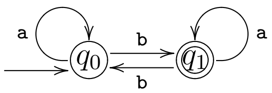
Modelos de Computação CC1004 2015/2016

Resolução de algumas questões

1. Sejam $\mathbf{K}$, $\mathbf{M}$ e $\mathbf{L}$ linguagens de alfabeto $\Sigma = \{\mathtt{a},\mathtt{b}\}$, com $\mathbf{K} = \{x \mid x \in \Sigma^\star$ e tem número ímpar de b's$\}$, $\mathbf{M} = \{x \mid x \in \Sigma^\star$ e tem abb como subpalavra$\}$ e $\mathbf{L} = \mathbf{K} \cap \mathbf{M} = \{x \mid x \in \mathbf{K}$ e $x \in \mathbf{M}\}$.

a) Desenhe o AFD mínimo que aceita $\mathbf{K}$.



b) Defina $\mathbf{K}$ por uma expressão regular (abreviada).

$$(\mathtt{a} + \mathtt{ba}^\star\mathtt{b})^\star\mathtt{ba}^\star$$

c) Indique uma expressão regular (abreviada) que descreva a linguagem $\mathbf{L}$.

$$(\mathtt{a} + \mathtt{ba}^\star\mathtt{b})^\star\mathtt{ba}^\star\mathtt{abb}(\mathtt{a} + \mathtt{ba}^\star\mathtt{b})^\star \;+\; (\mathtt{a} + \mathtt{ba}^\star\mathtt{b})^\star\mathtt{abb}(\mathtt{a} + \mathtt{ba}^\star\mathtt{b})^\star\mathtt{ba}^\star$$

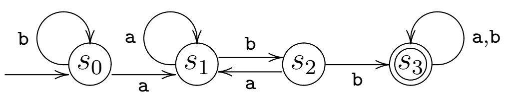
d) Apresente as regras (de produção) de uma GIC G que gere $\mathbf{L}$, com símbolo inicial A, e descreva informalmente $\mathcal{L}_X = \{w \mid X \Rightarrow_G^\star w$ e $w \in \Sigma^\star\}$, para cada variável X de G, com exceção de A.

$$\begin{array}{lcl} A & \rightarrow & Z\mathtt{b}X\mathtt{abb}Z \mid Z\mathtt{abb}Z\mathtt{b}X \\ Z & \rightarrow & \varepsilon \mid \mathtt{a}Z \mid \mathtt{b}X\mathtt{b}Z \\ X & \rightarrow & \varepsilon \mid \mathtt{a}X \end{array}$$

$\mathcal{L}_Z$ é formada pelas palavras de $\Sigma^\star$ com número par de b's.

$\mathcal{L}_X$ é constituída pelas palavras de $\Sigma^\star$ que não têm b's.

e) Desenhe o diagrama do AFD mínimo que reconhece $\mathbf{M}$ e descreva $\mathcal{L}_s = \{x \mid x \in \Sigma^\star$ e $\hat{\delta}(s_0, x) = s\}$ por uma expressão regular (abreviada), para cada estado s, sendo s_0 o estado inicial.



$\mathcal{L}_{s_0}$: $\mathtt{b}^\star$

$\mathcal{L}_{s_1}$: $\mathtt{b}^\star\mathtt{a}(\mathtt{a} + \mathtt{ba})^\star$

$\mathcal{L}_{s_2}$: $\mathtt{b}^\star\mathtt{a}(\mathtt{a} + \mathtt{ba})^\star\mathtt{b}$

$\mathcal{L}_{s_3}$: $\mathtt{b}^\star\mathtt{a}(\mathtt{a} + \mathtt{ba})^\star\mathtt{bb}(\mathtt{a} + \mathtt{b})^\star \equiv (\mathtt{a} + \mathtt{b})^\star\mathtt{abb}(\mathtt{a} + \mathtt{b})^\star$

f) Da análise da construção do AFD produto, pode-se concluir que o AFD mínimo que reconhece $\mathbf{L}$ tem no máximo *oito* estados e exatamente *um* estados finais/estado final. Complete a frase e justifique abaixo sucintamente as respostas, enunciando os resultados que as suportam.

O conjunto de estados do AFD produto é o produto cartesiano dos conjuntos de estados dos dois AFDs (segundo a construção geral, sem descartar os estados não acessíveis do inicial). Assim, existe um AFD que reconhece $\mathbf{L}$ e que tem $2 \times 4 = 8$ estados. Logo, o AFD mínimo tem no máximo oito estados.

