Gramáticas Independentes de Contexto e Autómatos de Pilha

Maria de Lurdes Azevedo Teixeira

Apontamentos para a disciplina de Autómatos e Linguagens Formais
do curso de Matemática e Ciências de Computação

Departamento de Matemática
Escola de Ciências da Universidade do Minho

2020

Estes apontamentos foram elaborados com o objetivo de servir de texto de apoio à segunda parte de uma disciplina da área de Teoria de Linguagens, atualmente com a designação de Autómatos e Linguagens Formais, do curso de Licenciatura em Ciências de Computação. A primeira parte do curso foi dedicada ao estudo dos seguintes temas: Indução Estrutural, Linguagens Regulares e Autómatos Finitos.

Índice

Capítulo 1

Gramáticas Independentes de Contexto

Considere-se o alfabeto $A = \{a, b\}$ e a linguagem L definida indutivamente por:

1. $\varepsilon \in L$
2. $a, b \in L$
3. se $x \in L$, então $axa, bxb \in L$

A definição de L é uma definição indutiva que permite facilmente identificar os seus elementos: são as palavras u sobre $\{a, b\}$ que verificam $u = u^I$. Usando indução estrutural pode demonstrar-se esta caracterização dos elementos de L.

Contudo L não é uma linguagem regular. Sejam $n \in \mathbb{N}$ e $u = a^n b a^n \in L$.

$$u = \begin{array}{|c|c|c|}\hline \qquad a^n \qquad & b & \qquad a^n \qquad \\ \hline\end{array}$$

$$u = \begin{array}{|c|c|c|}\hline \quad x \quad & \; y \; & \qquad\qquad z \qquad\qquad \\ \hline\end{array}$$

Supondo $x, y, z \in A^*$ tais que: $y \neq \varepsilon$, $|xy| \leq n$ e $u = xyz$, então, para qualquer $k \geq 2$, $xy^k z \notin L$, pois $y = a^l$, para $1 \leq l \leq n$, e, consequentemente, $xy^k z = a^{n+(k-1)l} b a^n$. Com base no lema da iteração, prova-se assim que L não é regular. Consequentemente, L não corresponde a uma expressão regular nem é reconhecida por um autómato finito.

Na primeira parte deste curso, foram estudadas várias formas, claras e elegantes, de caraterizar o conjunto das linguagens regulares. No entanto, um autómato finito é um modelo de computação muito restritivo que só reconhece linguagens com estrutura sintática muito rígida e elementar. Iremos de seguida estudar classes de linguagens mais ricas sintaticamente, definidas a partir de ferramentas que geram os seus elementos, e que são igualmente reconhecidas por modelos computacionais abstratos. A classe das linguagens regulares é apenas um caso particular destas classes.

1.1 Gramáticas

Entende-se por gramática uma ferramenta matemática que permite descrever uma linguagem, identificando um mecanismo gerador dos elementos da linguagem. A definição de gramática é próxima da definição indutiva de um conjunto.

Definição 1.1 *Uma gramática $\mathcal{G}$ é um quádruplo*

$$\mathcal{G} = (V, A, \mathcal{S}, P)$$

em que:

1. *V é um alfabeto, dito não terminal, constituído por elementos designados variáveis;*
2. *A é um alfabeto, dito terminal, constituído por elementos designados letras ou símbolos terminais e tal que $V \cap A = \emptyset$;*
3. *$\mathcal{S}$ é um elemento de V designado símbolo inicial;*
4. *P é um conjunto de produções, ou regras gramaticais, que é um subconjunto de $\big((V \cup A)^* V (V \cup A)^*\big) \times (V \cup A)^*$.*

De acordo com a notação usual, cada elemento $(\alpha, \beta) \in P$ representa-se por $\alpha \longrightarrow \beta$.

Exemplo 1.2

Seja $A = \{a, b\}$ e recorde-se a linguagem L definida no início deste capítulo indutivamente por:

1. $\varepsilon \in L$
2. $a, b \in L$
3. *se $x \in L$, então $axa, bxb \in L$*

Uma gramática que permita gerar L poderia ser descrita da seguinte forma:

$$\mathcal{G}_L = (V, A, \mathcal{S}, P)$$

em que:

- *$V = \{\mathcal{S}\}$;*
- *$A = \{a, b\}$;*
- *P é o conjunto de produções que, de acordo com a notação usual, se representam da seguinte forma:*

$$\begin{array}{lcl} \mathcal{S} & \longrightarrow & \varepsilon \\ \mathcal{S} & \longrightarrow & a \\ \mathcal{S} & \longrightarrow & b \\ \mathcal{S} & \longrightarrow & a\mathcal{S}a \\ \mathcal{S} & \longrightarrow & b\mathcal{S}b \end{array}$$

A partir da variável $\mathcal{S}$ e da utilização sequencial de produções pode-se obter, por exemplo, a palavra $ababa \in L$:

1. *utilizando*	$\mathcal{S} \longrightarrow a\mathcal{S}a$	*obtemos $a\mathcal{S}a$*
2. *utilizando*	$\mathcal{S} \longrightarrow b\mathcal{S}b$	*obtemos $ab\mathcal{S}ba$*
3. *utilizando*	$\mathcal{S} \longrightarrow a$	*obtemos $ababa$*

Isto traduz a afirmação de que a gramática $\mathcal{G}_L$ gera a palavra $ababa$.

Genericamente, aplicando as produções podem-se transformar expressões de $(V \cup A)^*$. Por exemplo, a expressão $a\mathcal{S}abb$ pode ser transformada pela produção $\mathcal{S} \longrightarrow \varepsilon$ na palavra $aabb$, e pode ser transformada pela produção $\mathcal{S} \longrightarrow b\mathcal{S}b$ na expressão $ab\mathcal{S}babb$.

Nesta gramática, os membros esquerdos das produções são todos iguais pelo que é possível simplificar a escrita das produções, utilizando o símbolo | que significa 'ou'. As produções de P poderiam então ser descritas de forma equivalente por:

$$\mathcal{S} \longrightarrow \varepsilon \mid a \mid b \mid a\mathcal{S}a \mid b\mathcal{S}b$$

A utilização do símbolo | significando 'ou' sempre que existirem duas ou mais produções com o lado esquerdo comum, como se viu no exemplo anterior, será generalizada.

Considere-se uma gramática $\mathcal{G} = (V, A, \mathcal{S}, P)$. Dados $\sigma_1 \in (V \cup A)^* V (V \cup A)^*$ e $\sigma_2 \in (V \cup A)^*$ diz-se que σ_2 *deriva diretamente de* σ_1 se, sendo $\sigma_1 = \gamma\alpha\gamma'$, onde γ, $\gamma' \in (V \cup A)^*$, então existe $\alpha \longrightarrow \beta \in P$ e $\sigma_2 = \gamma\beta\gamma'$. Em tal caso escreve-se

$$\sigma_1 \underset{\mathcal{G}}{\Rightarrow} \sigma_2$$

Sendo $k \in \mathbb{N}$, se $\sigma_1, \ldots, \in \sigma_k (V \cup A)^* V (V \cup A)^*$ e $\sigma_{k+1} \in (V \cup A)^*$ forem tais que, para todo o $i \in \{1, \ldots, k\}$,

$$\sigma_i \underset{\mathcal{G}}{\Rightarrow} \sigma_{i+1}$$

então diz-se que σ_{k+1} *deriva em k passos de* σ_1 e escreve-se

$$\sigma_1 \overset{k}{\underset{\mathcal{G}}{\Rightarrow}} \sigma_{k+1}$$

Genericamente, dados $\sigma \in (V \cup A)^* V (V \cup A)^*$ e $\sigma' \in (V \cup A)^*$, diz-se que σ' *deriva de* σ se $\sigma = \sigma'$ ou existir $k \in \mathbb{N}$ tal que σ' deriva em k passos de σ. Em tal caso escreve-se que

$$\sigma \overset{*}{\underset{\mathcal{G}}{\Rightarrow}} \sigma'$$

À sequência de passos elementares que permite concluir que $\sigma \overset{*}{\underset{\mathcal{G}}{\Rightarrow}} \sigma'$ chama-se *derivação* de σ' a partir de σ.

Recorde-se que o objetivo de introduzir o conceito de gramática era definir linguagens sobre um dado alfabeto. Neste contexto segue-se a definição de linguagem gerada por uma gramática.

Definição 1.3 *Considere-se a gramática $\mathcal{G} = (V, A, \mathcal{S}, P)$. A linguagem gerada por $\mathcal{G}$ é*

$$L(\mathcal{G}) = \{u \in A^* \mid \mathcal{S} \overset{*}{\underset{\mathcal{G}}{\Rightarrow}} u\}$$

Exemplo 1.4 *Considere-se a gramática $\mathcal{G}_L$ definida no exemplo 1.2.*

Podem traduzir-se os resultados obtidos dizendo que:

1. $\mathcal{S} \overset{*}{\underset{\mathcal{G}_L}{\Rightarrow}} ababa$ *ou, mais precisamente,* $\mathcal{S} \overset{3}{\underset{\mathcal{G}_L}{\Rightarrow}} ababa$*, o que constitui a prova de que* $ababa \in L(\mathcal{G}_L)$*;*

2. $a\mathcal{S}abb \underset{\mathcal{G}_L}{\Rightarrow} aabb$*;*

3. $a\mathcal{S}abb \underset{\mathcal{G}_L}{\Rightarrow} ab\mathcal{S}babb$*.* ■

Mais geralmente, para $\alpha \in (V \cup A)^*$, definem-se os seguintes conjuntos:

$$D(\alpha) = \{\beta \in (V \cup A)^* \mid \alpha \underset{\mathcal{G}}{\overset{*}{\Rightarrow}} \beta\}$$

$$L(\alpha) = \{u \in A^* \mid \alpha \underset{\mathcal{G}}{\overset{*}{\Rightarrow}} u\}$$

Assim, $L(\alpha) = D(\alpha) \cap A^*$ e $L(\mathcal{G}) = L(\mathcal{S})$.

Definição 1.5 *Duas gramáticas $\mathcal{G}_1$ e $\mathcal{G}_2$ dizem-se equivalentes se*

$$L(\mathcal{G}_1) = L(\mathcal{G}_2)$$

Exemplos 1.6

1. *Considere-se a gramática*

$$\mathcal{G} = (V, A, \mathcal{S}, P)$$

em que $V = \{\mathcal{S}, \mathcal{B}, \mathcal{C}\}$, $A = \{a, b, c\}$ e P é o seguinte conjunto de produções :

$$\begin{array}{ll} \mathcal{S} & \longrightarrow \mathcal{BC} \\ \mathcal{B} & \longrightarrow a\mathcal{B}b \mid \varepsilon \\ \mathcal{C} & \longrightarrow b\mathcal{C}c \mid \varepsilon \end{array}$$

Será que $a^2b^3c \in L(\mathcal{G})$? A resposta é sim porque

$$\mathcal{S} \underset{\mathcal{G}}{\Rightarrow} \mathcal{BC} \underset{\mathcal{G}}{\Rightarrow} a\mathcal{B}b\mathcal{C} \underset{\mathcal{G}}{\Rightarrow} aa\mathcal{B}bb\mathcal{C} \underset{\mathcal{G}}{\Rightarrow} aabb\mathcal{C} \underset{\mathcal{G}}{\Rightarrow} aabbb\mathcal{C}c \underset{\mathcal{G}}{\Rightarrow} aabbbc$$

E a palavra ab^3c será que é elemento de $L(\mathcal{G})$? Não. Note-se que a primeira produção só pode ser aplicada uma vez no primeiro passo da derivação. Para obter o prefixo a é necessário utilizar uma única vez a segunda produção e para obter o sufixo c é necessário utilizar uma única vez a quarta produção. Deste modo obtém-se igualmente duas ocorrências da letra b. Para obter uma terceira ocorrência de b seria necessário utilizar mais uma vez a segunda ou a quarta produção, o que implicaria acrescentar uma segunda ocorrência de a ou de c. Constata-se pois que é impossível derivar ab^3c.

De seguida determine-se a linguagem gerada por $\mathcal{G}$. Dado que os membros esquerdos das derivações são todos variáveis,

$$\begin{aligned} L(\mathcal{G}) &= L(\mathcal{BC}) \\ &= L(\mathcal{B}) \cdot L(\mathcal{C}) \\ &= \{a^n b^n \mid n \in \mathbb{N}_0\}\{b^m c^m \mid m \in \mathbb{N}_0\} \\ &= \{a^n b^{n+m} c^m \mid n, m \in \mathbb{N}_0\} \end{aligned}$$

uma vez que

$$\begin{aligned} L(\mathcal{B}) &= aL(\mathcal{B})b \cup \{\varepsilon\} \\ &= a\Big(aL(\mathcal{B})b \cup \{\varepsilon\}\Big)b \cup \{\varepsilon\} \\ &= a^2L(\mathcal{B})b^2 \cup \{\ ab,\ \varepsilon\} \\ &\ \vdots \\ &= a^{k+1}L(\mathcal{B})b^{k+1} \cup \{a^k b^k, \ldots,\ ab,\ \varepsilon\}, \quad \textit{para qualquer } k \in \mathbb{N}. \end{aligned}$$

Então

$$L(\mathcal{B}) = \{a^n b^n \mid n \in \mathbb{N}_0\}.$$

De modo análogo, se conclui que $L(\mathcal{C}) = \{b^m c^m \mid m \in \mathbb{N}_0\}$.

2. *Considere-se a gramática*

$$\mathcal{G}_r = (V, A, \mathcal{S}, P)$$

em que $V = \{\mathcal{S}, \mathcal{B}, \mathcal{C}\}$, $A = \{a, b, c\}$ *e* P *é o seguinte conjunto de produções:*

$$\begin{array}{ll} \mathcal{S} & \longrightarrow a\mathcal{S}\mathcal{B}\mathcal{C} \mid a\mathcal{B}\mathcal{C} \\ \mathcal{C}\mathcal{B} & \longrightarrow \mathcal{B}\mathcal{C} \\ a\mathcal{B} & \longrightarrow ab \\ b\mathcal{B} & \longrightarrow bb \\ b\mathcal{C} & \longrightarrow bc \\ c\mathcal{C} & \longrightarrow cc \end{array}$$

Com esta gramática pode-se gerar a palavra $a^2b^2c^2$*:*

$$\mathcal{S} \underset{\mathcal{G}_r}{\Rightarrow} a\mathcal{S}\mathcal{B}\mathcal{C} \underset{\mathcal{G}_r}{\Rightarrow} aa\mathcal{B}\mathcal{C}\mathcal{B}\mathcal{C} \underset{\mathcal{G}_r}{\Rightarrow} aa\mathcal{B}\mathcal{B}\mathcal{C}\mathcal{C} \underset{\mathcal{G}_r}{\Rightarrow} aab\mathcal{B}\mathcal{C}\mathcal{C} \underset{\mathcal{G}_r}{\Rightarrow} aabb\mathcal{C}\mathcal{C}$$

$$\underset{\mathcal{G}_r}{\Rightarrow} aabbc\mathcal{C} \underset{\mathcal{G}_r}{\Rightarrow} a^2b^2c^2$$

Para qualquer $n \in \mathbb{N}$, *procedendo de modo análogo viria*

$$\mathcal{S} \overset{n-1}{\underset{\mathcal{G}_r}{\Rightarrow}} a^{n-1}\mathcal{S}(\mathcal{B}\mathcal{C})^{n-1} \underset{\mathcal{G}_r}{\Rightarrow} a^n(\mathcal{B}\mathcal{C})^n \overset{*}{\underset{\mathcal{G}_r}{\Rightarrow}} a^n\mathcal{B}^n\mathcal{C}^n \underset{\mathcal{G}_r}{\Rightarrow} a^nb\mathcal{B}^{n-1}\mathcal{C}^n \overset{n-1}{\underset{\mathcal{G}_r}{\Rightarrow}} a^nb^n\mathcal{C}^n$$
$$\underset{\mathcal{G}_r}{\Rightarrow} a^nb^nc\mathcal{C}^{n-1} \overset{n-1}{\underset{\mathcal{G}_r}{\Rightarrow}} a^nb^nc^n$$

o que mostra que $\{a^nb^nc^n \mid n \in \mathbb{N}\} \subseteq L(\mathcal{G}_r)$. *Será que é válida a igualdade? Note-se que numa palavra gerada por* $\mathcal{G}_r$ *o número de ocorrências da letra* a *é igual ao número de aplicações da primeira produção mais uma, uma vez que aplicada a segunda produção não é possível voltar aplicar a primeira nem a segunda produções. De igual forma se conclui que as aplicações da primeira produção ocorrem no início da derivação da palavra, logo seguidas de uma única aplicação da segunda produção, ficando assim fixo o número de ocorrências da letra* a *e o comprimento das expressões que estão a ser derivadas. Mais ainda, por aplicação da primeira e da segunda produções são introduzidos igual número de símbolos* a, $\mathcal{B}$ *e* $\mathcal{C}$. *Temos também que a letra* a *nunca ocorre depois de um* $\mathcal{B}$ *ou* $\mathcal{C}$ *e, consequentemente nunca depois de um* b *ou de um* c. *Constata-se também que a terceira produção só permite trocar a ordem das variáveis* $\mathcal{B}$ *e* $\mathcal{C}$ *de modo a colocar* $\mathcal{B}$ *antes de* $\mathcal{C}$.

