Departamento de Ciência de Computadores FCUP

# Modelos de Computação (CC218) 2013/14

Exame (26.06.2014) *duração: 3h*

## Uma resolução

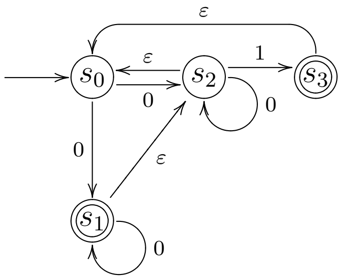
**Cotação: 2+2+1+2+1+1.5, 1+2+2, 1.5+2.5+1.5**

**1.** Seja $A = (\{s_0, s_1, s_2, s_3\}, \{0, 1\}, \delta, s_0, \{s_1, s_3\})$ um autómato finito não determinístico com transições por $\varepsilon$, em que a função $\delta$ é assim definida:

$$\begin{array}{llll}
\delta(s_0, 0) = \{s_1, s_2\} & \delta(s_1, 0) = \{s_1\} & \delta(s_2, 0) = \{s_2\} & \delta(s_3, 0) = \{\} \\
\delta(s_0, 1) = \{\} & \delta(s_1, 1) = \{\} & \delta(s_2, 1) = \{s_3\} & \delta(s_3, 1) = \{\} \\
\delta(s_0, \varepsilon) = \{\} & \delta(s_1, \varepsilon) = \{s_2\} & \delta(s_2, \varepsilon) = \{s_0\} & \delta(s_3, \varepsilon) = \{s_0\}
\end{array}$$

**a)** Represente o diagrama de transição do autómato $A$ e apresente a gramática linear à direita que se obtém de $A$ pelo algoritmo de conversão dado. Prove que essa gramática é ambígua.

*Resposta:*



A gramática obtida é $G = (\{V_{s_0}, V_{s_1}, V_{s_2}, V_{s_3}\}, \{\mathtt{0}, \mathtt{1}\}, P, V_{s_0})$, em que $P$ é constituído por:

$$\begin{array}{lcl}
V_{s_0} & \rightarrow & \mathtt{0}V_{s_1} \mid \mathtt{0}V_{s_2} \\
V_{s_1} & \rightarrow & \mathtt{0}V_{s_1} \mid V_{s_2} \mid \varepsilon \\
V_{s_2} & \rightarrow & \mathtt{0}V_{s_2} \mid \mathtt{1}V_{s_3} \mid V_{s_0} \\
V_{s_3} & \rightarrow & V_{s_0} \mid \varepsilon
\end{array}$$

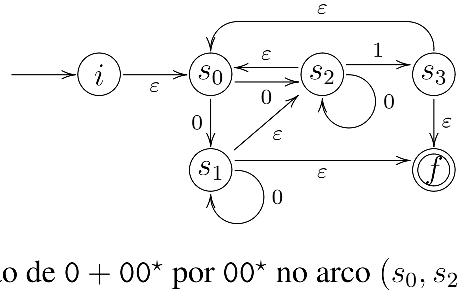
Esta gramática é ambígua porque, por exemplo, a palavra `01` admite duas derivações pela esquerda:

$$V_{s_0} \Rightarrow \mathtt{0}V_{s_1} \Rightarrow \mathtt{0}V_{s_2} \Rightarrow \mathtt{01}V_{s_3} \Rightarrow \mathtt{01}$$
$$V_{s_0} \Rightarrow \mathtt{0}V_{s_2} \Rightarrow \mathtt{01}V_{s_3} \Rightarrow \mathtt{01}$$

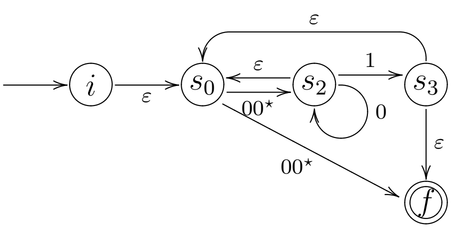
**b)** Por aplicação do algoritmo de eliminação de estados, determine uma expressão regular que descreva a linguagem $\mathcal{L}(A)$. Deverá apresentar os passos intermédios.

*Resposta:*

Introduzem-se dois novos estados $i$ (inicial) e $f$ (final), que serão os únicos que não serão eliminados.



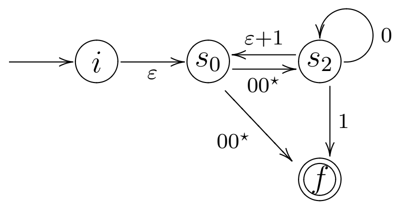
Por eliminação de $s_1$ e substituição de $\mathtt{0} + \mathtt{00}^\star$ por $\mathtt{00}^\star$ no arco $(s_0, s_2)$ resultante, obtém-se:



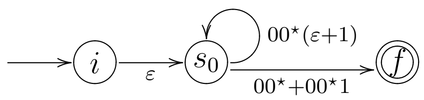
(Continua, v.p.f.)

