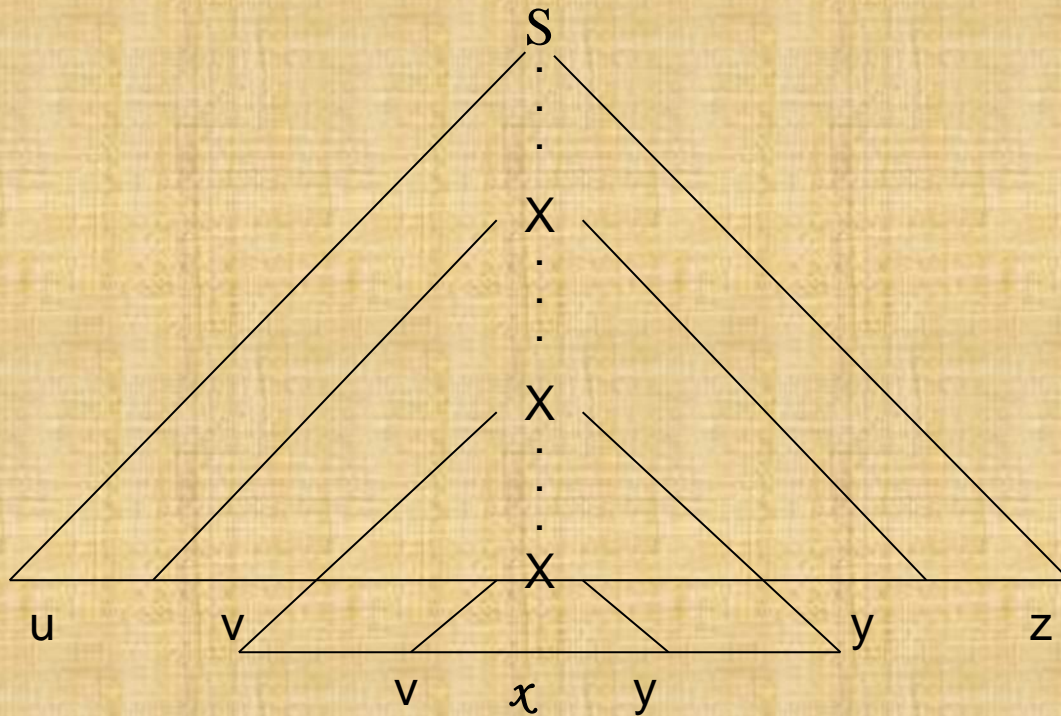
intrucat cei 2 subarbori au aceeaasi radacina (etichetata X), i.e. sunt generati de acelasi neterminat \rightarrow putem substitui pe unul cu celalalt obtinand tot un arbore de derivare corect

=> daca inlocuim, repetat:

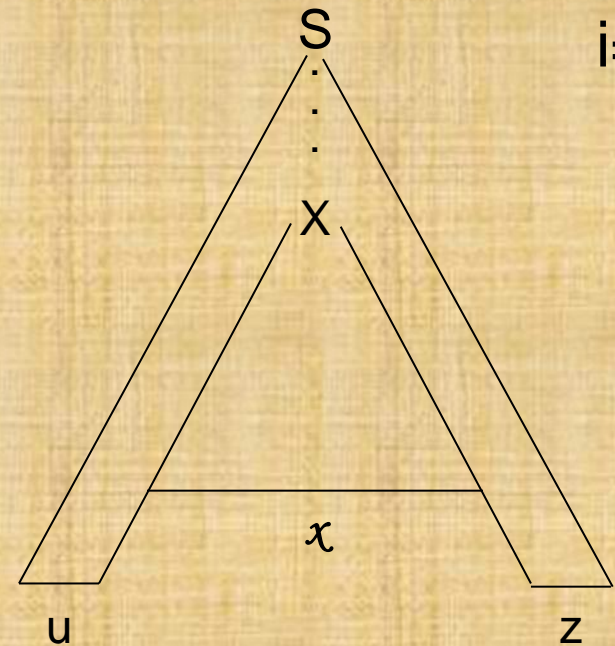
- ❑ cel mai mic arbore cu cel mai mare, obținem arborii de parsare ai cuvintelor uv^ixy^iz , $i > 1$;
- ❑ cel mai mare arbore cu cel mai mic, obținem arborele de parsare al cuvântului uxz (\Rightarrow cond. (ii));