No exemplo seguinte de derivação, verifica-se que a sequência de produções aplicada conduz a uma palavra de $(V \cup A)^*$, *a partir da qual não se pode continuar a derivação e que não é um elemento de* A^*.

$$\mathcal{S} \underset{\mathcal{G}_r}{\Rightarrow} a\mathcal{S}\mathcal{B}\mathcal{C} \underset{\mathcal{G}_r}{\Rightarrow} a^2(\mathcal{B}\mathcal{C})^2 \underset{\mathcal{G}_r}{\Rightarrow} a^2b\mathcal{C}\mathcal{B}\mathcal{C} \underset{\mathcal{G}_r}{\Rightarrow} a^2bc\mathcal{B}\mathcal{C}$$

Este exemplo torna claro que para uma derivação conduzir a uma palavra de A^* *é necessário 'colocar' todas as ocorrências de* $\mathcal{B}$ *antes de qualquer ocorrência de* $\mathcal{C}$. *No final desta ordenação, aplicações sucessivas das produções quatro a sete permitem substituir cada ocorrência do símbolo* $\mathcal{B}$ *por* b *e cada ocorrência do símbolo* $\mathcal{C}$ *por* c, *até obter uma palavra de* A^*.

Pode pois concluir-se que $L(\mathcal{G}_r) = \{a^nb^nc^n \mid n \in \mathbb{N}\}$.

3. *Considere-se a gramática*

$$\mathcal{G}_d = (V, A, \mathcal{S}, P)$$

em que $V = \{\mathcal{S}, \mathcal{B}\}$, $A = \{a,\, b\}$ *e* P *é o seguinte conjunto de produções:*

$$\begin{array}{ll} \mathcal{S} & \longrightarrow a\mathcal{S} \mid ba\mathcal{B} \\ \mathcal{B} & \longrightarrow ab\mathcal{B} \mid \varepsilon \end{array}$$

Uma palavra gerada por $\mathcal{G}_d$ *é* $aba(ab)^2$, *pois*

$$\mathcal{S} \underset{\mathcal{G}_d}{\Rightarrow} a\mathcal{S} \underset{\mathcal{G}_d}{\Rightarrow} aba\mathcal{B} \underset{\mathcal{G}_d}{\Rightarrow} abaab\mathcal{B} \underset{\mathcal{G}_d}{\Rightarrow} aba(ab)^2\mathcal{B} \underset{\mathcal{G}_d}{\Rightarrow} aba(ab)^2$$

Como são as palavras de $L(\mathcal{G}_d)$*?*

$$\begin{array}{ll} L(\mathcal{S}) & = aL(\mathcal{S}) \cup baL(\mathcal{B}) \\ & = a\big(aL(\mathcal{S}) \cup baL(\mathcal{B})\big) \cup baL(\mathcal{B}) \\ & = a^2L(\mathcal{S}) \cup abaL(\mathcal{B}) \cup baL(\mathcal{B}) \\ & = a^2L(\mathcal{S}) \cup \{aba,\, ba\}L(\mathcal{B}) \\ & \vdots \\ & = a^kL(\mathcal{S}) \cup \{a^{k-1}ba,\, \ldots, aba,\, ba\}L(\mathcal{B}),\ \textit{para qualquer } k \in \mathbb{N}. \end{array}$$

Então,

$$L(\mathcal{S}) = a^*baL(\mathcal{B}) = a^*ba(ab)^*$$

dado que, $L(\mathcal{B}) = abL(\mathcal{B}) \cup \{\varepsilon\} = \cdots = (ab)^nL(\mathcal{B}) \cup \{(ab)^{n-1}, \ldots, ab, \varepsilon\}$, *para qualquer* $n \in \mathbb{N}$, *ou seja,* $L(\mathcal{B}) = (ab)^*$.

4. *Considere-se agora a gramática*

$$\mathcal{G}_e = (V, A, \mathcal{S}, P)$$

em que $V = \{\mathcal{S}, \mathcal{C}\}$, $A = \{a,\, b\}$ *e* P *é o seguinte conjunto de produções:*

$$\begin{array}{ll} \mathcal{S} & \longrightarrow \mathcal{S}ab \mid \mathcal{C}ba \\ \mathcal{C} & \longrightarrow \mathcal{C}a \mid \varepsilon \end{array}$$

A palavra $aba(ab)^2$ *também é gerada por* $\mathcal{G}_e$, *pois*

$$\mathcal{S} \underset{\mathcal{G}_e}{\Rightarrow} \mathcal{S}ab \underset{\mathcal{G}_e}{\Rightarrow} \mathcal{S}abab \underset{\mathcal{G}_e}{\Rightarrow} \mathcal{C}ba(ab)^2 \underset{\mathcal{G}_e}{\Rightarrow} \mathcal{C}aba(ab)^2 \underset{\mathcal{G}_e}{\Rightarrow} aba(ab)^2$$

Como são as palavras de $L(\mathcal{G}_e)$*?*

$$\begin{array}{ll} L(\mathcal{S}) & = L(\mathcal{S})ab \cup L(\mathcal{C})ba \\ & = \big(L(\mathcal{S})ab \cup L(\mathcal{C})ba\big)ab \cup L(\mathcal{C})ba \\ & = L(\mathcal{S})(ab)^2 \cup L(\mathcal{C})\{baab,\, ba\} \\ & \vdots \\ & = L(\mathcal{S})(ab)^k \cup L(\mathcal{C})\{ba(ab)^{k-1},\, \ldots,\, baab,\, ba\},\ \textit{para qualquer } k \in \mathbb{N} \end{array}$$

Então,

$$L(\mathcal{S}) = L(\mathcal{C})ba(ab)^* = a^*ba(ab)^*$$

dado que $L(\mathcal{C}) = L(\mathcal{C})a \cup \{\varepsilon\} = \cdots = L(\mathcal{C})a^n \cup \{a^{n-1}, \ldots, a, \varepsilon\}$, *para qualquer* $n \in \mathbb{N}$, *ou seja,* $L(\mathcal{C}) = a^*$.

Conclui-se assim que $\mathcal{G}_e$ *é equivalente a* $\mathcal{G}_d$.

1.2 Classificação de gramáticas

Nesta secção faz-se uma rápida apresentação da classificação de gramáticas, de acordo com a forma das suas produções, apresentada em 1959 pelo linguista Avram Noam Chomsky e que se traduz numa sequência de tipos de gramáticas, vulgarmente conhecida por hierarquia de Chomsky.

Genericamente uma produção de uma gramática $\mathcal{G} = (V, A, \mathcal{S}, P)$ é uma expressão do tipo $\alpha \longrightarrow \beta$ onde $\alpha \in (V \cup A)^* \times V \times (V \cup A)^*$ e $\beta \in (V \cup A)^*$. Nos exemplos anteriores encontram-se produções de diversas formas: em 1.6(1) o membro esquerdo de cada produção é um elemento de V, por exemplo, $\mathcal{B} \longrightarrow a\mathcal{B}b$; em 1.6(2) temos a produção $a\mathcal{B} \longrightarrow ab$ em que o membro esquerdo é um elemento de AV e a produção $\mathcal{CB} \longrightarrow \mathcal{BC}$ em que o membro esquerdo é um elemento de V^2 . No primeiro caso, sempre que ocorra a variável $\mathcal{B}$ esta pode ser substituída pela expressão do lado direito da produção, $a\mathcal{B}b$. No segundo caso a produção $a\mathcal{B} \longrightarrow ab$ só será aplicada se a variável $\mathcal{B}$ ocorrer a seguir ao símbolo a e então $\mathcal{B}$ é substituído por b. Este tipo de análise conduz à identificação de diversos tipos de gramáticas.

Definição 1.7 *Uma gramática $\mathcal{G} = (V, A, \mathcal{S}, P)$ diz-se dependente de contexto se cada produção é de uma das formas:*

- $\alpha\mathcal{B}\gamma \longrightarrow \alpha\beta\gamma$*, onde* $\alpha, \gamma \in (V \cup A)^*$, $\beta \in (V \cup A)^+$ *e* $\mathcal{B} \in V$.
- $\mathcal{S} \longrightarrow \varepsilon$ *e, neste caso,* $\mathcal{S}$ *não aparecer no membro direito de outra produção.*

Uma linguagem diz-se *dependente de contexto* se pode ser gerada por uma gramática dependente de contexto.

Definição 1.8 *Uma gramática $\mathcal{G} = (V, A, \mathcal{S}, P)$ diz-se independente de contexto (ou livre de contexto) se cada produção é da forma:*

$$\mathcal{B} \longrightarrow \beta$$

onde $\mathcal{B} \in V$ *e* $\beta \in (V \cup A)^*$.

Em tal caso, abreviadamente, escreve-se que $\mathcal{G}$ é uma GIC.

Uma linguagem diz-se *independente de contexto* se existe uma GIC que a gere.

As gramáticas consideradas nos exemplos 1.6(1), 1.6(3) e 1.6(4) são gramáticas independentes de contexto, mas 1.6(2) não é.

Será que uma linguagem independente de contexto é também uma linguagem dependente de contexto?

Exemplo 1.9 *Considere-se a gramática $\mathcal{G} = (V, A, \mathcal{S}, P)$ definida no exemplo 1.6(1). Verifica-se que $\mathcal{G}$ não satisfaz as condições da definição 1.7. No entanto $\mathcal{G}$ é equivalente à gramática*

$$\mathcal{G}' = (V, A, \mathcal{S}, P')$$

em que P' é o seguinte conjunto de produções :

$$\begin{aligned} \mathcal{S} &\longrightarrow \mathcal{BC} \mid \mathcal{B} \mid \mathcal{C} \mid \varepsilon \\ \mathcal{B} &\longrightarrow a\mathcal{B}b \mid ab \\ \mathcal{C} &\longrightarrow b\mathcal{C}c \mid bc \end{aligned}$$

A gramática $\mathcal{G}'$ é dependente de contexto e independente de contexto.

Algo de semelhante se pode fazer com as gramáticas $\mathcal{G}_d$ e $\mathcal{G}_e$ definidas em 1.6(3) e 1.6(4), respetivamente. Sugere-se que o leitor faça esse exercício.

Com transformações do tipo das aqui implementadas é sempre possível construir uma gramática dependente de contexto equivalente a uma gramática independente de contexto, pelo que, qualquer linguagem independente de contexto é uma linguagem dependente de contexto.

Definição 1.10 *Uma gramática* $\mathcal{G} = (V, A, \mathcal{S}, P)$ *diz-se:*

- *linear à direita se cada produção é da forma* $\mathcal{B} \longrightarrow u$ *ou* $\mathcal{B} \longrightarrow u\mathcal{C}$*, onde* $\mathcal{B}, \mathcal{C} \in V$ *e* $u \in A^*$*;*

- *linear à esquerda se cada produção é da forma* $\mathcal{B} \longrightarrow u$ *ou* $\mathcal{B} \longrightarrow \mathcal{C}u$*, onde* $\mathcal{B}, \mathcal{C} \in V$ *e* $u \in A^*$*.*

Uma gramática diz-se regular se é linear à direita ou linear à esquerda.

Na secção 1.4 prova-se que a classe das linguagens geradas por gramáticas regulares é a classe das linguagens regulares.

As gramáticas $\mathcal{G}_d$ e $\mathcal{G}_e$ definidas em 1.6(3) e 1.6(4), respetivamente, são regulares, pois $\mathcal{G}_d$ é linear à direita e $\mathcal{G}_e$ é linear à esquerda. Obviamente qualquer gramática regular é uma gramática independente de contexto.

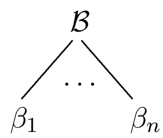
No caso de não ser colocada nenhuma restrição à forma das produções de uma gramática, os elementos da linguagem gerada por essa gramática obtém-se por aplicação sucessiva das produções da gramática. Uma linguagem gerada por uma gramática é uma *linguagem recursivamente enumerável.*

Temos então definida uma hierarquia de classes de linguagens, cuja ordem coincide com a relação de inclusão estrita.

LINGUAGENS REGULARES
LINGUAGENS INDEPENDENTES DE CONTEXTO
LINGUAGENS DEPENDENTES DE CONTEXTO
LINGUAGENS RECURSIVAMENTE ENUMERÁVEIS

1.3 Derivações, árvores e ambiguidade em GIC

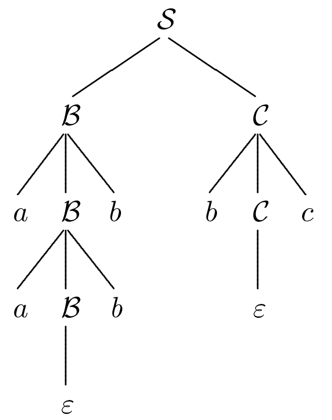
Uma derivação numa GIC pode ser representada graficamente de modo a evidenciar a estrutura da derivação. Numa tal representação, a uma derivação corresponde um grafo orientado em que os vértices são elementos de $V \cup A \cup \{\varepsilon\}$ e as arestas ligam variáveis que ocorrem no membro esquerdo de uma produção com cada um dos símbolos resultantes da aplicação dessa produção. Assim, se $\mathcal{B} \longrightarrow \beta_1 \cdots \beta_n$ é uma produção de uma gramática, uma sua aplicação deve ser representada por



Exemplos 1.11 *Considerem-se a gramática $\mathcal{G}$ definida em 1.6(1) e a palavra $a^2b^3c \in L(\mathcal{G})$. A representação gráfica correspondente à derivação*

$$\mathcal{S} \underset{\mathcal{G}}{\Rightarrow} \mathcal{BC} \underset{\mathcal{G}}{\Rightarrow} a\mathcal{B}b\mathcal{C} \underset{\mathcal{G}}{\Rightarrow} aa\mathcal{B}bb\mathcal{C} \underset{\mathcal{G}}{\Rightarrow} aabb\mathcal{C} \underset{\mathcal{G}}{\Rightarrow} aabbb\mathcal{C}c \underset{\mathcal{G}}{\Rightarrow} aabbbc$$

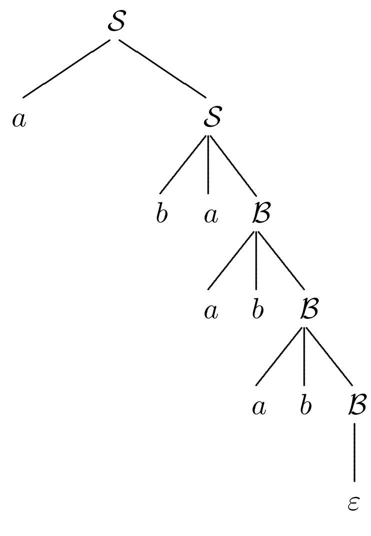
é



Considere-se agora a gramática $\mathcal{G}_d$ definida em 1.6(3) e a palavra $aba(ab)^2 \in L(\mathcal{G}_d)$. A representação gráfica correspondente à derivação

$$\mathcal{S} \underset{\mathcal{G}_d}{\Rightarrow} a\mathcal{S} \underset{\mathcal{G}_d}{\Rightarrow} aba\mathcal{B} \underset{\mathcal{G}_d}{\Rightarrow} abaab\mathcal{B} \underset{\mathcal{G}_d}{\Rightarrow} aba(ab)^2\mathcal{B} \underset{\mathcal{G}_d}{\Rightarrow} aba(ab)^2$$

é



Facilmente se conclui que em qualquer caso a representação gráfica da derivação de uma palavra a partir do símbolo inicial numa GIC é uma árvore, e que esta correspondência é uma função. No caso, em que uma palavra u é derivada a partir do símbolo inicial $\mathcal{S}$, a árvore correspondente terá como raiz um vértice etiquetado por $\mathcal{S}$ e as folhas serão letras de A, que concatenadas da esquerda para a direita dão a palavra u. Uma tal árvore será designada por *árvore de derivação de* u.

Exemplo 1.12 *Continuando a estudar a gramática $\mathcal{G}$ definida em 1.6(1), considere-se uma nova derivação da palavra a^2b^3c:*

$$\mathcal{S} \underset{\mathcal{G}}{\Rightarrow} \mathcal{BC} \underset{\mathcal{G}}{\Rightarrow} \mathcal{B}b\mathcal{C}c \underset{\mathcal{G}}{\Rightarrow} a\mathcal{B}bb\mathcal{C}c \underset{\mathcal{G}}{\Rightarrow} aa\mathcal{B}bbb\mathcal{C}c \underset{\mathcal{G}}{\Rightarrow} aa\mathcal{B}bbbc \underset{\mathcal{G}}{\Rightarrow} aabbbc$$

A representação gráfica desta nova derivação é a mesma árvore.