Para reconhecer $\mathbf{K} \cap \mathbf{M}$, o conjunto de estados finais do AFD produto é $\{(q_1, s_3)\}$ e, assim, tem um só elemento. Como $\mathbf{L} = \mathbf{K} \cap \mathbf{M} \neq \emptyset$, as palavras de $\mathbf{L}$ levam tal AFD do seu estado inicial (q_0, s_0) ao estado (q_1, s_3). Qualquer que seja o AFD A que reconheça $\mathbf{L}$, tem-se $\mathcal{C}_x \subseteq [x]$, sendo $\mathcal{C}_x$ e $[x]$ as classes de x para R_A e $R_{\mathbf{L}}$, segundo a notação dada nas aulas, o que implica que o AFD mínimo tem exatamente um estado final.

(Continua)

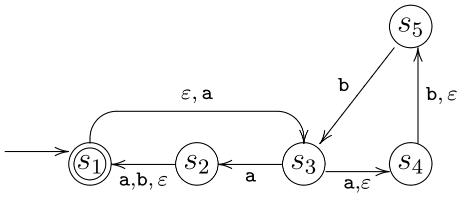
N.º [ ] Nome [ ]

g) Por aplicação do corolário do teorema de Myhill-Nerode, determine o AFD mínimo que reconhece **L**. (Em alternativa pode indicar um AFD que reconhece **L**, justificar a sua correção e minimizá-lo pelo algoritmo de Moore). Em ambos os casos, deve **justificar detalhadamente** os passos da construção.

[Resposta omitida.]

NB: No pressuposto de que os AFDs indicados em 1a) e 1e) estavam corretos, para a resposta alternativa, bastaria construir o AFD produto, seguindo o método dado nas aulas, e aplicar o algoritmo de Moore para minimizar esse AFD.

2. Seja A o AFND-ε representado pelo diagrama de transição seguinte.



a) Indique **o conjunto** de estados em que A pode estar após consumir `bbaab`. $\{s_1, s_3, s_4, s_5\}$

b) Desenhe o diagrama de transição do AFD equivalente, que se obtém pelo método de conversão (baseado em subconjuntos). Deve **obrigatoriamente manter as designações dos estados do AFD como conjuntos**.

[Resposta omitida.]

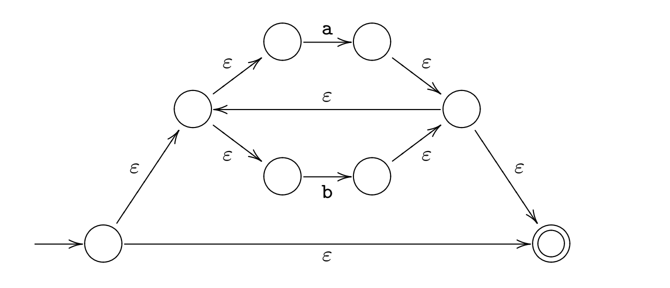
NB: O estado inicial do AFD é o $Fecho_\varepsilon(s_1) = \{s_1, s_3, s_4, s_5\}$. *O AFD tem outros dois estados:* $\{s_1, s_2, s_3, s_4, s_5\}$ *e* $\{s_3, s_4, s_5\}$. *Este último é o único que não é final. Os estados finais e o estado inicial têm de ser assinalados sempre. É também importante compreender que a resposta à alínea 2a) não pode ser inconsistente com o que se conclui da alínea 2b). De facto, as designações dos estados do AFD correspondem exatamente ao conjunto de estados em que o AFND-ε podia estar nas mesmas circunstâncias. É nisso que se baseia a correção do método de conversão dado!*

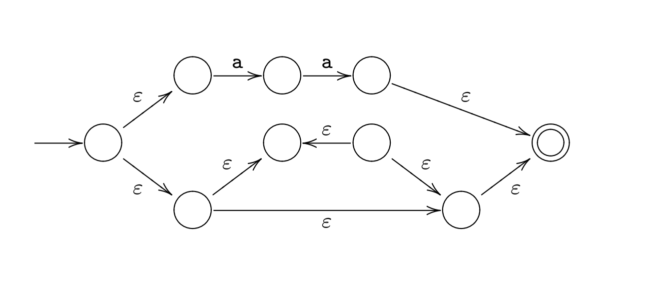
(Continua)

N.º ____ Nome ____

3. Sejam $r = ((\mathtt{a}+\mathtt{b})^\star)$ e $s = ((\mathtt{aa}) + (\emptyset^\star))$ expressões regulares sobre $\Sigma = \{\mathtt{a},\mathtt{b}\}$.

a) Desenhe os diagramas de transição dos autómatos finitos que resultam da aplicação do método de Thompson às expressões regulares r e s, segundo a construção dada nas aulas.





b) Usando a definição indutiva de expressão regular sobre Σ e de linguagem que a expressão descreve, prove que $\mathcal{L}((rs)) = \Sigma^\star$. **Apresente os passos intermédios.**

Usando a definição referida, tem-se:

$\mathcal{L}((rs)) = \mathcal{L}(r)\mathcal{L}(s)$.

