Departamento de Ciência de Computadores FCUP

# Modelos de Computação (CC1004) 2014/15

Exame Recurso (06.07.2015) *duração: 3h*

**Resolução de questões selecionadas (v1)** O alfabeto $\Sigma$ é $\{0, 1\}$, exceto em **3**, **6**, **7**.

**1.** Seja $r$ a expressão regular $((((1^\star) + (10)) + (11))^\star)$ sobre $\Sigma$.

**a)** [!!] Desenhe o diagrama de transição do AFND-$\varepsilon$ que resulta da aplicação do método de Thompson à expressão regular $r$, de acordo com a **construção definida nas aulas**. Desenhe um só diagrama e indique (à parte) os estados iniciais e finais das componentes associadas a **todas as sub-expressões** não elementares.

*Sugestão:* Rever os apontamentos e outras resoluções de testes/exames de 2014/15.

**b)** [!!] Indique uma expressão regular (**não abreviada**) equivalente $r$ mas mais simples. Justifique.

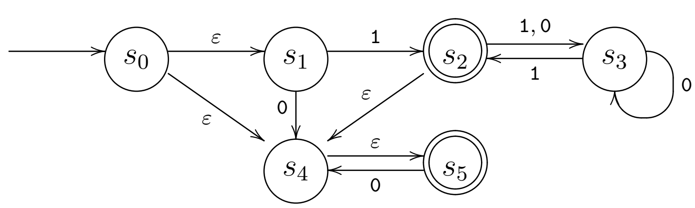
*Resposta:*

A expressão regular $r$ é equivalente à expressão regular $((1 + (10))^\star)$.

*Justificação:* Escrevendo $r$ na forma abreviada, isto é, como $(1^\star + 10 + 11)^\star$, vemos que $11 \in \mathcal{L}(1^\star)$ e, por isso, $r$ é equivalente a $(1^\star + 10)^\star$, sendo esta expressão equivalente a $(1 + 10)^\star$, por definição de fecho de Kleene. Acima, apresentamo-la na forma **não abreviada** (isto é, com todos os parentesis que, segundo a definição de expressão regular, deveria ter).

Importa salientar que $(1 + 10)^\star$ não é equivalente a $1^\star + (10)^\star$.

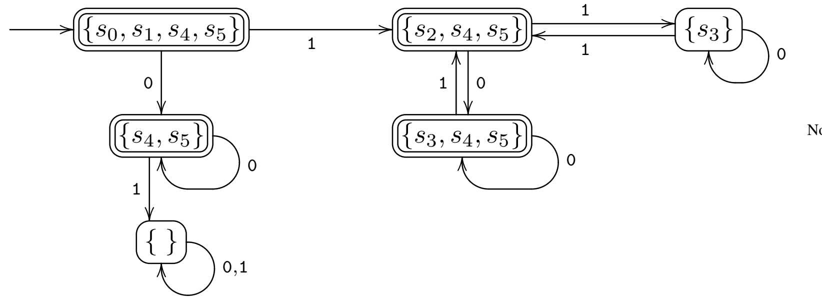
**2.** Seja $A$ o AFND-$\varepsilon$ representado pelo diagrama seguinte.




**a)** [!!] Determine o diagrama de transição do AFD $A'$, equivalente a $A$, que se obtém por aplicação do método de conversão baseado em subconjuntos. No diagrama de $A'$, tem **obrigatoriamente de usar conjuntos para designar os estados** e manter apenas estados acessíveis do estado inicial.

*Resposta:*

O estado inicial do AFD equivalente é $Fecho_\varepsilon(s_0) = \{s_0, s_1, s_4, s_5\}$ e o diagrama de transição é:




Notar que este AFD tem quatro estados finais.

**b)** [!!] Para cada uma das palavras $\varepsilon$, `1010` e `01`, indique: **(i)** o conjunto de estados em que o AFND-$\varepsilon$ $A$ se pode encontrar após consumir a palavra; **(ii)** se a palavra pertence a $\mathcal{L}(A)$; **(iii)** o estado em que o AFD $A'$ fica após consumir a palavra (use a designação dada em **2a)**); **(iv)** se o AFD $A'$ aceita a palavra.

