~ Seminar 6 ~

➢ Forma Normală Chomsky

O gramatică este (scrisă) în ***forma normală Chomsky*** dacă are doar producții care au în membrul stâng un neterminal, iar în membrul drept fie un terminal, fie două neterminale.
$\boldsymbol{A \rightarrow a}$
$\boldsymbol{A \rightarrow BC}$, unde $A, B, C \in N,\ a \in T$

În plus, dacă cuvântul vid **λ** trebuie generat de gramatică, atunci avem voie să avem și producția $\boldsymbol{S \rightarrow \lambda}$, dar atunci S nu are voie să apară în *membrul drept* al niciunei producții.

• *Algoritm:* Transformarea GIC → gramatică scrisă în F.N. Chomsky

▪ **<u>Pas 1:</u>** Se aplică **algoritmul de reducere** (se elimină simbolurile și producțiile nefolositoare):

a) Se elimină simbolurile și producțiile "**neutilizabile**".
Def: $X \in (N \cup T)$ este un simbol "neutilizabil" dacă nu există nicio derivare $\alpha X \beta \overset{*}{\Rightarrow} \alpha w \beta$ cu $\alpha, w, \beta \in T^*$.
(Adică plecând de la un cuvânt care conține simbolul X și aplicând oricât de multe producții, nu putem ajunge să generăm un cuvânt care sa nu conțină neterminale.)

Ideea intuitivă: Toate simbolurile din $T \cup \{\lambda\}$ sunt utilizabile. Găsim mulțimea simbolurilor din N care sunt **utilizabile**:
-- Plecăm de la cele care au cel puțin o producție cu membrul drept (de tip "caz de oprire") în T^*.
-- Iar apoi tot căutăm alte neterminale care au cel puțin o producție cu membrul drept în $(N \cup T)^*$, dar în care toate neterminalele fac deja parte din mulțimea celor utilizabile.

b) Se elimină simbolurile și producțiile "**inaccesibile**".
Def: $X \in (N \cup T)$ este un simbol "inaccesibil" dacă nu există nicio derivare $S \overset{*}{\Rightarrow} \alpha X \beta$, cu $\alpha, \beta \in (N \cup T)^*$.
(Adică plecând din S și aplicând oricât de multe producții, nu putem ajunge să generăm un cuvânt care să conțină simbolul X.)

Ideea intuitivă: Găsim mulțimea simbolurilor din N care sunt **accesibile**:
-- Plecăm de la neterminalele care apar în membrul drept al producțiilor lui S.
-- Apoi tot căutăm alte neterminale care apar în membrul drept al producțiilor acelor neterminale care fac deja parte din mulțimea celor accesibile.

- ***Exemplu:*** Aplicați algoritmul de transformare în F.N.Chomsky pe gramatica:

S → aABa | CD | bbAC
A → bc | d | λ
B → λ | E
C → A | dcabb | S
D → BAd | B
E → Ea | bbE
F → abc

Pas 1: Aplicăm ***algoritmul de reducere***.
a) Calculăm N_1 mulțimea neterminalelor ***utilizabile***.
$A \in N_1$ (pentru că A are cel puțin o producție cu membrul drept în T^*, de fapt are 3)
$B \in N_1$ (pentru că $B \rightarrow \lambda$ și $\lambda \in T^*$)
$C \in N_1$ (pentru că $C \rightarrow$ dcabb și dcabb $\in T^*$)
$F \in N_1$ (pentru că $F \rightarrow abc$ și $abc \in T^*$)

$S \in N_1$ (pentru că $S \rightarrow aABa$ și $aABa \in (T \cup N_1)^*$, la fel si pentru $S \rightarrow bbAC$)
$D \in N_1$ (pentru că $D \rightarrow B$ și $B \in T^*$, la fel și pentru $D \rightarrow BAd$)