*Resposta 1b) (cont.):*

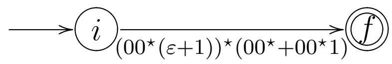
Por eliminação de $s_3$, obtém-se:




Por eliminação de $s_2$ e simplificação de $\mathtt{0}^\star\mathtt{0}^\star$ em $\mathtt{0}^\star$, obtém-se




Por eliminação de $s_0$, obtém-se

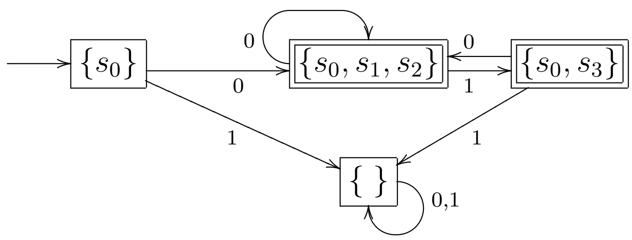



A linguagem $\mathcal{L}(A)$ pode ser descrita pela expressão regular $(\mathtt{00}^\star(\varepsilon+\mathtt{1}))^\star(\mathtt{00}^\star+\mathtt{00}^\star\mathtt{1})$.

Notar que esta expressão é equivalente a $(\mathtt{0}+\mathtt{01})^\star(\mathtt{0}+\mathtt{01})$ porque $(\mathtt{00}^\star(\varepsilon+\mathtt{1}))^\star \equiv (\mathtt{00}^\star+\mathtt{00}^\star\mathtt{1})^\star \equiv (\mathtt{0}+\mathtt{01})^\star$ e $(\mathtt{0}+\mathtt{01})^\star(\mathtt{00}^\star+\mathtt{00}^\star\mathtt{1}) \equiv (\mathtt{0}+\mathtt{01})^\star(\mathtt{0}^\star\mathtt{0}+\mathtt{0}^\star\mathtt{01}) \equiv (\mathtt{0}+\mathtt{01})^\star\mathtt{0}^\star(\mathtt{0}+\mathtt{01}) \equiv (\mathtt{0}+\mathtt{01})^\star(\mathtt{0}+\mathtt{01})$.

**c)** Apresente o diagrama de transição do autómato finito determinístico que se obtém de $A$ por aplicação do algoritmo de conversão dado. Nesse diagrama mantenha apenas os estados acessíveis do estado inicial.

*Resposta:*




NB: Os estados do AFD obtido pelo método de conversão correspondem a <u>subconjuntos</u> do conjunto de estados do AFND-$\varepsilon$ de que se partiu. As designações assim o indicam!

**d) Justifique a veracidade ou falsidade** de cada uma das afirmações seguintes:

(i) A palavra `00100` pode ser reconhecida pelo autómato $A$ e também pode não ser reconhecida.

(ii) Qualquer autómato finito determinístico que reconhece $\mathcal{L}(A)$ tem dois ou mais estados finais.

*Resposta:*

(i) Falsa. Por definição, uma palavra é reconhecida por um autómato finito se e só se o puder levar do estado inicial a algum estado final, sendo totalmente processada. Assim, cada palavra de $\Sigma^\star$ **ou é** reconhecida **ou não é** reconhecida por um autómato (ie., ou pertence à linguagem reconhecida pelo autómato ou pertence ao seu complementar).

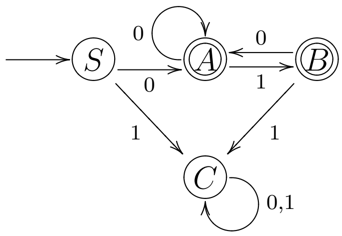
(De facto, $\mathtt{00100} \in \mathcal{L}(A)$, pois pode levar $A$ de $s_0$ a $s_1$, mas tal seria irrelevante para a resposta.)

(ii) Verdadeira. Se um AFD $A'$ que aceitasse $\mathcal{L}(A)$ só tivesse um estado final, as palavras `0` e `01` levariam $A'$ a esse estado final. Assim, as palavras `01` e `011` também levariam $A'$ a um mesmo estado, pois $A'$ é determinístico. Mas tal é absurdo porque $\mathtt{01} \in \mathcal{L}(A)$ e $\mathtt{011} \notin \mathcal{L}(A)$. Portanto, $A'$ tem de estar em estados (finais) distintos depois de consumir `0` e `01`. Logo, tem pelo menos dois estados finais.

**e)** Defina informalmente a linguagem $\mathcal{L}(A)$.

> *Resposta:*
>
> A linguagem reconhecida pelo autómato $A$ é o conjunto das sequências finitas de $\mathtt{0}$'s e $\mathtt{1}$'s que começam por $\mathtt{0}$ e não têm $\mathtt{1}$'s consecutivos.

**f)** Determine uma gramática independente de contexto que gere $\mathcal{L}(A)$, não seja ambígua e não tenha variáveis desnecessárias (i.e., variáveis que não produzem sequências de terminais ou não entram em derivações a partir do símbolo inicial). Justifique sucintamente os passos principais da resolução.