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

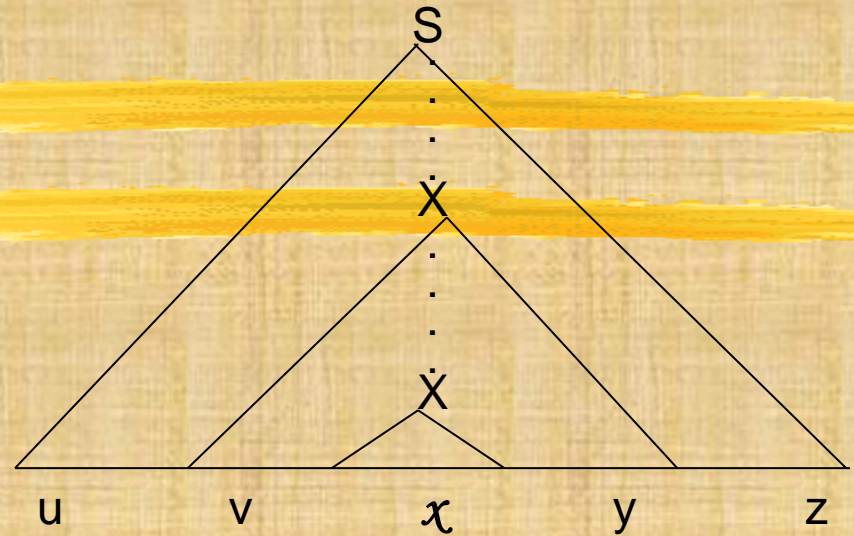
$i \geq 2$



$i=0$



\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT



Demonstratie (cont.)

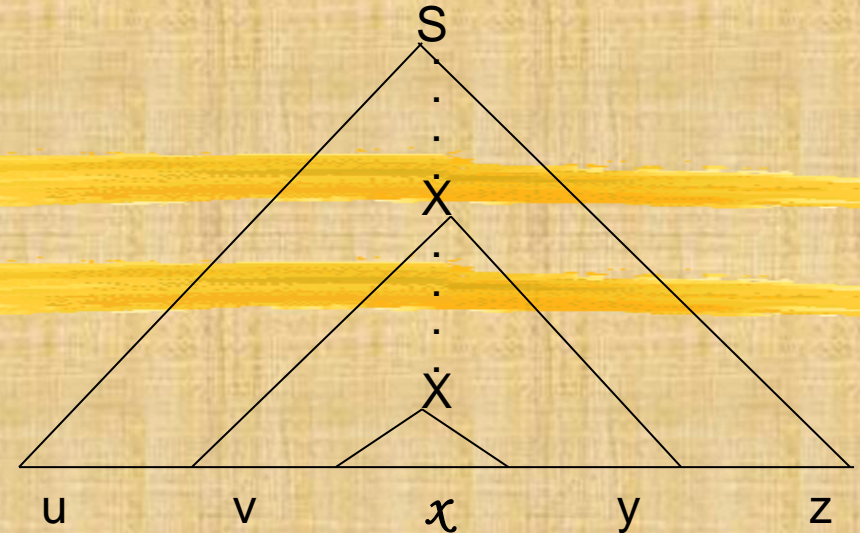
verificam cond.(iii) : $|vy| > 0$,

ppa $v=\varepsilon=y$

=> arborele de parsare obtinut prin inlocuirea celui mai mare subarboare de radacina X cu cel mai mic ar avea mai putine noduri decat τ și totusi ar genera cuvântul w

=> contradicție cu alegerea lui τ ca fiind arborele de parsare al lui w cu cel mai mic nr. de noduri.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT



Demonstratie (cont.)

verificam cond.(iv) : (iv) $|vxy| \leq p$

stim ca in τ , prima aparitie a lui X genereaza cel mai mare subarbore al lui w ,
și anume subarboarele asociat lui vxy ;

dar X a fost ales a.i. ambele sale aparitii sa fie cat mai aproape de frunza

=> aceste aparitii se afla printre ultimele $|V|+1$ noduri de pe drum (2)

in plus, drumul a fost ales ca fiind cel m. lung de la radacina la frunza (3)

din (2) și (3) rezulta ca inaltimea H a subarborelui de radacina X care
genereaza vxy este $H \leq |V|+2 \Rightarrow$

acest arbore poate genera un cuvant c , $|c| \leq b^{|V|+2} = p$.

q.e.d.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

1. Definitii

- gramatici și limbaje independente de context
- arbori de derivare
- proiectarea g.i.c.