$E \notin N_1$ (pentru că E nu are nicio producție cu membrul drept în $(T \cup N_1)^*$) => eliminăm din gramatică neterminalul E și toate producțiile în care apare acesta și obținem:
S → aABa | CD | bbAC
A → bc | d | λ
B → λ
C → A | dcabb | S
D → BAd | B
F → abc

b) Calculăm N_2 mulțimea neterminalelor ***accesibile***.
$S \in N_2$ (pentru că S este simbolul de start al gramaticii)
$A \in N_2$ și $B \in N_2$ (pentru că $S \in N_2$ și $S \rightarrow aABa$)
$C \in N_2$ și $D \in N_2$ (pentru că $S \in N_2$ și $S \rightarrow CD$)

$F \notin N_2$ (pentru că nu există niciun neterminal în N_2 care să aibă o producție în care să apară F în membrul drept) => eliminăm din gramatică neterminalul F și toate producțiile în care apare acesta și obținem:
S → aABa | CD | bbAC
A → bc | d | λ
B → λ
C → A | dcabb | S
D → BAd | B

- **Pas 2:** *Se elimină λ-producțiile.*

Obs: Dacă $\lambda \in L(G)$ (cuvântul vid trebuie să fie generat de gramatică), atunci avem voie să avem unica λ-producție a neterminalului de start, dar acesta nu are voie să mai apară în gramatică nicăieri (în membrul drept al producțiilor).
→ Se adaugă **un nou simbol de start S'** și producțiile $\boldsymbol{S' \rightarrow S \mid \lambda}$.

Vrem să nu mai avem în gramatică producţii care au ca membru drept cuvântul vid.
Pot exista două situaţii:

(i) Dacă neterminalul care are o λ–producţie NU are şi alte producţii, atunci
- va fi eliminată producţia lui;
- toate producţiile care au ca membru drept un cuvânt de lungime minim 2 în care apare acest neterminal → vor fi înlocuite prin eliminarea neterminalului din cuvinte;
- dacă membrul drept avea lungime 1 (adică era doar acest neterminal), atunci se înlocuieşte cu λ (dar şi această producţie va fi ulterior eliminată tot la „pasul 2”).

(ii) Dacă neterminalul care are o λ–producţie are şi alte producţii, atunci
- va fi eliminată doar λ–producţia lui;
- toate producţiile care au ca membru drept un cuvânt de lungime minim 2 în care apare acest neterminal → vor fi înlocuite atât de varianta în care cuvântul conţine neterminalul, cât şi de varianta în care neterminalul este eliminat din cuvânt;
- dacă membrul drept avea lungime 1 (adică era doar acest neterminal), atunci se înlocuieşte cu λ (dar şi această producţie va fi ulterior eliminată tot la „pasul 2”).

- ***Exemplu:***

Pas 2: *Se elimină λ-producțiile.*
Caz (i) → Eliminăm $B \to \lambda$ și neterminalul B => în toate producțiile în care apare B în membrul drept, ***înlocuim*** B cu λ (în cuvintele de lungime 1) sau îl ***eliminăm*** de tot pe B (în cuvintele de lungime ≥ 2).
S ⟶ aAa | CD | bbAC
A ⟶ bc | d | λ
C ⟶ A | dcabb | S
D ⟶ Ad | λ

Caz (ii)
→ Eliminăm $A \to \lambda$ => pentru toate producțiile în care apare A în membrul drept, ***păstrăm*** și variantele cu A (pentru că A are și alte producții non-lambda), dar ***adăugăm*** și producțiile în care înlocuim A cu λ (în cuvintele de lungime 1) sau îl eliminăm de tot pe A (în cuvintele de lungime ≥ 2).
S ⟶ aAa | aa | CD | bbAC | bbC
A ⟶ bc | d
C ⟶ A | λ | dcabb | S
D ⟶ Ad | d | λ

→ Eliminăm $C \to \lambda$ => pentru toate producțiile în care apare C în membrul drept, ***păstrăm*** și variantele cu C (pentru că C are și alte producții non-lambda), dar ***adăugăm*** și producțiile în care înlocuim C cu λ (în cuvintele de lungime 1) sau îl eliminăm de tot pe C (în cuvintele de lungime ≥ 2).
S ⟶ aAa | aa | CD | D | bbAC | bbA | bbC | bb
A ⟶ bc | d
C ⟶ A | dcabb | S
D ⟶ Ad | d | λ

→ Eliminăm $D \to \lambda$ => pentru toate producțiile în care apare D în membrul drept, ***păstrăm*** și variantele cu D (pentru că D are și alte producții non-lambda), dar ***adăugăm*** și producțiile în care înlocuim D cu λ (în cuvintele de lungime 1) sau îl eliminăm de tot pe D (în cuvintele de lungime ≥ 2).

