

1. Mostre que a gramática

$$G = (\{S\}, \{a, b\}, S, \{S \rightarrow \epsilon, S \rightarrow aSb\})$$

gera a linguagem

$$L = \{a^i b^i \mid i = 0, 1, \dots\}$$

2. Seja $G = (V_N, V_T, S, F)$ uma gramática com $V_N = \{S, A, B\}$, $V_T = \{a, b\}$, e com o seguinte conjunto de regras de derivação F :

$$\begin{array}{l} S \rightarrow aB \quad S \rightarrow bA \\ A \rightarrow a \quad A \rightarrow aS \quad A \rightarrow bAA \\ B \rightarrow b \quad B \rightarrow bS \quad B \rightarrow aBB \end{array}$$

Demonstre que esta gramática gera a linguagem

$$L = \{P \mid P \in \{a, b\}^+ \text{ e } N_a(P) = N_b(P)\}$$

com

$$\begin{array}{l} N_a(P) \text{ o número de ocorrências de } a \text{ em } P \\ N_b(P) \text{ o número de ocorrências de } b \text{ em } P \end{array}$$

3. Mostre que a linguagem gerada pela gramática

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

com F o seguinte conjunto de regras de derivação

$$\begin{array}{l} S \rightarrow aAB \quad bB \rightarrow a \quad Ab \rightarrow SBb \\ Aa \rightarrow SaB \quad B \rightarrow SA \quad B \rightarrow ab \end{array}$$

é a linguagem vazia.

Ajuda: Veja que para toda a palavra derivável a partir de S fica sempre pelo menos um não terminal

4. Descubra uma gramática livre do contexto para cada uma das seguintes linguagens:
- (a) $L = \{a^i b^j \mid i = 0, 1, \dots \text{ e } j \geq i\}$
 - (b) $L = \{a^i b^j a^j b^i \mid i = 0, 1, \dots \text{ e } j = 0, 1, \dots\}$
 - (c) $L = \{a^i b^i a^j b^j \mid i = 0, 1, \dots \text{ e } j = 0, 1, \dots\}$
 - (d) $L = \{a^i b^i \mid i = 0, 1, \dots\} \cup \{b^j a^j \mid j = 0, 1, \dots\}$
 - (e) $L = \{PP^{-1} \mid P \in \{a, b\}^*\}$
 - (f) $L = \{a^i b^j c^{i+j} \mid i = 0, 1, \dots \text{ e } j = 0, 1, \dots\}$
5. Utilizando os programas `grep`, `egrep`, e `fgrep`, conforme for mais apropriado, selecione linhas de um dado texto que contenham:
- (a) números inteiros (e.g. 23).
 - (b) importâncias em Euros (e.g. 25,00Euros).
 - (c) números reais (e.g. 23,5 ou 2,35E1).
 - (d) identificadores terminados por números (e.g. folha1).
 - (e) palavras que contenham duas letras “a” ou “b” seguidas (e.g. aa).
 - (f) palavras que contenham duas letras (quaisquer) iguais seguidas.
6. Pretende-se construir um programa que, de uma forma simples, nos dê uma indicação da origem dos acessos ao nosso servidor de páginas da rede. Escreva um programa em *C* que, utilizando os comandos `egrep` e `wc` de forma conveniente construa um gráfico de barras com a percentagem de acessos por domínio (por exemplo .pt, .com, ...).
- Nota: o registo dos acesso ao servidor `apache` é guardado no ficheiro:
`/var/log/apache/access.log`.
7. Utilizando a pesquisa com expressões regulares do editor (X)`emacs`, procure as ocorrências de:
- (a) números entre parêntesis (e.g. (1)).
 - (b) palavras começadas por letras maiúsculas (e.g Folha).
 - (c) palavras em que duas ocorrências da letra “a” estejam separadas só por uma outra (qualquer) letra (e.g. bata).
8. Utilizando o mecanismo de *procura e substituição utilizando expressões regulares* do (X)`emacs`, faça as seguintes alterações:
- (a) Dado datas escritas no formato `mm:dd:aa`, converta-as para o formato `dd/mm/aa`.
 - (b) Dado nomes no formato `<nome><num inteiro com zeros à esquerda>.jpeg` os altere para `<nome><num inteiro>.jpg`