*Resposta:*

O AFD $A'$ é equivalente ao AFND-$\varepsilon$ $A$ (o que significa que $\mathcal{L}(A) = \mathcal{L}(A')$) e, de acordo com o método de conversão, as designações dos estados do AFD indicam os conjuntos estados em que o AFND-$\varepsilon$ pode estar em circunstâncias iguais. Nas condições do enunciado, não há diferença entre as respostas a **(i)** e **(iii)**, sendo também semelhantes as respostas a **(ii)** e **(iv)**:

| | (i) e (iii) | (iv) e (ii) |
|---|---|---|
| $\varepsilon$ | $\{s_0, s_1, s_4, s_5\}$ | aceite, ou seja, $\varepsilon$ pertence a $\mathcal{L}(A)$ |
| `1010` | $\{s_3, s_4, s_5\}$ | aceite, ou seja, `1010` pertence a $\mathcal{L}(A)$ |
| `01` | $\{\}$ | não aceite, ou seja, `01` não pertence a $\mathcal{L}(A)$ |

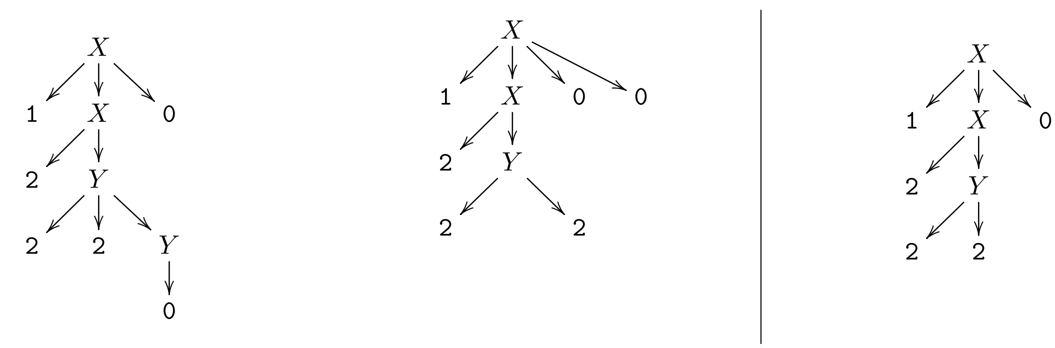
**3.** Seja $G = (\{X, Y\}, \{0, 1, 2\}, P, X)$ uma gramática independente de contexto com $P$ dado por:

$$\begin{aligned} X &\rightarrow \mathtt{1}X\mathtt{0} \mid \mathtt{1}X\mathtt{00} \mid \mathtt{2}Y \mid \mathtt{2} \\ Y &\rightarrow \mathtt{22} \mid \mathtt{2} \mid \mathtt{0} \mid \mathtt{22}Y \end{aligned}$$

**a)** [!!] Apresente todas as árvores de derivação das palavras `122200` e `12220` pertencentes a $\mathcal{L}(G)$.

*Resposta:*

Existem duas árvores de derivação para `122200` e apenas uma para `12220`



**b)** [!!] Averigue se `120000` ou `220` pertence à linguagem gerada pela gramática. Justifique.

*Resposta:*

Nenhuma das palavras pertence a $\mathcal{L}(G)$ pois:

- $X \not\Rightarrow^{\star}$ `120000` porque as palavras de $\mathcal{L}(G)$ com apenas um `1` têm no máximo três `0`'s, já que apenas uma das regras $X \rightarrow \mathtt{1}X\mathtt{0}$ ou $X \rightarrow \mathtt{1}X\mathtt{00}$ poderia ser usada na derivação. Depois, para se poder introduzir mais `0`'s, seria necessário usar $X \rightarrow \mathtt{2}Y$, mas não se obteria `120000`, pois nenhuma palavra que seja derivada de $Y$ tem dois ou mais `0`'s.
- $X \not\Rightarrow^{\star}$ `220` porque, como a palavra não tem `1`'s, temos de usar $X \rightarrow \mathtt{2}Y$ e a palavra `20` não pode ser derivada de $Y$.

