# LUCRAREA 1

## Gramatici

**Cuprinsul lucrării:**

- Recapitulare noţiuni: alfabet, gramatică, limbaj.
- Generarea unui anumit limbaj de către o gramatică.
- Scrierea unei gramatici pentru un limbaj dat

Unele exemple din acest capitol au fost preluate şi adaptate din [3], [8] şi [12].

## 1.1 Breviar teoretic

În teoria limbajelor formale, un alfabet reprezintă o mulţime finită şi nevidă de elemente denumite simboluri. Fie $V$ un astfel de alfabet. O submulţime $L \subseteq V$ este un limbaj (formal) peste alfabetul $V$ (sau V-limbaj) dacă $L$ are o descriere (matematică) finită.
O gramatică este un sistem $G = (N, T, S, P)$, unde:

- $N$ şi $T$ sunt două alfabete disjuncte
- $N$ este mulţimea simbolurilor neterminale
- $T$ este mulţimea simbolurilor terminale
- $S \in N$ este simbolul de start (simbolul neterminal iniţial)

- $P$ este o mulţime finită de reguli (producţii) de forma $x \to y$, unde $x, y \in (N \cup T)^*$ şi $x$ conţine cel puţin un simbol neterminal.

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

- Spunem că $v$ este derivat direct (într-un pas) din $u$ 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 $u_1 \Rightarrow u2... \Rightarrow u^{\mathrm{n}}, n > 1$, spunem ca un este derivat din $\mathrm{u}_1$ în G şi notăm $u_1 \Rightarrow +u_{\mathrm{n}}$.
- Scriem $u \Rightarrow^* v$ dacă $u \Rightarrow +v$ sau $u = v$.

Limbajul generat de gramatica $G$ este:

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

Două gramatici $G_1$ şi $G_2$ sunt echivalente dacă:

$$L(G_1) = L(G_2)$$

*Exemplul 1:*

Pentru limbajul $L = \{a^{\mathrm{n}} b^{2\mathrm{n}} | n \geq 1\}$ putem afirma:

1. Definiţia inductivă:
   - $abb \in L$
   - Daca $X \in L$, atunci $aXbb \in L$
   - Niciun alt cuvânt nu face parte din L
2. Definiţia generativă (definirea gramticii care generează limbajul):
   - $G = (\{X\}, \{a, b\}, X, P), undeP = \{X \to aXbb, X \to abb\}$
   - Derivarea cuvântului $\mathrm{a}^3\mathrm{b}^6$: $X \Rightarrow aXbb \Rightarrow a(aXbb)bb \Rightarrow aa(abb)bbbb$

*Exemplul 2:*

Limbajul $L = \{a^{\mathrm{n}} b^{\mathrm{n}} c^{\mathrm{n}} | n \geq 1\}$ este generat de următoarea gramatică:

- $G = (N, T, S, P)$, $N = \{S, X\}$, $T = \{a, b, c\}$, unde $P$ constă din:
  1. $S \to abc$
  2. $S \to aSXc$
  3. $cX \to Xc$
  4. $bX \to bb$
- Derivarea cuvântului $\mathrm{a}^3\mathrm{b}^3\mathrm{c}^3$: $S \Rightarrow (2)aSXc \Rightarrow (2)aaSXcXc \Rightarrow (1)aaabcXcXc \Rightarrow (3)aaabXccXc \Rightarrow (4)aaabbccXc \Rightarrow (3)aaabbcXcc \Rightarrow (3)aaabbXccc \Rightarrow (4)aaabbbccc = a^3b^3c^3$

## 1.2 Probleme rezolvate

**Problema 1** Definiţi limbajul generat de gramatica G definită prin producţiile:

$$S \to aSa|aBa$$
$$B \to bB|b$$

*Rezolvare:*
În astfel de cazuri, gramatica este definită doar de conţinutul mulţimii P (mulţimea producţiilor), restul elementelor fiind deduse din context, astfel:

- Simbolul de start este S (simbolul din stânga al primei producţii, poate fi notat cu altă literă decât S)
- Simbolurile terminale sunt literele mici (sau cifre)
- Simbolurile neterminale sunt litere mari

Se observă faptul că prima regulă, $S \to aSa$, construieşte în mod recursiv un număr egal de simboluri $a$ la începutul şi sfârşitul şirului. Se poate ieşi din bucla recursivă prin aplicarea celei de-a doua reguli, $S \to aBa$, care în cazul în care este prima regulă aplicată asigură ca şirul generat va avea cel puţin un simbol $a$ la început şi unul la sfârşit. Apoi regula $B \to bB$ generează recursiv un număr oarecare de simboluri $b$. Pentru a reduce neterminalul $B$ din şir şi a obţine un cuvânt al limbajului, se aplică ultima regulă $B \to b$, care asigură prezenţa a cel puţin unui singur $b$ în interiorul cuvântului. Astfel, limbajul generat este:

$$L(G) = \{a^{\mathrm{n}}b^{\mathrm{m}}a^{\mathrm{n}}|n > 0, m > 0\}$$

**Problema 2** Proiectaţi o gramatică pentru a genera limbajul:

$$L(G) = \{a^{\mathrm{n}}b^{\mathrm{m}}c^{\mathrm{m}}d^{2\mathrm{n}}|n \geq 0, m > 0\}$$

*Rezolvare:*
Se poate observa pe baza legăturii dintre numărul de simboluri $a$ şi $d$, de la începutul şi respectiv sfârşitul cuvintelor limbajului, faptul că este necesară o regulă recursivă pentru a le genera. În cadrul acestei reguli, numărul de simboluri $d$ de la sfârşitul şirului trebuie să fie dublu faţă de numărul de simboluri $a$ de la început: $S \to aSdd$. Mai este necesară o regulă pentru a genera simbolurile $b$ şi $c$, folosind încă un simbol neterminal: $S \to A, A \to bAc$. Pentru a încheia recursivitatea se aplică regula $A \to bc$, care garantează faptul că subşirul $bc$ este prezent în toate cuvintele limbajului. Aşadar, regulile gramaticii sunt:

$$S \to aSdd|A$$
$$A \to bAc|bc$$

## 1.3 Probleme propuse

1. Proiectaţi o gramatică pentru a genera limbajul tuturor cuvintelor formate din alfabetul 0,1 care încep şi se termină cu acelaşi simbol.

2. Proiectaţi o gramatică pentru a genera limbajul tuturor cuvintelor formate din alfabetul 0,1 care conţin subşirul 001.

3. Proiectaţi o gramatică pentru a genera limbajul următor:

$$L(G) = \{a^{\mathrm{n}}b^{\mathrm{m}} | 0 \leq n \leq m \leq 2n\}$$

4. Pentru gramatica definită prin producţiile de mai jos, determinaţi limbajul generat:

$$S \rightarrow abScB|\lambda$$
$$B \rightarrow bB|b$$

5. Fie gramatica definită de producţiile:

$$S \rightarrow abScB|\lambda$$
$$B \rightarrow bB|b$$

   Ce limbaj generează?

6. Fie gramatica definită de producţiile:

$$S \rightarrow bTbb$$
$$T \rightarrow bTbb|Acccb$$
$$A \rightarrow aAc|\lambda$$

   Ce limbaj generează?