Limbaje Formale, Automate şi Compilatoare

Curs 1

2018-19

Limbaje Formale, Automate şi Compilatoare - Curs 1

1. Prezentare curs
2. Limbaje formale
3. Mecanisme de generare a limbajelor: gramatici
4. Ierarhia lui Chomsky
5. Limbaje şi gramatici de tip 3 (regulate)
6. Proprietăţi de închidere pentru familia de limbaje regulate

Limbaje Formale, Automate şi Compilatoare

Titulari curs:

- O. Captarencu: otto@info.uaic.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;

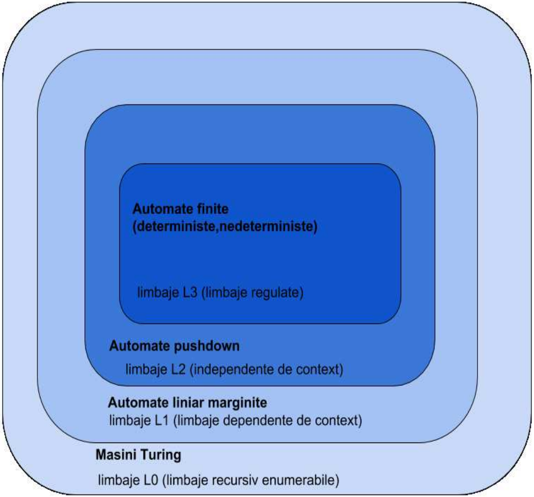
  Punctajul final se obţine astfel:

  P = 3 * AS + 3 * AL + 2 * T1 + 2 * T2
- Condiţii miminale de promovare: AS $\geq$ 5, AL $\geq$ 5, $T1 \geq 5$, $T2 \geq 5$;
- Punctaj minim pentru promovare: P $\geq$ 50;
- Nota finală se va stabili conform criteriilor ECTS;

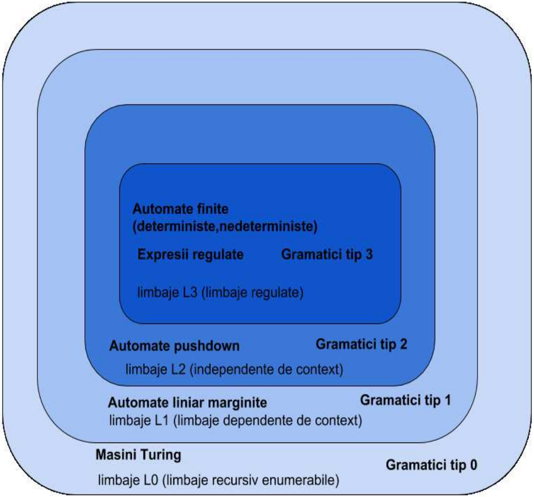
Sistem evaluare

- AS = activitatea la seminar (max 10 puncte):
  - două teste scrise
- 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)



Tematica cursului (partea I)



Tematica cursului (partea I)

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

Tematica cursului (partea II)

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

Bibliografie (selecţii)

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

Limbaje Formale, Automate şi Compilatoare - Curs 1

1. Prezentare curs
2. Limbaje formale
3. Mecanisme de generare a limbajelor: gramatici
4. Ierarhia lui Chomsky
5. Limbaje şi gramatici de tip 3 (regulate)
6. Proprietăţi de închidere pentru familia de limbaje regulate

Alfabet, cuvânt, mulţime de cuvinte

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

Alfabet, cuvânt, mulţime de cuvinte

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

Alfabet, cuvânt, multţime de cuvinte

- Alfabet: V o mulţime finită (elementele lui V = simboluri)
- Cuvânt: şir finit de simboluri
  - cuvântul nul este notat cu ϵ sau λ.
- Lungimea unui cuvânt u: numarul simbolurilor sale. Notaţie: $|u|$.