**c)** [!!] Apresente a noção de gramática ambígua e averigue se $G$ é ambígua.

*Resposta:*

Uma gramática $\mathcal{G}$ é ambígua se alguma palavra de $\mathcal{L}(\mathcal{G})$ admitir mais do que uma árvore de derivação segundo $\mathcal{G}$. Consequentemente, de **3a)** concluimos que a gramática dada é ambígua.

**d)** [!!] Indique uma expressão regular (abreviada) que descreva $\{y \mid y \in \{0,1,2\}^\star,\ Y \Rightarrow^\star y\}$

*Resposta:*

A linguagem gerada a partir de $Y$ é descrita pela expressão regular $(22)^\star(22+2+0)$.

**e)** [!] Apresente a forma genérica das palavras de $\mathcal{L}(G)$. Explique como a deduziu (use $\Rightarrow$, $\Rightarrow^n$, ou $\Rightarrow^\star$).

*Resposta:*

A forma geral das palavras de $\mathcal{L}(G)$ é $1^{p+q}2w0^{p+2q}$, com $p,q \geq 0$ e $w \in \mathcal{L}(\varepsilon + (22)^\star(22+2+0))$, pois

$X \Rightarrow^{p+q} 1^{p+q}X0^{p+2q} \Rightarrow 1^{p+q}2Y0^{p+2q} \Rightarrow^\star 1^{p+q}2y0^{p+2q}$, para $y \in \mathcal{L}((22)^\star(22+2+0))$, ou

$X \Rightarrow^{p+q} 1^{p+q}X0^{p+2q} \Rightarrow 1^{p+q}20^{p+2q}$,

sendo $p$ e $q$ o número de vezes que as regras $X \to 1X0$ e $X \to 1X00$ são usadas (por qualquer ordem).

Estas formas correspondem a $1^n2w0^m$, com $0 \leq n \leq m \leq 2n$, com $w \in \mathcal{L}(\varepsilon + (22)^\star(22+2+0)) = \mathcal{L}(2^\star + (22)^\star 0)$

**f)** [!!] Prove que a linguagem independente de contexto $\mathcal{L}(G)$ é não ambígua.

*Resposta:*

Uma LIC é não ambígua se for gerada por alguma GIC não ambígua. A ambiguidade de $G$ resulta da possibilidade de intercalar a aplicação das regras $X \to 1X0$ e $X \to 1X00$ e ainda do facto de $Y$ poder acrescentar um $0$ à palavra.

A gramática $G' = (\{X,K,Z,R\},\{0,1,2\},P',X)$ gera $\mathcal{L}(G)$ e resolve esses problemas, sendo $P'$ dado por:

$$\begin{array}{lcl}
X & \to & 1X00 \mid 1K0 \mid 2Z \mid 2R0 \\
K & \to & 1K0 \mid 2Z \\
Z & \to & 2Z \mid \varepsilon \\
R & \to & 22R \mid \varepsilon
\end{array}$$

Esta gramática resulta da análise da forma genérica $1^q1^p2w0^p0^{2q}$, com $p,q \geq 0$ e $w \in \mathcal{L}(2^\star + (22)^\star 0)$. Depois de aplicar a regra $X \to 1K0$, já não é possível usar a regra $X \to 1X00$. Por outro lado, a regra $X \to 2R0$ só é usada se a palavra gerada for da forma $1^q2w0^{2q+1}$, com $q \geq 0$ e $w \in \mathcal{L}((22)^\star)$. A gramática $G'$ é não ambígua porque cada palavra de $\mathcal{L}(G')$ tem uma só derivação pela esquerda em $G'$ (não admite mesmo mais do que uma derivação):

- para $1^{p+q}22^r0^{p+2q}$, com $q \geq 0$, $p \geq 1$ e $r \geq 0$, é necessário aplicar $q$ vezes a regra $X{\to}1X00$, depois $X{\to}1K0$, a seguir $p-1$ vezes a regra $K{\to}1K0$, depois $K{\to}2Z$, depois $r$ vezes a regra $Z{\to}2Z$ e finalmente $Z{\to}\varepsilon$;

$$X \Rightarrow^q 1^qX0^{2q} \Rightarrow 1^q1K00^{2q} \Rightarrow^{p-1} 1^{p+q}K0^{p+2q} \Rightarrow 1^{p+q}2Z0^{p+2q} \Rightarrow^r 1^{p+q}22^rZ0^{p+2q} \Rightarrow 1^{p+q}22^r0^{p+2q}$$