S → aAa | aa | CD | C | D | λ | bbAC | bbA | bbC | bb
A → bc | d
C → A | dcabb | S
D → Ad | d

!! Gramatica trebuie să genereze λ (pentru că avem $S \to \lambda$), dar simbolul de start NU are voie să apară în membrul drept al nici unei producții (avem $C \to S$). => ***Adăugăm un nou simbol de start*** S' cu producțiile (S' → S | λ).

→ Eliminăm $S \to \lambda$ => pentru toate producțiile în care apare S în membrul drept, ***păstrăm*** și variantele cu S (pentru că S are și alte producții non-lambda), dar ***adăugăm*** și producțiile în care înlocuim S cu λ (în cuvintele de lungime 1) sau îl eliminăm de tot pe S (în cuvintele de lungime ≥ 2).

- *Observăm* că ar fi reapărut λ-producția $C \to \lambda$, dar nu o mai introducem pentru că am avut-o deja și am eliminat-o, deci am obținut deja toate producțiile care rezultau din eliminarea ei.

S' → S | λ
S → aAa | aa | CD | C | D | bbAC | bbA | bbC | bb
A → bc | d
C → A | dcabb | S
D → Ad | d

- **Pas 3:** *Se elimină redenumirile.*

Vrem să nu mai avem în gramatică producţii care au ca membru drept un neterminal.

Înlocuim neterminalul din dreapta cu toate cuvintele care sunt membru drept în producţiile sale.

Verificăm dacă acest neterminal din dreapta mai apare în dreapta în cadrul altor producţii:

- Daca nu, atunci eliminăm toate producţiile pe care le avea.
- Dacă mai apare în cuvinte de lungime minim 2, arunci trebuie să i le păstrăm.
- Dacă apare în cuvânt de lungime 1, înseamnă că este tot o redenumire şi va fi eliminată tot la „pasul 3”.

- ***Exemplu:***

Pas 3: *Se elimină redenumirile.*

→ Eliminăm $C \to A$ => Pentru toate producțiile $A \to \alpha$, adăugăm producțiile $C \to \alpha$.

S' → S | λ
S → aAa | aa | CD | C | D | bbAC | bbA | bbC | bb
A → bc | d
C → bc | d | dcabb | S
D → Ad | d

→ Eliminăm $S \to D$ => Pentru toate producțiile $D \to \alpha$, adăugăm producțiile $S \to \alpha$.

S' → S | λ
S → aAa | aa | CD | C | Ad | d | bbAC | bbA | bbC | bb

A → bc | d
C → bc | d | dcabb | S
D → Ad | d

→ Eliminăm $S \rightarrow C$ => Pentru toate producțiile $C \rightarrow \alpha$, adăugăm producțiile $S \rightarrow \alpha$ (atenție, adăugăm doar producțiile care nu există deja).
Observăm că ar apărea și producția $S \rightarrow S$, dar nu o adăugăm pentru că e inutilă.
S' → S | λ
S → aAa | aa | CD | bc | dcabb | Ad | d | bbAC | bbA | bbC | bb
A → bc | d
C → bc | d | dcabb | S
D → Ad | d

→ Eliminăm $C \rightarrow S$ => Pentru toate producțiile $S \rightarrow \alpha$, adăugăm producțiile $C \rightarrow \alpha$ (care nu există deja).
S' → S | λ
S → aAa | aa | CD | bc | dcabb | Ad | d | bbAC | bbA | bbC | bb
A → bc | d
C → bc | d | dcabb | aAa | aa | CD | Ad | bbAC | bbA | bbC | bb
D → Ad | d

Observație: sortăm alfabetic cuvintele de pe fiecare rând, pentru a fi mai ușor de urmărit în continuare.
S' → S | λ
S → aa | aAa | Ad | bb | bbA | bbAC | bbC | bc | CD | d | dcabb
A → bc | d
C → aa | aAa | Ad | bb | bbA | bbAC | bbC | bc | CD | d | dcabb
D → Ad | d