Do exemplo anterior, conclui-se que é possível a uma árvore corresponder mais do que uma derivação. Qual será, no caso geral, a diferença entre derivações a que corresponde uma mesma árvore? A diferença está na ordem da aplicação de algumas produções, mais claramente, na ordenação dos passos elementares das derivações representados em ramos independentes da árvore. Derivações a que corresponde a mesma árvore consideram-se *essencialmente iguais.*

Exemplo 1.13 *Considere-se a gramática*

$$\mathcal{G}_a = (V, A, \mathcal{S}, P)$$

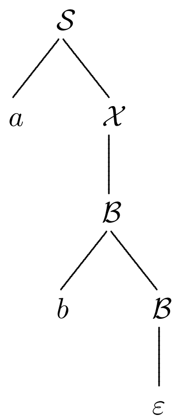
em que $V = \{\mathcal{S}, \mathcal{B}, \mathcal{C}, \mathcal{X}\}$, $A = \{a, b\}$ *e* P *é o seguinte conjunto de produções:*

$$\begin{array}{ll} \mathcal{S} & \longrightarrow a\mathcal{X} \\ \mathcal{X} & \longrightarrow \mathcal{B} \mid \mathcal{C} \\ \mathcal{B} & \longrightarrow b\mathcal{B} \mid \varepsilon \\ \mathcal{C} & \longrightarrow a\mathcal{C} \mid b \end{array}$$

A palavra ab *é gerada por* $\mathcal{G}_a$ *pois*

$$\mathcal{S} \underset{\mathcal{G}_a}{\Rightarrow} a\mathcal{X} \underset{\mathcal{G}_a}{\Rightarrow} a\mathcal{B} \underset{\mathcal{G}_a}{\Rightarrow} ab\mathcal{B} \underset{\mathcal{G}_a}{\Rightarrow} ab$$

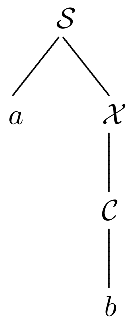
A árvore de derivação correspondente é



Alternativamente poder-se-ia ter efectuado a derivação

$$\mathcal{S} \underset{\mathcal{G}_a}{\Rightarrow} a\mathcal{X} \underset{\mathcal{G}_a}{\Rightarrow} a\mathcal{C} \underset{\mathcal{G}_a}{\Rightarrow} ab$$

a que corresponde a árvore



Definição 1.14 *Uma GIC $\mathcal{G} = (V, A, \mathcal{S}, P)$ diz-se ambígua se existir pelo menos uma palavra $u \in L(\mathcal{G})$ à qual se podem associar duas árvores de derivação distintas.*

Definição 1.15 *Uma linguagem L independente de contexto diz-se ambígua se qualquer gramática que gere L for uma GIC ambígua.*

Exemplo 1.16 *Recorde-se a gramática definida no exemplo 1.13. De acordo com a definição 1.14, $\mathcal{G}_a$ classifica-se como gramática ambígua. E $L(\mathcal{G}_a)$ será uma linguagem ambígua?*

$$L(\mathcal{G}_a) = L(\mathcal{S}) = aL(\mathcal{X}) = a\big(L(\mathcal{B}) \cup L(\mathcal{C})\big) = a(b^* \cup a^*b)$$

Considere-se a gramática $\mathcal{G}_{na} = (V, A, \mathcal{S}, P_{na})$ em que P_{na} é o seguinte conjunto de produções:

$$\begin{array}{ll} \mathcal{S} & \longrightarrow a\mathcal{X} \\ \mathcal{X} & \longrightarrow \mathcal{B} \mid a\mathcal{C}b \\ \mathcal{B} & \longrightarrow b\mathcal{B} \mid \varepsilon \\ \mathcal{C} & \longrightarrow a\mathcal{C} \mid \varepsilon \end{array}$$

$$L(\mathcal{G}_{na}) = L(\mathcal{S}) = aL(\mathcal{X}) = a\big(L(\mathcal{B}) \cup aL(\mathcal{C})b\big) = a(b^* \cup aa^*b)$$

Note-se que

$$a(b^* \cup a^*b) = ab^* \cup a^+b = ab^* \cup ab \cup aa^+b = ab^* \cup a^2a^*b = a(b^* \cup aa^*b)$$

pelo que $L(\mathcal{G}_a) = L(\mathcal{G}_{na})$. Uma vez que $\mathcal{G}_{na}$ não é uma gramática ambígua (verifique), $L(\mathcal{G}_a)$ não é uma linguagem ambígua .

Neste contexto, dada uma GIC ambígua, a questão fundamental é saber se existe outra GIC equivalente e não ambígua. Este tipo de problema é indecidível, pois não existe um algoritmo que permita associar a uma qualquer gramática ambígua uma não ambígua equivalente.

1.4 Gramáticas regulares e linguagens regulares

Na primeira parte deste curso, estudaram-se linguagens regulares e viu-se que, dado um alfabeto, cada linguagem regular corresponde a uma expressão regular. Na secção 1.2, definição 1.10, as gramáticas de um determinado tipo foram classificadas como gramáticas regulares. Não se trata de uma coincidência de nomes, mas sim do facto de existir uma correspondência entre gramáticas regulares e linguagens regulares, que se traduz nas seguintes afirmações:

1. se $\mathcal{G}$ é uma gramática regular, então $L(\mathcal{G})$ é uma linguagem regular;

2. se L é uma linguagem regular, então existe uma gramática regular $\mathcal{G}$, tal que $L = L(\mathcal{G})$.

Vai-se verificar a validade destas duas afirmações, começando por se constatar a equivalência entre gramáticas lineares à direita e gramáticas lineares à direita cujas produções tem formas mais simples.

Proposição 1.17 *Seja $\mathcal{G} = (V, A, \mathcal{S}, P)$ uma gramática linear à direita. Então, existe uma gramática $\mathcal{G}' = (V', A, \mathcal{S}, P')$ equivalente a $\mathcal{G}$, tal que as produções de P' são da forma*

$$\mathcal{B} \longrightarrow \varepsilon \quad ou \quad \mathcal{B} \longrightarrow a\mathcal{C}$$

onde $\mathcal{B}, \mathcal{C} \in V'$ e $a \in A$.

Demonstração:
Recorde-se que cada produção $\alpha \longrightarrow \beta$ de $\mathcal{G}$ é de uma das formas:

$$\mathcal{B} \longrightarrow u \quad \text{ou} \quad \mathcal{B} \longrightarrow u\mathcal{C},$$

onde $\mathcal{B}, \mathcal{C} \in V$ e $u \in A^*$.

Nesta demonstração iremos construir a gramática $\mathcal{G}'$ a partir da gramática $\mathcal{G}$ aplicando, sucessivamente, os passos que a seguir se descrevem.

Para cada produção de $\mathcal{G}$ do tipo:

1. $\mathcal{B} \longrightarrow a_1 \cdots a_n$, com $n \in \mathbb{N}$, acrescentam-se n novas variáveis $\mathcal{A}_1, \ldots, \mathcal{A}_n$ e substitui-se a produção em causa pelas seguintes $n+1$ produções:

$$\begin{array}{ll} \mathcal{B} & \longrightarrow a_1\mathcal{A}_1 \\ \mathcal{A}_1 & \longrightarrow a_2\mathcal{A}_2 \\ & \vdots \\ \mathcal{A}_{n-1} & \longrightarrow a_n\mathcal{A}_n \\ \mathcal{A}_n & \longrightarrow \varepsilon \end{array}$$

2. $\mathcal{B} \longrightarrow a_1 \cdots a_n\mathcal{C}$, com $n \in \mathbb{N}$, acrescentam-se $n-1$ novas variáveis $\mathcal{B}_1, \ldots, \mathcal{B}_{n-1}$ e substitui- se a produção em causa pelas seguintes n produções:

$$\begin{array}{ll} \mathcal{B} & \longrightarrow a_1\mathcal{B}_1 \\ \mathcal{B}_1 & \longrightarrow a_2\mathcal{B}_2 \\ & \vdots \\ \mathcal{B}_{n-2} & \longrightarrow a_{n-1}\mathcal{B}_{n-1} \\ \mathcal{B}_{n-1} & \longrightarrow a_n\mathcal{C} \end{array}$$

3. $\mathcal{B} \longrightarrow \mathcal{C}$ e a variável $\mathcal{C}$ não ocorre no membro esquerdo de nenhuma produção de $\mathcal{G}$, então elimina-se a produção (nenhuma palavra pode ser obtida como resultado de tal derivação pois $L(\mathcal{C}) = \emptyset$)

4. $\mathcal{B} \longrightarrow \mathcal{C}$ e, para $i = 1, \ldots, n$, $\mathcal{C} \longrightarrow \gamma_i$ são produções de $\mathcal{G}$, então substitui-se a produção $\mathcal{B} \longrightarrow \mathcal{C}$ pelas n produções:

$$\mathcal{B} \longrightarrow \gamma_1 | \cdots | \gamma_n$$

   e a cada uma das novas produções aplica-se o processo descrito nos pontos 1 a 4.

Assim, define-se $\mathcal{G}' = (V', A, \mathcal{S}, P')$ uma gramática tal que:

- V' é igual à união de V com o conjunto das novas variáveis introduzidas nas diversas aplicações dos passos 1 e 2;

- P' é o conjunto de produções que foram obtidas a partir de P pelas substituições descritas acima.

Facilmente se verifica que as produções de P' são todas do tipo $\mathcal{B} \longrightarrow \varepsilon$ ou $\mathcal{B} \longrightarrow a\mathcal{C}$, para $\mathcal{B}, \mathcal{C} \in V'$ e $a \in A$.

Atendendo a que as novas variáveis definidas em cada passo são todas distintas entre si e distintas das variáveis de V, conclui-se também que $\mathcal{G}$ é equivalente a $\mathcal{G}'$. Note-se que em cada passo uma produção é substituída por outras equivalentes, no sentido em que, antes e depois da substituição, a partir de uma qualquer expressão $\alpha \in (V \cup A)^*$, se podem derivar exatamente as mesmas expressões. ■

Proposição 1.18 *Sejam A um alfabeto e L uma linguagem regular sobre A. Então existe uma gramática linear à direita que gera L.*

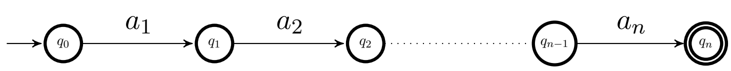
Demonstração:
Se $L \subseteq A^*$ é regular, então existe um autómato determinista $\mathcal{A} = (Q, A, \delta, q_0, F)$ que reconhece L. Considere-se a gramática $\mathcal{G}_{\mathcal{A}} = (V, A, \mathcal{S}, P)$ definida por:

- $V = Q$,
- $\mathcal{S} = q_0$,
- P é constituído por:

$$\begin{array}{ll} q \longrightarrow \varepsilon & \text{se } q \in F \\ p \longrightarrow aq & \text{se } p,\, q \in Q \;\text{ e }\; \delta(p, a) = q \end{array}$$

Então, $\mathcal{G}_{\mathcal{A}}$ é uma gramática linear à direita, em que as produções são todas do tipo descrito na proposição 1.17.

Seja $u \in L$. Então, $\delta^*(q_0, u) \in F$, ou seja, existe um caminho de etiqueta $u = a_1 \cdots a_n$ de vértice inicial q_0 e vértice final q_n pertencente a F.



Logo, existem as seguintes produções em $\mathcal{G}_{\mathcal{A}}$:

$$\begin{array}{l} q_0 \longrightarrow a_1 q_1 \\ q_1 \longrightarrow a_2 q_2 \\ \quad\vdots \\ q_{n-1} \longrightarrow a_n q_n \\ q_n \longrightarrow \varepsilon \end{array}$$

Consequentemente

$$q_0 \underset{\mathcal{G}_{\mathcal{A}}}{\Rightarrow} a_1 q_1 \underset{\mathcal{G}_{\mathcal{A}}}{\Rightarrow} a_1 a_2 q_2 \underset{\mathcal{G}_{\mathcal{A}}}{\Rightarrow} \cdots \underset{\mathcal{G}_{\mathcal{A}}}{\Rightarrow} a_1 \cdots a_n q_n \underset{\mathcal{G}_{\mathcal{A}}}{\Rightarrow} a_1 \cdots a_n \tag{1.1}$$

é uma derivação em $\mathcal{G}_{\mathcal{A}}$, pelo que $u \in L(\mathcal{G}_{\mathcal{A}})$.

Reciprocamente, seja $u = a_1 \cdots a_n \in L(\mathcal{G}_{\mathcal{A}})$. Então existe uma derivação $q_0 \overset{n}{\underset{\mathcal{G}_{\mathcal{A}}}{\Rightarrow}} u$ que será forçosamente do tipo 1.1, para algumas variáveis $q_1, \ldots q_n \in V$. Consequentemente, existe um caminho do estado q_0 para o estado q_n, de etiqueta u, e $q_n \in F$, pois do último passo da derivação de u conclui-se que $q_n \longrightarrow \varepsilon \in P$. Isto significa que $\mathcal{A}$ reconhece u, ou seja, $u \in L$.

Conclui-se então que $L = L(\mathcal{G}_{\mathcal{A}})$. ■

Proposição 1.19 *Seja $\mathcal{G} = (V, A, \mathcal{S}, P)$ uma gramática linear à direita. Então $L(\mathcal{G})$ é uma linguagem regular.*

Demonstração:
Podemos admitir, pela proposição 1.17, que uma produção de P é do tipo $\mathcal{B} \longrightarrow a\mathcal{C}$ ou $\mathcal{B} \longrightarrow \varepsilon$, com $\mathcal{B}, \mathcal{C} \in V$ e $a \in A$.

Considere-se o autómato finito $\mathcal{A}_{\mathcal{G}} = (Q, A, \delta, q_0, F)$ tal que

- $Q = V$,
- $q_0 = \mathcal{S}$,
- $F = \{q \in V \mid q \longrightarrow \varepsilon \in P\}$,
- δ é definida por $q \in \delta(p, a)$ se $p \longrightarrow aq \in P$ com $p, q \in V$ e $a \in A$.

Assim $\mathcal{A}_{\mathcal{G}}$ é um autómato finito, possivelmente não determinístico e não completo, tal que

$$\begin{aligned} L(\mathcal{G}) &= \{x \in A^* \mid \mathcal{S} \underset{\mathcal{G}}{\overset{*}{\Rightarrow}} x\} \\ &= \{x \in A^* \mid \exists n \in \mathbb{N}_0\ \exists a_1, \ldots, a_n \in A\ \exists \mathcal{A}_1, \ldots, \mathcal{A}_n \in V, \\ &\qquad \mathcal{S} \underset{\mathcal{G}}{\Rightarrow} a_1\mathcal{A}_1 \underset{\mathcal{G}}{\Rightarrow} \cdots \underset{\mathcal{G}}{\Rightarrow} a_1 \cdots a_n\mathcal{A}_n \underset{\mathcal{G}}{\Rightarrow} a_1 \cdots a_n = x\} \\ &= \{x \in A^* \mid \exists n \in \mathbb{N}_0\ \exists a_1, \ldots, a_n \in A, \delta^*(\mathcal{S}, a_1 \cdots a_n) \in F\ \wedge\ x = a_1 \cdots a_n\} \\ &= \{x \in A^* \mid \delta^*(\mathcal{S}, x) \in F\} \\ &= L(\mathcal{A}_{\mathcal{G}}) \end{aligned}$$

Conclui-se deste modo que $L(\mathcal{G})$ é regular, visto que é reconhecida por um autómato finito. ■

Das proposições 1.18 e 1.19 resulta imediatamente a equivalência estabelecida na proposição seguinte.

Proposição 1.20 *Uma linguagem L sobre um alfabeto A é regular se e só se é gerada por uma gramática linear à direita.* ■

Resultados análogos às proposições 1.17, 1.18 e 1.19 podem ser obtidos para gramáticas lineares à esquerda e consequentemente obter a proposição seguinte.

Proposição 1.21 *Uma linguagem L sobre um alfabeto A é regular se e só se é gerada por uma gramática linear à esquerda.* ■

Pode então concluir-se que a classe das linguagens lineares à direita coincide com a classe das linguagens lineares à esquerda, e que cada uma das proposições 1.20 e 1.21 é equivalente a afirmar que uma linguagem L sobre um alfabeto A é regular se e só se é gerada por uma gramática regular.

1.5 Exercícios

1. Seja $\mathcal{G} = (\{\mathcal{S}, \mathcal{B}\}, \{a, b\}, \mathcal{S}, P)$ a gramática com produções

$$\begin{array}{lcl} \mathcal{S} & \to & a\mathcal{S}b \mid \mathcal{B} \\ \mathcal{B} & \to & b\mathcal{B} \mid b \end{array}$$

   Prove que $L(\mathcal{G}) = \{a^n b^m : 0 \leq n < m\}$.

