

Limbaje Formale, Automate și Compilatoare

Curs 1

2017-18

Limbaje Formale, Automate și Compilatoare - Curs 1

- 1 **Prezentare curs**
- 2 Limbaje formale
- 3 Mecanisme de generare a limbajelor: gramatici
- 4 Ierarhia lui Chomsky
- 5 Limbaje și gramatici de tip 3 (regulate)

Limbaje Formale, Automate și Compilatoare

Titulari curs:

- O. Captarencu: otto@info.uaic.iasi.ro
<http://profs.info.uaic.ro/~otto/lfac.html>
- A. Moruz:mmoruz@info.uaic.ro

Sistem evaluare

- 7 seminarii, 6 laboratoare;
- **AS** = activitatea la seminar (max 10 puncte);
- **AL** = activitatea la laborator (max 10 puncte);
- **T1, T2** teste scrise în săptămânile 8, respectiv în sesiune;

Punctajul final se obține astfel:

$$P = 3 * AS + 3 * AL + 2 * T1 + 2 * T2$$

- Condiții minime de promovare: **$AS \geq 5$, $AL \geq 5$, $T1 \geq 4$, $T2 \geq 4$** ;
- Punctaj minim pentru promovare: **$P \geq 50$** ;
- Nota finală se va stabili conform criteriilor ECTS;

Sistem evaluare

- **AS** = activitatea la seminar (max 10 puncte):
 - două teste scrise
 - până la 2 puncte bonus pentru activitatea din timpul seminarului
- **AL** = activitatea la laborator (max 10 puncte):
 - 1 test laborator, 1 proiect (note de la 0 la 10)
 - **AL** = media celor 2 note

Tematica cursului (partea I)

- Limbaje și gramatici
- Limbaje regulate; gramatici, automate , expresii regulate
- Limbaje independente de context; gramatici, automate pushdown

Tematica cursului (partea II)

- Limbaje de programare: proiectare și implementare
- Analiza lexicală
- Analiza sintactică
- Traducere în cod intermediar

Tematica seminarului

- Exemple de limbaje și gramatici
- Automate finite deterministe, nedeterministe, cu epsilon-tranziții - Exemple
- Expresii regulate
- Gramatici independente de context, arbori de derivare, eliminarea simbolurilor inutile, eliminarea regulilor de ștergere, a redenumirilor
- Forma normală Chomsky, algoritmul CYK
- Automate pushdown - exemple

Tematica laboratorului

- Analiza lexicală folosind instrumente de tip LEX
- Analiza sintactică folosind instrumente de tip YACC
- Interpretor construit cu LEX și YACC

Bibliografie (selecții)

- ❶ A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman: Compilers: Principles, Techniques, and Tools. Boston: Addison-Wesley, 2007
- ❷ Gh. Grigoras. Constructia compilatoarelor - Algoritmi fundamentali, Ed. Universitatii Al. I. "Cuza Iasi", ISBN 973-703-084-2, 274 pg., 2005
- ❸ Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D. (2006). Introduction to Automata Theory, Languages, and Computation (3rd ed.). Addison-Wesley
- ❹ J. Toader - Limbaje formale și automate, Editura Matrix Rom, Bucuresti, 1999.
- ❺ J. Toader, S. Andrei - Limbaje formale și teoria automatelor. Teorie și practică, Editura Universitatii "Al. I. Cuza", Iasi, 2002.

Limbaje Formale, Automate și Compilatoare - Curs 1

- 1 Prezentare curs
- 2 Limbaje formale**
- 3 Mecanisme de generare a limbajelor: gramatici
- 4 Ierarhia lui Chomsky
- 5 Limbaje și gramatici de tip 3 (regulate)

Alfabet, cuvânt, mulțime de cuvinte

- **Alfabet:** V o mulțime finită (elementele lui V = simboluri)

Alfabet, cuvânt, mulțime de cuvinte

- **Alfabet:** V o mulțime finită (elementele lui V = simboluri)
- **Cuvânt:** șir finit de simboluri
 - cuvântul nul este notat cu ϵ sau λ .