Acum se vede clar că neterminalele S și C au exact aceleași producții, deci putem simplifica gramatica prin eliminarea neterminalului C și a producțiilor lui, și prin înlocuirea lui C cu S în toate celelalte apariții.
S' → S | λ
S → aa | aAa | Ad | bb | bbA | bbAS | bbS | bc | d | dcabb | SD
A → bc | d
D → Ad | d

→ Eliminăm $S' \rightarrow S$ => Pentru toate producțiile $S \rightarrow \alpha$, adăugăm producțiile $S' \rightarrow \alpha$.
S' → aa | aAa | Ad | bb | bbA | bbAS | bbS | bc | d | dcabb | SD | λ
S → aa | aAa | Ad | bb | bbA | bbAS | bbS | bc | d | dcabb | SD
A → bc | d
D → Ad | d

- **<u>Pas 4:</u>** Se aplică din nou *algoritmul de reducere* (vezi Pas 1).

- ***Exemplu:***

Observăm că nu avem ce modifica, nu există neterminale neutilizabile sau inaccesibile.

- **Pas 5:** *Se adaugă neterminale noi pentru terminalele din cuvinte de lungime >1.*

Vrem ca terminalele să apară doar singure în membrul drept. De aceea, peste tot unde apar în componenţa unui cuvânt de lungime minim 2, le vom înlocui cu un neterminal nou şi vom adăuga producţia de la neterminalul nou la terminalul pe care l-a înlocuit.

- ***Exemplu:***

Pas 5: *Se adaugă neterminale noi pentru terminalele din cuvinte de lungime >1.*

$S' \to X_1X_1 \mid X_1AX_1 \mid AX_4 \mid X_2X_2 \mid X_2X_2A \mid X_2X_2AS \mid X_2X_2S \mid X_2X_3 \mid d \mid X_4X_3X_1X_2X_2 \mid SD \mid \lambda$
$S \to X_1X_1 \mid X_1AX_1 \mid AX_4 \mid X_2X_2 \mid X_2X_2A \mid X_2X_2AS \mid X_2X_2S \mid X_2X_3 \mid d \mid X_4X_3X_1X_2X_2 \mid SD$
$A \to X_2X_3 \mid d$
$D \to AX_4 \mid d$
$X_1 \to a \ ; \ X_2 \to b \ ; \ X_3 \to c \ ; \ X_4 \to d$

- **Pas 6:** *Se adaugă neterminale noi pentru „spargerea” cuvintelor de lungime >2.*

Vrem ca în dreapta să avem cuvinte formate din exact două neterminale. De aceea, unde avem cuvinte mai lungi, păstrăm doar primul neterminal din cuvânt şi îi alipim un neterminal nou, iar noul neterminal va avea o producţie cu membrul drept cuvântul pe care l-a înlocuit. Reluăm procedeul până când toate cuvintele ajung la lungimea 2.
Obs: Fiecare producţie care avea un cuvânt de lungime k va fi înlocuită de k–1 producţii cu cuvinte de lungime 2.

- ***Exemplu:***

Pas 6: *Se adaugă neterminale noi pentru „spargerea” cuvintelor de lungime >2.*

$S' \to X_1X_1 \mid X_1Y_1 \mid AX_4 \mid X_2X_2 \mid X_2Y_2 \mid X_2Y_3 \mid X_2Y_5 \mid X_2X_3 \mid d \mid X_4Y_6 \mid SD \mid \lambda$
$S \to X_1X_1 \mid X_1Y_1 \mid AX_4 \mid X_2X_2 \mid X_2Y_2 \mid X_2Y_3 \mid X_2Y_5 \mid X_2X_3 \mid d \mid X_4Y_6 \mid SD$
$A \to X_2X_3 \mid d$
$D \to AX_4 \mid d$
$X_1 \to a \ ; \ X_2 \to b \ ; \ X_3 \to c \ ; \ X_4 \to d$
$Y_1 \to AX_1 \ ; \ Y_2 \to X_2A \ ; \ Y_3 \to X_2Y_4 \ ; \ Y_4 \to AS \ ; \ Y_5 \to X_2S$
$Y_6 \to X_3Y_7 \ ; \ Y_7 \to X_1Y_8 \ ; \ Y_8 \to X_2X_2$