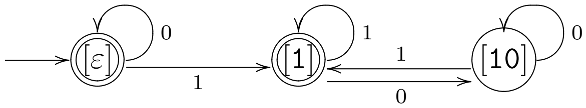
> *Resposta:*
>
> É conhecido que a gramática linear à direita que se obtém por conversão de um AFD é não ambígua (cada derivação pela esquerda corresponde a um percurso no AFD, que é único por o AFD ser determinístico).
>
> Assim, por conversão do AFD determinado em **1c)**, obtém-se uma gramática não ambígua. Essa gramática tem um símbolo desnecessário, pois a variável $C$ (correspondente ao estado designado por $\{\}$, em **1c)**) não produz sequências de terminais. Por remoção dessa variável, obtém-se a gramática apresentada à direita.
>
> 
> 
>
> $G' = (\{S, A, B\}, \{0, 1\}, P', S)$ com $P'$ dado por
>
> $$\begin{array}{lcl} S & \to & \mathtt{0}A \\ A & \to & \mathtt{0}A \mid \mathtt{1}B \mid \varepsilon \\ B & \to & \mathtt{0}A \mid \varepsilon \end{array}$$
>
> NB: A expressão regular simplificada, $(\mathtt{0}+\mathtt{01})^\star(\mathtt{0}+\mathtt{01})$, sugere uma gramática mais simples ainda: $X \to \mathtt{0} \mid \mathtt{01} \mid \mathtt{0}X \mid \mathtt{01}X$.

**2.** Seja $\mathcal{M}$ o autómato finito determinístico **mínimo** que reconhece a linguagem $L$ de alfabeto $\{\mathtt{0}, \mathtt{1}\}$ descrita pela expressão regular $\mathtt{0}^\star\mathtt{1}^\star + \mathtt{0}^\star\mathtt{1}\mathtt{1}^\star\mathtt{0}(\mathtt{0}+\mathtt{1})^\star\mathtt{1}$.

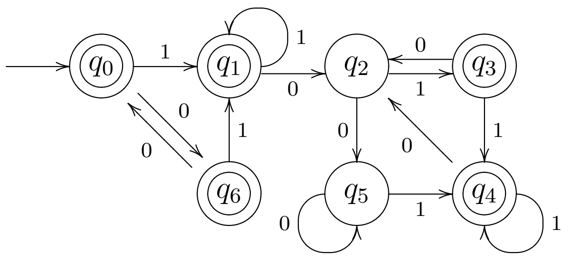
**a)** Defina **informalmente** a linguagem $L$. Explique sucintamente.

> *Resposta:*
>
> A linguagem $L$ é o conjunto das sequências finitas de $\mathtt{0}$'s e $\mathtt{1}$'s que se tiverem algum $\mathtt{0}$ depois de um $\mathtt{1}$ então terminam em $\mathtt{1}$. *Explicação*: A expressão $\mathtt{0}^\star\mathtt{1}^\star$ descreve as sequências que não têm $\mathtt{0}$'s depois de $\mathtt{1}$'s. As palavras descritas por $\mathtt{0}^\star\mathtt{1}\mathtt{1}^\star\mathtt{0}(\mathtt{0}+\mathtt{1})^\star\mathtt{1}$ são as que têm algum $\mathtt{0}$ depois de algum $\mathtt{1}$ mas que terminam em $\mathtt{1}$, pois são da forma $xy$ com $x \in \mathcal{L}(\mathtt{0}^\star\mathtt{1}\mathtt{1}^\star\mathtt{0})$ e $y \in \mathcal{L}((\mathtt{0}+\mathtt{1})^\star\mathtt{1})$.

**b)** Por aplicação do teorema de Myhill-Nerode, determine o diagrama de transição de $\mathcal{M}$. Justifique.

> *Resposta:*
>
> 
> 
>
> | | | | |
> |---|---|---|---|
> | $\delta([\varepsilon], \mathtt{1})$ | $\stackrel{\text{def}}{=}$ | $[\mathtt{1}]$ | $[\mathtt{1}] \neq [\varepsilon]$ porque $\mathtt{10} \notin L$ e $\varepsilon\mathtt{0} \in L$ |
> | $\delta([\varepsilon], \mathtt{0})$ | $\stackrel{\text{def}}{=}$ | $[\mathtt{0}] = [\varepsilon]$ | $(\mathtt{0}, \varepsilon) \in R_L$ porque $\mathtt{0}z \in L$ se e só se $z \in L$ |
> | $\delta([\mathtt{1}], \mathtt{0})$ | $\stackrel{\text{def}}{=}$ | $[\mathtt{10}]$ | $(\mathtt{10}, \varepsilon) \notin R_L$ e $(\mathtt{10}, \mathtt{1}) \notin R_L$ porque $\mathtt{10} \notin L$, $\varepsilon \in L$ e $\mathtt{1} \in L$ |
> | $\delta([\mathtt{1}], \mathtt{1})$ | $\stackrel{\text{def}}{=}$ | $[\mathtt{11}] = [\mathtt{1}]$ | $(\mathtt{1}, \mathtt{11}) \in R_L$ pois $\mathtt{1}z \in L \Leftrightarrow z \in \mathcal{L}(\mathtt{1}^\star + \mathtt{1}^\star\mathtt{0}(\mathtt{0}+\mathtt{1})^\star\mathtt{1}) \Leftrightarrow \mathtt{11}z \in L$ |
> | $\delta([\mathtt{10}], \mathtt{0})$ | $\stackrel{\text{def}}{=}$ | $[\mathtt{100}] = [\mathtt{10}]$ | $(\mathtt{100}, \mathtt{10}) \in R_L$ pois $\mathtt{100}z \in L \Leftrightarrow z \in \mathcal{L}((\mathtt{0}+\mathtt{1})^\star\mathtt{1}) \Leftrightarrow \mathtt{10}z \in L$ |
> | $\delta([\mathtt{10}], \mathtt{1})$ | $\stackrel{\text{def}}{=}$ | $[\mathtt{101}] = [\mathtt{1}]$ | $(\mathtt{101}, \mathtt{1}) \in R_L$ porque $\mathtt{101}z \in L \Leftrightarrow z \in \mathcal{L}(\mathtt{1}^\star + \mathtt{1}^\star\mathtt{0}(\mathtt{0}+\mathtt{1})^\star)$ |

**c)** Minimize o autómato finito determinístico representado abaixo e diga se é equivalente a $\mathcal{M}$. Deverá apresentar os passos intermédios relevantes.