Alfabet, cuvânt, mulțime de cuvinte

- **Alfabet:** V o mulțime finită (elementele lui V = simboluri)
- **Cuvânt:** șir finit de simboluri
 - cuvântul nul este notat cu ϵ sau λ .
- Lungimea unui cuvânt u : numărul simbolurilor sale. Notăție: $|u|$.
 $|\epsilon| = 0$

Alfabet, cuvânt, mulțime de cuvinte

- **Alfabet:** V o mulțime finită (elementele lui V = simboluri)
- **Cuvânt:** șir finit de simboluri
 - cuvântul nul este notat cu ϵ sau λ .
- Lungimea unui cuvânt u : numărul simbolurilor sale. Notăție: $|u|$.
 $|\epsilon| = 0$
- V^* - mulțimea tuturor cuvintelor peste alfabetul V , inclusiv ϵ .
 $\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \dots\}$

Alfabet, cuvânt, mulțime de cuvinte

- **Alfabet:** V o mulțime finită (elementele lui V = simboluri)
- **Cuvânt:** șir finit de simboluri
 - cuvântul nul este notat cu ϵ sau λ .
- Lungimea unui cuvânt u : numărul simbolurilor sale. Notăție: $|u|$.
 $|\epsilon| = 0$
- V^* - mulțimea tuturor cuvintelor peste alfabetul V , inclusiv ϵ .
 $\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \dots\}$
- V^+ - mulțimea tuturor cuvintelor nenule peste alfabetul V
 $\{0, 1\}^+ = \{0, 1, 00, 01, 10, 11, 000, 001, \dots\}$

Operații pe cuvinte

- **Concatenarea** a doua cuvinte x, y : cuvântul $x \cdot y$ obținut din simbolurile lui x , în ordinea în care apar, urmate de cele ale lui y de asemenea în ordinea în care apar:

$$x = 0100, y = 100, x \cdot y = 0100100$$

$$x = 000, y = \epsilon, x \cdot y = 000$$

Operații pe cuvinte

- **Concatenarea** a doua cuvinte x, y : cuvântul $x \cdot y$ obținut din simbolurile lui x , în ordinea în care apar, urmate de cele ale lui y de asemenea în ordinea în care apar:

$$x = 0100, y = 100, x \cdot y = 0100100$$

$$x = 000, y = \epsilon, x \cdot y = 000$$

- Concatenarea este asociativă

Operații pe cuvinte

- **Concatenarea** a doua cuvinte x, y : cuvântul $x \cdot y$ obținut din simbolurile lui x , în ordinea în care apar, urmate de cele ale lui y de asemenea în ordinea în care apar:
 $x = 0100, y = 100, x \cdot y = 0100100$
 $x = 000, y = \epsilon, x \cdot y = 000$
- Concatenarea este asociativă
- (V^*, \cdot) este monoid (ϵ este element neutru), se numește monoidul liber generat de V .

Operații pe cuvinte

- **Concatenarea** a doua cuvinte x, y : cuvântul $x \cdot y$ obținut din simbolurile lui x , în ordinea în care apar, urmate de cele ale lui y de asemenea în ordinea în care apar:
 $x = 0100, y = 100, x \cdot y = 0100100$
 $x = 000, y = \epsilon, x \cdot y = 000$
- Concatenarea este asociativă
- (V^*, \cdot) este monoid (ϵ este element neutru), se numește monoidul liber generat de V .
- Cuvântul v este un **prefix** al cuvântului u dacă $\exists w \in V^* : u = vw$; dacă $w \in V^+$, atunci v este un **prefix propriu** al lui u .

Operații pe cuvinte

- **Concatenarea** a doua cuvinte x, y : cuvântul $x \cdot y$ obținut din simbolurile lui x , în ordinea în care apar, urmate de cele ale lui y de asemenea în ordinea în care apar:
 $x = 0100, y = 100, x \cdot y = 0100100$
 $x = 000, y = \epsilon, x \cdot y = 000$
- Concatenarea este asociativă
- (V^*, \cdot) este monoid (ϵ este element neutru), se numește monoidul liber generat de V .
- Cuvântul v este un **prefix** al cuvântului u dacă $\exists w \in V^* : u = vw$; dacă $w \in V^+$, atunci v este un **prefix propriu** al lui u .
- Cuvântul v este un **sufix** al cuvântului u dacă $\exists w \in V^* : u = wv$; dacă $w \in V^+$, atunci v este un **sufix propriu** al lui u .

