

# Gramáticas

Definições

Hierarquia de Chomsky

Exemplos de gramáticas

Formalmente, as gramáticas são caracterizadas como quádruplas ordenadas

$$G = ( V_n, V_t, P, S )$$

onde:

$V_n$  representa o vocabulário não terminal da gramática. Este vocabulário corresponde ao conjunto de todos os símbolos dos quais a gramática se vale para definir as leis de formação das sentenças da linguagem.

**V<sub>t</sub>** é o vocabulário terminal, contendo os símbolos que constituem as sentenças da linguagem. Dá-se o nome de terminais aos elementos de V<sub>t</sub>.

**P** representa o conjunto de todas as leis de formação utilizadas pela gramática para definir a linguagem.

Para tanto, cada construção parcial, representada por um não-terminal, é definida como um conjunto de regras de formação relativas à definição do não-terminal a ela referente. A cada uma destas regras de formação que compõem o conjunto P dá-se o nome de produção da gramática.

Assumimos  $V_n \cap V_t = \emptyset$ . Convencionamos que  $V_n \cup V_t = V$   
Cada produção  $P$  tem a forma:

$$\alpha \rightarrow \beta \quad \alpha \in V^+; \quad \beta \in V^*$$

$S \in V_n$  denota a principal categoria gramática de  $G$ ; é dito o símbolo inicial ou o axioma da gramática. Indica onde se inicia o processo de geração de sentenças.

Ex.1:  $G = (\{S, A, B\}, \{a, b\}, P, S)$

$P: \{S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow b\}$

# Notação/Convenções

- Letras do alfabeto latino maiúsculas  $\{A,B,..Z\}$ : **variáveis**
- Letras do começo do alfabeto latino minúsculas  $\{a,b,c,.. \}$ : **terminais**
- Letras do fim do alfabeto latino minúsculas  $\{t,u,v,x,z\}$ : **cadeias de terminais**
- Letras gregas minúsculas  $\{\alpha,\beta,\gamma,\delta,\epsilon, \dots, \omega\}$ : **cadeias de terminais e não terminais**

alfa, beta, gama, delta, épsilon, zeta, eta, teta, iota, kapa, lâmbda, mi, ni, xi, ômicron, pi, rô, sigma, tau, úpsilon, fi, qui, psi, ômega

Definida uma gramática  $G$ , qual é a linguagem gerada por ela?

Precisaremos das relações  $\Rightarrow_G$  (deriva diretamente) e  $\Rightarrow_{G^*}$  (deriva) definidas entre as cadeias de  $V^*$

**Def1.** Se  $\alpha \rightarrow \beta$  é uma produção de  $P$  e  $\gamma$  (gama) e  $\delta$  (delta) são cadeias quaisquer de  $V^*$ , então  $\gamma \alpha \delta \Rightarrow_G \gamma \beta \delta$  (deriva diretamente na gramática  $G$ ).

Dizemos que a produção  $\alpha \rightarrow \beta$  é aplicada à cadeia  $\gamma \alpha \delta$  para obter  $\gamma \beta \delta$ . A relação  $\Rightarrow_G$  relaciona cadeias exatamente quando a segunda é obtida a partir da primeira pela aplicação de uma **única produção**.

No Ex.1.:  $S \Rightarrow_G AB$  ;  $aB \Rightarrow_G ab$       **ou**  
 $S \Rightarrow_G AB \Rightarrow_G aB \Rightarrow_G ab$

**Def2.** Suponha que  $\alpha_1 \alpha_2 \alpha_3 \dots \alpha_m$  são cadeias de  $V^*$  e  $\alpha_1 \Rightarrow_G \alpha_2, \alpha_3 \Rightarrow_G \alpha_4, \dots, \alpha_{m-1} \Rightarrow_G \alpha_m$ .  
Então dizemos que  $\alpha_1 \Rightarrow_G^* \alpha_m$  (**deriva**).  
Aplicamos algum número de produções de P.  
Por convenção  $\alpha \Rightarrow_G^* \alpha$  para a cadeia  $\alpha$ .

No Ex.1.:  $S \Rightarrow_G^* ab$ ;  
 $S \Rightarrow_G^* aB$ ;  
 $AB \Rightarrow_G^* ab$ ;  
 $ab \Rightarrow_G^* ab$

**Def3. Forma sentencial:** uma cadeia  $\alpha$  composta de terminais e não terminais se  $S \Rightarrow^* \alpha$

No Ex.1: aB, AB, S, ab são formas sentenciais.