$\mathcal{L}(r) = (\mathcal{L}((\mathtt{a}+\mathtt{b})))^\star = (\mathcal{L}(\mathtt{a}) \cup \mathcal{L}(\mathtt{b}))^\star = (\{\mathtt{a}\} \cup \{\mathtt{b}\})^\star = \{\mathtt{a},\mathtt{b}\}^\star$, ou seja, $\mathcal{L}(r) = \Sigma^\star$.

$\mathcal{L}(s) = \mathcal{L}((\mathtt{aa})) \cup \mathcal{L}((\emptyset^\star)) = \mathcal{L}(\mathtt{a})\mathcal{L}(\mathtt{a}) \cup (\mathcal{L}(\emptyset))^\star = \{\mathtt{aa}\} \cup \emptyset^\star$.

Por definição de fecho de Kleene, $\emptyset^\star = \{\varepsilon\}$. Logo, $\mathcal{L}(s) = \{\mathtt{aa}\} \cup \{\varepsilon\} = \{\varepsilon, \mathtt{aa}\}$.

Como $\varepsilon \in \mathcal{L}(s)$, então $\mathcal{L}(r) \subseteq \mathcal{L}(r)\mathcal{L}(s)$. Ou seja, $\Sigma^\star \subseteq \mathcal{L}(r)\mathcal{L}(s)$. Portanto, $\mathcal{L}(r)\mathcal{L}(s) = \Sigma^\star$, pois, por definição de linguagem de alfabeto Σ, tem-se $\mathcal{L}(r)\mathcal{L}(s) \subseteq \Sigma^\star$.

4. Seja $G = (\{S, X, Y\}, \{\mathtt{a},\mathtt{b}\}, P, S)$, com P dado por:

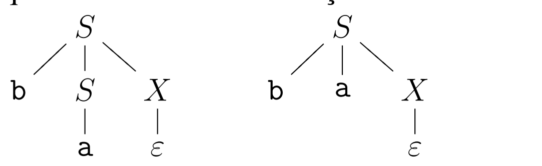
$$\begin{aligned} S &\to \mathtt{aa}YY \mid \mathtt{b}SX \mid \mathtt{ba}X \mid \mathtt{a} \\ X &\to \varepsilon \mid \mathtt{b}X \\ Y &\to \mathtt{a}YY \mid \mathtt{b} \end{aligned}$$

a) Diga, justificando, se $\mathtt{aabbb} \in \mathcal{L}(G)$.

$\mathtt{aabbb} \notin \mathcal{L}(G)$ porque para tentar derivar a palavra seria necessário aplicar a regra $S \to \mathtt{aa}YY$. A seguir, como não se pode introduzir mais a's, só se pode usar a regra $Y \to \mathtt{b}$, mas a palavra que se obteria seria aabb.

b) Prove que G é ambígua.

Por exemplo, a palavra ba tem mais do que uma árvore de derivação:



c) Apresente a noção de GIC na forma normal de Chomsky.

GIC em que as regras são da forma $A \to BC$ ou $A \to a$, com B, C variáveis e $a \in \Sigma$, quaisquer.

d) Por conversão de G, determine uma GIC G' equivalente a G mas sem variáveis que gerem ε. A seguir, converta G' à forma normal de Chomsky.

$$\begin{aligned} S &\to \mathtt{aa}YY \mid \mathtt{b}SX \mid \mathtt{ba}X \mid \mathtt{a} \\ S &\to \mathtt{b}S \mid \mathtt{ba} \\ X &\to \mathtt{b} \mid \mathtt{b}X \\ Y &\to \mathtt{a}YY \mid \mathtt{b} \end{aligned}$$

$$\begin{aligned} S &\to AZ \mid BK \mid BM \mid \mathtt{a} & Z &\to AR \\ S &\to BS \mid BA & R &\to YY \\ X &\to \mathtt{b} \mid BX & K &\to SX \\ Y &\to AW \mid \mathtt{b} & M &\to AX \\ A &\to \mathtt{a} & W &\to YY \\ B &\to \mathtt{b} & & \end{aligned}$$

(Continua)

N.º [ ] Nome [ ]

5. Considere a linguagem $L = \{y \mid y \text{ é capícua}\} \cap \{y \mid y \text{ tem número ímpar de } \mathtt{b}\text{'s ou começa por } \mathtt{c}\}$, de alfabeto $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

a) Use o lema da repetição ou o teorema de Myhill-Nerode para provar que L não é regular.