- Fie V un alfabet. O submulțime $L \subseteq V^*$ este un **limbaj** (formal) peste alfabetul V (sau V -limbaj) dacă L are o descriere (matematică) finită.
- O descriere poate fi:

- Fie V un alfabet. O submulțime $L \subseteq V^*$ este un **limbaj** (formal) peste alfabetul V (sau V -limbaj) dacă L are o descriere (matematică) finită.
- O descriere poate fi:
 - neformală (în limbaj natural):
 - mulțimea cuvintelor peste alfabetul $\{0, 1\}$ care contin un număr par de 0.
 - $L = \{x \in V^+ : |x| \text{ este par}\}$.
 - $\{a^n b^n | n \in \mathbb{N}\}$.
 - $\{w \in \{0, 1\}^* | w \text{ se termină în } 00\}$.

- Fie V un alfabet. O submulțime $L \subseteq V^*$ este un **limbaj** (formal) peste alfabetul V (sau V -limbaj) dacă L are o descriere (matematică) finită.
- O descriere poate fi:
 - neformală (în limbaj natural):
 - mulțimea cuvintelor peste alfabetul $\{0, 1\}$ care contin un numar par de 0.
 - $L = \{x \in V^+ : |x| \text{ este par}\}$.
 - $\{a^n b^n | n \in \mathbb{N}\}$.
 - $\{w \in \{0, 1\}^* | w \text{ se termina in } 00\}$.
 - formală (descriere matematică):
 - o descriere inductivă a cuvintelor
 - o descriere generativă a cuvintelor (gramatică generativă)
 - o descriere a unei metode de recunoaștere a cuvintelor din limbaj (automat finit, automat pushdown, etc.)

Operații cu limbaje

- Operațiile cu mulțimi (reuniune, intersecție etc)
- Produs de limbaje: $L_1 \cdot L_2 = \{u \cdot v \mid u \in L_1, v \in L_2\}$
- Iterația (produsul Kleene): $L^* = \bigcup_{n \geq 0} L^n$, unde:
 - $L^0 = \{\epsilon\}$
 - $L^{n+1} = L^n \cdot L$
- $L^R = \{w^R \mid w \in L\}$; dacă $w = a_1 a_2 \dots a_n$, atunci $w^R = a_n \dots a_2 a_1$

Limbaje Formale, Automate și Compilatoare - Curs 1

- 1 Prezentare curs
- 2 Limbaje formale
- 3 Mecanisme de generare a limbajelor: gramatici**
- 4 Ierarhia lui Chomsky
- 5 Limbaje și gramatici de tip 3 (regulate)

Gramatici

Definiție 1

O gramatica este un sistem $G = (N, T, S, P)$, unde:

- *N și T sunt două alfabete disjuncte:*
 - *N este multimea neterminalilor*
 - *T este multimea terminalilor*
- *$S \in N$ este simbolul de start (neterminalul inițial)*
- *P este o multime finită de reguli (producții) de forma $x \rightarrow y$, unde $x, y \in (N \cup T)^*$ și x conține cel puțin un neterminal.*

Derivare

Definiție 2

Fie $G = (N, T, S, P)$ o gramatică și $u, v \in (N \cup T)^*$.

Spunem că v este derivat direct (într-un pas) de la u prin aplicarea regulii $x \rightarrow y$, și notăm $u \Rightarrow v$, dacă $\exists p, q \in (N \cup T)^*$ astfel încât $u = pxq$ și $v = pyq$.

Derivare

Definiție 2

Fie $G = (N, T, S, P)$ o gramatică și $u, v \in (N \cup T)^*$.

Spunem că v este derivat direct (într-un pas) de la u prin aplicarea regulii $x \rightarrow y$, și notăm $u \Rightarrow v$, dacă $\exists p, q \in (N \cup T)^*$ astfel încât $u = pxq$ și $v = pyq$.

- Dacă $u_1 \Rightarrow u_2 \dots \Rightarrow u_n, n > 1$, spunem ca u_n este derivat din u_1 în G și notăm $u_1 \Rightarrow^+ u_n$.

Derivare

Definiție 2

Fie $G = (N, T, S, P)$ o gramatică și $u, v \in (N \cup T)^*$.