➢ *Lema de pompare pentru limbaje independente de context*

Fie L un limbaj independent de context.
Atunci $\exists p \in \mathbb{N}$ (număr natural) astfel încât pentru $\forall \alpha \in L$ cuvânt, cu $|\alpha| \geq p$, există o descompunere $\alpha = u \cdot v \cdot w \cdot x \cdot y$ cu proprietățile:

(1) $|v \cdot w \cdot x| \leq p$
(2) $|v \cdot x| \geq 1$
(3) $u \cdot v^i \cdot w \cdot x^i \cdot y \in L, \ \forall i \geq 0.$

→ ***Schema demonstrației***

- Vrem să demonstrăm că L **nu este** limbaj independent de context, folosind lema de pompare ***negată***.

- Presupunem prin reducere la absurd că L **este** limbaj independent de context.
  Atunci $\exists p \in \mathbb{N}$ și putem aplica lema de pompare. *(În continuare negăm afirmația lemei.)*
- ($\exists$) Alegem un cuvânt α din limbajul L astfel încât $|\alpha| \geq p$, $\forall p \in \mathbb{N}$. *Este important ca pentru acel α să NU poată fi construit un automat push-down (sau o gramatică independentă de context), pentru că altfel NU vom putea obține contradicția dorită.*
- Știm că $\alpha = u \cdot v \cdot w \cdot x \cdot y$. Vom presupune proprietățile (1) și (2) îndeplinite și vom găsi o contradicție pentru (3).
- ($\forall u, v, w, x, y$) Trebuie să analizăm pe rând ***orice*** împărțire posibilă a lui α în cele 5 componente (altfel spus, trebuie să poziționăm vwx în toate modurile posibile în α). Pentru ***fiecare*** caz, trebuie să alegem ($\exists$) câte un număr natural *i* (nu neapărat același pentru toate cazurile) astfel încât cuvântul $\beta = u \cdot v^i \cdot w \cdot x^i \cdot y \notin L$, rezultând o contradicție a lemei (a proprietății (3)), deci a presupunerii că limbajul L era independent de context.
  (Atenție, demonstrația este completă doar dacă obținem contradicție pentru fiecare posibilitate de descompunere a lui α, nu doar pe unele cazuri.)

- ***Exemple:*** Demonstrați că următoarele limbaje NU sunt independente de context.

$L1 = \{a^n b^n c^n \mid n \geq 0\}$

$L2 = \{a^n b^m c^n d^m \mid n \geq 0, m \geq 0\}$ *[făcut la seminar]*

$L3 = \{z \cdot z \mid z \in \{a, b\}^*\}$

$L4 = \{a^n b^m c^r \mid n > m > r \geq 0\}$ *[făcut la seminar]*

$L5 = \{a^r \mid r \text{ este număr prim}\}$

✓ ***Demonstrație*** *pentru* $\boldsymbol{L1 = \{a^n b^n c^n \mid n \geq 0\} \notin CFL}$

Presupunem prin reducere la absurd că L1 este limbaj independent de context (CFL). Atunci $\exists p \in \mathbb{N}$ și putem aplica lema de pompare. *(În continuare negăm afirmația lemei.)*

Alegem cuvântul $\boldsymbol{\alpha = a^p b^p c^p \in L1}$, cu $|\alpha| = 3p \geq p, \forall p \in \mathbb{N}$. Conform lemei, cuvântul poate fi scris sub forma $\alpha = u \cdot v \cdot w \cdot x \cdot y$ astfel încât $|v \cdot w \cdot x| \leq p$ și $|v \cdot x| \geq 1$, deci $1 \leq |vx| \leq p$ **(*)**.