*Resposta:*

Aplica-se o algoritmo de Moore para efetuar a minimização. Como nenhum estado final é equivalente a um estado não final, a tabela inicial é a que se encontra à esquerda. A tabela final encontra-se à direita.

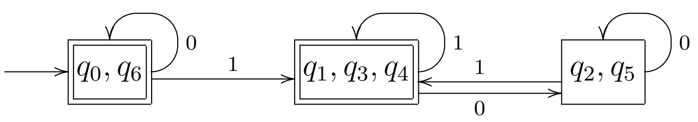
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $q_0$ | $=$ | | | | | | |
| $q_1$ | | $=$ | | | | | |
| $q_2$ | **X** | **X** | $=$ | | | | |
| $q_3$ | | | **X** | $=$ | | | |
| $q_4$ | | | **X** | | $=$ | | |
| $q_5$ | **X** | **X** | | **X** | **X** | $=$ | |
| $q_6$ | | | **X** | | | **X** | $=$ |
| | $q_0$ | $q_1$ | $q_2$ | $q_3$ | $q_4$ | $q_5$ | $q_6$ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $q_0$ | $=$ | | | | | | |
| $q_1$ | $X$ | $=$ | | | | | |
| $q_2$ | **X** | **X** | $=$ | | | | |
| $q_3$ | $X$ | $\cancel{?}$ $=$ | **X** | $=$ | | | |
| $q_4$ | $X$ | $(q_1, q_3)$ $=$ | **X** | $(q_2, q_5)$ $=$ | $=$ | | |
| $q_5$ | **X** | **X** | $\cancel{?}$ $=$ | **X** | **X** | $=$ | |
| $q_6$ | $=$ | $X$ | **X** | $X$ | $X$ | **X** | $=$ |
| | $q_0$ | $q_1$ | $q_2$ | $q_3$ | $q_4$ | $q_5$ | $q_6$ |

Justificação (sequência de passos que conduziram à tabela representada à da direita):

$(q_0, q_1)$ : $q_0 \not\equiv q_1$ porque $\delta(q_0, \mathsf{0}) = q_6 \not\equiv q_2 = \delta(q_1, \mathsf{0})$.
$(q_0, q_3)$ : $q_0 \not\equiv q_3$ porque $\delta(q_0, \mathsf{0}) = q_6 \not\equiv q_2 = \delta(q_3, \mathsf{0})$.
$(q_0, q_4)$ : $q_0 \not\equiv q_4$ porque $\delta(q_0, \mathsf{0}) = q_6 \not\equiv q_2 = \delta(q_4, \mathsf{0})$.
$(q_0, q_6)$ : $q_0 \equiv q_6$ pois são indistinguíveis já que $\delta(q_0, \mathsf{1}) = \delta(q_6, \mathsf{1})$ e $\delta(q_0, \mathsf{0}) = q_6$ e $\delta(q_6, \mathsf{0}) = q_0$.
$(q_1, q_3)$ : $q_1 \equiv q_3$ se e só se $q_1 \equiv q_4$ pois $\delta(q_1, \mathsf{0}) = \delta(q_3, \mathsf{0})$ e $\delta(q_1, \mathsf{1}) = q_1$ e $\delta(q_3, \mathsf{1}) = q_4$.
Assinalou-se $(q_1, q_3)$ na entrada de $(q_1, q_4)$ para recordar que a decisão estava pendente.
$(q_1, q_4)$ : $q_1 \equiv q_4$ pois são indistinguíveis já que $\delta(q_1, \mathsf{0}) = \delta(q_4, \mathsf{0})$ e $\delta(q_4, \mathsf{1}) = q_4$ e $\delta(q_1, \mathsf{1}) = q_1$.
Portanto, também $q_1 \equiv q_3$.
$(q_1, q_6)$ : $q_1 \not\equiv q_6$ pois $\delta(q_1, \mathsf{0}) = q_2 \not\equiv q_0 = \delta(q_6, \mathsf{0})$,
$(q_2, q_5)$ : $q_2 \equiv q_5$ se e só se $q_3 \equiv q_4$ pois $\delta(q_2, \mathsf{0}) = \delta(q_5, \mathsf{0})$ e $\delta(q_2, \mathsf{1}) = q_3$ e $\delta(q_5, \mathsf{1}) = q_4$.
Assinalou-se $(q_2, q_5)$ na entrada de $(q_3, q_4)$ para recordar que a decisão estava pendente.
$(q_3, q_4)$ : $q_3 \equiv q_4$ pois são indistinguíveis já que $\delta(q_3, \mathsf{0}) = \delta(q_4, \mathsf{0})$ e $\delta(q_3, \mathsf{1}) = \delta(q_4, \mathsf{1})$.
Portanto, também $q_2 \equiv q_5$.
$(q_3, q_6)$ : $q_3 \not\equiv q_6$ pois já que $\delta(q_3, \mathsf{0}) = q_2 \not\equiv q_0 = \delta(q_6, \mathsf{0})$.
$(q_4, q_6)$ : $q_4 \not\equiv q_6$ porque $\delta(q_4, \mathsf{0}) = q_2 \not\equiv q_0 = \delta(q_6, \mathsf{0})$.