Spunem că v este derivat direct (într-un pas) de la u prin aplicarea regulii $x \rightarrow y$, și notăm $u \Rightarrow v$, dacă $\exists p, q \in (N \cup T)^*$ astfel încât $u = pxq$ și $v = pyq$.

- Dacă $u_1 \Rightarrow u_2 \dots \Rightarrow u_n, n > 1$, spunem ca u_n este derivat din u_1 în G și notăm $u_1 \Rightarrow^+ u_n$.
- Scriem $u \Rightarrow^* v$ dacă $u \Rightarrow^+ v$ sau $u = v$.

Limбай generat

Definiție 3

Limбайul generat de gramatica G este:

$$L(G) = \{w \in T^* \mid S \Rightarrow^+ w\}$$

Limбай generat

Definiție 3

Limбайul generat de gramatica G este:

$$L(G) = \{w \in T^* \mid S \Rightarrow^+ w\}$$

Definiție 4

Două gramatici G_1 și G_2 sunt echivalente dacă $L(G_1) = L(G_2)$.

Exemplu

- $L = \{a^n b^n | n \geq 1\}$
- Definiția inductivă:
 - $ab \in L$
 - Dacă $X \in L$, atunci $aXb \in L$
 - Nici un alt cuvânt nu face parte din L

Exemplu

- $L = \{a^n b^n | n \geq 1\}$
- Definiția inductivă:
 - $ab \in L$
 - Dacă $X \in L$, atunci $aXb \in L$
 - Nici un alt cuvânt nu face parte din L
- Definiția generativă:
 - $G = (\{X\}, \{a, b\}, X, P)$, unde $P = \{X \rightarrow aXb, X \rightarrow ab\}$
 - Derivarea cuvântului $a^3 b^3$:

$$X \Rightarrow aXb \Rightarrow a(aXb)b \Rightarrow aa(ab)bb$$

Exemplu

- $L = \{a^n b^n c^n | n \geq 1\}$
- $= (N, T, S, P)$, $N = \{S, X\}$, $T = \{a, b, c\}$, P constă din:

① $S \rightarrow abc$

② $S \rightarrow aSXc$

③ $cX \rightarrow Xc$

④ $bX \rightarrow bb$

- Derivarea cuvântului $a^3 b^3 c^3$:

$$S \Rightarrow^{(2)} a\underline{S}Xc \Rightarrow^{(2)} aa\underline{S}XcXc \Rightarrow^{(1)} aaabc\underline{c}XcXc \Rightarrow^{(3)}$$

$$aaab\underline{X}ccXc \Rightarrow^{(4)} aaabbcc\underline{X}c \Rightarrow^{(3)} aaabb\underline{c}Xcc \Rightarrow^{(3)}$$

$$aaabb\underline{X}ccc \Rightarrow^{(4)} aaabbbccc = a^3 b^3 c^3$$

Limbae Formale, Automate și Compilatoare - Curs 1

- 1 Prezentare curs
- 2 Limbaje formale
- 3 Mecanisme de generare a limbajelor: gramatici
- 4 Ierarhia lui Chomsky**
- 5 Limbaje și gramatici de tip 3 (regulate)

Ierarhia lui Chomsky

1 Gramatici de tip 0 (generale)

Nu exista restrictii asupra regulilor

Ierarhia lui Chomsky

1 Gramatici de tip 0 (generale)

Nu exista restrictii asupra regulilor

2 Gramatici de tip 1 (dependente de context)

reguli de forma $pxq \rightarrow pyq$ unde $x \in N$, $y \neq \epsilon$, $p, q \in (N \cup T)^*$,
 $S \rightarrow \epsilon$, caz în care S nu apare în dreapta regulilor

Ierarhia lui Chomsky

1 Gramatici de tip 0 (generale)

Nu exista restrictii asupra regulilor

2 Gramatici de tip 1 (dependente de context)

reguli de forma $pxq \rightarrow pyq$ unde $x \in N$, $y \neq \epsilon$, $p, q \in (N \cup T)^*$,
 $S \rightarrow \epsilon$, caz în care S nu apare în dreapta regulilor