- para $1^q22^r0^{2q}$, com $q \geq 0$ e $r \geq 0$, é necessário aplicar $q$ vezes a regra $X{\to}1X00$, depois a regra $X{\to}2Z$, e em seguida $r$ vezes a regra $Z{\to}2Z$ e finalmente $Z{\to}\varepsilon$;

$$X \Rightarrow^q 1^qX0^{2q} \Rightarrow 1^q2Z0^{2q} \Rightarrow^r 1^q22^rZ0^{2q} \Rightarrow 1^q22^r0^{2q}$$

- para $1^q22^{2r}0^{2q+1}$, com $q \geq 0$ e $r \geq 0$, é necessário aplicar $q$ vezes a regra $X{\to}1X00$, depois $X{\to}2R0$, a seguir $r$ vezes a regra $R{\to}22R$ e finalmente $R{\to}\varepsilon$.

$$X \Rightarrow^q 1^qX0^{2q} \Rightarrow 1^q2R00^{2q} \Rightarrow^k 1^q22^{2k}R0^{2q+1} \Rightarrow 1^q22^{2k}0^{2q+1}$$

**g)** [!] Converta $G$ à forma normal de Chomsky e aplique o algoritmo CYK para decidir se $12000 \in \mathcal{L}(G')$, sendo $G'$ tal gramática . Explique em detalhe a construção da **primeira** e da **quarta** linha da tabela e, a partir da tabela, indique todas as subpalavras de $z$ de $12000$ tais que $z \in \mathcal{L}(G') = \mathcal{L}(G)$.

*Sugestão:* Rever os apontamentos e outras resoluções de testes/exames de 2014/15.

**(Continua, v.p.f.)**

**4.** Seja $L = \{\mathtt{1}^n\mathtt{0}w\mathtt{1}^k \mid n \geq 1, k \geq 1, w \in \{\mathtt{0},\mathtt{1}\}^\star$ e $w$ tem número ímpar de $\mathtt{0}$'s$\}$.

**a)** [!] Averigue se o *fecho de Kleene de* $L$ é uma linguagem regular e, se for, caraterize-o ou por uma expressão regular (abreviada) ou por um autómato finito.

*Resposta:*

A linguagem das palavras que têm número ímpar de $\mathtt{0}$'s pode ser descrita pela expressão regular $(\mathtt{1} + \mathtt{01}^\star\mathtt{0})^\star\mathtt{01}^\star$.

Assim, $L$ pode ser descrita expressão regular $\mathtt{11}^\star\mathtt{0}(\mathtt{1} + \mathtt{01}^\star\mathtt{0})^\star\mathtt{01}^\star\mathtt{11}^\star$, a qual é equivalente a $\mathtt{11}^\star\mathtt{0}(\mathtt{1} + \mathtt{01}^\star\mathtt{0})^\star\mathtt{011}^\star$ e, portanto, $(\mathtt{11}^\star\mathtt{0}(\mathtt{1} + \mathtt{01}^\star\mathtt{0})^\star\mathtt{011}^\star)^\star$ descreve $L^\star$, o que quer dizer que $L^\star$ é uma linguagem regular.

**b)** [!] Seja $R_L$ a relação de equivalência definida em $\Sigma^\star$ por $(x, y) \in R_L$ se e só se $xz \in L \Leftrightarrow yz \in L$, para todo $z \in \Sigma^\star$. Quantas classes de equivalência $\mathtt{101}$, $\mathtt{10010}$, $\mathtt{0011}$, $\mathtt{1111}$ e $\mathtt{11010100}$ definem? Justifique.

*Resposta:*

$\mathtt{101}z \in L$ se e só se $z$ tem número ímpar de $\mathtt{0}$'s e termina em $\mathtt{1}$.

$\mathtt{10010}z \in L$ se e só se $z$ tem número ímpar de $\mathtt{0}$'s e termina em $\mathtt{1}$.

$\mathtt{0011}z \notin L$, para todo $z \in \{\mathtt{0},\mathtt{1}\}^\star$.