Observăm că subșirul *vwx* (de lungime maxim p) nu poate conține simultan litere de a, b și c (pentru că lungimea ar depăși cele p litere de b) => *vwx* poate conține doar 1 sau 2 tipuri de litere, deci există 5 cazuri:

Caz I: $vwx \in a^+$ => $\boldsymbol{vx = a^k}$, din (*) => $1 \leq k \leq p$.
Alegem $i = 2$ => $\boldsymbol{\beta} = u \cdot v^2 \cdot w \cdot x^2 \cdot y = \boldsymbol{a^{p+k} b^p c^p} \in \boldsymbol{L1}$
<=> $|\beta|_a = |\beta|_b$ <=> $p + k = p$ <=> $k = 0$, <u>contradicție</u> cu $1 \leq k$ **(rel 1).**

Caz II: : $vwx \in b^+$ => $\boldsymbol{vx = b^k}$ și
Caz III: : $vwx \in c^+$ => $\boldsymbol{vx = c^k}$
se tratează analog cu Caz I => contradicții **(rel 2 și rel 3)**.

Caz IV: $vwx \in a^+b^+$ => $\boldsymbol{vx = a^k b^s}$, din (*) => $1 \le k + s \le p$.
Alegem $i = 0$ => $\boldsymbol{\beta} = u \cdot v^0 \cdot w \cdot x^0 \cdot y = u \cdot w \cdot y = \boldsymbol{a^{p-k} b^{p-s} c^p} \in \boldsymbol{L1}$
<=> $|\beta|_a = |\beta|_b = |\beta|_c$ <=> $p - k = p - s = p$
<=> $k = s = 0$, contradicție cu $1 \le k + s$ **(rel 4)**.

Caz V: $vwx \in b^+c^+$ => $\boldsymbol{vx = b^k c^s}$
se tratează analog cu Caz IV => contradicție **(rel 5)**.

Concluzie: Din relațiile 1 – 5 (am reușit să obținem contradicții pentru *toate* 5 cazurile de descompunere) rezultă că presupunerea făcută este falsă => $L1 \notin CFL$. ■

✓ ***Demonstrație*** *pentru* $\boldsymbol{L3 = \{z \cdot z \mid z \in \{a, b\}^*\} \notin CFL}$

Presupunem că L3 este limbaj independent de context, rezultă că există acel număr natural *p* din lemă *(numit lungimea de pompare)*.

Alegem $\alpha = \boldsymbol{a^p b^p a^p b^p} = \underbrace{a \dots a}_{p}\underbrace{b \dots b}_{p}\underbrace{a \dots a}_{p}\underbrace{b \dots b}_{p} \in L3$ => $|\alpha| = 4p \ge p, \forall p \in \mathbb{N}$.

Avem $\alpha = u \cdot v \cdot w \cdot x \cdot y$ astfel încât $|v \cdot w \cdot x| \le p$ și $|v \cdot x| \ge 1$,
=> $1 \le |vx| \le p$ *(pentru că w are voie să fie inclusiv cuvântul vid).*

Caz I: Dacă vwx este în prima jumătate a lui α, atunci alegem $i = 2$ și rezultă după pompare cuvântul $\beta = u \cdot v^2 \cdot w \cdot x^2 \cdot y$ => $|\beta| = |\alpha| + |vx|$ => $|\beta|$ este mai mare decât $|\alpha|$ cu maxim *p* litere (și minim *o* literă) => jumătatea cuvântului se mută spre stânga cu maxim *p*/2 poziții (și minim 1/2), deci nu va mai fi între un "b" și un "a", ci va fi între doi de "b" *(indiferent dacă pompăm doar a-uri, doar b-uri, sau amândouă în prima jumătate a cuvântului)*. Rezultă că cuvântul β începe cu litera "a", dar prima literă din a doua lui jumătate este "b", deci cuvântul $\beta \notin L3$ *(pentru că nu este de forma $z \cdot z$)*, contradicție cu proprietatea (3) din lemă. ***(rel. 1)***

Caz II: Dacă vwx este în a doua jumătate a lui α, atunci alegem $i = 2$ și rezultă după pompare cuvântul $\beta = u \cdot v^2 \cdot w \cdot x^2 \cdot y$ => $|\beta| = |\alpha| + |vx|$ => $|\beta|$ este mai mare decât $|\alpha|$ cu maxim *p* litere (și minim *o* literă) => jumătatea cuvântului se mută spre dreapta cu maxim *p*/2 poziții (și minim 1/2), deci nu va mai fi între un "b" și un "a", ci va fi între doi de "a" *(indiferent dacă pompăm doar a-uri, doar b-uri, sau amândouă în a doua jumătate a cuvântului)*. Rezultă că cuvântul β se termină cu litera "b", dar ultima literă din prima lui jumătate este "a", deci cuvântul $\beta \notin L3$ *(pentru că nu este de forma $z \cdot z$)*, contradicție cu proprietatea (3) din lemă. ***(rel. 2)***