3 Gramatici de tip 2 (independente de context)

reguli de forma $A \rightarrow y$ unde $A \in N$ și $y \in (N \cup T)^*$

Ierarhia lui Chomsky

1 Gramatici de tip 0 (generale)

Nu exista restrictii asupra regulilor

2 Gramatici de tip 1 (dependente de context)

reguli de forma $pxq \rightarrow pyq$ unde $x \in N$, $y \neq \epsilon$, $p, q \in (N \cup T)^*$,
 $S \rightarrow \epsilon$, caz în care S nu apare în dreapta regulilor

3 Gramatici de tip 2 (independente de context)

reguli de forma $A \rightarrow y$ unde $A \in N$ și $y \in (N \cup T)^*$

4 Gramatici de tip 3 (regulate)

reguli $A \rightarrow u$ sau $A \rightarrow uB$ unde $A, B \in N$ și $u \in T^*$.

Exemple

Ce tip au urmatoarele gramatici?

- $G = (N, T, S, P)$, $N = \{S, A, B\}$, $T = \{a, b, c\}$, P :

$$(1) S \rightarrow aaAc$$

$$(2) aAc \rightarrow aAbBc$$

$$(3) bB \rightarrow bBc$$

$$(4) Bc \rightarrow Abc$$

$$(5) A \rightarrow a$$

- $G = (N, T, S, P)$, $N = \{S, X\}$, $T = \{a, b, c\}$, P :

$$(1) S \rightarrow abc$$

$$(2) S \rightarrow aSXc$$

$$(3) cX \rightarrow Xc$$

$$(4) bX \rightarrow bb$$

Exemple

- Fie

$$G = (\{E\}, \{a, +, -, (,)\}, E, \{E \rightarrow a, E \rightarrow (E + E), E \rightarrow (E - E)\}).$$

- Ce tip are gramatica G ?
- Construiti derivari din E pentru cuvintele $(a + a)$ si $((a + a) - a)$
- Cuvantul $(a + a - a)$ poate fi derivat din E ?
- Descrieti limbajul $L(G)$

- Fie $G = (\{A, B\}, \{a, b\}, A, \{A \rightarrow aA, A \rightarrow B, B \rightarrow bB, B \rightarrow \epsilon\})$

- Ce tip are gramatica G ?
- Descrieti limbajul $L(G)$

Clasificarea limbajelor

- Un limbaj L este de tipul j dacă există o gramatică G de tipul j astfel încât $L(G) = L$, unde $j \in \{0, 1, 2, 3\}$.
- Vom nota cu \mathcal{L}_j clasa limbajelor de tipul j , unde $j \in \{0, 1, 2, 3\}$.
- Din ierarhia lui Chomsky: $\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$
- Incluziunile sunt stricte:
 - orice limbaj de tip $j + 1$ este și de tip $j \in \{0, 1, 2\}$
 - există limbaje de tip j care nu sunt de tip $j + 1$, $j \in \{0, 1, 2\}$

Proprietăți

- Fiecare din familiile \mathcal{L}_j cu $0 \leq j \leq 3$ conține toate limbajele finite
- Fiecare din familiile \mathcal{L}_j cu $0 \leq j \leq 3$ este închisă la operația de reuniune:

$$L_1, L_2 \in \mathcal{L}_j \implies L_1 \cup L_2 \in \mathcal{L}_j,$$

$$\forall j : 0 \leq j \leq 3$$

Notății alternative pentru gramatici de tip 2: BNF

The syntax of C in Backus-Naur Form

```
<translation-unit> ::= {<external-declaration>}*
```

```
<external-declaration> ::= <function-definition>
                          | <declaration>
```

```
<function-definition> ::= {<declaration-specifier>}* <declarator> {<declaration>}* <compound-statement>
```

```
<declaration-specifier> ::= <storage-class-specifier>
                          | <type-specifier>
                          | <type-qualifier>
```

```
<storage-class-specifier> ::= auto
                          | register
                          | static
                          | extern
                          | typedef
```

```
<type-specifier> ::= void
                  | char
                  | short
                  | int
```

gramatici DTD

- generează mulțimea documentelor XML cu o anumită structură (limbaj independent de context)

```
<!ELEMENT family (person)+>  
<!ELEMENT person (name, address)*>  
<!ELEMENT name (#PCDATA)>  
<!ELEMENT address (#PCDATA)>
```

gramatici DTD

- Un "cuvânt" din limbajul generat de gramtica DTD:

```
<?xml version = "1.0">
<!DOCTYPE family SYSTEM "family.dtd">
<family>
  <person>
    <name>John</name>
    <address>First address</address>
    <address>Second address</address>
  </person>
  <person>
    <name>Sam</name>
  </person>
  <person>
    <name>Sarah</name>
    <address>First address</address>
  </person>
</family>
```