$\mathtt{1111}z \in L$ se e só se $z$ tem número par de $\mathtt{0}$'s, maior ou igual a dois, e termina em $\mathtt{1}$.

$\mathtt{11010100}z \in L$ se e só se $z$ tem número par de $\mathtt{0}$'s, possivelmente zero, e termina em $\mathtt{1}$.

Portanto, as palavras definem quatro classes de equivalência: $[\mathtt{101}] = [\mathtt{10010}]$, $[\mathtt{0011}]$, $[\mathtt{1111}]$, e $[\mathtt{11010100}]$.

**c)** [!!] O conjunto das classes de equivalência de $R_L$ é finito ou infinito? Justifique.

*Resposta:*

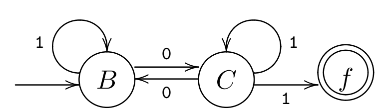
Pelo teorema de Myhill-Nerode, sabemos que $\Sigma^\star/R_L$ (o conjunto das classes de equivalência de $R_L$) é finito se e só se $L$ é uma linguagem regular. Em **4a)**, indicámos uma expressão regular que descreve $L$. Portanto, $\Sigma^\star/R_L$ é finito.

**d)** [!] Indique uma gramática independente de contexto **não ambígua** que gere $L$. Justifique sucintamente porque é que a gramática gera $L$ e é não ambígua.

*Resposta:*

A linguagem $L$ é gerada pela gramática $G = (\{S, A, B, C\}, \{\mathtt{0},\mathtt{1}\}, P, S)$, com $P$ dado por:

$$
\begin{array}{rcl}
S & \to & AB \\
A & \to & \mathtt{1}A \mid \mathtt{10} \\
B & \to & \mathtt{1}B \mid \mathtt{0}C \\
C & \to & \mathtt{1}C \mid \mathtt{0}B \mid \mathtt{1}
\end{array}
$$




A forma geral das palavras de $L$, definida como $\mathtt{1}^n\mathtt{0}w\mathtt{1}^k$, onde $w \in \{\mathtt{0},\mathtt{1}\}^\star$ tem número ímpar de $\mathtt{0}$'s, $n \geq 1$ e $k \geq 1$, pode ser re-escrita como $\mathtt{1}^n\mathtt{0}w'\mathtt{1}$, onde $w' \in \{\mathtt{0},\mathtt{1}\}^\star$ é qualquer sequência com número ímpar de $\mathtt{0}$'s.

A gramática $G$ tem por base essa forma e decompõe a palavra como $\underbrace{\mathtt{1}^n\mathtt{0}}_{A}\underbrace{w'\mathtt{1}}_{B}$.

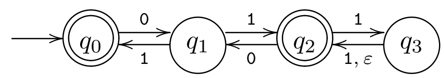
A linguagem gerada a partir de $A$ é $\mathcal{L}(\mathtt{1}^\star\mathtt{10})$ e, para gerar $\mathtt{1}^n\mathtt{0}$, é necessário aplicar $n-1$ vezes a regra $A \to \mathtt{1}A$ e depois a regra $A \to \mathtt{10}$. A linguagem gerada a partir de $B$ é constituída pelas palavras que têm número ímpar de $\mathtt{0}$'s e terminam em $\mathtt{1}$. A linguagem gerada a partir de $C$ é constituída pelas palavras que têm número par de $\mathtt{0}$'s e terminam em $\mathtt{1}$. As regras para $B$ e $C$ traduzem o autómato apresentado acima à direita, o qual, quando restringido aos estados $B$ e $C$, seria o AFD mínimo que reconhece as palavras que têm número ímpar de $\mathtt{0}$'s, se $C$ fosse assinalado como estado final. Existe em $G$ uma única derivação para o sufixo $w'\mathtt{1}$ a partir da variável $B$, a qual se pode identificar com o único percurso existente de $B$ até $f$ com consumo de $w'\mathtt{1}$, no autómato finito representado (a transição final nesse percurso seria substituída pela aplicação da regra $C \to \mathtt{1}$, que termina a derivação). Portanto, a gramática gera $L$ e é não ambígua.