Caz III: Dacă vwx intersectează mijlocul cuvântului α (vwx conține cel puțin una din cele 2 litere din mijloc), atunci alegem $i = 0$ și rezultă după pompare cuvântul $\beta = u \cdot v^0 \cdot w \cdot x^0 \cdot y = u \cdot w \cdot y$ care este de forma $a^p b^{p-s} a^{p-r} b^p$, cu $1 \leq s + r \leq p$. Rezultă că cuvântul β are mai puțini de "b" în prima parte decât la final sau are mai puțini de "a" în a doua parte decât la început (sau ambele), deci nu este de forma $z \cdot z$ => $\beta \notin L3$, contradicție cu proprietatea (3) din lemă. ***(rel. 3)***

Concluzie: Din relațiile 1 – 3 (am reușit să obținem contradicții pentru *toate* 3 cazurile de descompunere) rezultă că presupunerea făcută este falsă => $L3 \notin CFL$. ■

✓ ***Demonstrație*** *pentru* $\boldsymbol{L5 = \{a^r \mid r \text{ este număr prim}\} \notin CFL}$

Presupunem că L5 este limbaj independent de context, rezultă că există acel număr natural *p* din lemă.
Alegem $\boldsymbol{\alpha = a^n}$, unde $\boldsymbol{n =}$ **nr. prim**, $\boldsymbol{n \geq p + 2}$ *(cel mai mic număr natural este 0, dar cel mai mic număr prim este 2)* => $|\alpha| = n \geq p + 2 \geq p, \forall p \in \mathbb{N}$.

Avem $\alpha = u \cdot v \cdot w \cdot x \cdot y$ astfel încât $|v \cdot w \cdot x| \leq p$ și $|v \cdot x| \geq 1$.
Fie $\boldsymbol{vx = a^k}$ => $|vx| = k$.
Din relațiile de mai sus rezultă $1 \leq k = |vx| \leq |vwx| \leq p \leq n - 2$ **(rel. *)**.

Trebuie să alegem un număr natural *i* pentru care după pompare rezultă cuvântul $\beta = u \cdot v^i \cdot w \cdot x^i \cdot y$ despre care trebuie să demonstrăm că NU aparține limbajului L5, adică să arătăm că lungimea cuvântului β ***nu poate fi un număr prim***.

Avem $|\beta| = |u \cdot v^i \cdot w \cdot x^i \cdot y| = |u \cdot v^1 \cdot w \cdot x^1 \cdot y| + |v^{i-1} \cdot x^{i-1}|$

$= |uvwxy| + (i - 1) * |vx| = |\alpha| + (i - 1) * |vx| = n + (i - 1) * k = n - k + i * k$

Pentru că avem nevoie să dăm un factor comun în această expresie, observăm că putem alege $\boldsymbol{i = n + 1}$ => $|\beta| = n + (n + 1 - 1) * k = n + n * k = \boldsymbol{n * (k + 1)}$.
Dar știm că $n =$ nr. prim, deci $\boldsymbol{n \geq 2}$, iar conform (*) rezultă $\boldsymbol{k + 1 \geq 2}$.
Pentru că $|\beta|$ este egal cu produsul a două numere ≥ 2 => $\boldsymbol{|\beta| \neq}$ **nr. prim** => $\beta \notin L5$, contradicție cu proprietatea (3) din lemă.

Concluzie: am reușit să obținem contradicție pentru singurul caz posibil de descompunere, rezultă că presupunerea făcută este falsă => $L5 \notin CFL$. ■

Observație:

Am fi putut alege și $\boldsymbol{i = n - k}$ => $|\beta| = n - k + (n - k) * k = (n - k) * (k + 1)$, care conform inegalităților din relația (*) este tot un produs de două numere ≥ 2.