2. Ambiguitate

3. Forme normale

- definitii
- exemple
- aducerea la forma normala Chomsky

4. Lema de pompare

5. Operatii de inchidere

- \mathcal{L}_2 este inchisa la reuniune, concatenare, operatia *, omomorfism
- \mathcal{L}_2 nu este inchisa la intersectie si complementara

6. Automatul pushdown;

echivalenta cu gramatica independenta de context.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Definitii 26

$\forall L_1, L_2 \subseteq \Sigma^*$:

$L_1 \cup L_2 = \{w \in \Sigma^* \mid w \in L_1 \text{ sau } w \in L_2\},$

$L_1 \cap L_2 = \{w \in \Sigma^* \mid w \in L_1 \text{ si } w \in L_2\},$

$L_1 \setminus L_2 = \{w \in \Sigma^* \mid w \in L_1 \text{ si } w \notin L_2\},$

$L_1 \circ L_2 = \{w_1 w_2 \in \Sigma^* \mid w_1 \in L_1 \text{ si } w_2 \in L_2\},$

$mi(L) = \{mi(w) \mid w \in L\}.$

$$\bar{L} = \{w \in \Sigma^* \mid w \notin L\} = \Sigma^* \setminus L,$$

$$L^n: \begin{cases} L^0 = \varepsilon & \text{si} \\ L^{n+1} = L \cdot L^n = L \cdot L^n, & \forall n \in N, \end{cases}$$

$$L \cdot \emptyset = \emptyset \cdot L = \emptyset,$$

$$L \cdot \{\varepsilon\} = \{\varepsilon\} \cdot L = L,$$

$$L^* = \bigcup_{i=0}^{\infty} L^i, \quad L^+ = \bigcup_{i=1}^{\infty} L^i.$$

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Lema 27

\mathcal{L}_2 este închisă la reuniune, concatenare și operația $*$.

Demonstratie

Fie $L_1, L_2 \in \mathcal{L}_2$, $L_i = L(G_i)$, unde $G_i = (V_i, \Sigma_i, S_i, P_i)$, $\forall i=1,2$:

$L_1 \cup L_2$ este generat de $G' = (V', \Sigma', S', P')$, unde:

$$V' = V_1 \cup V_2 \cup \{S_0\},$$

$$\Sigma' = \Sigma_1 \cup \Sigma_2,$$

$$S' = S_0,$$

$$P' = P_1 \cup P_2 \cup \{S_0 \rightarrow S_1, S_0 \rightarrow S_2\}.$$

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

$L_1 \circ L_2$ este generat de $G'=(V', \Sigma', S', P')$, unde:

$$V'=V_1 \cup V_2 \cup \{S_0\},$$

$$\Sigma' = \Sigma_1 \cup \Sigma_2,$$

$$S'=S_0,$$

$$P'=P_1 \cup P_2 \cup \{S_0 \rightarrow S_1 S_2\};$$

L^* este generat de $G'=(V', \Sigma', S', P')$, unde:

$$V'=V \cup \{S'\},$$

$$\Sigma' = \Sigma,$$

$$S'=\text{simbolul de start},$$

$$P'=P \cup \{S' \rightarrow S'S\};$$

Evident, toate cele 3 gramatici sunt independente de context

q.e.d.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Lema 28

\mathcal{L}_2 este inchisa la operatia mirror

Demonstratie

(i) Fie $L \in \mathcal{L}_2$, $L = L(G)$, unde $G = (V, \Sigma, S, P)$;

putem pp ca G este in FNC

\Rightarrow construim $G' = (V, \Sigma, S, P')$, unde

$P' = \{A \rightarrow CB \mid \exists A \rightarrow BC \text{ in } G\} \cup \{A \rightarrow a \mid \forall A \rightarrow a \text{ in } G\}$