Concluiu-se que $q_0 \equiv q_6$, $q_1 \equiv q_3 \equiv q_4$ e $q_2 \equiv q_5$ e, consequentemente, o AFD mínimo equivalente ao AFD representado é:




É evidente que, se substituir os nomes dos estados, este autómato é exatamente o que se obteve em **2b)**. Portanto, o AFD dado em 2c) é equivalente a $\mathcal{M}$.

**3.** Sejam $L$ e $M$ as linguagens de alfabeto $\{\mathtt{a},\mathtt{b},\mathtt{c}\}$ assim definidas.

$$\begin{array}{rcl} L &=& \{\mathtt{c}x\mathtt{c}y \mid x,y \in \{\mathtt{a},\mathtt{b}\}^\star \text{ e } 1 \leq |x| \leq 2|y|\} \\ M &=& \{\mathtt{c}x\mathtt{c}y\mathtt{c}z \mid x,y \in \{\mathtt{b}\}\{\mathtt{b}\}^\star,\ z \in \{\mathtt{a}\}^\star,\ |y| = |x| \text{ e } |z| = |x| + |y|\} \end{array}$$

**Resolva apenas uma das duas alíneas seguintes:**

**a)** Por aplicação do lema da repetição para linguagens regulares, mostre que $L$ não é regular.

**b)** Por aplicação do teorema de Myhill-Nerode, mostre que $M$ não é regular.

*Resposta 3a):*

Seja $n \geq 1$ qualquer (notação do Lema). Tome-se $x = \mathtt{c}\mathtt{b}^{2n}\mathtt{c}\mathtt{b}^n \in L$. Tem-se $|x| = 3n+2 \geq n$. Seja a decomposição de $x$ denotada por $uvw$, como no enunciado do Lema, com $|uv| \leq n$ e $v \neq \varepsilon$. Tal prefixo $uv$ termina sempre antes do segundo $\mathtt{c}$ de $x$. Assim, existem duas possibilidades para $uv$:

- $u = \varepsilon$ e $v$ começa por $\mathtt{c}$. Neste caso, se se tomar $i = 0$ (i.e., cortar $v$), tem-se $uv^iw \notin L$ porque perderia tal $\mathtt{c}$.

- $u \neq \varepsilon$ e $v$ é da forma $\mathtt{b}^k$, para algum $k$ tal que $1 \leq k < n$. Neste caso, se se repetir $v$, i.e., tomar $i = 2$, então $uv^iw \notin L$, pois o número de símbolos entre os dois $\mathtt{c}$'s seria $2n+k$. Assim, como $k \geq 1$, excederia o dobro do número de símbolos que $uv^iw$ teria à direita do segundo $\mathtt{c}$ (que continuaria a ser $n$ pois sendo $|uv| \leq n$, o sufixo $w$ terminaria sempre em $\mathtt{b}^{n+1}\mathtt{c}\mathtt{b}^n$).

Conclui-se que $L$ não satisfaz a condição do lema da repetição e, consequentemente, não é regular.

NB: na prova de que uma linguagem não verifica a condição do lema, a palavra $x$ escolhida terá de depender de $n$ e nem o valor de $n$ nem a decomposição $uv$ de $x$ podem ser concretizados.

*Resposta 3b):*

Pelo teorema de Myhill-Nerode, $M$ é regular se e só se o conjunto das classes de equivalência de $R_M$ é finito. Mas, se $n$ e $m$ forem inteiros positivos distintos então $(\mathtt{c}\mathtt{b}^n, \mathtt{c}\mathtt{b}^m) \notin R_M$, pois $\mathtt{c}\mathtt{b}^n\mathtt{c}\mathtt{b}^n\mathtt{c}\mathtt{a}^{2n} \in M$ e $\mathtt{c}\mathtt{b}^m\mathtt{c}\mathtt{b}^n\mathtt{c}\mathtt{a}^{2n} \notin M$ (o que quer dizer que existe $z \in \Sigma^\star$ tal que $\mathtt{c}\mathtt{b}^n z \in M$ e $\mathtt{c}\mathtt{b}^m z \notin M$). Assim, cada palavra da forma $\mathtt{c}\mathtt{b}^n$, com $n \geq 1$, pertence a uma classe distinta. Logo, o conjunto das classes de $R_M$ é infinito e, consequentemente, $M$ não é regular.

**Resolva apenas uma das duas alíneas seguintes:**

**c)** Determine uma gramática independente de contexto que gere $L$. A gramática pode ser ambígua. Justifique sucintamente a correção dessa gramática e apresente uma árvore de derivação para cabcbba.

(NB: pode ser útil resolver a alínea **c)** para responder a **e)**)

**d)** Determine um autómato de pilha que reconheça $L$ por pilha vazia. Explique sucintamente o significado dos estados e de que modo garantem a correção do autómato. Usando a relação $\vdash$ (de mudança de configuração), mostre que cbbbcb não é aceite por esse autómato mas cbcb é aceite.