Prova pelo Lema da Repetição: Dado $n > 0$, se tomarmos $z = \mathtt{c}^n\mathtt{b}\mathtt{c}^n$, temos $z \in L$ e $|z| \geq n$ e, sendo $u, v, w \in \Sigma^\star$ quaisquer, com $z = uvw$ e $|uv| \leq n$ e $v \neq \varepsilon$, o prefixo uv está à esquerda de $\mathtt{b}$ em z. Assim, para $i = 0$, a palavra uv^iw não pertence a L, pois $uv^0w = uw = \mathtt{c}^{n-|v|}\mathtt{b}\mathtt{c}^n$ e não é capícua, dado que $|v| \geq 1$. Isto significa que L não satisfaz a condição do lema da repetição para linguagens regulares. Portanto, não é regular.

Prova pelo teorema de Myhill-Nerode: Tem-se $(\mathtt{c}^n, \mathtt{c}^m) \notin R_L$, quaisquer que sejam n e m tais que $0 < m < n$, pois para $z = \mathtt{b}\mathtt{c}^n$ tem-se $\mathtt{c}^n z \in L$ mas $\mathtt{c}^m z \notin L$. Tal significa que as classes de equivalência das palavras $\mathtt{c}^k$, com $k > 0$, são disjuntas duas a duas. Portanto, o conjunto das classes de equivalência de R_L é infinito, dado que inclui $\{[\mathtt{c}^k] \mid k > 0\}$.

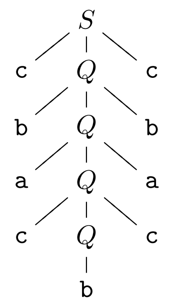
b) Indique uma GIC G **não ambígua** que gere L e represente a(s) árvore(s) de derivação de `cbacbcabc`.

As regras são:

$$
\begin{array}{lcl}
S & \to & \mathtt{c}Q\mathtt{c} \mid \mathtt{c} \\
S & \to & \mathtt{b}T\mathtt{b} \mid \mathtt{a}T\mathtt{a} \mid \mathtt{b} \\
T & \to & \mathtt{b} \mid \mathtt{b}T\mathtt{b} \mid \mathtt{a}T\mathtt{a} \mid \mathtt{c}T\mathtt{c} \\
Q & \to & \varepsilon \mid \mathtt{a} \mid \mathtt{b} \mid \mathtt{c} \\
Q & \to & \mathtt{b}Q\mathtt{b} \mid \mathtt{a}Q\mathtt{a} \mid \mathtt{c}Q\mathtt{c}
\end{array}
$$

O símbolo inicial é S.

A única árvore de derivação para `cbacbcabc` é:



c) Explique como é que garantiu a não ambiguidade de G.

Adaptámos a GIC dada nas aulas para a linguagem das capícuas (que era uma GIC não ambígua). Para não introduzir ambiguidade, as regras para S separam as capícuas que começam por `c` das capícuas que começam por `a` ou `b`. Estas últimas têm comprimento ímpar e o símbolo `b` no meio, enquanto que as começam por `c` podem ter comprimento par ou ímpar (e qualquer símbolo no meio se tiverem comprimento ímpar).

A variável T gera as capícuas da forma $w\mathtt{b}w^R$, com $w \in \Sigma^\star$.

A variável Q gera as capícuas de alfabeto Σ, i.e., as palavras da forma wtw^R, com $w \in \Sigma^\star$ e $t \in \Sigma \cup \{\varepsilon\}$.

Resolva apenas uma das alíneas d) , e), e f)

d) Apresente um autómato de pilha que reconheça a linguagem $L \cap \{y \mid y \text{ tem número ímpar de } \mathtt{b}\text{'s}\}$. Pode escolher o critério de aceitação (ou pilha vazia ou estados finais), mas deve indicar a sua opção. **Indique sucintamente a interpretação de cada estado e as ideias principais subjacentes**.

e) Prove que a linguagem $L \cap \{x \mid x \text{ tem igual número de } \mathtt{a}\text{'s e } \mathtt{c}\text{'s}\}$ não satisfaz a condição do lema da repetição para LICs para nenhum $n > 0$.

f) Apresente uma máquina de Turing que reconheça a linguagem $L \cap \{x \mid x \text{ tem igual número de } \mathtt{a}\text{'s e } \mathtt{c}\text{'s}\}$. O símbolo branco é $\bullet$ e a máquina pode destruir a palavra. Descreva **as ideias principais** do algoritmo.

Use o verso da folha para responder à questão. [Resposta omitida.]

(Fim)