2. Considere a gramática $\mathcal{G} = (\{\mathcal{S}\}, \{a, b\}, \mathcal{S}, P)$ com produções

$$\mathcal{S} \to a\mathcal{S}b \mid \mathcal{S}\mathcal{S} \mid ab \mid ba$$

   Mostre que

   (a) $(ab)^2a^2b^2, a^3b^2a^3bab^4 \in L(\mathcal{G})$;

   (b) se $u \in L(\mathcal{G})$, então $|u|_a = |u|_b$;

   (c) se $u \in L(\mathcal{G})$, então b^2 não é prefixo de u.

3. Considere o alfabeto $\{a, b\}$.

   (a) Construa gramáticas que gerem as linguagens:

   $i)L_1 = \{a^n b^{2n} \mid n > 0\}$ ii)$L_2 = (abb \cup b)^*(ab)^*$
   iii)$L_3 = \{a^i b^j a \mid i > j > 0\}$ iv)$L_4 = \{a^i b^j a^k \mid j \geq (i + k),\ i, j, k \in \mathbb{N}\}$

   (b) Justifique que as linguagens dadas são independentes de contexto.

   (c) Elabore derivações e respetivas árvores de derivação de modo a provar que: $a^3b^6,\ a^4b^8 \in L_1$, $ab^4ab,\ ab^4ab^2ab^2 \in L_2$, $a^6b^5a,\ a^7b^2a \in L_3$ e $a^3b^7a^2 \in L_4$.

4. Indique, justificando, qual é a linguagem gerada pela gramática:

   (a) $\mathcal{G}_1 = (\{\mathcal{S}, \mathcal{A}\}, \{a, b\}, \mathcal{S}, P)$ com produções

$$\begin{array}{ll} \mathcal{S} & \to a\mathcal{A}bab\mathcal{A}a \\ \mathcal{A} & \to \mathcal{A}a \mid \mathcal{A}b \mid \varepsilon \end{array}$$

   (b) $\mathcal{G}_2 = (\{\mathcal{S}, \mathcal{A}, \mathcal{B}\}, \{a, b\}, \mathcal{S}, P)$ com produções

$$\begin{array}{ll} \mathcal{S} & \to a\mathcal{S}a \mid b\mathcal{S}b \mid \mathcal{A} \\ \mathcal{A} & \to a\mathcal{B}b \mid b\mathcal{B}a \\ \mathcal{B} & \to a\mathcal{B}a \mid b\mathcal{B}b \mid a \mid b \mid \varepsilon \end{array}$$

   (c) $\mathcal{G}_3 = (\{\mathcal{S}, \mathcal{A}\}, \{a, b, c, d\}, \mathcal{S}, P)$ com produções

$$\begin{array}{lcl} \mathcal{S} & \to & ab\mathcal{S}c \mid \mathcal{A} \\ \mathcal{A} & \to & c\mathcal{A}d \mid cd \end{array}$$

(d) $\mathcal{G}_4 = (V,\ ,\ \mathcal{S},\ P)$ definida por:

$$V = \{\mathcal{S}, \mathcal{B}, \mathcal{B}_0, \mathcal{B}_1, \mathcal{C}, \mathcal{C}_0, \mathcal{C}_1\} \qquad\qquad A = \{a,\ b,\ c\}$$

$$\begin{array}{lcl} \mathcal{S} & \rightarrow & \mathcal{B} \mid \mathcal{C} \\ \\ \mathcal{B} & \rightarrow & \mathcal{B}_0\mathcal{B}_1 \\ \mathcal{B}_0 & \rightarrow & a\mathcal{B}_0 b \mid \varepsilon \\ \mathcal{B}_1 & \rightarrow & \mathcal{B}_1 c \mid \varepsilon \\ \\ \mathcal{C} & \rightarrow & \mathcal{C}_0\mathcal{C}_1 \\ \mathcal{C}_0 & \rightarrow & a\mathcal{C}_0 \mid \varepsilon \\ \mathcal{C}_1 & \rightarrow & b\mathcal{C}_1 c \mid \varepsilon \end{array}$$

(e) $\mathcal{G}_5 = (V,\ A,\ \mathcal{S},\ P)$ definida por:

$$V = \{\mathcal{S}, \mathcal{B}, \mathcal{C}, \mathcal{D}\} \qquad\qquad A = \{b,\ c,\ d\}$$

$$\begin{array}{lcl} \mathcal{S} & \rightarrow & \mathcal{B}\mathcal{C} \\ \mathcal{B} & \rightarrow & b\mathcal{B} \mid \varepsilon \\ \mathcal{C} & \rightarrow & c\mathcal{C}d \mid \mathcal{D} \\ \mathcal{D} & \rightarrow & \mathcal{D}d \mid d \end{array}$$

5. Considerando as gramáticas definidas no exercício 4, elabore derivações que justifiquem que:

   (a) $a(ba)^3 \in L(\mathcal{G}_1)$;

   (b) $(b^2a)^2bab^5 \in L(\mathcal{G}_2)$;

   (c) $(ab)^2c^3d^3c^2 \in L(\mathcal{G}_3)$;

   (d) $a^2b^2c^3,\ a^3b^2c^2 \in L(\mathcal{G}_4)$;

   (e) $b^3c^2d^3,\ b^2c^3d^5\ \in L(\mathcal{G}_5)$.

6. Sejam $A = \{a, b\}$ e $L = \{x \in A^* : |x|_a = |x|_b\}$.

   (a) Mostre que se x é uma palavra de L que começa e acaba com o mesmo símbolo s, para algum $s \in A$, então $x = sus'vs$, com $u \in L$, $v \in A^*$ e $s' \in A \setminus \{s\}$.

   (b) Seja $\mathcal{G} = (\{\mathcal{S}\}, \{a, b\}, \mathcal{S}, P)$ a gramática com produções

   $$\mathcal{S} \rightarrow \mathcal{S}a\mathcal{S}b\mathcal{S} \mid \mathcal{S}b\mathcal{S}a\mathcal{S} \mid \varepsilon$$

   Usando indução matemática e o resultado da alínea (a), prove que $L(\mathcal{G}) = L$.

7. Mostre que uma gramática independente de contexto $\mathcal{G} = (\{\mathcal{S}\}, \{a, b\}, \mathcal{S}, P)$ que contém as produções

   $$\mathcal{S} \rightarrow a \mid \mathcal{S}a \mid b\mathcal{S}\mathcal{S} \mid \mathcal{S}\mathcal{S}b \mid \mathcal{S}b\mathcal{S}$$

   é ambígua.

8. Mostre que as gramáticas independentes de contexto, sobre o alfabeto $\{a, b\}$, cujas produções se apresentam a seguir são ambíguas e, em cada caso, encontre uma gramática independente de contexto equivalente e que não seja ambígua.

   (a) $\mathcal{S} \rightarrow \mathcal{SS} \mid a \mid b$ (b) $\mathcal{S} \rightarrow a\mathcal{S}b \mid aa\mathcal{S}b \mid \varepsilon$

   (c) $$\begin{array}{lcl} \mathcal{S} & \rightarrow & \mathcal{A} \mid \mathcal{B} \\ \mathcal{A} & \rightarrow & a\mathcal{A}b \mid ab \\ \mathcal{B} & \rightarrow & ab\mathcal{B} \mid \varepsilon \end{array}$$

   (d) $$\begin{array}{lcl} \mathcal{S} & \rightarrow & \mathcal{ABA} \\ \mathcal{A} & \rightarrow & a\mathcal{A} \mid \varepsilon \\ \mathcal{B} & \rightarrow & b\mathcal{B} \mid \varepsilon \end{array}$$

9. Justifique detalhadamente a seguinte afirmação: 'Dada uma gramática regular ambígua existe uma gramática regular não ambígua equivalente'.

10. Seja $\mathcal{G}$ a gramática independente de contexto cujo conjunto de produções é:

$$\mathcal{S} \rightarrow a\mathcal{S} \mid \mathcal{S}b \mid a \mid b$$

    Mostre, usando indução matemática, que nenhuma palavra de $L(\mathcal{G})$ tem o factor ba.

11. Seja $\mathcal{G}$ uma gramática independente de contexto cujo conjunto de produções é:

$$\mathcal{S} \rightarrow a\mathcal{S}b \mid ab \mid \mathcal{SS}$$

    Mostre, usando indução matemática, que nenhuma palavra de $L(\mathcal{G})$ tem como prefixo abb.

12. Considere a gramática $\mathcal{G} = (V, A, \mathcal{S}, P)$ definida por:

$$V = \{\mathcal{S}, \mathcal{B}, \mathcal{C}, \mathcal{D}\} \qquad A = \{b, c, d\}$$

$$\begin{array}{lcl} \mathcal{S} & \rightarrow & \mathcal{BC} \\ \mathcal{B} & \rightarrow & b\mathcal{B} \mid \varepsilon \\ \mathcal{C} & \rightarrow & c\mathcal{C}d \mid \mathcal{D} \\ \mathcal{D} & \rightarrow & \mathcal{D}d \mid d \end{array}$$

    (a) Verifique que se $u \in L(\mathcal{G})$, então $b^n u \in L(\mathcal{G})$, para qualquer $n \in \mathbb{N}$.
    (b) Determine a linguagem $L(\mathcal{G})$.

13. Sejam L_1 e L_2 as linguagens geradas pelas gramáticas independentes de contexto $\mathcal{G}_1 = (V_1, \Sigma, \mathcal{S}_1, P_1)$ e $\mathcal{G}_2 = (V_2, \Sigma, \mathcal{S}_2, P_2)$, respectivamente. (Suponha sem perda de generalidade que $V_1 \cap V_2 = \emptyset$.)

    (a) Mostre que a gramática $\mathcal{G}_C = (V_C, \Sigma, \mathcal{S}_C, P_C)$ onde
    $V_C = V_1 \cup V_2 \cup \{\mathcal{S}_C\}$; $\mathcal{S}_C \notin V_1 \cup V_2$; $P_C = P_1 \cup P_2 \cup \{\mathcal{S}_C \rightarrow \mathcal{S}_1\mathcal{S}_2\}$
    gera a linguagem L_1L_2.
    (b) Mostre que a gramática $\mathcal{G}_U = (V_U, \Sigma, \mathcal{S}_U, P_U)$, onde
    $V_U = V_1 \cup V_2 \cup \{\mathcal{S}_U\}$; $\mathcal{S}_U \notin V_1 \cup V_2$; $P_U = P_1 \cup P_2 \cup \{\mathcal{S}_U \rightarrow \mathcal{S}_1 \mid \mathcal{S}_2\}$;
    gera a linguagem $L_1 \cup L_2$.

(c) Mostre que a gramática $\mathcal{G}^* = (V, \Sigma, \mathcal{S}, P)$ onde

$$V = V_1 \cup \{\mathcal{S}\}; \qquad \mathcal{S} \notin V_1; \qquad P = P_1 \cup \{\mathcal{S} \to \mathcal{S}_1\mathcal{S} \mid \varepsilon\}$$

gera a linguagem L_1^*.

(d) Conclua que se L_1 e L_2 são linguagens independentes de contexto, então $L_1 \cup L_1$, L_1L_2 e L_1^* também são linguagens independentes de contexto.

(e) A seguir apresentam-se construções de gramáticas $\mathcal{G}'_C$, $\mathcal{G}'_U$ e $\mathcal{G}'^*$, respetivamente:

(Para $\mathcal{G}'_C$)	$V_C = V_1 \cup V_2$;	$\mathcal{S}_C = \mathcal{S}_1$;	$P_C = P_1 \cup P_2 \cup \{\mathcal{S}_1 \to \mathcal{S}_1\mathcal{S}_2\}$
(Para $\mathcal{G}'_U$)	$V_U = V_1 \cup V_2$;	$\mathcal{S}_U = \mathcal{S}_1$;	$P_U = P_1 \cup P_2 \cup \{\mathcal{S}_1 \to \mathcal{S}_2\}$
(Para $\mathcal{G}'^*$)	$V = V_1$;	$\mathcal{S} = \mathcal{S}_1$;	$P = P_1 \cup \{\mathcal{S}_1 \to \mathcal{S}_1\mathcal{S}_1 \mid \varepsilon\}$.

Diga se

i. $\mathcal{G}'_C$ é equivalente a $\mathcal{G}_C$,

ii. $\mathcal{G}'_U$ é equivalente a $\mathcal{G}_U$.

iii. $\mathcal{G}'^*$ é equivalente a $\mathcal{G}^*$,

e justifique.

14. Considere o alfabeto $\{a, b, c\}$. Construa gramáticas que gerem as linguagens:

(a) $L_1 = \{a^ib^jc^k \mid j = i \text{ ou } j = k,\ i, j, k \in \mathbb{N}\}$;

(b) $L_2 = \{a^ib^jc^k \mid i < j \text{ ou } i > k,\ i, j, k \in \mathbb{N}\}$;

(c) L_1L_2;

(d) $L_3 = \{a^ib^jc^k \mid i \neq (j + k),\ i, j, k \in \mathbb{N}\}$.

15. Tendo em conta a alínea (b) do exercício 13, mostre que toda a linguagem regular é uma linguagem independente de contexto.

16. Considere o alfabeto $\{a, b, c\}$.

(a) Construa uma gramática que gera a linguagem constituída por todas as palavras que contêm um número par de ocorrências da letra a.

(b) A gramática que escreveu é regular? Se não, verifique que a linguagem dada é uma linguagem regular e escreva uma gramática regular que a gere.

17. Mostre que uma linguagem $L \subseteq \Sigma^*$ sobre um alfabeto Σ é regular se e só se $L = L(\mathcal{G})$ para alguma gramática linear à esquerda $\mathcal{G}$.

18. Cada um dos conjuntos de produções que se apresentam a seguir define uma gramática independente de contexto sobre o alfabeto $\{a, b\}$ que, embora não seja regular, gera uma linguagem regular. Em cada caso, encontre uma expressão regular que represente a linguagem, indique um autómato finito e determinista que reconheça a linguagem, e determine uma gramática regular que gere a linguagem.

(a)
$$\begin{aligned} \mathcal{S} &\to \mathcal{A}ab\mathcal{B} \\ \mathcal{A} &\to a\mathcal{A} \mid b\mathcal{A} \mid \varepsilon \\ \mathcal{B} &\to \mathcal{B}ab \mid \mathcal{B}b \mid ab \mid b \end{aligned}$$

(b)
$$\begin{aligned} \mathcal{S} &\to \mathcal{A}\mathcal{A}\mathcal{S} \mid ab \mid aab \\ \mathcal{A} &\to ab \mid ba \mid \varepsilon \end{aligned}$$

19. Considere as linguagens:

i. $L_1 = \{a^i b^j c^k \mid i > j \geq k \geq 0\}$
ii. $L_2 = \{a^i b^j c^k \mid i > \text{máx}(j, k) \geq 0\}$

Para $i = 1, 2$,

(a) Construa uma gramática $\mathcal{G}_i$ que gere L_i.

(b) Classifique $\mathcal{G}_i$.

Capítulo 2

Autómatos de Pilha

As linguagens regulares são reconhecidas por autómatos finitos, mecanismos abstratos com memória fornecida pelos estados do autómato. Os estados são em número finito, que cresce com a complexidade da linguagem. No entanto o reconhecimento de uma linguagem L não regular, sobre um alfabeto A, requer uma memória não limitada, pois o conjunto das linguagens do tipo $u^{-1}L$, para $u \in A^*$, é infinito (recordar construção do autómato minimal que reconhece uma linguagem regular).

Recorde-se a linguagem L definida no Exemplo 1.2. O reconhecimento de uma palavra requer a memorização de uma quantidade de informação que depende diretamente do comprimento da palavra. Considere-se um exemplo de uma linguagem com uma estrutura mais simples.

Exemplo 2.1 *Seja $\mathcal{G} = (V, A, \mathcal{S}, P)$ tal que:*

- $V = \{\mathcal{S}\}$*;*
- $A = \{a,\, b,\, c\}$*;*
- *P é o conjunto de produções que, de acordo com a notação usual, se representam da seguinte forma:*

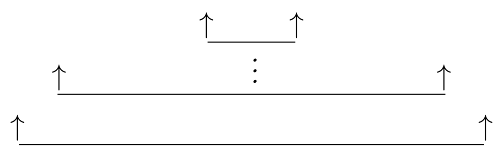
$$\mathcal{S} \longrightarrow a\mathcal{S}a \mid b\mathcal{S}b \;\mid c$$

A linguagem gerada por esta gramática é

$$L(\mathcal{G}) = \{u \in A^* \mid u = xcx^{\mathrm{I}},\; x \in \{a,b\}^*\}$$

Sendo $n \in \mathbb{N}_0$, reconhecer uma palavra de comprimento $2n+1$ desta linguagem implica memorizar as n ocorrências de letras do prefixo de comprimento n, para comparar com as ocorrências no sufixo de comprimento n.

a	b	$\cdots$	a	c	a	$\cdots$	b	a



Um modo simples de proceder é ir lendo a palavra e memorizar cada ocorrência até ocorrer o símbolo c, de forma a que a última letra guardada é a última lida. Depois *continuar a leitura e comparar a letra da primeira posição do sufixo que falta ler com a última letra guardada. Em caso de igualdade, continuar a leitura e apagar a letra da memória. Caso contrário, interromper a leitura. A identificação com sucesso de uma palavra de $L(\mathcal{G})$ significa que a palavra foi lida na totalidade e que a memória não contém mais informação.*

Este caso mostra como no reconhecimento de uma linguagem independente de contexto, pode não existir um limite para a quantidade de informação a guardar, a qual varia com o comprimento de cada palavra. Para não introduzir um número infinito de estados, a solução passa por adicionar uma memória extra, potencialmente infinita, onde é armazenada a informação necessária. A consideração do exemplo acima leva a pensar numa memória com um funcionamento tipo $LIFO$ (*Last In First Out*).