Uma forma sentencial,  $\alpha$ , é uma **sentença** de G se  $S \Rightarrow^* \alpha$  e  $\alpha \in Vt^*$  (são composta de terminais). Ou seja, as cadeias geradas pela gramática são as sentenças de G.

**Def4.** A Linguagem L gerada por uma gramática G é definida como o conjunto de cadeias geradas por G. Ou seja,

$$L(G) = \{x \mid x \in Vt^* \text{ e } S \Rightarrow_G^* x\} \text{ ou } \{x \mid x \text{ é sentença de } G\}$$

1. A cadeia consiste somente de terminais
2. A cadeia pode ser derivada a partir do símbolo inicial da gramática

**Def5.** Duas gramáticas G1 e G2 são equivalentes sse  
 $L(G1) = L(G2)$

# Exemplos de Gramáticas

$G1 = (\{S\}, \{0, 1\}, P1, S)$

$P1: \{$   
1.  $S \rightarrow 0S1$   
2.  $S \rightarrow 01$   
 $\}$

Qual é a linguagem gerada por  $G1$ ? Aplicamos o **processo de derivação** para saber  $L(G1)$ , que é o processo de obtenção de cadeias a partir de uma gramática.

$G2 = (\{S, B, C\}, \{a, b, c\}, P2, S)$

$P2: \{$   
1.  $S \rightarrow aSBC$   
2.  $S \rightarrow aBC$   
3.  $CB \rightarrow BC$   
4.  $aB \rightarrow ab$   
5.  $bB \rightarrow bb$   
6.  $bC \rightarrow bc$   
7.  $cC \rightarrow cc$   
 $\}$

$L(G2) = ?$

# G1

- A menor cadeia gerada é 01:  $S \Rightarrow^2 01$
- Se aplicarmos n-1 vezes a produção 1, seguida da produção 2 teremos:
  - $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 0^3S1^3 \Rightarrow^*$
  - $0^{n-1}S1^{n-1} \Rightarrow 0^n1^n$
  - Portanto,  $L(G1) = \{0^n1^n \mid n \geq 1\}$   
ou  $S \Rightarrow^* 0^n1^n$



# Tipos de Gramáticas

- Chamamos o tipo de gramática que definimos **de tipo 0 ou Recursivamente enumerável/ Estrutura de Frase/ Irrestritas.**

$$\alpha \rightarrow \beta \quad \alpha \in V^+; \beta \in V^*$$

- Não há restrições nas regras de produção.

$G = ( \{S,A,B,C,D,E\}, \{a\}, P, S )$

$P = \{$  1.  $S \rightarrow ACaB$

2.  $Ca \rightarrow aaC$

3.  $CB \rightarrow DB$

4.  $CB \rightarrow E$

5.  $aD \rightarrow Da$

6.  $AD \rightarrow AC$

7.  $aE \rightarrow Ea$

8.  $AE \rightarrow \lambda \}$

$L(G) = ?$

- Menor cadeia:  $aa$

$S \Rightarrow^1 ACaB \Rightarrow^2 AaaCB \Rightarrow^4 AaaE \Rightarrow^7 AaEa \Rightarrow^7 AEaa \Rightarrow^8 aa$

- A e B servem como marcadores da esq e dir para as formas sentenciais.
- C é o marcador que se move através da cadeia de a's entre A e B, **dobrando** seu número pela produção 2.
- Quando C alcança o marcador à direita B, ele se torna um D ou E pela produção 3 ou 4.
- Se um D é escolhido, então ele migra à esquerda pela produção 5 até que o marcador à esq, A, seja alcançado.
- Nesse ponto, D se torna C de novo pela produção 6 e o processo recomeça.
- Se um E é escolhido, o marcador à direita (B) é consumido.
- O E migra à esquerda pela produção 7 e consome o marcador à esq pela produção 8.

$$L(G) = \{a^{2^n} \mid n \text{ é um inteiro positivo}\}$$

Até este ponto não foi imposta qualquer restrição sobre a gramática ou sobre as produções que denotam as leis de formação da linguagem que está sendo definida.

As gramáticas gerais têm limitações em relação à sua aplicabilidade no contexto do estudo dos **compiladores**, devido às dificuldades que acarretam em seu tratamento, sendo que as linguagens de programação de interesse não exigem toda a generalidade que as gramáticas gerais definidas acima são capazes de oferecer.