*Resposta 3c):*

Ideia: As palavras são da forma $\mathtt{c}x\mathtt{c}y$ com $1 \leq |x| \leq 2|y|$, pelo que, qualquer símbolo de $y$ pode dar origem a dois símbolos de $x$ no máximo. Comece-se por esgotar $x$, emparelhando cada dois símbolos de $x$ com um de $y$, processando $x$ da esquerda para a direita e $y$ da direita para a esquerda. No fim, poderá sobrar um símbolo de $x$ se $|x|$ for ímpar. Esse símbolo terá de ser ainda emparelhado com um símbolo de $y$. Finalmente, só restará uma subpalavra $\mathtt{c}y'$, com $y' \in \{\mathtt{a},\mathtt{b}\}^\star$.

(Continua, v.p.f.)

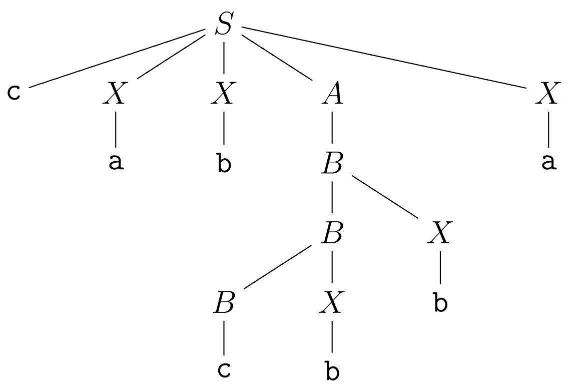
*Resposta 3c) (cont.):*

Tal ideia conduz à gramática $G_1 = (\{S, A, B, X\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, P_1, S)$, com $P_1$ dado por:

$$
\begin{array}{rcl}
S & \rightarrow & \mathtt{c}XXAX \mid \mathtt{c}XBX \\
A & \rightarrow & XXAX \mid XBX \mid B \\
B & \rightarrow & BX \mid \mathtt{c} \\
X & \rightarrow & \mathtt{a} \mid \mathtt{b}
\end{array}
$$

As regras para $S$ garantem que a palavra começa por c e que $|x| \geq 1$, além do que se disse acima sobre o emparelhamento de símbolos. A primeira regra para $A$ procede ao esgotamento dos símbolos de $x$, podendo sobrar ainda um símbolo (que será tratado pela segunda regra) ou nenhum (terceira regra). A variável $B$ produzirá $\mathtt{c}y'$. Pode-se concluir que $G_1$ não é ambígua (embora não fosse pedido).

A árvore de derivação da palavra cabcbba é:



*Resposta 3d):*

Ideia: Coloca um símbolo Y na pilha por cada par de símbolos de $x$ e, se $|x|$ for ímpar, coloca ainda mais um Y. Depois de detetar o segundo c, retira um Y por cada símbolo de $y$. Quando volta a ter apenas o símbolo inicial Z na pilha, continua a processar símbolos de $y$, se existirem, ou retira Z (para aceitar a palavra se $y$ chegar ao fim). Inicialmente, está no estado $s_0$ e tem Z na pilha.

$$
\begin{array}{lclcll}
\delta(s_0, \mathtt{c}, \mathtt{Z}) & = & \{(s_1, \mathtt{Z})\} & & & \text{\% garante c inicial} \\
\delta(s_1, \mathtt{a}, \mathtt{Z}) & = & \{(s_2, \mathtt{YZ})\} & = & \delta(s_1, \mathtt{b}, \mathtt{Z}) & \text{\% em } s_1 \text{, está a carregar a pilha e } |x| \text{ é par} \\
\delta(s_2, \mathtt{a}, \mathtt{Y}) & = & \{(s_1, \mathtt{Y})\} & = & \delta(s_2, \mathtt{b}, \mathtt{Y}) & \text{\% em } s_2 \text{, está a carregar a pilha e } |x| \text{ é ímpar} \\
\delta(s_2, \mathtt{c}, \mathtt{Y}) & = & \{(s_3, \mathtt{Y})\} & & & \text{\% encontra segundo c (por ter Y garante } |x| \geq 1) \\
\delta(s_1, \mathtt{c}, \mathtt{Y}) & = & \{(s_3, \mathtt{Y})\} & & & \text{\% encontra segundo c (por ter Y garante } |x| \geq 1) \\
\delta(s_3, \mathtt{a}, \mathtt{Y}) & = & \{(s_3, \varepsilon)\} & = & \delta(s_3, \mathtt{b}, \mathtt{Y}) & \text{\% em } s_3 \text{, está a descarregar a pilha} \\
\delta(s_3, \mathtt{a}, \mathtt{Z}) & = & \{(s_3, \mathtt{Z})\} & = & \delta(s_3, \mathtt{b}, \mathtt{Z}) & \text{\% Y's esgotados, mas aceita o resto de } y \\
\delta(s_3, \varepsilon, \mathtt{Z}) & = & \{(s_3, \varepsilon)\} & & & \text{\% pilha vazia para poder aceitar a palavra}
\end{array}
$$

A palavra cbcb é reconhecida porque $(s_0, \mathtt{cbcb}, \mathtt{Z}) \vdash^\star (s_3, \varepsilon, \varepsilon)$ pois:

$$(s_0, \mathtt{cbcb}, \mathtt{Z}) \vdash (s_1, \mathtt{bcb}, \mathtt{Z}) \vdash (s_2, \mathtt{cb}, \mathtt{YZ}) \vdash (s_3, \mathtt{b}, \mathtt{YZ}) \vdash (s_3, \varepsilon, \mathtt{Z}) \vdash (s_3, \varepsilon, \varepsilon)$$