No caso da linguagem descrita no Exemplo 1.2, à partida não está identificada a posição intermédia da palavra, a partir da qual se deve passar a efetuar comparações. O processo para reconhecer tal linguagem terá que estar ligado a um mecanismo não determinista, pois em certas etapas há duas possibilidades: comparar a ocorrência seguinte com ocorrências memorizadas, considerando que foi ultrapassada a posição central, ou continuar a guardar informação prosseguindo a leitura da palavra para atingir a posição central.

O reconhecimento de linguagens não regulares exige pois o recurso a outro tipo de mecanismos abstratos. No caso das linguagens independentes de contexto, essas máquinas possuem um conjunto de estados finito e uma memória externa de tipo $LIFO$, a pilha. A combinação entre o funcionamento do conjunto de estados e o da pilha permite ultrapassar o problema do espaço de memória e definir o comportamento do mecanismo. Em geral, estes mecanismos, que se designam por autómatos de pilha, têm um carater não determinista.

2.1 Autómatos de pilha

Definição 2.2 *Um autómato de pilha (não determinístico) é um* 7*-uplo*

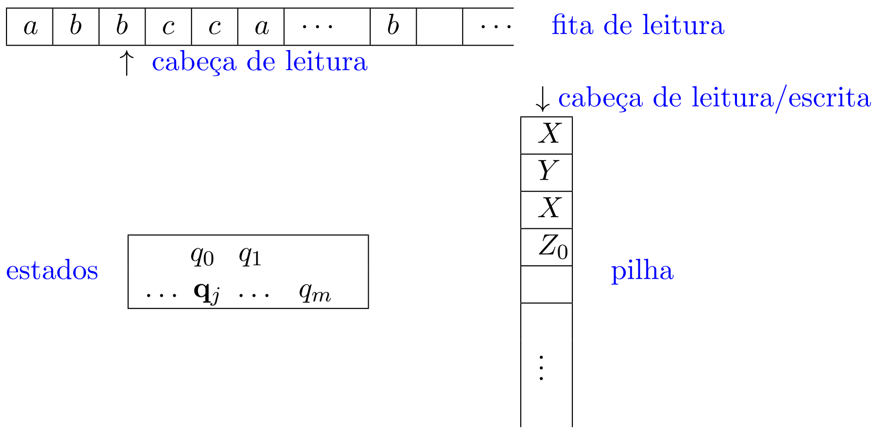
$$\mathcal{M} = (Q, A, \Sigma, \delta, q_0, Z_0, F)$$

em que:

1. Q *é um conjunto dos estados, que é finito e não vazio;*
2. A *é um alfabeto;*
3. Σ *é um conjunto finito de símbolos designado alfabeto da pilha;*
4. $\delta : Q \times (A \cup \{\varepsilon\}) \times \Sigma \longrightarrow \mathcal{P}_{fin}(Q \times \Sigma^*)$ *é uma função dita função de transição;*
5. $q_0 \in Q$ *é o estado inicial;*
6. $Z_0 \in \Sigma$ *é o símbolo inicial da pilha;*
7. $F \subseteq Q$ *é um conjunto de estados finais, também designados estados de aceitação.*

De seguida iremos analisar o funcionamento de um autómato de pilha.

Considere-se a seguinte representação de um autómato de pilha, num determinado momento:



onde $m \in \mathbb{N}$, $Q = \{q_0, q_1, \ldots, q_j, \ldots, q_m\}$, $A = \{a, b, c\}$, $X, Y, Z_0 \in \Sigma$ e Z_0 é o símbolo inicial da pilha. A palavra que se pretende reconhecer é a palavra $u = abbcca \cdots b$ que se encontra na fita de leitura, já foi lido o prefixo ab, a informação entretanto guardada é o conteúdo da pilha, $XYXZ_0$, e o estado do autómato é $\mathbf{q}_j$. A cabeça de leitura da fita, indicada pela seta, desloca-se para a direita à medida que vai lendo a palavra de entrada. A cabeça da pilha tem funções de leitura e de escrita no topo da pilha. Dependendo do estado em que o autómato se encontre, da letra que se encontre na posição de leitura da fita e do símbolo que esteja no topo da pilha, o autómato pode mover-se mudando de estado, avançando na fita de leitura ou alterando o conteúdo da pilha.

Para caraterizar a situação de um autómato de pilha num certo momento e determinar o seu comportamento futuro, até decidir se uma dada palavra é aceite, ou não, é necessário conhecer o estado em que se encontra, o sufixo que falta ler e a informação armazenada na pilha.

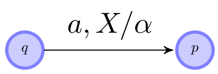
Definição 2.3 *Uma configuração de um autómato de pilha $\mathcal{M} = (Q, A, \Sigma, \delta, q_0, Z_0, F)$ é um triplo*

$$(q, w, \alpha) \in Q \times A^* \times \Sigma^*$$

Dizer que o autómato está na configuração (q, w, α) significa que o estado em que se encontra é q, o sufixo que falta ler é w e que a informação armazenada na pilha é α.

Na figura acima a configuração do autómato é $(q_j, bcca \ldots b, XYXZ_0)$.

A função de transição δ define os movimentos do autómato identificando todas as passagens permitidas de uma configuração para outra, ou seja, define como efetuar uma transição entre configurações. Formalmente, define-se uma transição como sendo um quíntuplo (q, a, X, p, α), em que $q, p \in Q$, $a \in A \cup \{\varepsilon\}$, $X \in \Sigma$ e $\alpha \in \Sigma^*$, tal que $(p, \alpha) \in \delta(q, a, X)$. Graficamente tal transição seria representada por:



Considerem-se $n \in \mathbb{N}_0$ e $u = a_1 \cdots a_n$ uma palavra de comprimento n que é colocada na fita de leitura do autómato de pilha $\mathcal{M} = (Q, A, \Sigma, \delta, q_0, Z_0, F)$. Recorde-se que a palavra vazia é elemento neutro para a concatenação, pelo que $u = a_1\varepsilon a_2\varepsilon \cdots \varepsilon a_n$. Assim, em qualquer passo, poderá ser possível decidir não avançar na fita de leitura, interpretando esse facto como correspondendo à leitura de ε. Quando se inicia a leitura de u, a configuração inicial de $\mathcal{M}$ é (q_0, u, Z_0). Em cada etapa do processo de leitura de u a configuração é (p, w, α), com $w = a_i \cdots a_n$ ou $w = \varepsilon$, e verifica-se que:

1. se $\alpha = \varepsilon$, nenhuma transição é possível;
2. se $\alpha = X\alpha'$, para $X \in \Sigma$, $\alpha' \in \Sigma^*$ e se lê a_i, então a transição é determinada pela escolha de um elemento $(q, \gamma) \in \delta(p, a_i, X)$ e a configuração passa a ser $(q, a_{i+1} \cdots a_n, \gamma\alpha')$;
3. se $\alpha = X\alpha'$, para $X \in \Sigma$, $\alpha' \in \Sigma^*$ e não se lê a_i, ou se $w = \varepsilon$, então a transição é determinada pela escolha de um elemento $(q, \gamma) \in \delta(p, \varepsilon, X)$ e a configuração passa a ser $(q, w, \gamma\alpha')$;
4. se $\delta(p, a_i, X) \cup \delta(p, \varepsilon, X) = \emptyset$, nenhuma transição é possível.

Exemplos 2.4

1. *Com base nos conceitos introduzidos pode-se pensar em construir um autómato que reconheça a linguagem* $L = \{xcx^{\mathrm{I}} \mid x \in \{a, b\}^*\}$ *sobre o alfabeto* $\{a, b, c\}$ *já estudada no Exemplo 2.1. Seja* $\mathcal{M} = (Q, \{a, b, c\}, \Sigma, \delta, q_0, Z_0, F)$ *um tal autómato. Então a função* δ *poderia ser definida do seguinte modo: para* $X \in \{a, b\}$,

$$\left.\begin{array}{ll} \delta(q_0, a, Z_0) & = \{(q_0, aZ_0)\} \\ \delta(q_0, b, Z_0) & = \{(q_0, bZ_0)\} \\ \delta(q_0, c, Z_0) & = \{(q_f, Z_0)\} \end{array}\right\} \textit{transições a partir da configuração inicial}$$

$$\left.\begin{array}{ll} \delta(q_0, a, X) & = \{(q_0, aX)\} \\ \delta(q_0, b, X) & = \{(q_0, bX)\} \end{array}\right\} \textit{transições antes de encontrar a posição central}$$

$$\left.\delta(q_0, c, X) = \{(q_1, X)\}\right\} \textit{transição quando se atinge a posição central}$$

$$\left.\begin{array}{ll} \delta(q_1, a, a) & = \{(q_1, \varepsilon)\} \\ \delta(q_1, b, b) & = \{(q_1, \varepsilon)\} \end{array}\right\} \textit{transições após encontrar a posição central}$$

$$\left.\delta(q_1, \varepsilon, Z_0) = \{(q_f, Z_0)\}\right\} \textit{transição para uma configuração final}$$

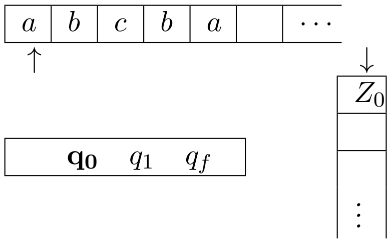
$$\delta(q, x, Y) = \emptyset \quad \textit{nos restantes casos}$$

De acordo com a definição dada para δ *pode-se concluir que* $Q = \{q_0, q_1, q_f\}$, $F = \{q_f\}$ *e* $\Sigma = \{Z_0, a, b\}$.

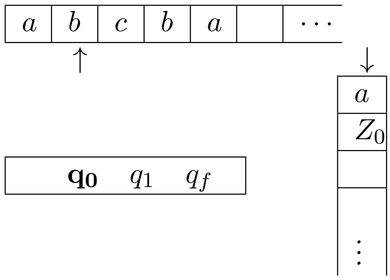
Considere-se a palavra abcba. Como é que o autómato se comporta para verificar que esta palavra pertence à linguagem? Para responder a esta questão deve-se recorrer à definição de δ, *e estudar todas as hipóteses de aplicação possíveis.*
A configuração inicial é $(q_0, abcba, Z_0)$. *Existe uma única hipótese para iniciar o funcionamento do autómato de pilha pois* $\delta(q_0, a, Z_0) \cup \delta(q_0, \varepsilon, Z_0) = \{(q_0, aZ_0)\}$. *Com base em que* $(q_0, aZ_0) \in \delta(q_0, a, Z_0)$, *na primeira etapa o autómato lê a primeira letra e transita para* $(q_0, bcba, aZ_0)$. *Depois, sucessivamente, o autómato lê a segunda letra e transita para* (q_0, cba, baZ_0), *pois* $\delta(q_0, b, a) \cup \delta(q_0, \varepsilon, a) = \{(q_0, ba)\}$ *e* $(q_0, ba) \in \delta(q_0, b, a)$, *lê a terceira letra, c, e uma vez que foi atingida a posição central vai passar a comparar a informação guardada com a o sufixo que falta ler, muda de estado e transita para* (q_1, ba, baZ_0), *dado que* $\delta(q_0, c, b) \cup \delta(q_0, \varepsilon, b) =$

$\{(q_1, b)\}$ e $(q_1, b) \in \delta(q_0, c, b)$. As transições seguintes estão também definidas de modo único e são a transição para (q_1, a, aZ_0), seguidamente para (q_1, ε, Z_0) e finalmente para (q_f, ε, Z_0). A partir desta configuração não é possível mais nenhuma transição. Esta deve ser considerada uma leitura com sucesso: atingiu-se um estado final e a palavra foi lida. As representações abaixo ilustram este processo.

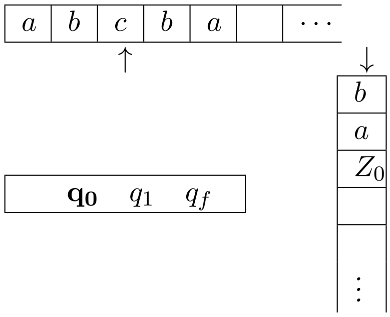
Situação inicial



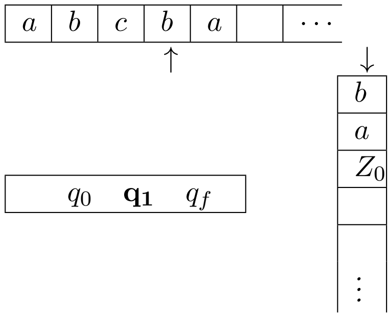
Final primeira etapa



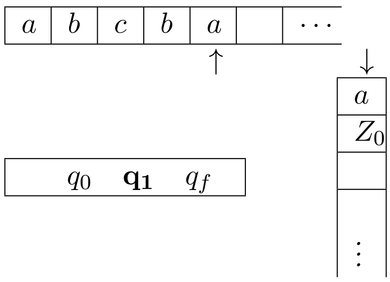
Final segunda etapa



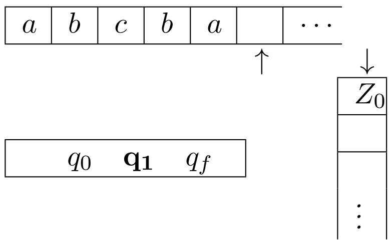
Final terceira etapa



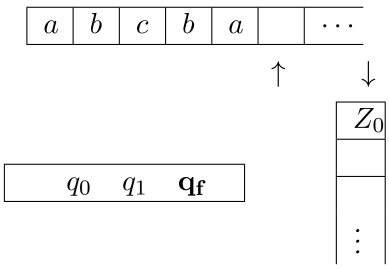
Final quarta etapa



Final quinta etapa



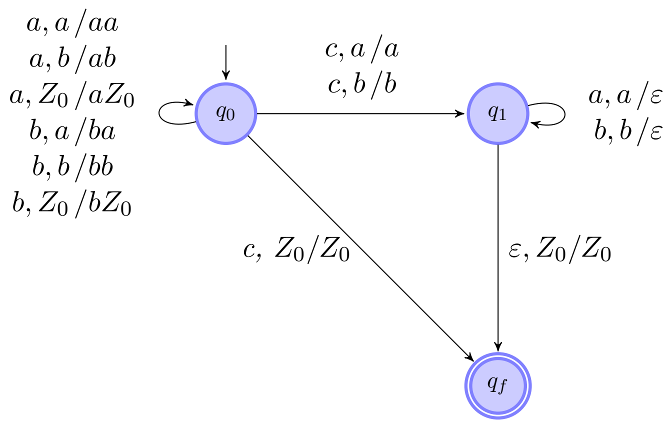
Situação final



O que aconteceria se a palavra da fita de leitura fosse abcb? Obviamente, que o autómato não deveria ler com sucesso. De modo análogo ao analisado para a palavra abcba, facilmente se conclui que da configuração inicial $(q_0, abcb, Z_0)$ *se atinge a configuração* (q_1, ε, aZ_0)*. A partir daqui a função transição não permite mais alterações da configuração e a configuração atingida significa que a leitura foi efetuada mas sem sucesso.*

O que aconteceria se a palavra em causa fosse abcbaa?