**Resolva apenas DOIS dos problemas 5.–8. (se resolver mais, os últimos não são avaliados).**

**5.** [!] Aplique o método de eliminação de estados para obter uma expressão regular (abreviada) que descreva a linguagem reconhecida pelo o autómato representado à direita.

Na resposta deve:

- começar a eliminação de estados por $q_3$ **seguido de** $q_1$ **e** $q_0$;
- apresentar os **passos intermédios** e, sempre que for óbvio, **simplificar as expressões** (após indicar as originais).




*Sugestão:* Rever os apontamentos e outras resoluções de testes/exames de 2014/15.

**6.** [!] Defina um autómato de pilha que aceite a linguagem das palavras de $\{\mathtt{0},\mathtt{1},\mathtt{2}\}^\star$ que *têm mais dois* `1`*'s do que* `0`*'s ou mais dois* `2`*'s do que* `0`*'s*. O critério de aceitação é por **pilha vazia**. Descreva sucintamente a ideia do algoritmo e ilustre o reconhecimento das palavras `2112` e `121201` pelo autómato.

*Resposta:*

A linguagem é $\{x \mid x \in \Sigma^\star$ tem mais dois `1`'s do que `0`'s$\} \cup \{x \mid x \in \Sigma^\star$ tem mais dois `2`'s do que `0`'s$\}$, com $\Sigma = \{\mathtt{0},\mathtt{1},\mathtt{2}\}$.

Assim, o autómato de pilha que apresentamos a seguir, com estado inicial $s_0$ e símbolo inicial $Z$, é uma adaptação do autómato dado na aulas para reconhecimento das sequências de `a`'s e `b`'s que têm o mesmo número de `a`'s e `b`'s.

$$
\begin{array}{lcl}
\delta(s_0,\varepsilon,Z) & = & \{(s_1,Z),(q_1,Z)\} \\
\delta(s_1,\mathtt{2},Z) & = & \{(s_1,Z)\} \\
\delta(s_1,\mathtt{0},Z) & = & \{(s_1,AZ)\} \\
\delta(s_1,\mathtt{1},Z) & = & \{(s_1,BZ)\} \\
\delta(s_1,\mathtt{2},A) & = & \{(s_1,A)\} \\
\delta(s_1,\mathtt{2},B) & = & \{(s_1,B)\} \\
\delta(s_1,\mathtt{1},A) & = & \{(s_1,\varepsilon)\} \\
\delta(s_1,\mathtt{0},B) & = & \{(s_1,\varepsilon)\} \\
\delta(s_1,\mathtt{0},A) & = & \{(s_1,AA)\} \\
\delta(s_1,\mathtt{1},B) & = & \{(s_1,BB)\} \\
\delta(s_1,\varepsilon,B) & = & \{(s_2,\varepsilon)\} \\
\delta(s_2,\varepsilon,B) & = & \{(s_3,\varepsilon)\} \\
\delta(s_3,\varepsilon,Z) & = & \{(s_3,\varepsilon)\}
\end{array}
\qquad
\begin{array}{lcl}
\delta(q_1,\mathtt{1},Z) & = & \{(q_1,Z)\} \\
\delta(q_1,\mathtt{0},Z) & = & \{(q_1,AZ)\} \\
\delta(q_1,\mathtt{2},Z) & = & \{(q_1,BZ)\} \\
\delta(q_1,\mathtt{1},A) & = & \{(q_1,A)\} \\
\delta(q_1,\mathtt{1},B) & = & \{(q_1,B)\} \\
\delta(q_1,\mathtt{2},A) & = & \{(q_1,\varepsilon)\} \\
\delta(q_1,\mathtt{0},B) & = & \{(q_1,\varepsilon)\} \\
\delta(q_1,\mathtt{0},A) & = & \{(q_1,AA)\} \\
\delta(q_1,\mathtt{2},B) & = & \{(q_1,BB)\} \\
\delta(q_1,\varepsilon,B) & = & \{(s_2,\varepsilon)\}
\end{array}
$$

No estado $s_0$ passa ao estado $s_1$ ou a $q_1$ para tratar os dois subconjuntos de forma independente (excepto no fim).