A palavra cbbbcb não é reconhecida porque a pilha não fica vazia quando o autómato termina o seu processamento, já que se tem necessariamente o seguinte:

$$(s_0, \mathtt{cbbbcb}, \mathtt{Z}) \vdash (s_1, \mathtt{bbbcb}, \mathtt{Z}) \vdash (s_2, \mathtt{bbcb}, \mathtt{YZ}) \vdash (s_1, \mathtt{bcb}, \mathtt{YZ}) \vdash (s_2, \mathtt{cb}, \mathtt{YYZ}) \vdash (s_3, \mathtt{b}, \mathtt{YYZ}) \vdash (s_3, \varepsilon, \mathtt{YZ})$$

**Resolva apenas uma das três alíneas seguintes:**

**e)** Defina uma gramática $G$ que gere $L$ e que esteja na forma normal de Chomsky. Explique sucintamente. Aplique o algoritmo CYK para mostrar que $\texttt{cabcbba} \in \mathcal{L}(G)$.

**f)** Apresente uma máquina de Turing que reconheça $M$. A máquina **não deve** repor o estado inicial da fita. Indique o significado dos estados.

**g)** Mostre que $M$ não é independente de contexto.

*Resposta 3e):*

NB: Pode-se obter uma gramática por conversão do autómato de pilha (pelo método dado no curso), mas seria demasiado trabalhoso. Por isso, assumir-se-á como ponto de partida a gramática obtida em **3c)**.

Em **3c)**, obteve-se a gramática $G_1 = (\{S, A, B, X\}, \{\texttt{a}, \texttt{b}, \texttt{c}\}, P_1, S)$ com

$$\begin{array}{lcl} S & \to & \texttt{c}XXAX \mid \texttt{c}XBX \\ A & \to & XXAX \mid XBX \mid B \\ B & \to & BX \mid \texttt{c} \\ X & \to & \texttt{a} \mid \texttt{b} \end{array}$$

Para reduzir $G_1$ à forma normal de Chomsky, introduz-se uma nova variável $C$ para não ter terminais no lado direito de regras que não são da forma $V \to t$, com $t \in \Sigma$. Remove-se também a produção unitária $A \to B$, criando duas regras novas para $A$ (que resultam da substituição de $B$ em $A \to B$ pelo lado direito das produções de $B$).

$$\begin{array}{lcl} S & \to & CXXAX \mid CXBX \\ A & \to & XXAX \mid XBX \mid BX \mid \texttt{c} \\ B & \to & BX \mid \texttt{c} \\ C & \to & \texttt{c} \\ X & \to & \texttt{a} \mid \texttt{b} \end{array}$$

Para obter uma gramática equivalente mas na forma normal de Chomsky, basta agora transformar todas as regras do tipo $V \to \gamma$ com $|\gamma| \geq 3$, usando variáveis auxiliares.

$$\begin{array}{lcl} S & \to & CR_1 \mid CW_1 \\ A & \to & XR_2 \mid XW_2 \mid BX \mid \texttt{c} \\ B & \to & BX \mid \texttt{c} \\ C & \to & \texttt{c} \\ X & \to & \texttt{a} \mid \texttt{b} \end{array} \qquad \begin{array}{lcl} R_1 & \to & XR_2 \\ R_2 & \to & XR_3 \\ R_3 & \to & AX \\ W_1 & \to & XW_2 \\ W_2 & \to & BX \end{array}$$

A tabela que resulta da análise da palavra cabcbba pelo algoritmo CYK encontra-se abaixo. Sendo $\texttt{cabcbba} = x_1x_2x_3x_4x_5x_6x_7$, a célula na linha $i$ e coluna $j$ contém o conjunto das possíveis categorias da subpalavra $x_j \dots x_{j+i-1}$. Conclui-se que $\texttt{cabcbba} \in \mathcal{L}(G)$ pois $S$ está no topo da tabela (i.e.,, na célula $(7, 1)$, que caracteriza $x_1 \dots x_7$).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| #7 | $S$ | | | | | | |
| #6 | $S$ | $A, R_1, R_2, R_3$ | | | | | |
| #5 | $S$ | $A, R_1, R_2, R_3$ | $A, R_2, R_3, W_1$ | | | | |
| #4 | $\emptyset$ | $A, R_1$ | $A, R_2, R_3, W_1$ | $R_3, A, B, W_2$ | | | |
| #3 | $R_3, A, B, W_2$ | $\emptyset$ | $R_2, W_1, A$ | $R_3, A, B, W_2$ | $\emptyset$ | | |
| #2 | $R_3, A, B, W_2$ | $\emptyset$ | $\emptyset$ | $R_3, A, B, W_2$ | $\emptyset$ | $\emptyset$ | |
| #1 | $A, B, C$ | $X$ | $X$ | $A, B, C$ | $X$ | $X$ | $X$ |
| | c | a | b | c | b | b | a |