  $|\epsilon| = 0$

Alfabet, cuvânt, mulţime de cuvinte

- Alfabet: V o mulţime finită (elementele lui V = simboluri)
- Cuvânt: şir finit de simboluri
  - cuvântul nul este notat cu ϵ sau λ.
- Lungimea unui cuvânt u: numarul simbolurilor sale. Notaţie: $|u|$.

  $|\epsilon| = 0$
- V^* - multimea tuturor cuvintelor peste alfabetul V, inclusiv ϵ.

  $\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$

Alfabet, cuvânt, multţime de cuvinte

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

Operaţii pe cuvinte

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

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

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

Operaţii pe cuvinte

- Concatenarea a doua cuvinte x, y: cuvântul $x \cdot y$ obţinut 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

Operaţii pe cuvinte

- Concatenarea a doua cuvinte x, y: cuvântul $x \cdot y$ obţinut 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 numeşte monoidul liber generat de V.

Operaţii pe cuvinte

- Concatenarea a doua cuvinte x, y: cuvântul $x \cdot y$ obţinut 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 numeşte 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.

Operaţii pe cuvinte

- Concatenarea a doua cuvinte x, y: cuvântul $x \cdot y$ obţinut 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 numeşte 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 submulţime $L \subseteq V^*$ 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 submulţime $L \subseteq V^*$ 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):
    - multimea cuvintelor peste alfabetul $\{0, 1\}$ care contin un numar par de 0.
    - $L = \{x \in V^+ : |x| \ \textit{este par}\}$.
    - $\{a^n b^n | n \in N\}$.
    - $\{w \in \{0, 1\}^* | w \ \textit{se termina in} \ 00\}$.

- Fie V un alfabet. O submulţime $L \subseteq V^*$ 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):
    - multimea cuvintelor peste alfabetul $\{0, 1\}$ care contin un numar par de 0.
    - $L = \{x \in V^+ : |x| \textit{ este par}\}$.
    - $\{a^n b^n | n \in N\}$.
    - $\{w \in \{0, 1\}^* | w \textit{ se termina in } 00\}$.
  - formală (descriere matematică):
    - o descriere inductivă a cuvintelor
    - o descriere generativă a cuvintelor (gramatică generativă)
    - o descriere a unei metode de recunoaştere a cuvintelor din limbaj (automat finit, automat pushdown, etc.)

Operaţii cu limbaje

- Operatiile cu multimi (reuniune, intersectie etc)
- Produs de limbaje: $L_1 \cdot L_2 = \{u \cdot v | u \in L_1, v \in L_2\}$

  Exemplu:

  $L_1 = \{a^n, n \geq 1\}$, $L_2 = \{b^n, n \geq 1\}$

  $L_1 \cdot L_2 = \{a^n b^m, n \geq 1, m \geq 1\}$
- Iteraţia (produsul Kleene): $L^* = \bigcup_{n \geq 0} L^n$, unde:
  - $L^0 = \{\epsilon\}$
  - $L^{n+1} = L^n \cdot L$

  Exemplu:

  $L = \{a\}$, $L^0 = \{\epsilon\}, L^1 = L, L^2 = \{aa\}, \ldots, L^n = \{a^n\}$

  $L^* = \{a^n, n \geq 0\}$

Limbaje Formale, Automate şi Compilatoare - Curs 1

1. Prezentare curs
2. Limbaje formale
3. Mecanisme de generare a limbajelor: gramatici
4. Ierarhia lui Chomsky
5. Limbaje şi gramatici de tip 3 (regulate)
6. Proprietăţi de închidere pentru familia de limbaje regulate

Gramatici

Definiţie 1

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

- N *şi* T *sunt două alfabete disjuncte:*
  - N *este multimea neterminalilor*
  - T *este multimea terminalilor*
- $S \in N$ *este simbolul de start (neterminalul iniţial)*
- P *este o multime finita de reguli (producţii) de forma* $x \to y$*, unde* $x, y \in (N \cup T)^*$ *şi* x *conţine cel puţin un neterminal.*

Derivare

Definiţie 2

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

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

Derivare

Definiţie 2

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

- Daca $u_1 \Rightarrow u_2 \ldots \Rightarrow u_n, n > 1$, spunem ca u_n este derivat din u_1 în G şi notam $u1 \Rightarrow^+ u_n$.