Torna-se atraente o estudo de casos particulares, de aplicação mais restrita, porém suficiente para resolver os problemas levantados ao se projetar compiladores para linguagens de interesse. Sendo assim, dividimos as gramáticas em quatro classes, que serão vistas a seguir.

# Classes Gramaticais

Conforme as restrições impostas ao formato das produções de uma gramática, a classe de linguagens que tal gramática gera varia correspondentemente. A teoria mostra que há quatro classes de gramáticas capazes de gerar quatro classes correspondentes de linguagens, de acordo com a denominada **Hierarquia de Chomsky**:

Gramáticas Irrestritas ou Tipo 0

Gramáticas Sensíveis ao Contexto ou Tipo 1

Gramáticas Livres de Contexto ou Tipo 2

Gramáticas Regulares ou Tipo 3

# Linguagens LEF

As linguagens geradas pelas Gramáticas com Estrutura de Frase ou do Tipo 0 são chamadas de Linguagens com Estrutura de Frase (LEF) ou Linguagens do Tipo 0.

## Gramáticas Sensíveis ao/Dependentes de Contexto ou Tipo 1

Se às regras de substituição for imposta a restrição de que nenhuma substituição possa reduzir o comprimento da forma sentencial à qual a substituição é aplicada, cria-se uma classe de gramáticas ditas sensíveis ao contexto. As gramáticas que obedecem a estas restrições pertencem, na hierarquia de Chomsky, ao conjunto das **Gramáticas Sensíveis ao Contexto (GSC) ou do Tipo 1**.

Para as GSC, as produções são todas da forma

$\alpha \rightarrow \beta$ , com  $|\alpha| \leq |\beta|$  (produções não decrescentes)  
onde  $\alpha, \beta \in (V_n \cup V_t)^+$

Alguns autores colocam as produções de uma GDC como:

$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$  com  $\alpha_1, \alpha_2, \beta \in V^*$ ,  $\beta \neq \lambda$  e  $A \in V_n$

Para motivar o nome sensível ao contexto desde que a produção  $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$  permite que  $A$  seja trocado por  $\beta$  no contexto de  $\alpha_1$  e  $\alpha_2$ .

A gramática do Ex 2 é uma GSC e também as variações dela abaixo:

$G_1 = (\{A, B, C\}, \{a, b, c\}, P_1, A)$

$P_1 = \{ A \rightarrow abc \quad A \rightarrow aBbc$

$Bb \rightarrow bB \quad Bc \rightarrow Cbcc$

$bC \rightarrow Cb \quad aC \rightarrow aaB$

$aC \rightarrow aa \}$

$G_2 = (\{S, C\}, \{a, b, c\}, P_2, S)$

$P_2 = \{ S \rightarrow abc$

$ab \rightarrow aabbC$

$Cb \rightarrow bC$

$Cc \rightarrow cc \}$

# Linguagens LSC

As linguagens geradas pelas Gramáticas Sensíveis ao Contexto ou do Tipo 1 são chamadas de Linguagens Sensíveis ao Contexto (LSC) ou Linguagens do Tipo 1.

## Resultado 1:

Toda gramática do tipo 1 é também do tipo 0.

## Corolário 1:

Toda LSC é também uma LEF (mas nem toda LEF é LSC).

## Gramáticas Livres de Contexto ou Tipo 2

As Gramáticas Livres de Contexto (GLC) ou do Tipo 2 são aquelas cujas regras de produção são da forma:

$$A \rightarrow \alpha \quad \text{onde } A \in V_n, \alpha \in V^+$$

Ou seja, quando do lado esquerdo da regra há apenas um símbolo não-terminal (uma variável)

A gramática do Ex 1 é uma GLC. Outro exemplo:

$$G = (\{S, A, B\}, \{a, b\}, P, S)$$

$$P = \{S \rightarrow aB \mid bA$$

$$A \rightarrow a \mid aS \mid bAA$$

$$B \rightarrow b \mid bS \mid aBB \}$$

$$L(G) = ?$$

- Menores cadeias: ab e ba

$S \Rightarrow aB \Rightarrow ab$

$S \Rightarrow bA \Rightarrow ba$

$S \Rightarrow aB \Rightarrow abS \Rightarrow abbA \Rightarrow abba$

$\Rightarrow abaB \Rightarrow abab$

$\Rightarrow aaBB \Rightarrow aabb$

$S \Rightarrow bA \Rightarrow baS \Rightarrow baaB \Rightarrow baab$

$\Rightarrow babA \Rightarrow baba$

$\Rightarrow bbAA \Rightarrow bbaa$

$L(G) = \{w \in \{a,b\}^+ \mid nro(a) = nro(b)\}$

Todas as combinações de cadeias em  $V^+$  com  $nro(a) = nro(b)$

# Linguagens LLC

As linguagens geradas pelas Gramáticas Livres de Contexto ou do Tipo 2 são chamadas de Linguagens Livres de Contexto (LLC) ou Linguagens do Tipo 2.