*Resposta 3f):*

Ideia: Para reconhecer as palavras da forma $\texttt{c}x\texttt{c}y\texttt{c}z$ com $x, y \in \{\texttt{b}\}\{\texttt{b}\}^\star$, $z \in \{\texttt{a}\}^\star$, $|y| = |x|$ e $|z| = |x| + |y| = 2|x|$, a máquina vai "cortar" um símbolo de $y$ e dois símbolos de $z$ por cada símbolo de $x$. Para facilitar a localização do último símbolo de $x$ que foi cortado, usa X para cortar símbolos de $x$ e Y para cortar símbolos de $y$ e de $z$. O estado inicial é $s_0$ e • é o símbolo branco. O estado designado por "aceita" será o único estado final.

| | |
|---|---|
| $(s_0, \texttt{c}, \text{cortaX}, \texttt{c}, d)$ | % garante c inicial |
| $(\text{cortaX}, \texttt{b}, \text{procY}, \texttt{X}, d)$ | % garante $\|x\| \geq 1$ (corta um símbolo de $x$ e vai procurar $y$) |
| $(\text{procY}, \texttt{b}, \text{procY}, \texttt{b}, d)$ | % passa restantes b's de $x$ se existirem |
| $(\text{procY}, \texttt{c}, \text{cortaY}, \texttt{c}, d)$ | % encontra zona de $y$ e vai tentar cortar um símbolo |
| $(\text{cortaY}, \texttt{Y}, \text{cortaY}, \texttt{Y}, d)$ | % pode passar por símbolos já foram cortados |
| $(\text{cortaY}, \texttt{b}, \text{procZ}, \texttt{Y}, d)$ | % corta um símbolo de $y$ e vai procurar $z$ |
| $(\text{procZ}, \texttt{b}, \text{procZ}, \texttt{b}, d)$ | % passa restantes b's de $y$ se existirem |
| $(\text{procZ}, \texttt{c}, \text{cortaZ}, \texttt{c}, d)$ | % encontra zona de $z$ e vai tentar cortar dois símbolos |
| $(\text{cortaZ}, \texttt{Y}, \text{cortaZ}, \texttt{Y}, d)$ | % pode passar por símbolos já foram cortados |
| $(\text{cortaZ}, \texttt{a}, \text{corta2Z}, \texttt{Y}, d)$ | % corta um e vai tentar cortar outro |
| $(\text{corta2Z}, \texttt{a}, \text{maisX}, \texttt{Y}, e)$ | % após corte, vai para a esquerda ver se $x$ ainda tem símbolos |
| $(\text{maisX}, \texttt{Y}, \text{maisX}, \texttt{Y}, e)$ | |
| $(\text{maisX}, \texttt{c}, \text{maisX}, \texttt{c}, e)$ | |
| $(\text{maisX}, \texttt{b}, \text{maisX}, \texttt{b}, e)$ | |
| $(\text{maisX}, \texttt{X}, \text{procX}, \texttt{X}, d)$ | % encontra novamente zona de $x$ |
| $(\text{procX}, \texttt{b}, \text{procY}, \texttt{X}, d)$ | % $x$ ainda não terminou (corta mais um símbolo e vai procurar $y$) |
| $(\text{procX}, \texttt{c}, \text{fimX}, \texttt{X}, d)$ | % $x$ acabou; há que verificar que $y$ e $z$ contêm apenas Y's |
| $(\text{fimX}, \texttt{Y}, \text{fimX}, \texttt{Y}, d)$ | |
| $(\text{fimX}, \texttt{c}, \text{fimX}, \texttt{c}, d)$ | |
| $(\text{fimX}, \bullet, \text{aceita}, \bullet, e)$ | % nada restava em $y$ e $z$; pára em estado final (aceita a palavra) |

NB: A máquina descrita acima é determinística e pára sempre. As palavras que a fazem encravar (isto é, parar) num dos estados não finais não são reconhecidas.

*Resposta 3g):*

Seja $n \geq 1$ qualquer. Tome-se a palavra $z = \texttt{c}\texttt{b}^{2n}\texttt{c}\texttt{b}^{2n}\texttt{c}\texttt{a}^{4n} \in M$. Claramente, $|z| = 8n + 3 \geq n$. Em qualquer decomposição de $z$ tal que $z = uvwxy$ com $|vwx| \leq n$ e $vx \neq \varepsilon$, a palavra $vwx$ interseta no máximo dois dos três blocos que constituem $z$ (assume-se que cada c inicia um bloco). Assim, se se cortar $v$ e $x$, isto é, se $i = 0$, obtém-se uma palavra que perde algum c ou que não satisfaz a condição que define a relação entre o comprimento desses três blocos, pois um dos blocos permanece inalterado e pelo menos um dos restantes perde elementos. De acordo com a definição de $M$, os dois primeiros teriam de ter o mesmo comprimento e o terceiro teria o dobro menos uma unidade. Consequentemente, para $i = 0$, a palavra $uv^iwx^iy$ não pertencerá a $M$, qualquer que seja a decomposição $uvwxy$ de $z$, tal que $|vwx| \leq n$ e $vx \neq \varepsilon$. Portanto, a linguagem $M$ não é independente de contexto pois não satisfaz a condição do lema da repetição para linguagens independentes de contexto.

NB: na prova de que uma linguagem não verifica a condição do lema, a palavra $z$ escolhida terá de depender de $n$ e nem o valor de $n$ nem a decomposição $uvwxy$ de $z$ podem ser concretizados.

(Fim)