O diagrama das transições do autómato $\mathcal{M}$ *é o seguinte:*



2. *Considere-se agora a linguagem*

$$L = \{u\, u^{\mathrm{I}} \mid u \in A^*\}$$

sobre o alfabeto $A = \{a, b\}$*. Seja* $\mathcal{M} = (Q, \{a, b\}, \Sigma, \delta, q_0, Z_0, F)$ *um autómato de pilha que reconhece* L*. A função* δ *poderia ser definida do seguinte modo:*

$$\left.\begin{aligned}\delta(q_0, a, Z_0) &= \{(q_0, aZ_0)\}\\ \delta(q_0, b, Z_0) &= \{(q_0, bZ_0)\}\\ \delta(q_0, a, b) &= \{(q_0, ab)\}\\ \delta(q_0, b, a) &= \{(q_0, ba)\}\end{aligned}\right\}\ \textit{transições antes de encontrar o centro da palavra}$$

$$\left.\begin{aligned}\delta(q_0, a, a) &= \{(q_0, aa), (q_1, \varepsilon)\}\\ \delta(q_0, b, b) &= \{(q_0, bb), (q_1, \varepsilon)\}\end{aligned}\right\}\ \begin{array}{l}\textit{transição no momento em que é possível}\\ \textit{que se esteja a passar o centro}\end{array}$$

$$\left.\begin{aligned}\delta(q_1, a, a) &= \{(q_1, \varepsilon)\}\\ \delta(q_1, b, b) &= \{(q_1, \varepsilon)\}\end{aligned}\right\}\ \textit{transições após encontrar o centro}$$

$$\left.\begin{aligned}\delta(q_0, \varepsilon, Z_0) &= \{(q_1, Z_0)\}\\ \delta(q_1, \varepsilon, Z_0) &= \{(q_f, Z_0)\}\end{aligned}\right\}\ \textit{transições para uma configuração final}$$

$$\delta(q, x, Y) = \emptyset \qquad \textit{nos restantes casos}$$

De acordo com a definição dada para δ *pode-se concluir que* $Q = \{q_0, q_1, q_f\}$, $F = \{q_f\}$ *e* $\Sigma = \{Z_0, a, b\}$.

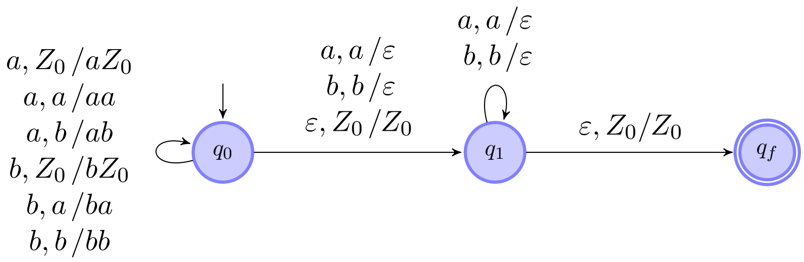
Considere-se a palavra baab. Para analisar o comportamento de $\mathcal{M}$ *quando é colocada na fita de leitura a palavra baab deve aplicar-se a definição de* δ *e estudar todas as transições possíveis. A configuração inicial é* $(q_0, baab, Z_0)$*. Existem várias possibilidades para iniciar o funcionamento do autómato de pilha, que se baseiam* *no facto de que* $\delta(q_0, b, Z_0) = \{(q_0, bZ_0)\}$ *e que* $\delta(q_0, \varepsilon, Z_0) = \{(q_1, Z_0)\}$*. Assim na primeira etapa o autómato ou lê a primeira letra* b *e transita para* (q_0, aab, bZ_0)*, ou não avança na fita de leitura e transita para* $(q_1, baab, Z_0)$*. As duas hipóteses deverão ser consideradas para verificar se alguma delas conduz a uma configuração final que se considere de leitura com sucesso.*

Supondo que se considera a segunda hipótese, $\mathcal{M}$ *transita para* $(q_1, baab, Z_0)$ *e seguidamente, dado que* $\delta(q_1, b, Z_0) \cup \delta(q_1, \varepsilon, Z_0) = \{(q_f, Z_0)\}$*, há uma única hipótese de atingir uma configuração final* $(q_f, baab, Z_0)$*. A hipótese aqui considerada não conduziu a uma leitura com sucesso, pois a palavra não foi lida na totalidade. É pois necessário explorar a outra possibilidade.*

Se $\mathcal{M}$ *lê a primeira letra, a configuração no final da primeira etapa é* (q_0, aab, bZ_0) *e a leitura pode prosseguir de uma única forma, usando o facto de que* $\delta(q_0, a, b) = \{(q_0, ab)\}$*.* $\mathcal{M}$ *transita para a configuração* (q_0, ab, abZ_0)*. Dado que* $\delta(q_0, a, a) = \{(q_0, aa), (q_1, \varepsilon)\}$*, neste momento existem dois movimentos possíveis para o autómato. Novamente, se devem estudar as duas possibilidades. O processo continua e é claro que é a escolha de* (q_1, ε) *que conduz à configuração* (q_1, b, bZ_0) *que será bem sucedida. As configurações a atingir seguidamente seriam escolhidas de forma única e seriam:* (q_1, ε, Z_0) *e* (q_f, ε, Z_0)*.*

Como atuaria o autómato dado com a palavra baabaab? E com aabbaa?

O diagrama das transições deste autómato de pilha $\mathcal{M}$ *é o seguinte:*



Definição 2.5 *No conjunto das configurações de um autómato de pilha*

$$\mathcal{M} = (Q, A, \Sigma, \delta, q_0, Z_0, F)$$

define-se a relação $\underset{\mathcal{M}}{\vdash}$ *do seguinte modo:*

1. $(p, aw, X\alpha) \underset{\mathcal{M}}{\vdash} (q, w, \gamma\alpha)$ *se* $(q, \gamma) \in \delta(p, a, X)$*;*

2. $(p, aw, X\alpha) \underset{\mathcal{M}}{\vdash} (q, aw, \gamma\alpha)$ *e* $(p, \varepsilon, X\alpha) \underset{\mathcal{M}}{\vdash} (q, \varepsilon, \gamma\alpha)$ *se* $(q, \gamma) \in \delta(p, \varepsilon, X)$

onde $p, q \in Q$*,* $a \in A$*,* $w \in A^*$*,* $X \in \Sigma$*,* $\alpha, \gamma \in \Sigma^*$*.*

Na definição anterior, em ambos os casos, a expressão traduz o facto de que o autómato de pilha $\mathcal{M}$ se movimenta transitando da configuração indicada à esquerda para a configuração indicada à direita do símbolo $\underset{\mathcal{M}}{\vdash}$, num único passo.

Por $\overset{+}{\underset{\mathcal{M}}{\vdash}}$ define-se o fecho transitivo de $\underset{\mathcal{M}}{\vdash}$ e por $\overset{*}{\underset{\mathcal{M}}{\vdash}}$ define-se o fecho reflexivo e transitivo de $\underset{\mathcal{M}}{\vdash}$. O símbolo $\underset{\mathcal{M}}{\vdash}$ lê-se deriva diretamente e o símbolo $\overset{*}{\underset{\mathcal{M}}{\vdash}}$ lê-se deriva.

Genericamente, em cada etapa, o autómato de pilha transita de uma configuração para outra com base numa escolha, o que traduz o não determinismo do autómato de pilha. Intuitivamente, o autómato deveria ser classificado de determinista se a partir de uma qualquer configuração existisse no máximo uma transição possível.

Definição 2.6 *Um autómato de pilha $\mathcal{M} = (Q, A, \Sigma, \delta, q_0, Z_0, F)$ diz-se determinista se*

1. $|\delta(q, a, X)| \leq 1$, *para quaisquer* $q \in Q$, $a \in A \cup \{\varepsilon\}$, $X \in \Sigma$;

2. *para cada* $q \in Q$, $X \in \Sigma$ *e* $a \in A$, *se* $\delta(q, a, X) \neq \emptyset$, *então* $\delta(q, \varepsilon, X) = \emptyset$.

Exemplos 2.7

1. *Considere-se o autómato do exemplo 2.4(1). A análise então efetuada para a palavra abcba traduz-se na seguinte sequência de derivações:*

$$\begin{array}{ll} (q_0, abcba, Z_0) & \underset{\mathcal{M}}{\vdash} (q_0, bcba, aZ_0) \underset{\mathcal{M}}{\vdash} (q_0, cba, baZ_0) \underset{\mathcal{M}}{\vdash} (q_1, ba, baZ_0) \underset{\mathcal{M}}{\vdash} (q_1, a, aZ_0) \\ & \underset{\mathcal{M}}{\vdash} (q_1, \varepsilon, Z_0) \underset{\mathcal{M}}{\vdash} (q_f, \varepsilon, Z_0) \end{array}$$

Para as palavras abcb e abcbaa as derivações possíveis são, respetivamente,

$$\begin{array}{ll} (q_0, abcb, Z_0) & \underset{\mathcal{M}}{\vdash} (q_0, bcb, aZ_0) \underset{\mathcal{M}}{\vdash} (q_0, cb, baZ_0) \underset{\mathcal{M}}{\vdash} (q_1, b, baZ_0) \underset{\mathcal{M}}{\vdash} (q_1, \varepsilon, aZ_0) \\ e & \\ (q_0, abcbaa, Z_0) & \underset{\mathcal{M}}{\vdash} (q_0 bcbaa, aZ_0) \underset{\mathcal{M}}{\vdash} (q_0, cbaa, baZ_0) \underset{\mathcal{M}}{\vdash} (q_1, baa, baZ_0) \\ & \underset{\mathcal{M}}{\vdash} (q_1, aa, aZ_0) \underset{\mathcal{M}}{\vdash} (q_1, a, Z_0) \end{array}$$

Facilmente se verifica que $\mathcal{M}$ é um autómato determinista. Esse facto é notório ao elaborar as derivações pois em cada passo existe apenas uma única transição possível para o autómato até atingir uma configuração final a partir da qual não é possível efetuar mais transições.

2. *Considere-se o autómato do exemplo 2.4(2). Da análise efetuada para a palavra baab conclui-se que são possíveis as seguintes sequências de derivações:*

$$(q_0, baab, Z_0) \underset{\mathcal{M}}{\vdash} (q_1, baab, Z_0) \underset{\mathcal{M}}{\vdash} (q_f, baab, Z_0)$$

$$\begin{array}{ll} (q_0, baab, Z_0) & \underset{\mathcal{M}}{\vdash} (q_0, aab, bZ_0) \underset{\mathcal{M}}{\vdash} (q_0, ab, abZ_0) \underset{\mathcal{M}}{\vdash} (q_1, b, bZ_0) \underset{\mathcal{M}}{\vdash} (q_1, \varepsilon, Z_0) \\ & \underset{\mathcal{M}}{\vdash} (q_f, \varepsilon, Z_0) \end{array}$$

$$(q_0, baab, Z_0) \underset{\mathcal{M}}{\vdash} (q_0, aab, bZ_0) \underset{\mathcal{M}}{\vdash} (q_0, ab, abZ_0) \underset{\mathcal{M}}{\vdash} (q_0, b, aabZ_0) \underset{\mathcal{M}}{\vdash} (q_0, \varepsilon, baabZ_0)$$

Analisando estas derivações pode concluir-se que apenas no segundo caso a leitura ocorreu com sucesso. No entanto, existem várias sequências de derivações possíveis *a partir de uma única configuração inicial. Trata-se pois de um autómato de pilha não determinista.*

Serão as palavras baabaab e $b^3ab^4ab^3$ aceites por este autómato? Para responder seria necessário, em cada caso, efetuar todas as sequências de derivações possíveis a partir da configuração inicial e verificar se alguma conduz à configuração final (q_f, ε, Z_0).

Por *linguagem reconhecida por um autómato de pilha* entende-se o conjunto de todas as palavras sobre o alfabeto de símbolos de entrada que o autómato aceita, isto é, todas as palavras que sendo colocadas na fita de leitura são lidas pelo autómato com sucesso. Dois autómatos de pilha dizem-se equivalentes se reconhecem a mesma linguagem.

Importa clarificar o que se entende por ser aceite pelo autómato, isto é, definir as configurações do autómato que indicam uma leitura com sucesso. Nos exemplos anteriores em que condições se considerou que a leitura terminou com sucesso? Em que estado se encontrava o autómato? Em que posição se encontrava a cabeça de leitura? Qual era o conteúdo da pilha?

Existem dois critérios para definir a configuração que corresponde ao facto de uma dada palavra ter sido aceite por um autómato de pilha: um recorre à noção de estados finais, outro considera que o processo termina com sucesso se a pilha estiver vazia. Em qualquer caso, considera-se que uma leitura com sucesso implica que a palavra foi lida na totalidade. Cada autómato de pilha utiliza apenas um destes dois critérios, e a sua construção, nomeadamente a definição da função de transição, depende do critério escolhido. Nos exemplos anteriores recorreu-se à introdução de estados finais.

Definição 2.8 *Sejam $\mathcal{M} = (Q, A, \Sigma, \delta, q_0, Z_0, F)$ um autómato de pilha e $u \in A^*$. Diz-se que:*

1. *$\mathcal{M}$ aceita u pelo critério dos estados finais se*

$$(q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q, \varepsilon, X)$$

para algum $X \in \Sigma^*$ e $q \in F$;

2. *$\mathcal{M}$ aceita u pelo critério de pilha vazia se*

$$(q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q, \varepsilon, \varepsilon)$$

para algum $q \in Q$.

Definição 2.9 *Nas condições da Definição 2.8 temos que:*

1. *a linguagem reconhecida por $\mathcal{M}$ segundo o critério dos estados finais é:*

$$L_{EF}(\mathcal{M}) = \{u \in A^* \mid \exists q \in F,\ \exists X \in \Sigma^*,\ (q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q, \varepsilon, X)\}$$

2. *a linguagem reconhecida por $\mathcal{M}$ segundo o critério de pilha vazia é:*

$$L_{PV}(\mathcal{M}) = \{u \in A^* \mid \exists q \in Q,\ (q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q, \varepsilon, \varepsilon)\}$$

Exemplos 2.10

1. *Nos exemplos estudados em 2.4 o critério utilizado foi o dos estados finais, isto é, em cada caso teríamos* $L = L_{EF}(\mathcal{M})$. *Mas, nesses casos, a opção do critério de pilha vazia implica apenas uma pequena alteração na descrição da função de transição.*

   Por exemplo, no caso 2.4(1), um autómato de pilha que reconhece a linguagem L *pelo critério de pilha vazia é* $\mathcal{M}_v = (\{q_0, q_1\}, \{a, b, c\}, \{Z_0, a, b\}, \delta_v, q_0, Z_0, \emptyset)$ *em que a função* δ_v *poderia ser definida do seguinte modo: para* $X \in \{a, b\}$,

   $$\left.\begin{array}{ll}\delta_v(q_0, a, Z_0) & = \{(q_0, aZ_0)\} \\ \delta_v(q_0, b, Z_0) & = \{(q_0, bZ_0)\}\end{array}\right\} \text{ transições a partir da configuração inicial}$$

   $$\left.\begin{array}{ll}\delta_v(q_0, a, X) & = \{(q_0, aX)\} \\ \delta_v(q_0, b, X) & = \{(q_0, bX)\}\end{array}\right\} \text{ transições antes de encontrar a posição central}$$

   $$\delta_v(q_0, c, X) = \{(q_1, X)\} \ \} \text{ transição quando se atinge a posição central}$$

   $$\left.\begin{array}{ll}\delta_v(q_1, a, a) & = \{(q_1, \varepsilon)\} \\ \delta_v(q_1, b, b) & = \{(q_1, \varepsilon)\}\end{array}\right\} \text{ transições após encontrar a posição central}$$

   $$\left.\begin{array}{ll}\delta_v(q_0, c, Z_0) & = \{(q_0, \varepsilon)\} \\ \delta_v(q_1, \varepsilon, Z_0) & = \{(q_1, \varepsilon)\}\end{array}\right\} \text{ transição para uma configuração de pilha vazia}$$

   $$\delta_v(q, x, Y) = \emptyset \quad \text{nos restantes casos}$$

   A palavra abcba seria aceite por $\mathcal{M}_v$ *pois:*

   $$\begin{array}{ll}(q_0, abcba, Z_0) & \underset{\mathcal{M}_v}{\vdash} (q_0, bcba, aZ_0) \underset{\mathcal{M}_v}{\vdash} (q_0, cba, baZ_0) \underset{\mathcal{M}_v}{\vdash} (q_1, ba, baZ_0) \\ & \underset{\mathcal{M}_v}{\vdash} (q_1, a, aZ_0) \underset{\mathcal{M}_v}{\vdash} (q_1, \varepsilon, Z_0) \underset{\mathcal{M}_v}{\vdash} (q_1, \varepsilon, \varepsilon)\end{array}$$

   Para as palavras abcb e abcbaa as derivações possíveis são, respetivamente,

   $$(q_0, abcb, Z_0) \quad \underset{\mathcal{M}}{\vdash} (q_0, bcb, aZ_0) \underset{\mathcal{M}}{\vdash} (q_0, cb, baZ_0) \underset{\mathcal{M}}{\vdash} (q_1, b, baZ_0) \underset{\mathcal{M}}{\vdash} (q_1, \varepsilon, aZ_0)$$

   e

   $$\begin{array}{ll}(q_0, abcbaa, Z_0) & \underset{\mathcal{M}}{\vdash} (q_0 bcbaa, aZ_0) \underset{\mathcal{M}}{\vdash} (q_0, cbaa, baZ_0) \underset{\mathcal{M}}{\vdash} (q_1, baa, baZ_0) \\ & \underset{\mathcal{M}}{\vdash} (q_1, aa, aZ_0) \underset{\mathcal{M}}{\vdash} (q_1, a, Z_0) \underset{\mathcal{M}}{\vdash} (q_1, a, \varepsilon)\end{array}$$

   De notar as semelhanças entre as sequências de movimentos realizados por $\mathcal{M}_v$ *e por* $\mathcal{M}$.