Derivare

Definiţie 2

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

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

- Daca $u_1 \Rightarrow u_2 \ldots \Rightarrow u_n, n > 1$, spunem ca u_n este derivat din u_1 în G şi notam $u1 \Rightarrow^+ u_n$.
- Scriem $u \Rightarrow^* v$ dacă $u \Rightarrow^+ v$ sau $u = v$.

Limbaj generat

Definiţie 3

Limbajul generat de gramatica G este:

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

Limbaj generat

Definiţie 3

Limbajul generat de gramatica G este:

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

Definiţie 4

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

Exemplu

- $G = (N, T, S, P)$, $N = \{S, X, A\}$, $T = \{a, b\}$, P constă din:
  1. $S \rightarrow aXb$
  2. $aX \rightarrow aAb$
  3. $Xb \rightarrow bA$
  4. $aA \rightarrow aa$
  5. $A \rightarrow \epsilon$
- $L(G) = \{ab, abb, aabb\}$
- Gramatică echivalentă cu un singur neterminal ?
- Ce limbaj generează gramatica dacă sunt eliminate utlimele două reguli?

Exemplu

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

Exemplu

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

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

Exemplu

- $L = \{a^n b^n c^n | n \geq 1\}$
- $G = (N, T, S, P)$, $N = \{S, X\}$, $T = \{a, b, c\}$, P constă din:
  1. $S \rightarrow abc$
  2. $S \rightarrow aSXc$
  3. $cX \rightarrow Xc$
  4. $bX \rightarrow bb$
- Derivarea cuvântului $a^3b^3c^3$:

  $S \Rightarrow^{(2)} a\underline{S}Xc \Rightarrow^{(2)} aa\underline{S}XcXc \Rightarrow^{(1)} aaab\underline{cX}cXc \Rightarrow^{(3)}$
  $aaa\underline{bX}ccXc \Rightarrow^{(4)} aaabbc\underline{cX}c \Rightarrow^{(3)} aaabb\underline{cX}cc \Rightarrow^{(3)}$
  $aaab\underline{bX}ccc \Rightarrow^{(4)} aaabbbccc = a^3b^3c^3$

Limbaje Formale, Automate şi Compilatoare - Curs 1

1. Prezentare curs
2. Limbaje formale
3. Mecanisme de generare a limbajelor: gramatici
4. Ierarhia lui Chomsky
5. Limbaje şi gramatici de tip 3 (regulate)
6. Proprietăţi de închidere pentru familia de limbaje 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 $A \in N$ şi $y \in (N \cup T)^*$

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 $pxq \to pyq$ unde $x \in N, y \neq \epsilon, p, q \in (N \cup T)^*$,
   $S \to \epsilon$, caz în care S nu apare în dreapta regulilor
3. **Gramatici de tip 2 (independente de context)**
   reguli de forma $A \to y$ unde $A \in N$ şi $y \in (N \cup T)^*$
4. **Gramatici de tip 3 (regulate)**
   reguli $A \to u$ sau $A \to uB$ unde $A, B \in N$ şi $u \in T^*$.

Exemple

Tip 1: $pxq \to pyq$ unde $x \in N$, $y \neq \epsilon$, $p, q \in (N \cup T)^*$, $S \to \epsilon$

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

  (1)$S \to aaAc$

  (2)$aAc \to aAbBc$

  (3)$bB \to bBc$

  (4)$Bc \to Abc$

  (5)$A \to a$

  Gramatica tip 1

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

  (1)$S \to abc$

  (2)$S \to aSXc$

  (3)$cX \to Xc$ (nu este regulă de tip 1!, gramatica va fi de tip 0)

  (4)$bX \to bb$

Exemple

Tip 2: $A \rightarrow y$ unde $A \in N$ şi $y \in (N \cup T)^*$

Tip3: $A \rightarrow u$ sau $A \rightarrow uB$ unde $A, B \in N$ şi $u \in T^*$.