$L(G') = \text{mi}(L)$, i.e. : $S_G \Rightarrow^* w \Leftrightarrow S_{G'} \Rightarrow^* \text{mi}(w)$, adica:

$S \Rightarrow x_1 \Rightarrow x_2 \Rightarrow \dots \Rightarrow x_n$ este o derivare in $G \Leftrightarrow$

$S \Rightarrow \text{mi}(x_1) \Rightarrow \text{mi}(x_2) \Rightarrow \dots \Rightarrow \text{mi}(x_n)$ este o derivare in G'

demonstratie prin inductie dupa $n = \text{nr pasi ai derivarii lui } x \text{ in } G$.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

$n=1$

$\Rightarrow x = \varepsilon$ sau $x \in \Sigma \Rightarrow$ evident

(se aplica productia $x \rightarrow \varepsilon$ sau $x \rightarrow a$ dar $a = mi(a)$ deci avem si productia corespunzatoare in G' .)

$n \geq 1$

Pp. afirmatia adevarata pt derivarea cuvintului x in $n \geq 1$ pasi si demonstram pt o derivare cu $n+1$ pasi:

Fie $S_G \Rightarrow^* x$, respectiv $S_{G'} \Rightarrow^* mi(x)$ in $n+1$ pasi \Rightarrow

1) Prima productie este obligatoriu de forma $S \rightarrow AB$, respectiv $S \rightarrow BA$

2) $\exists y \in \Sigma^*$ a.i. $x = yB \Rightarrow mi(x) = mi(yB) = Bmi(y)$

\Rightarrow lg derivarii lui y in G (și a lui $mi(y)$ in G') este n

\Downarrow cf. ip.ind. pt y

\exists o derivare pt y in G iff \exists o derivare pt $mi(y)$ in G' .

\Downarrow compunere cu $S \rightarrow AB$ ($S \rightarrow BA$)

$\exists S_G \Rightarrow^* x \Leftrightarrow S_{G'} \Rightarrow^* mi(x).$

q.e.d.₅₀

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Lema 30

\mathcal{L}_2 NU este inchisa la intersectie si complementara

Demonstratie

(i) Ppa. ca \mathcal{L}_2 este inchisa la \cap .

Fie $G_1 = (\{S, T\}, \{a, b, c\}, \{S \rightarrow Sc \mid Tc, T \rightarrow aTb \mid ab\}, S)$

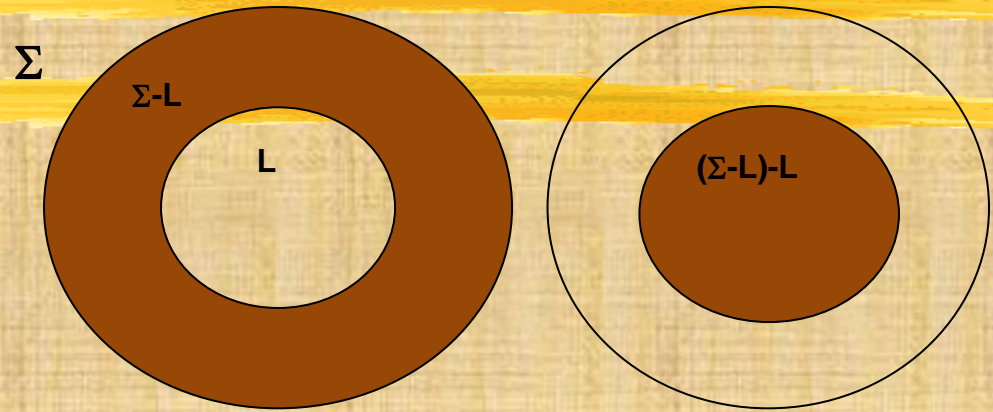
$G_2 = (\{S, T\}, \{a, b, c\}, \{S \rightarrow aS \mid aT, T \rightarrow bTc \mid bc\}, S)$

$\Rightarrow L(G_1) = \{a^n b^n c^m \mid n, m \geq 1\} \in \mathcal{L}_2$

$L(G_2) = \{a^m b^n c^n \mid n, m \geq 1\} \in \mathcal{L}_2$

Observam ca $L(G_1) \cap L(G_2) = \{a^n b^n c^n \mid n \geq 1\} \notin \mathcal{L}_2$ (cf. Lema pompare).