   De facto, verifica-se que $L_{PV}(\mathcal{M}_v) = L_{EF}(\mathcal{M})$.

2. *Seja* $L = \{a^n b^m \mid n \geq m, \ n, m \in \mathbb{N}_0\}$ *uma linguagem sobre o alfabeto* $\{a, b\}$. *Um autómato de pilha que reconhece* L *com base no critério de estados finais é* $\mathcal{M}_L = (\{q_0, q_1\}, \{a, b\}, \{Z_0, a\}, \delta_L, q_0, Z_0, \{q_1\})$ *onde a função* δ_L *é definida por:*

   $$\left.\begin{array}{ll}\delta_L(q_0, a, X) & = \{(q_0, aX)\} \\ \delta_L(q_0, \varepsilon, X) & = \{(q_1, X)\}\end{array}\right\} \text{ transições antes de encontrar a primeira ocorrência de } b$$

   $$\delta_L(q_1, b, a) = \{(q_1, \varepsilon)\} \ \} \text{ transições após encontrar a primeira ocorrência de } b$$

   $$\delta_L(q, x, X) = \emptyset \quad \text{nos restantes casos}$$

onde $X \in \Sigma$ *e* $x \in A$.

Trata-se de um autómato não determinista pois $\delta_L(q_0, \varepsilon, X) \neq \emptyset$ *e* $\delta_L(q_0, a, X) \neq \emptyset$.

Será que o autómato $\mathcal{M}_L$ *reconhece as palavras* $\varepsilon, a^4b^2,\ a^2ba,\ b^2a^3,\ a^3$ *ou* ab^2*?*

- $(q_0, \varepsilon, Z_0) \underset{\mathcal{M}_L}{\vdash} (q_1, \varepsilon, Z_0)$
- $(q_0, a^4b^2, Z_0) \underset{\mathcal{M}_L}{\vdash} (q_0, a^3b^2, aZ_0) \underset{\mathcal{M}_L}{\vdash} (q_0, a^2b^2, a^2Z_0) \underset{\mathcal{M}_L}{\vdash} (q_0, ab^2, a^3Z_0)$
  $\underset{\mathcal{M}_L}{\vdash} (q_0, b^2, a^4Z_0) \underset{\mathcal{M}_L}{\vdash} (q_1, b^2, a^4Z_0) \underset{\mathcal{M}_L}{\vdash} (q_1, b, a^3Z_0) \underset{\mathcal{M}_L}{\vdash} (q_1, \varepsilon, a^2Z_0)$
- $\begin{cases} (q_0, a^2ba, Z_0) \underset{\mathcal{M}_L}{\vdash} (q_1, a^2ba, Z_0) \\ (q_0, a^2ba, Z_0) \underset{\mathcal{M}_L}{\vdash} (q_0, aba, aZ_0) \underset{\mathcal{M}_L}{\vdash} (q_1, aba, aZ_0) \\ (q_0, a^2ba, Z_0) \underset{\mathcal{M}_L}{\vdash} (q_0, aba, aZ_0) \underset{\mathcal{M}_L}{\vdash} (q_0, ba, a^2Z_0) \underset{\mathcal{M}_L}{\vdash} (q_1, ba, a^2Z_0) \\ \qquad\qquad\quad \underset{\mathcal{M}_L}{\vdash} (q_1, a, aZ_0) \end{cases}$
- $(q_0, b^2a^3, Z_0) \underset{\mathcal{M}_L}{\vdash} (q_1, b^2a^3, Z_0)$
- $(q_0, a^3, Z_0) \underset{\mathcal{M}_L}{\vdash} (q_0, a^2, aZ_0) \underset{\mathcal{M}_L}{\vdash} (q_0, a, a^2Z_0) \underset{\mathcal{M}_L}{\vdash} (q_0, \varepsilon, a^3Z_0) \underset{\mathcal{M}_L}{\vdash} (q_1, \varepsilon, a^3Z_0)$
- $\begin{cases} (q_0, ab^2, Z_0) \underset{\mathcal{M}_L}{\vdash} (q_0, b^2, aZ_0) \underset{\mathcal{M}_L}{\vdash} (q_1, b^2, aZ_0) \underset{\mathcal{M}_L}{\vdash} (q_1, b, Z_0) \\ (q_0, ab^2, Z_0) \underset{\mathcal{M}_L}{\vdash} (q_1, ab^2, Z_0) \end{cases}$

Pode assim concluir-se que ε, a^4b^2 *e* a^3 *são aceites. As palavras* a^2ba, b^2a^3 *e* ab^2 *não são aceites, pois foram tentadas todas as derivações possíveis e nenhuma conduziu a uma configuração do tipo* $(q_1, \varepsilon, \alpha)$.

Note-se que o estado q_0 *só permite a leitura das ocorrências da letra* a*, enquanto que o estado* q_1 *só permite ler as ocorrências da letra* b *desde que no topo da pilha esteja um* a.

Genericamente, para $n \geq m$*, tem-se*

$$(q_0, a^nb^m, Z_0) \overset{*}{\underset{\mathcal{M}_L}{\vdash}} (q_0, b^m, a^nZ_0) \underset{\mathcal{M}_L}{\vdash} (q_1, b^m, a^nZ_0) \overset{*}{\underset{\mathcal{M}_L}{\vdash}} (q_1, \varepsilon, a^{n-m}Z_0)$$

ou seja, $L \subseteq L_{EF}(\mathcal{M}_L)$.

Pode também constatar-se que as palavras, diferentes de ε *aceites por* $\mathcal{M}_L$ *começam com a letra* a*, podem ou não ter ocorrências de* b*, e que as ocorrências de* b *só podem acontecer depois de todas as ocorrências de* a *e em número inferior ou igual. Equivalentemente dir-se-ia que* $\mathcal{M}_L$ *não aceita palavras de sufixo* b*, nem com fator* ba*, nem palavras* u *tais que* $|u|_a < |u|_b$*, isto é,* $L_{EF}(\mathcal{M}_L) \subseteq L$.

3. *Considere-se a linguagem* $L = \{wa^{|w|} \mid w \in \{a,b\}^+\}$ *sobre o alfabeto* $\{a,b\}$*. Um autómato de pilha que reconhece* L *com base no critério de pilha vazia é* $\mathcal{M} = (\{q_0,q_1\}, \{a,b\}, \{Z_0,a,b\}, \delta, q_0, Z_0, \emptyset)$ *onde a função* δ *é definida por: para* $X \in \{a,b\}$,

$$\begin{array}{ll}
\delta(q_0,a,Z_0) & =\{(q_0,aZ_0)\}\\
\delta(q_0,b,Z_0) & =\{(q_0,bZ_0)\}\\
\delta(q_0,a,X) & =\{(q_0,aX),(q_1,\varepsilon)\}\\
\delta(q_0,b,X) & =\{(q_0,bX)\}\\
\delta(q_1,a,X) & =\{(q_1,\varepsilon)\}\\
\delta(q_1,\varepsilon,Z_0) & =\{(q_1,\varepsilon)\}
\end{array}$$

Trata-se de um autómato não determinista pois $|\delta(q_0,a,X)|=2$.

O autómato $\mathcal{M}$ *aceita a palavra* b^2a^6 *pois existe a sequência:*

$$\begin{array}{ll}
(q_0,b^2a^6,Z_0) & \underset{\mathcal{M}}{\vdash} (q_0,ba^6,bZ_0) \underset{\mathcal{M}}{\vdash} (q_0,a^6,b^2Z_0) \underset{\mathcal{M}}{\vdash} (q_0,a^5,ab^2Z_0)\\
& \underset{\mathcal{M}}{\vdash} (q_0,a^4,a^2b^2Z_0) \underset{\mathcal{M}}{\vdash} (q_1,a^3,ab^2Z_0) \underset{\mathcal{M}}{\vdash} (q_1,a^2,b^2Z_0)\\
& \underset{\mathcal{M}}{\vdash} (q_1,a,bZ_0) \underset{\mathcal{M}}{\vdash} (q_1,\varepsilon,Z_0) \underset{\mathcal{M}}{\vdash} (q_1,\varepsilon,\varepsilon)
\end{array}$$

No entanto, existem outras sequências a partir da mesma configuração inicial, por exemplo:

$$(q_0,b^2a^6,Z_0) \underset{\mathcal{M}}{\vdash} (q_0,ba^6,bZ_0) \underset{\mathcal{M}}{\vdash} (q_0,a^6,b^2Z_0) \underset{\mathcal{M}}{\vdash} (q_1,a^5,bZ_0) \underset{\mathcal{M}}{\vdash} (q_1,a^4,Z_0)$$

$$\begin{array}{ll}
(q_0,b^2a^6,Z_0) & \underset{\mathcal{M}}{\vdash} (q_0,ba^6,bZ_0) \underset{\mathcal{M}}{\vdash} (q_0,a^6,b^2Z_0) \underset{\mathcal{M}}{\vdash} (q_0,a^5,ab^2Z_0)\\
& \underset{\mathcal{M}}{\vdash} (q_1,a^4,b^2Z_0) \underset{\mathcal{M}}{\vdash} (q_1,a^3,bZ_0) \underset{\mathcal{M}}{\vdash} (q_1,a^2,Z_0)
\end{array}$$

Genericamente, para $w=vx\in\{a,b\}^+$ *e* $x\in\{a,b\}$*, tem-se*

$$(q_0,wa^{|w|},Z_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q_0,a^{|w|},w^{\mathrm{I}}Z_0) \underset{\mathcal{M}}{\vdash} (q_1,a^{|w|-1},v^{\mathrm{I}}Z_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q_1,\varepsilon,Z_0) \underset{\mathcal{M}}{\vdash} (q_1,\varepsilon,\varepsilon)$$

ou seja, $L\subseteq L_{PV}(\mathcal{M})$.

Inversamente, se u *é aceite por* $\mathcal{M}$*, então* $(q_0,u,Z_0)\overset{*}{\underset{\mathcal{M}}{\vdash}}(q_1,\varepsilon,\varepsilon)$. *Notar que:*

(i) a única forma de esvaziar a pilha é a partir de uma configuração do estado q_1 *não lendo nada e tendo no topo da pilha* Z_0,

(ii) numa configuração com estado q_1 *só é possível a leitura da letra* a *ou a transição para pilha vazia.*

Então

$$(q_0,u,Z_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q_0,a^n,xv^{\mathrm{I}}Z_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q_1,a^{n-1},v^{\mathrm{I}}Z_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q_1,\varepsilon,Z_0) \underset{\mathcal{M}}{\vdash} (q_1,\varepsilon,\varepsilon)$$

para $n\in\mathbb{N}$, $v\in\{a,b\}^*$ *tal que* $|v|=n-1$. *Em movimentos entre configurações com o estado* q_0*, tudo o que é lido na fita é guardado na pilha, pelo que*

$$u=vxa^n$$

e $|vx|=n$. *Assim,* $u\in L$.

2.2 Equivalência entre critérios para reconhecer palavras

Da experiência obtida na análise do exemplo 2.10(1), pode-se pensar que as diferenças entre os dois critérios são pouco significativas, mas em 2.10(2) a construção de um autómato que reconhece a mesma linguagem pelo critério de pilha vazia não é tão simples. O processo utilizado no primeiro caso não é generalizável, mas será que existe um autómato de pilha que reconhece a linguagem dada em 2.10(2) pelo critério de pilha vazia? E será que existe um autómato de pilha que reconhece a linguagem definida em 2.10(3) pelo critério de estados finais? A resposta é afirmativa em ambos os casos.

Os dois resultados que se seguem, Proposições 2.11 e 2.13, têm como consequência a equivalência entre os dois critérios e as suas demonstrações fornecem métodos que permitem a partir de um autómato de pilha que reconhece uma linguagem segundo um critério construir outro autómato de pilha que reconhece a mesma linguagem segundo o outro critério.

Proposição 2.11 *Sejam $L \subseteq A^*$ e $\mathcal{M}_1 = (Q_1, A, \Sigma_1, \delta_1, q_0, Z_0, F_1)$ um autómato de pilha tal que $L = L_{PV}(\mathcal{M}_1)$. Então, existe um autómato de pilha $\mathcal{M}$ tal que*

$$L = L_{EF}(\mathcal{M})$$

Demonstração:
O objetivo é construir o autómato de pilha $\mathcal{M}$ a partir de $\mathcal{M}_1$ introduzindo um estado final, digamos q_f, em que um movimento para uma configuração em que a pilha fica vazia dá lugar a um movimento para uma configuração com o estado q_f. Note-se que F_1 pode ser o conjunto vazio.

Considere-se

$$\mathcal{M} = (Q, A, \Sigma, \delta, q_0', U_0, \{q_f\})$$

tal que:

- $Q = Q_1 \cup \{q_0', q_f\}$, onde $q_0', q_f \notin Q_1$;
- $\Sigma = \Sigma_1 \cup \{U_0\}$, onde $U_0 \notin \Sigma_1$;
- $\delta : Q \times (A \cup \{\varepsilon\}) \times \Sigma \longrightarrow \mathcal{P}_{fin}(Q \times \Sigma^*)$ é definida por:

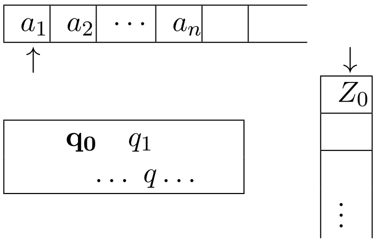
$$\begin{array}{ll}
\delta(q_0', \varepsilon, U_0) = \{(q_0, Z_0U_0)\} & \\
\delta(q, a, X) = \delta_1(q, a, X), & \text{para quaisquer } q \in Q_1,\ a \in A \cup \{\varepsilon\},\ X \in \Sigma_1 \\
\delta(q, \varepsilon, U_0) = \{(q_f, U_0)\}, & \text{para qualquer } q \in Q_1 \\
\delta(q, a, X) = \emptyset & \text{nos restantes casos}
\end{array}$$

De seguida demonstra-se que $\mathcal{M}$ está nas condições do enunciado. Seja $u \in L_{PV}(\mathcal{M}_1)$. Então

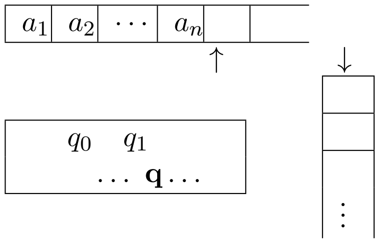
$$(q_0, u, Z_0) \underset{\mathcal{M}_1}{\overset{*}{\vdash}} (q, \varepsilon, \varepsilon) \tag{2.1}$$

para algum $q \in Q_1$. Na figura seguinte ilustra-se a situação inicial e final do autómato $\mathcal{M}_1$ no processo de reconhecer a palavra $u = a_1 \cdots a_n$.

Situação inicial



Situação final

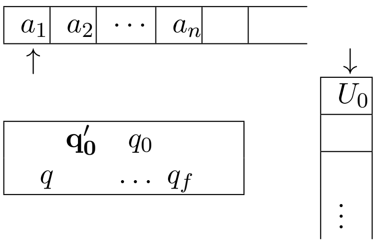


Da definição de δ e de (2.1) resulta que

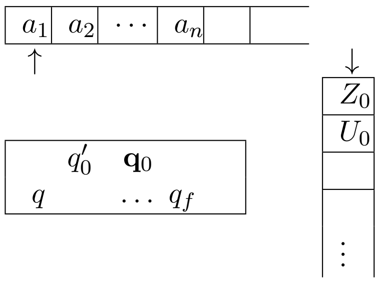
$$(q_0', u, U_0) \underset{\mathcal{M}}{\vdash} (q_0, u, Z_0U_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q, \varepsilon, U_0) \underset{\mathcal{M}}{\vdash} (q_f, \varepsilon, U_0) \tag{2.2}$$

pelo que $u \in L_{EF}(\mathcal{M})$. A seguinte sequência de representações do autómato $\mathcal{M}$ ilustra a sequência de transições (2.2).