Em $s_1$, coloca um $A$ na pilha por cada `0` e coloca um $B$ na pilha por cada `1`, a menos que tenha já $B$ ou $A$ como topo da pilha. Com $A$ no topo da pilha, retira esse $A$ se consome um `1` e com $B$ no topo da pilha, retira esse $B$ se consome `0`. Do mesmo modo, em $q_1$, coloca um $A$ na pilha por cada `0` e coloca um $B$ na pilha por cada `2`, a menos que tenha já $B$ ou $A$ como topo da pilha. Com $A$ no topo da pilha, retira esse $A$ se consome um `2` e com $B$ no topo da pilha, retira esse $B$ se consome `0`. Em $s_1$, pode consumir `2`'s livremente, do mesmo modo que em $q_1$ permite o consumo de `1`'s livremente.

Em $s_1$, a existência de $B$'s na pilha indica que há um excesso de `1`'s relativamente aos `0`'s. Analogamente, em $q_1$, a existência de $B$'s na pilha indica que há um excesso de `2`'s relativamente aos `0`'s. Por essa razão, sempre que tem $B$ no topo da pilha, assume um comportamento não determinístico, podendo passar ao estado $s_2$, retirando tal $B$ e não consumindo qualquer símbolo da palavra, e depois a $s_3$ se ainda puder retirar um segundo $B$ e finalmente a pilha vazia (retirando $Z$). Se puder concluir esta sequência de transições com sucesso e nada restar da palavra dada, a palavra pertence à linguagem dada.

As seguintes sequências de mudanças de configuração (válidas para o AP descrito) levam ao reconhecimento das palavras `2112` e `121201`, respetivamente:

$(s_0,\mathtt{2112},Z) \vdash (s_1,\mathtt{2112},Z) \vdash (s_1,\mathtt{112},Z) \vdash (s_1,\mathtt{12},BZ) \vdash (s_1,\mathtt{2},BBZ) \vdash (s_1,\varepsilon,BBZ) \vdash (s_2,\varepsilon,BZ) \vdash (s_3,\varepsilon,Z) \vdash (s_3,\varepsilon,\varepsilon)$

$(s_0,\mathtt{121201},Z) \vdash (s_1,\mathtt{121201},Z) \vdash (s_1,\mathtt{21201},BZ) \vdash (s_1,\mathtt{1201},BZ) \vdash (s_1,\mathtt{201},BBZ) \vdash (s_1,\mathtt{01},BBZ) \vdash (s_1,\mathtt{1},BZ) \vdash (s_1,\varepsilon,BBZ) \vdash (s_2,\varepsilon,BZ) \vdash (s_3,\varepsilon,Z) \vdash (s_3,\varepsilon,\varepsilon)$

**7.** [!!] Recordando a demonstração do corolário do teorema de Myhill-Nerode, que define o AFD mínimo para uma dada linguagem regular $L$ de alfabeto $\Sigma$, explique o que garante que $|\Sigma^\star/R_L|$ é menor ou igual que o número de estados de qualquer AFD $A$ tal que $\mathcal{L}(A) = L$.

*Resposta:*

Qualquer que seja o AFD $A = (S, \Sigma, \delta, s_0, F)$ tal que $\mathcal{L}(A) = L$, a conclusão de que $|\Sigma^\star/R_L| \leq |S|$ resulta de três observações (resultados demonstrados):

- que, sem perda de generalidade, podiamos supor que todos os estados do AFD $A$ são acessíveis do seu estado inicial pois, caso contrário, os estados não acessíveis podiam ser trivialmente descartados e o AFD resultante seria equivalente a $A$ e teria menos estados;
- que, nessas condições, existe uma bijeção entre $S$ e o conjunto das classes de equivalência da relação $R_A$ definida em $\Sigma^\star$ por $(x, y) \in R_A$ se e só se $\hat{\delta}(s_0, x) = \hat{\delta}(s_0, y)$;
- que se $(x, y) \in R_A$ então $(x, y) \in R_L$, o que implica que $\mathcal{C}_x \subseteq [x]$, para todo $x \in \Sigma^\star$, sendo $\mathcal{C}_x$ a classe equivalência de $x$ segundo $R_A$ e $[x]$ a classe equivalência $x$ segundo $R_L$; assim se concluiu que cada classe de $R_L$ é ou uma classe de $R_A$ ou resulta da união de classes de $R_A$ e, consequentemente, o número de classes de equivalência de $R_L$ é menor ou igual que o número de classes de equivalência de $R_A$, o que significa que $|\Sigma^\star/R_L| \leq |\Sigma^\star/R_A| = |S|$.