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT



(ii) Ppa ca \mathcal{L}_2 este inchisa la complementara;

fie $L = \{a^k b^j c^n \mid k, j, n \geq 1, k \neq j \neq n\}$; $L \in \mathcal{L}_3 \subset \mathcal{L}_2$: $L = L(G)$, unde

$G = (\{S, A, B, C\}, \{a, b, c\}, S, \{S \rightarrow aA, A \rightarrow aA \mid bB, B \rightarrow bB \mid cC, C \rightarrow cC \mid \varepsilon\})$

$\Rightarrow C(L) = \{a^k b^j c^n \mid k, j, n \geq 1, k = j = n\} = \{a^n b^n c^n \mid n \geq 1\} \notin \mathcal{L}_2$

Alta dem: fie $L \in \mathcal{L}_2 \Rightarrow L = \Sigma^* \cap L$ dar $\Sigma^* \cap L = C(L)$.

ppa ca \mathcal{L}_2 este inchisa la complementara $\Rightarrow \mathcal{L}_2$ este inchisa la intersectie

\Rightarrow contradictie cu (i)

q.e.d.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

1. Definitii

- gramatici și limbaje independente de context
- arbori de derivare
- proiectarea g.i.c.

2. Ambiguitate

3. Forme normale

- definitii
- exemple
- aducerea la forma normala Chomsky

4. Lema de pompare

5. Operatii de inchidere

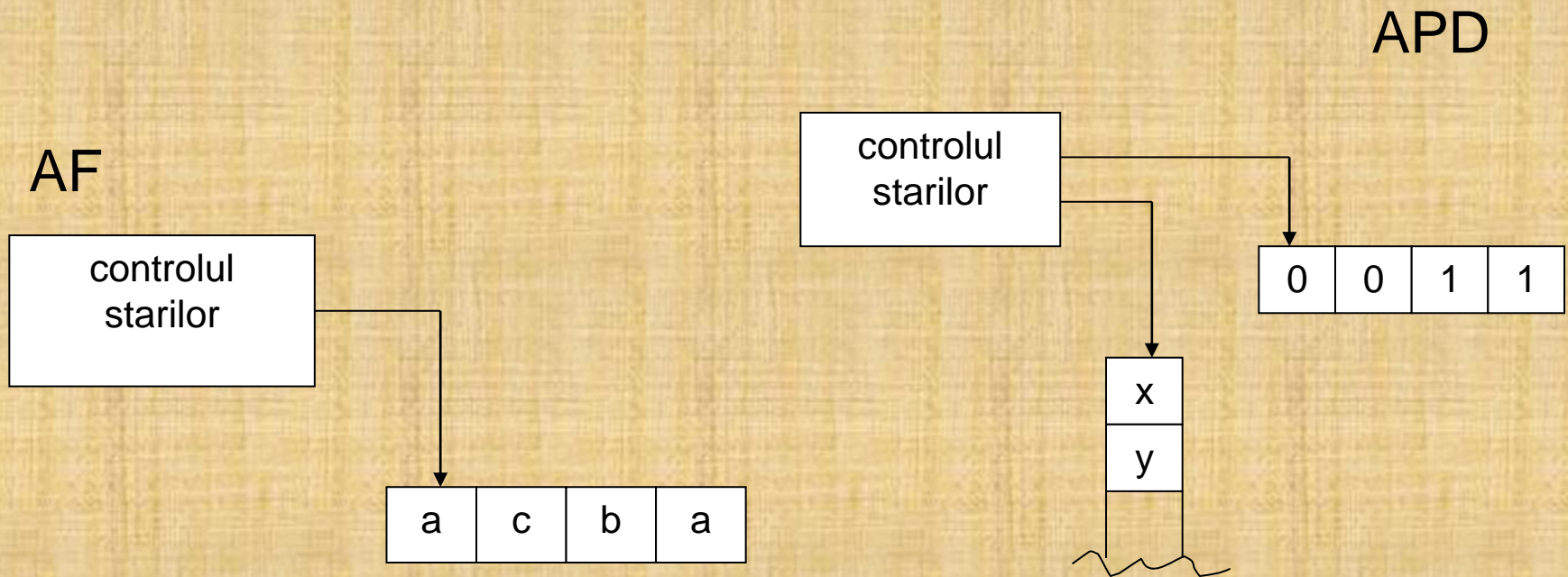
- \mathcal{L}_2 este inchisa la reuniune, concatenare, operatia *, omomorfism
- \mathcal{L}_2 nu este inchisa la intersectie si complementara

6. Automatul pushdown;

echivalenta cu gramatica independenta de context.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

APD reprezinta un nou model de calculabilitate
Stiva



$\{0^n 1^n \mid n \in \mathbb{N}\}$.