## Resultado 2:

Toda gramática do tipo 2 é também do tipo 1.

## Corolário 2:

Toda LLC é também uma LSC (mas nem toda LSC é uma LLC).

# BNF

Outra maneira de se representar as Gramáticas Livres de Contexto é através da **Forma Normal de Backus**.

Neste caso,  $\rightarrow$  é substituído por  $::=$  e os não terminais são ladeados por  $\langle \rangle$

No caso de repetições de lado esquerdo:

$\langle A \rangle ::= a_1$

$\langle A \rangle ::= a_2$

$\vdots$

$\langle A \rangle ::= a_n$

escreve-se:  $\langle A \rangle ::= a_1 \mid a_2 \mid \dots \mid a_n$

Os símbolos  $\langle, \rangle, ::=, \mid$  formam a metalinguagem, ou seja, são símbolos que não fazem parte da linguagem mas ajudam a descrevê-la.

Exemplo:

$G = \{Vn, Vt, P, S\}$  onde:

$Vn = \{<sentença, <sn>, <sv>, <artigo>, <substantivo>, <verbo>\}$

$Vt = \{o, a, peixe, comeu, isca\}$

$S = <sentença>$

$P = \{$

1.  $<sentença> ::= <sn> <sv>$

2.  $<sn> ::= <artigo> <substantivo>$

3.  $<sv> ::= <verbo> <sn>$

4.  $<artigo> ::= o | a$

5.  $<verbo> ::= mordeu$

6.  $<substantivo> ::= peixe | isca \}$

Exercícios:

- a) verifique se a cadeia “a isca mordeu o peixe” é uma sentença de  $L(G)$ .
- b) Dê exemplos de sentenças de  $L(G)$ .

## Mais GLC:

$G = (\{S\}, \{a, +, *, (, )\}, P, S)$

$P = \{$   
     $S \rightarrow S * S$   
     $S \rightarrow S + S$   
     $S \rightarrow (S)$   
     $S \rightarrow a \}$

$L(G) =$  conjunto das expressões aritméticas envolvendo  
 $*, +, (, )$  e  $a$ .

Um exemplo de cadeia formada por esta gramática é  
 $a * (a + a)$ .

Processo inverso: Dada uma  $L(G)$  definir a gramática  $G$ .

$$L(G) = \{a^m b^n \mid m \geq 1, n \geq 1\}$$

$L(G) = \{a^m b^n \mid m \geq 1, n \geq 1\}$  ou  $a^+ b^+$

Resp.:

$G = (\{S, A, B\}, \{a, b\}, P, S)$

$P = \{S \rightarrow AB$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b\}$

*Obs.:* Caso geral:

Se  $S \Rightarrow \alpha S \mid \beta$  então  $L(G) = \alpha^* \beta$

$$L(G) = \{a^n b^n \mid n \geq 1\}$$

$$L(G) = \{a^n b^n \mid n \geq 1\}$$

$$G = (\{S\}, \{a, b\}, P, S)$$

$$P = \{S \rightarrow aSb \mid ab\}$$

## Gramáticas Regulares ou Tipo 3

Aplicando-se mais uma restrição sobre a forma das produções, pode-se criar uma nova classe de gramáticas, as **Gramáticas Regulares (GR)**, de grande importância no estudo dos compiladores por possuírem propriedades adequadas para a obtenção de reconhecedores simples. Nas GRs, as produções são restritas às formas seguintes:

$A \rightarrow aB$  ou  $A \rightarrow a$  (linear à direita)

OU

$A \rightarrow Ba$  ou  $A \rightarrow a$  (linear à esquerda)

onde  $A, B \in V_n$  e  $a \in V_t$

Tem que escolher uma das duas formas acima.

# Linguagens LR

As linguagens geradas pelas Gramáticas Regulares ou do Tipo 3 são chamadas de Linguagens Regulares (LR) ou Linguagens do Tipo 3.