- G:

  $(1) x \rightarrow axb$

  $(2) x \rightarrow \epsilon$

  (Gramatică tip 2)

- G:

  $(1) x \rightarrow ax$

  $(2) x \rightarrow bx$

  $(3) x \rightarrow \epsilon$

  (Gramatică tip 3)

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 daca exista o gramatica G de tipul j astfel incat $L(G) = L$, unde $j \in \{0, 1, 2, 3\}$.
- Vom nota cu $\mathcal{L}_j$ clasa limbajelor de tipul j, unde $j \in \{0, 1, 2, 3\}$.
- Are loc: $\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$
- Incluziunile sunt stricte:
    - orice limbaj de tip $j + 1$ este si de tip $j \in \{0, 1, 2\}$
    - exista limbaje de tip j care nu sunt de tip $j + 1$, $j \in \{0, 1, 2\}$

Proprietăţi

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

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

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

Notaţii 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-statemen

<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 mulţimea documentelor XML cu o anumită structură (limbaj independent de context)

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

gramatici DTD

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

```
<?xml verson = "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" maxOccures="unbounded">
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Limbaje Formale, Automate şi Compilatoare - Curs 1

1 Prezentare curs

2 Limbaje formale

3 Mecanisme de generare a limbajelor: gramatici

4 Ierarhia lui Chomsky

5 Limbaje şi gramatici de tip 3 (regulate)

6 Proprietăţi de închidere pentru familia de limbaje regulate

Gramatici de tip 3

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

  Unde P este:

  $D \to 0D|1D|2D|\dots|9D$

  $D \to 0|1|\dots|9$

Exemple

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

Exemple

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

Exemple

- Fie gramatica $G = (\{A, B\}, \{l, d\}, A, P)$ unde P este:
  $A \rightarrow lB,\ B \rightarrow lB|dB|\epsilon$ (l = 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 in formă normală daca regulile sale sunt de forma $A \rightarrow a$ sau $A \rightarrow aB$, unde $a \in T$, si, eventual $S \rightarrow \epsilon$ (caz in care S nu apare in dreapta regulilor).
- Pentru orice gramatica de tip 3 exista o gramatica echivalenta in forma normala.

Forma normală

- Obtinerea gramaticii in forma normala echivalenta cu o gramatica de tip 3:
  - Se poate arata ca pot fi eliminate regulile de forma $A \to B$ (redenumiri) si cele de forma $A \to \epsilon$ (reguli de stergere), cu exceptia, eventual a regulii $S \to \epsilon$.
  - Orice regula de forma $A \to a_1 a_2 \ldots a_n$ se inlocuieste cu $A \to a_1 B_1, B_1 \to a_2 B_2, \ldots, B_{n-2} \to a_{n-1} B_{n-1}, B_{n-1} \to a_n$, $n > 1$, $B_1, \ldots, B_{n-1}$ fiind neterminali noi.
  - Orice regula de forma $A \to a_1 a_2 \ldots a_n B$ se inlocuieste cu $A \to a_1 B_1$, $B_1 \to a_2 B_2, \ldots, B_{n-2} \to a_{n-1} B_{n-1}, B_{n-1} \to a_n B$, $n > 1$, $B_1, \ldots, B_{n-1}$ fiind neterminali noi
  - Transformarile care se fac nu modifica limbajul generat de gramatica

Limbaje Formale, Automate şi Compilatoare - Curs 1

1. Prezentare curs
2. Limbaje formale
3. Mecanisme de generare a limbajelor: gramatici
4. Ierarhia lui Chomsky
5. Limbaje şi gramatici de tip 3 (regulate)
6. Proprietăţi de închidere pentru familia de limbaje regulate

Fie L, L_1, L_2 limbaje de tip 3 (regulate).

Atunci, urmatoarele limbaje sunt de asemenea de tip 3:

- $L_1 \cup L_2$
- $L_1 \cdot L_2$
- L^*
- $L_1 \cap L_2$
- $L_1 \setminus L_2$