ℒFA: C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Observatii 32: Deosebiri intre APD si AFD:

(i) La APD trebuie sa luam in considerare:

- ✓ setul de stari,
- ✓ banda de intrare,
- ✓ stiva;

Simbolurile scrise in stiva pot fi preluate din acelasi alfabet / din alt alfabet decat cel al benzii de intrare

=> in definitia APD vom avea 2 alfabete: Σ si Γ ,
urmatoarea actiune a APD este determinata de:

- ✓ starea crt. a APD,
- ✓ simbolul citit de pe banda de intrare
- ✓ simbolul aflat in varful stivei,

(ii) APD poate intalni si pe banda de intrare si in stiva smb. vid, ε =>

$$\text{dom}(\delta) = Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\})$$

(iii) Urmatoarea actiune a APD poate consta in trecerea intr-o noua stare si EVENTUAL scrierea unui simbol in stiva;

In plus, modelul APD fiind intrinsec nedeterminist,

APD poate trece in diferite stari => $\text{codom}(\delta) = \mathcal{P}(Q \times (\Gamma \cup \{\varepsilon\}))$.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Definitie 33

Automat pushdown = APD = $(Q, \Sigma, \Gamma, \delta, q_0, F)$, unde:

Q = multime finita, nevida, ale carei elemente se numesc stari;

Σ = multime finita, nevida, numita alfabet de intrare, ale carei elemente se numesc simboluri, ($\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$);

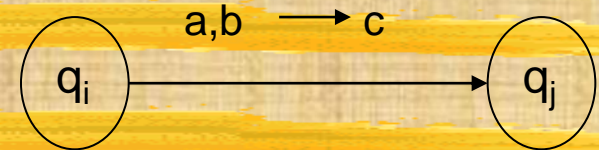
Γ = multime finita, nevida, numita alfabetul stivei, ($\Gamma_\epsilon = \Gamma \cup \{\epsilon\}$);

$\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow \mathcal{P}(Q \times \Gamma_\epsilon)$, numita functia de tranzitie;

$q_0 \in Q$, numita starea initiala;

$F \subseteq Q$ numita multimea starilor finale.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT



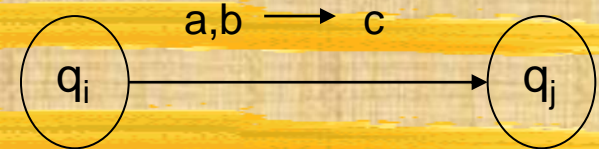
Notatie 34

\Leftrightarrow APD, aflat in starea q_i ,

- ✓ citește smb. a de pe banda de intrare,
- ✓ extrage smb. b din stiva,
- ✓ depune smb. c în stivă;

- Dacă $a = \varepsilon \Rightarrow$ APD face tranziția și fără să citească nimic de pe banda de intrare,
- Dacă $b = \varepsilon \Rightarrow$ APD face tranziția și fără să extragă nimic din stivă,
- Dacă $c = \varepsilon \Rightarrow$ APD face tranziția și fără să depună nimic în stivă.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT



Observatie 35

(i) Definitia formală a APD NU conține nici un mecanism explicit de testare a vidării stivei \Rightarrow

METODA STANDARD:

- ✓ se utilizează un smb. special din Γ , \$, care este depus în stivă de la început,
- ✓ când acest smb. este întâlnit (când s-a ajuns în vârful stivei) înseamnă că stivă s-a golit;

(ii) Analog, definiția formală a APD NU conține nici un mecanism explicit de testare a terminării secvenței de intrare \Rightarrow

METODA STANDARD:

- ✓ trecerea într-o stare finală.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