**Resultado 3:**

Toda gramática do tipo 3 é também do tipo 2.

**Corolário 3:**

Toda LR é também uma LLC (mas nem toda LLC é LR).

## Exemplo 1:

$G = (\{S\}, \{a, b\}, P, S)$

$P = \{$

$S \rightarrow aS$

$S \rightarrow b \}$

Exemplo 1:

$G = (\{S\}, \{a, b\}, P, S)$

$P = \{$

$S \rightarrow aS$

$S \rightarrow b \}$

Resp.:  $L(G) = \{a^n b \mid n \geq 0\}$  ou  $a^*b$

## Exemplo 2:

$G = (\{S, A\}, \{a, b, c\}, P, S)$

$P = \{$

$S \rightarrow aS \mid bA$

$A \rightarrow c \}$

Exemplo 2:

$G = (\{S, A\}, \{a, b, c\}, P, S)$

$P = \{$

$S \rightarrow aS \mid bA$

$A \rightarrow c \}$

Resp.:  $L(G) = \{a^nbc \mid n \geq 0\}$

### Exemplo 3:

$G = ( \{ \langle \text{Dig} \rangle, \langle \text{Int} \rangle \}, \{ +, -, 0, \dots, 9 \}, P, \langle \text{Int} \rangle )$

$P = \{ \langle \text{Int} \rangle ::= + \langle \text{Dig} \rangle \mid - \langle \text{Dig} \rangle$

$\langle \text{Dig} \rangle ::= 0 \langle \text{Dig} \rangle \mid 1 \langle \text{Dig} \rangle \mid \dots \mid 9 \langle \text{Dig} \rangle \mid 0 \mid 1 \mid 2 \mid \dots \mid 9 \}$

### Exemplo 3:

$G = ( \{ \langle \text{Dig} \rangle, \langle \text{Int} \rangle \}, \{ +, -, 0, \dots, 9 \}, P, \langle \text{Int} \rangle )$

$P = \{ \langle \text{Int} \rangle ::= + \langle \text{Dig} \rangle \mid - \langle \text{Dig} \rangle$

$\langle \text{Dig} \rangle ::= 0 \langle \text{Dig} \rangle \mid 1 \langle \text{Dig} \rangle \mid \dots \mid 9 \langle \text{Dig} \rangle \mid 0 \mid 1 \mid 2 \mid \dots \mid 9 \}$

**Resp.:**

**$L(G) = \text{conj. números inteiros com sinal } \pm[0..9]^+$**

## Exemplo 4:

$G = ( \{A,B,C\}, \{0,1\}, P, A )$

$P = \{ A \rightarrow 0B \mid 0$

$B \rightarrow 1C$

$C \rightarrow 0B \mid 0 \}$

## Exemplo 4:

$G = ( \{A, B, C\}, \{0, 1\}, P, A )$

$P = \{ A \rightarrow 0B \mid 0$

$B \rightarrow 1C$

$C \rightarrow 0B \mid 0 \}$

Resp.:

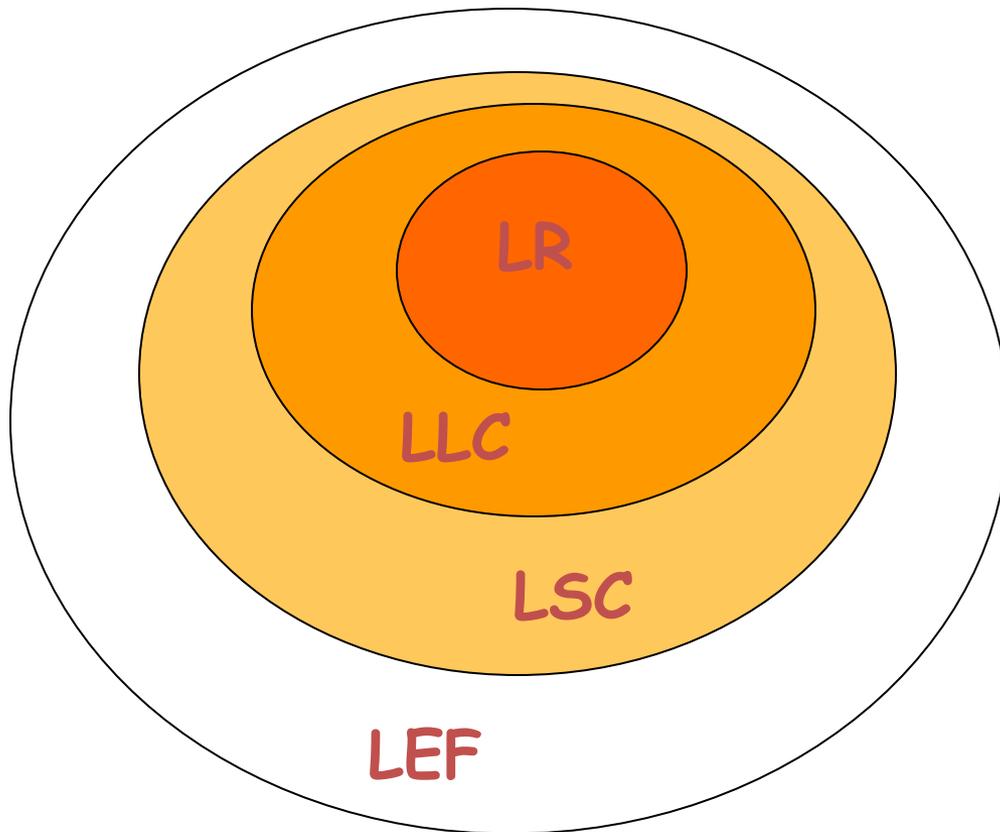
$L(G) = \{0(10)^*\}$

### Hierarquia de Chomsky

Em termos gerais, para  $n \in \{0, 1, 2, 3\}$  pode-se afirmar que uma *linguagem* de qualquer tipo pode ser classificada também como sendo de tipo menor, de acordo com a **Hierarquia de Chomsky**.

Uma linguagem do tipo  $n$  é caracterizada pela existência de alguma gramática do tipo  $n$  que a descreva.

## Hierarquia de Chomsky



**LR** = Linguagens  
Regulares

**LLC** = Linguagens  
Livres de Contexto

**LSL** = Linguagens  
Sensíveis ao Contexto

**LEF** = Linguagens com  
Estrutura de Frase

# Gramáticas e reconhecedores

<b>Gramáticas</b>	<b>Reconhecedores</b>
Irrestrita	Máquina de Turing
Sensível ao contexto	Máquina de Turing com memória limitada
Livre de contexto	Autômato a pilha
Regular	Autômato finito

# Linguagens e Reconhecedores

Linguagem	Gramática	Reconhecedor	Tempo para reconhecer $w$ ; $ w =n$
Tipo 0: Linguagens Computáveis ou Recursivamente Enumeráveis	Gramáticas com Estrutura de Frase	Máquinas de Turing	NP-completo
Tipo 1: Sensíveis ao Contexto	Gramáticas Sensíveis ao Contexto	Máquinas de Turing com memória limitada	Exponencial: $O(2^n)$
Tipo 2: Livres de Contexto	Gramáticas Livres de Contexto	Autômatos à Pilha	Polinomial Espaço: $O(n)$ ; Tempo: Geral: $O(n^3)$ ; Não-ambíguas: $O(n^2)$ ; Se $P = A \rightarrow aB$ ou $A \rightarrow Ba$ ou $A \rightarrow a$ : $O(n)$
Tipo 3: Conjuntos Regulares	Gramáticas Regulares	Autômatos Finitos	Linear: $O(n)$ e $O( E )$ (no tamanho do AF)

Classifique as gramáticas, dê a quádrupla e a  $L(G)$  e diga se as ling são finitas/infinitas

$$1) \quad E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid F \\ F \rightarrow 0 \mid 1 \mid \dots \mid 9$$

$$2) \quad A \rightarrow BC \\ BC \rightarrow CB \\ B \rightarrow b \\ C \rightarrow a$$

$$3) \quad A \rightarrow 0A \mid B \\ B \rightarrow 1B \mid \lambda$$

4)  $S \rightarrow 0A$

$A \rightarrow 1S \mid 1$

5)  $S \rightarrow 0A$

$A \rightarrow 1B$

$B \rightarrow 1S \mid 1$

6)  $L(G6) = \{111(00)^n \mid n \geq 0\}$

$G6 = ?$

7)  $L(G7) = \{a^n b^n c^i \mid n \geq 1 \text{ e } i \geq 0\}$

$G7 = ?$

8)  $L(G8) = \{a^j b^n c^n \mid n \geq 1 \text{ e } j \geq 0\}$

$G8 = ?$

9) Utilize o software JFLAP com os exemplos acima