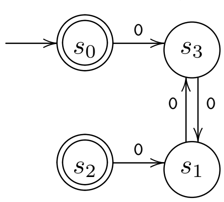
**8.** [!!] A tabela indicada abaixo, à direita, foi construída na fase inicial da aplicação do algoritmo de Moore a um dado AFD $A = (S, \{\mathtt{0}, \mathtt{1}\}, \delta, s_0, F)$, com $\delta$ função (total) de $S \times \{\mathtt{0}, \mathtt{1}\}$ em $S$. Justificando a resposta, analise as possibilidades de o AFD $A$ coincidir com o AFD mínimo equivalente a $A$, sabendo que:

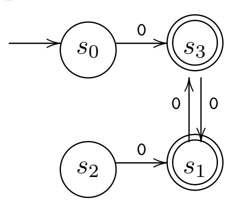
- $\delta(s_0, \mathtt{0}) = s_3 = \delta(s_1, \mathtt{0})$ e $\delta(s_2, \mathtt{0}) = s_1 = \delta(s_3, \mathtt{0})$,
- se $x \in \mathcal{L}((\mathtt{0} + \mathtt{1})^\star \mathtt{1})$ então $x \in \mathcal{L}(A)$,
- se $x \notin \mathcal{L}((\mathtt{0} + \mathtt{1})^\star \mathtt{1})$, nada nos foi dito sobre se $x \in \mathcal{L}(A)$ ou não,
- todos os estados em $S = \{s_0, s_1, s_2, s_3\}$ são acessíveis de $s_0$.

| | | | | |
|---|---|---|---|---|
| $s_0$ | $\equiv$ | | | |
| $s_1$ | $X$ | $\equiv$ | | |
| $s_2$ | | $X$ | $\equiv$ | |
| $s_3$ | $X$ | | $X$ | $\equiv$ |
| | $s_0$ | $s_1$ | $s_2$ | $s_3$ |

*Resposta:*

As entradas não preenchidas na tabela correspondem aos pares $(s_k, s_j)$, com $s_k \neq s_j$, e $s_k, s_j \in F$ ou $s_k, s_j \notin F$. Assim, de acordo com a informação dada sobre $\delta$, há que analisar duas situações possíveis:

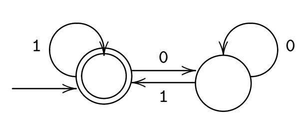



É dito que todos os estados de $A$ são acessíveis de $s_0$. Assim, podemos concluir que haverá uma transição por $\mathtt{1}$ de $s_0$ ou de $s_3$ ou de $s_1$ para $s_2$, pois só assim $s_2$ será acessível de $s_0$, dado que, sendo $A$ um AFD, não podem existir outras transições por $\mathtt{0}$, além das indicadas.

Daí resulta que $s_2$ terá que ser um estado final porque se sabe que qualquer palavra que termina em $\mathtt{1}$ tem de ser reconhecida por $A$. Portanto, podemos descartar a análise da segunda situação, pois os estados finais só podem ser $s_0$ e $s_2$.

Como todas as transições por $\mathtt{0}$ terminam em $s_1$ ou $s_3$, nenhuma palavra que termine em $\mathtt{0}$ é aceite pelo AFD $A$. Por outro lado, como nenhuma palavra que termina em $\mathtt{1}$ é rejeitada e $s_2$ e $s_0$ são estados finais, segue que $\mathcal{L}(A) = \mathcal{L}(\varepsilon + (\mathtt{0} + \mathtt{1})^\star \mathtt{1})$. Assim, o AFD $A$ nunca será igual ao AFD mínimo equivalente a $A$ pois, em todos os casos, $s_0 \equiv s_2$ e $s_1 \equiv s_3$, uma vez que o AFD mínimo para $\mathcal{L}(\varepsilon + (\mathtt{0} + \mathtt{1})^\star \mathtt{1})$ é:




**(Fim)**