Teorema 36

Fie $L \subseteq \Sigma^* \Rightarrow (\exists G \in \mathcal{G}_2: L=L(G) \Leftrightarrow \exists A \in \text{APD}: L=L(A))$

Demonstratie “ \Rightarrow ”

Fie $L \subseteq \Sigma^*$ și fie $G=(V, \Sigma, S, P) \in \mathcal{G}_2$ ai. $L(G)=L$;

putem defini un APD $R=(Q, \Sigma_\varepsilon, \Gamma_\varepsilon, \delta, q_0, F)$ cu ajutorul lui G astfel:

$Q = \{q_{\text{start}}, q_{\text{loop}}, q_{\text{accept}}\} \cup E$:

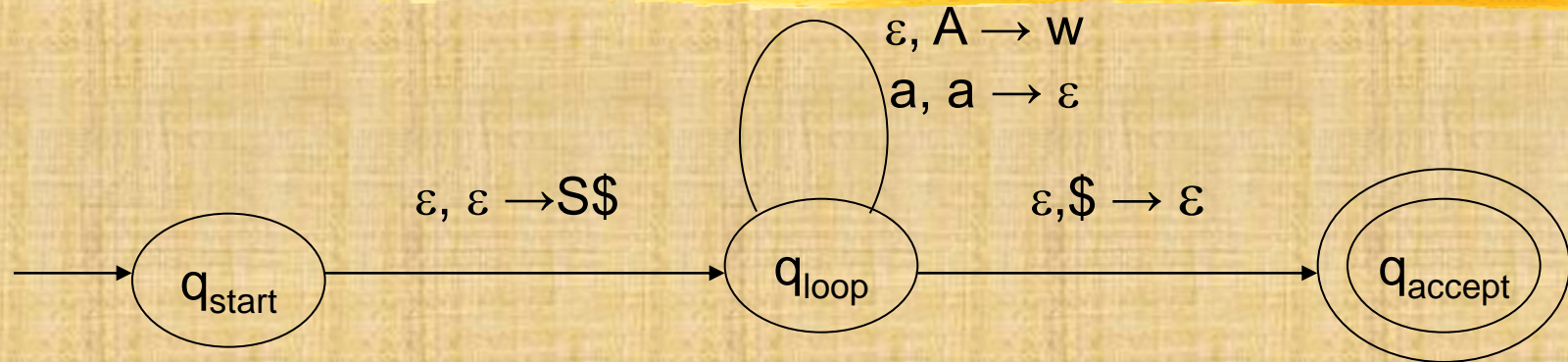
E = multimea starilor auxiliare necesare implementarii depunerii
in stiva a secventelor intermediare din derivarea $S \Rightarrow^* w, w \in L$;

$\Sigma_\varepsilon, \Gamma_\varepsilon$ depind de limbajul L considerat;

$q_0 = q_{\text{start}}$;

$F = \{q_{\text{accept}}\}$;

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT



$\delta : Q \times \Sigma_{\varepsilon} \times \Gamma_{\varepsilon} \rightarrow \mathcal{P}(Q \times \Gamma_{\varepsilon})$ definita prin:

$$\delta(q_{\text{start}}, \varepsilon, \varepsilon) = \{(q_{\text{loop}}, S\$)\}$$

$$\delta(q_{\text{loop}}, \varepsilon, A) = \{(q_{\text{loop}}, w) \mid \exists A \rightarrow w \in P\}$$

$$\delta(q_{\text{loop}}, a, a) = \{(q_{\text{loop}}, \varepsilon)\}$$

$$\delta(q_{\text{loop}}, \varepsilon, \$) = \{(q_{\text{accept}}, \varepsilon)\} .$$

q.e.d.

\mathcal{LFA} : C4 – LIMBAJE INDEPENDENTE DE CONTEXT

1. Definitii

- gramatici și limbaje independente de context
- arbori de derivare
- proiectarea g.i.c.

2. Ambiguitate

3. Forme normale

- definitii
- exemple
- aducerea la forma normala Chomsky

4. Lema de pompare

5. Operatii de inchidere

- \mathcal{L}_2 este inchisa la reuniune, concatenare, operatia *, omomorfism
- \mathcal{L}_2 nu este inchisa la intersectie si complementara

6. Automatul pushdown;

echivalenta cu gramatica independenta de context.