XML Schema

- - rol similar gramaticilor DTD

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="family">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="address" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```


Limbae Formale, Automate și Compilatoare - Curs 1

- 1 Prezentare curs
- 2 Limbae formale
- 3 Mecanisme de generare a limbajelor: gramatici
- 4 Ierarhia lui Chomsky
- 5 Limbae și gramatici de tip 3 (regulate)**

Gramatici de tip 3

- O gramatică $G = (N, T, S, P)$ este de tip 3 dacă regulile sale au forma: $A \rightarrow u$ sau $A \rightarrow uB$ unde $A, B \in N$ și $u \in T^*$.
- Exemplu: $G = (\{D\}, \{0, 1, \dots, 9\}, D, P)$

Unde P este:

$$D \rightarrow 0D \mid 1D \mid 2D \mid \dots \mid 9D$$

$$D \rightarrow 0 \mid 1 \mid \dots \mid 9$$

Exemple

- Fie gramatica $G = (\{A, B\}, \{l, d\}, A, P)$ unde P este:
 $A \rightarrow lB, B \rightarrow lB \mid dB \mid \epsilon$ (l = litera, d = cifra)

Exemple

- Fie gramatica $G = (\{A, B\}, \{I, d\}, A, P)$ unde P este:
 $A \rightarrow IB, B \rightarrow IB|dB| \epsilon$ (I = litera, d = cifra)
 $L(G)$: multimea identificatorilor
- Fie gramatica $G = (\{A, B\}, \{+, -, d\}, A, P)$ unde P este:
 $A \rightarrow +dB| - dB| \epsilon, B \rightarrow dB| \epsilon$ (d = cifra)

Exemple

- Fie gramatica $G = (\{A, B\}, \{I, d\}, A, P)$ unde P este:
 $A \rightarrow IB, B \rightarrow IB|dB| \epsilon$ (I = litera, d = cifra)
 $L(G)$: multimea identificatorilor
- Fie gramatica $G = (\{A, B\}, \{+, -, d\}, A, P)$ unde P este:
 $A \rightarrow +dB| - dB| dB, B \rightarrow dB| \epsilon$ (d = cifra)
 $L(G)$: multimea constantelor intregi

Forma normală

- O gramatică de tip 3 este în formă normală dacă regulile sale sunt de forma $A \rightarrow a$ sau $A \rightarrow aB$, unde $a \in T$, și, eventual $S \rightarrow \epsilon$ (caz în care S nu apare în dreapta regulilor).
- Pentru orice gramatică de tip 3 există o gramatică echivalentă în forma normală.

Forma normală

- Obținerea gramaticii în forma normală echivalentă cu o gramatică de tip 3:
 - Se poate arata că pot fi eliminate regulile de forma $A \rightarrow B$ (redenumiri) și cele de forma $A \rightarrow \epsilon$ (reguli de ștergere), cu excepția, eventual a regulii $S \rightarrow \epsilon$.
 - Orice regulă de forma $A \rightarrow a_1 a_2 \dots a_n$ se înlocuiește cu $A \rightarrow a_1 B_1, B_1 \rightarrow a_2 B_2, \dots, B_{n-2} \rightarrow a_{n-1} B_{n-1}, B_{n-1} \rightarrow a_n$, $n > 1$, B_1, \dots, B_{n-1} fiind neterminali noi.
 - Orice regulă de forma $A \rightarrow a_1 a_2 \dots a_n B$ se înlocuiește cu $A \rightarrow a_1 B_1, B_1 \rightarrow a_2 B_2, \dots, B_{n-2} \rightarrow a_{n-1} B_{n-1}, B_{n-1} \rightarrow a_n B$, $n > 1$, B_1, \dots, B_{n-1} fiind neterminali noi
 - Transformările care se fac nu modifică limbajul generat de gramatică