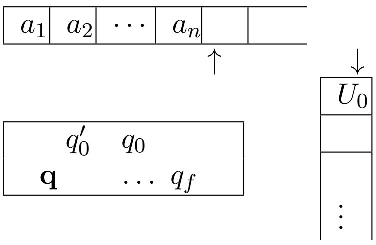
Situação inicial



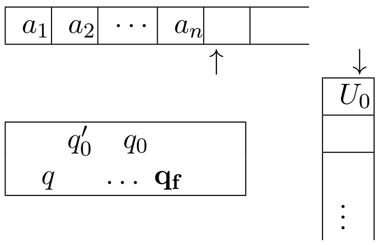
Final da primeira etapa



Final da penúltima etapa



Situação final



Reciprocamente, se $u \in L_{EF}(\mathcal{M})$, então

$$(q_0', u, U_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q_f, \varepsilon, \alpha) \tag{2.3}$$

Note-se que, pela definição de δ, a única configuração de $\mathcal{M}$ que engloba q_0' é a inicial, (q_0', u, U_0), e que a única configuração de $\mathcal{M}$ que engloba q_f é a final, (q_f, ε, U_0). Então, em (2.3), $\alpha = U_0$. Assim, dado que $\delta(q_0', \varepsilon, U_0) = \{(q_0, Z_0U_0)\}$ e $\delta(q, \varepsilon, U_0) = \{(q_f, U_0)\}$, para qualquer $q \in Q_1$, em qualquer caso, a primeira e a última transições estão fixas. Consequentemente, (2.3) implica que existe $q \in Q_1$ tal que

$$(q_0', u, U_0) \underset{\mathcal{M}}{\vdash} (q_0, u, Z_0U_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q, \varepsilon, U_0) \underset{\mathcal{M}}{\vdash} (q_f, \varepsilon, U_0) \tag{2.4}$$

Consequentemente,

$$(q_0, u, Z_0) \underset{\mathcal{M}_1}{\overset{*}{\vdash}} (q, \varepsilon, \varepsilon)$$

ou seja, $u \in L_{PV}(\mathcal{M}_1)$. ■

Exemplo 2.12 *Sejam $L_c = \{a^i b^j c^i \mid i, j \in \mathbb{N}_0\}$ uma linguagem sobre o alfabeto $\{a, b, c\}$ e $\mathcal{M}_c = (\{q_0, q_1, q_2\}, \{a, b, c\}, \{Z_0, a\}, \delta, q_0, Z_0, \emptyset)$ um autómato de pilha que reconhece com base no critério de pilha vazia, sendo a função δ definida por:*

$$\begin{array}{ll}
\left.\begin{array}{ll}
\delta(q_0, a, Z_0) & = \{(q_0, aZ_0)\} \\
\delta(q_0, a, a) & = \{(q_0, aa)\}
\end{array}\right\} & \textit{transições antes de encontrar a primeira ocorrência de } b \\
\left.\begin{array}{ll}
\delta(q_0, b, a) & = \{(q_1, a)\} \\
\delta(q_0, b, Z_0) & = \{(q_1, Z_0)\} \\
\delta(q_1, b, a) & = \{(q_1, a)\} \\
\delta(q_1, b, Z_0) & = \{(q_1, Z_0)\}
\end{array}\right\} & \textit{transições após encontrar a primeira ocorrência de } b \\
\left.\begin{array}{ll}
\delta(q_0, c, a) & = \{(q_2, \varepsilon)\} \\
\delta(q_1, c, a) & = \{(q_2, \varepsilon)\} \\
\delta(q_2, c, a) & = \{(q_2, \varepsilon)\}
\end{array}\right\} & \textit{transições após encontrar a primeira ocorrência de } c \\
\left.\begin{array}{ll}
\delta(q_0, \varepsilon, Z_0) & = \{(q_0, \varepsilon)\} \\
\delta(q_1, \varepsilon, Z_0) & = \{(q_1, \varepsilon)\} \\
\delta(q_2, \varepsilon, Z_0) & = \{(q_2, \varepsilon)\}
\end{array}\right\} & \textit{transições para esvaziar a pilha} \\
\delta(q, x, X) \quad = \emptyset \quad \textit{nos restantes casos} &
\end{array}$$

Verifica-se que o autómato $\mathcal{M}_c$ aceita as palavras a^3bc^3, a^2c^2, b^4, ε e ab^3c pois existem as sequências de derivações:

- $$\begin{aligned}(q_0, a^3bc^3, Z_0) &\underset{\mathcal{M}_c}{\vdash} (q_0, a^2bc^3, aZ_0) \underset{\mathcal{M}_c}{\vdash} (q_0, abc^3, a^2Z_0) \underset{\mathcal{M}_c}{\vdash} (q_0, bc^3, a^3Z_0) \\ &\underset{\mathcal{M}_c}{\vdash} (q_1, c^3, a^3Z_0) \underset{\mathcal{M}_c}{\vdash} (q_2, c^2, a^2Z_0) \underset{\mathcal{M}_c}{\vdash} (q_2, c, aZ_0) \\ &\underset{\mathcal{M}_c}{\vdash} (q_2, \varepsilon, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_2, \varepsilon, \varepsilon)\end{aligned} \tag{2.5}$$
- $$\begin{aligned}(q_0, a^2c^2, Z_0) &\underset{\mathcal{M}_c}{\vdash} (q_0, ac^2, aZ_0) \underset{\mathcal{M}_c}{\vdash} (q_0, c^2, a^2Z_0) \underset{\mathcal{M}_c}{\vdash} (q_2, c, aZ_0) \underset{\mathcal{M}_c}{\vdash} (q_2, \varepsilon, Z_0) \\ &\underset{\mathcal{M}_c}{\vdash} (q_2, \varepsilon, \varepsilon)\end{aligned}$$
- $(q_0, b^4, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, b^3, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, b^2, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, b, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, \varepsilon, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, \varepsilon, \varepsilon)$
- $(q_0, \varepsilon, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_0, \varepsilon, \varepsilon)$
- $$\begin{aligned}(q_0, ab^3c, Z_0) &\underset{\mathcal{M}_c}{\vdash} (q_0, b^3c, aZ_0) \underset{\mathcal{M}_c}{\vdash} (q_1, b^2c, aZ_0) \underset{\mathcal{M}_c}{\vdash} (q_1, bc, aZ_0) \underset{\mathcal{M}_c}{\vdash} (q_1, c, aZ_0) \\ &\underset{\mathcal{M}_c}{\vdash} (q_2, \varepsilon, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_2, \varepsilon, \varepsilon)\end{aligned}$$

No entanto, não aceita as palavras ca, cba, b^2c^2 nem $b^2a^3c^3$, por exemplo, pois tentando todas as sequências de derivações possíveis,

- $(q_0, ca, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_0, ca, \varepsilon)$
- $(q_0, cba, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_0, cba, \varepsilon)$

- $$\begin{cases} (q_0, b^2c^2, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_0, b^2c^2, \varepsilon) \\ (q_0, b^2c^2, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, bc^2, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, bc^2, \varepsilon) \\ (q_0, b^2c^2, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, bc^2, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, c^2, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, c^2, \varepsilon) \end{cases}$$

- $$\begin{cases} (q_0, b^2a^3c^3, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_0, b^2a^3c^3, \varepsilon) \\ (q_0, b^2a^3c^3, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, ba^3c^3, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, ba^3c^3, \varepsilon) \\ (q_0, b^2a^3c^3, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, ba^3c^3, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, a^3c^3, Z_0) \underset{\mathcal{M}_c}{\vdash} (q_1, a^3c^3, \varepsilon) \end{cases}$$

De modo análogo, verifica-se que $\mathcal{M}_c$ não reconhece as palavras $abac^2$, $a2bc^3a$, ba^2c^2, a^2c^2b, a^2bc, ab^2c^2 e a^3c, por exemplo.

Genericamente, $\mathcal{M}_c$ não reconhece palavras de prefixo c, nem as que admitem ba, ca ou cb como fatores, o que se deve ao facto de que, respetivamente, $\delta(q_0, c, X) = \emptyset$ e $\delta(q_1, a, X) = \delta(q_2, a, X) = \delta(q_2, b, X) = \emptyset$. Logo as palavras aceites por $\mathcal{M}_c$ são do tipo $u = a^ib^jc^k$, $i, j, k \in \mathbb{N}_0$. No entanto, $\mathcal{M}_c$ também não reconhece u se $|u|_a \neq |u|_c$, isto é, em que $i \neq k$. Em tal caso, as derivações que correspondem a ler prefixos de u de comprimento máximo são

$$\begin{array}{ll} (q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}_c}{\vdash}} (q_0, \varepsilon, a^iZ_0) & \textit{se } i > k = j = 0 \\ (q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}_c}{\vdash}} (q_1, \varepsilon, a^iZ_0) & \textit{se } i > k = 0 \textit{ e } j \neq 0 \\ (q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}_c}{\vdash}} (q_2, \varepsilon, a^{i-k}Z_0) & \textit{se } i > k > 0 \\ (q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}_c}{\vdash}} (q_0, c^k, Z_0) & \textit{se } 0 = i < k \\ (q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}_c}{\vdash}} (q_2, c^{k-i}, Z_0) & \textit{se } i < k \end{array}$$

pelo que o autómato não reconhece u. Constata-se assim que $L_{PV}(\mathcal{M}_c) = L_c$.

Aplicando a proposição 2.11 poderíamos construir um autómato equivalente que reconhece segundo o critério de estados finais.

Considere-se $\mathcal{M}_f = (\{q'_0, q_0, q_1, q_2, q_f\}, \{a, b, c\}, \{Z_0, U_0, a\}, \delta_f, q'_0, U_0, \{q_f\})$ tal que:

$$\begin{array}{l} \delta_f(q'_0, \varepsilon, U_0) = \{(q_0, Z_0U_0)\} \\ \left.\begin{array}{ll} \delta_f(q_0, a, Z_0) & = \{(q_0, aZ_0)\} \\ \delta_f(q_0, a, a) & = \{(q_0, aa)\} \\ \delta_f(q_0, b, a) & = \{(q_1, a)\} \\ \delta_f(q_0, b, Z_0) & = \{(q_1, Z_0)\} \\ \delta_f(q_1, b, a) & = \{(q_1, a)\} \\ \delta_f(q_1, b, Z_0) & = \{(q_1, Z_0)\} \\ \delta_f(q_0, c, a) & = \{(q_2, \varepsilon)\} \\ \delta_f(q_1, c, a) & = \{(q_2, \varepsilon)\} \\ \delta_f(q_2, c, a) & = \{(q_2, \varepsilon)\} \\ \delta_f(q_0, \varepsilon, Z_0) & = \{(q_0, \varepsilon)\} \\ \delta_f(q_1, \varepsilon, Z_0) & = \{(q_1, \varepsilon)\} \\ \delta_f(q_2, \varepsilon, Z_0) & = \{(q_2, \varepsilon)\} \end{array}\right\} \quad \textit{transições equivalentes às de } \mathcal{M}_c \\ \delta_f(q, \varepsilon, U_0) = \{(q_f, U_0)\}, \; \textit{para qualquer } q \in \{q_0, q_1, q_2\} \end{array}$$

$\delta_f(q, x, X) \quad = \emptyset \qquad$ *nos restantes casos*

A proposição 2.11 permite concluir que $L_{EF}(\mathcal{M}_f) = L_c$.

A palavra a^3bc^3 *é aceite por* $\mathcal{M}_f$ *dado que:*

$$\begin{aligned}(q_0', a^3bc^3, U_0) &\underset{\mathcal{M}_f}{\vdash} (q_0, a^3bc^3, Z_0U_0)\\ &\underset{\mathcal{M}_f}{\vdash} (q_0, a^2bc^3, aZ_0U_0) \underset{\mathcal{M}_f}{\vdash} (q_0, abc^3, a^2Z_0U_0) \underset{\mathcal{M}_f}{\vdash} (q_0, bc^3, a^3Z_0U_0)\\ &\underset{\mathcal{M}_f}{\vdash} (q_1, c^3, a^3Z_0U_0 \underset{\mathcal{M}_f}{\vdash} (q_2, c^2, a^2Z_0U_0) \underset{\mathcal{M}_f}{\vdash} (q_2, c, aZ_0U_0)\\ &\underset{\mathcal{M}_f}{\vdash} (q_2, \varepsilon, Z_0U_0) \underset{\mathcal{M}_f}{\vdash} (q_2, \varepsilon, U_0)\\ &\underset{\mathcal{M}_f}{\vdash} (q_f, \varepsilon, U_0)\end{aligned}$$

Sugere-se a comparação desta derivação com a efetuada em 2.5 no caso do autómato $\mathcal{M}_c$.

Proposição 2.13 *Sejam* $L \subseteq A^*$ *e* $\mathcal{M}_1 = (Q_1, A, \Sigma_1, \delta_1, q_0, Z_0, F_1)$ *um autómato de pilha tais que* $L = L_{EF}(\mathcal{M}_1)$. *Então, existe um autómato de pilha* $\mathcal{M}$ *tal que*

$$L = L_{PV}(\mathcal{M})$$

Demonstração:

O objetivo é construir o autómato de pilha $\mathcal{M}$ a partir de $\mathcal{M}_1$ acrescentando novas transições que permitem esvaziar a pilha quando se atinge um estado final e na fita de leitura se tem a palavra vazia.

Considere-se $\mathcal{M} = (Q, A, \Sigma, \delta, q_0', U_0, \emptyset)$ tal que:

- $Q = Q_1 \cup \{q_0',\ q_e\}$, onde $q_0', q_e \notin Q_1$;
- $\Sigma = \Sigma_1 \cup \{U_0\}$, onde $U_0 \notin \Sigma_1$;
- $\delta : Q \times (A \cup \{\varepsilon\}) \times \Sigma \longrightarrow \mathcal{P}_{fin}(Q \times \Sigma^*)$ é definida por:

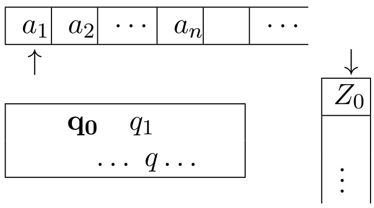
$$\begin{array}{ll}\delta(q_0', \varepsilon, U_0) = \{(q_0, Z_0U_0)\} & \\ \delta(q, a, X) = \delta_1(q, a, X) & \text{para quaisquer } q \in Q_1,\ a \in A,\ X \in \Sigma_1\\ \delta(q, \varepsilon, X) = \delta_1(q, \varepsilon, X) & \text{para quaisquer } q \in Q_1 \setminus F_1,\ X \in \Sigma_1\\ \delta(q, \varepsilon, X) = \delta_1(q, \varepsilon, X) \cup \{(q_e, X)\} & \text{para quaisquer } q \in F_1,\ X \in \Sigma_1\\ \delta(q_e, \varepsilon, X) = \{(q_e, \varepsilon)\} & \\ \delta(q, a, X) = \emptyset & \text{nos restantes casos}\end{array}$$

Seja $u \in L_{EF}(\mathcal{M}_1)$. Então, existem $q \in F_1$ e $\alpha \in \Sigma_1^*$ tais que

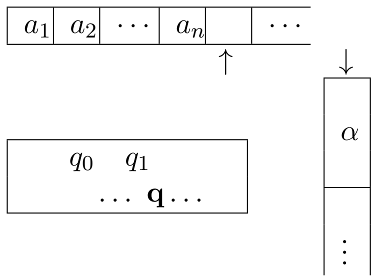
$$(q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}_1}{\vdash}} (q, \varepsilon, \alpha) \tag{2.6}$$

Na figura seguinte ilustra-se a situação inicial e final do autómato $\mathcal{M}_1$ no processo de reconhecer a palavra $u = a_1 \cdots a_n$.

Situação inicial



Situação final



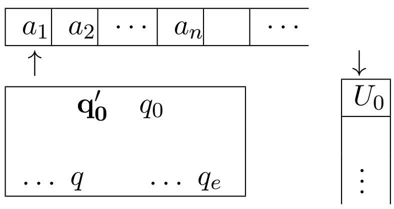
Como $\delta(q_0', \varepsilon, U_0) = \{(q_0, Z_0U_0)\}$ e $\delta(q, \varepsilon, X) = \{(q_e, X)\}$, então (2.6) implica

$$(q_0', u, U_0) \underset{\mathcal{M}}{\vdash} (q_0, u, Z_0U_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q, \varepsilon, \alpha U_0) \underset{\mathcal{M}}{\vdash} (q_e, \varepsilon, \alpha U_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q_e, \varepsilon, \varepsilon) \tag{2.7}$$

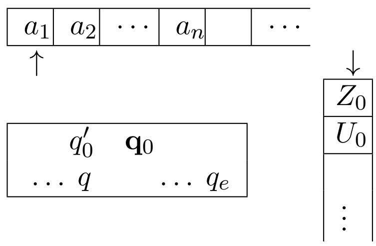
pelo que $u \in L_{PV}(\mathcal{M})$.

A sequência de representações do autómato $\mathcal{M}$ apresentada de seguida ilustra a sequência de transições (2.7).

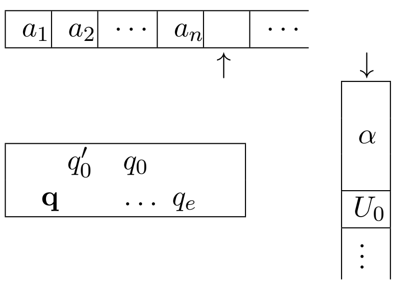
Situação inicial



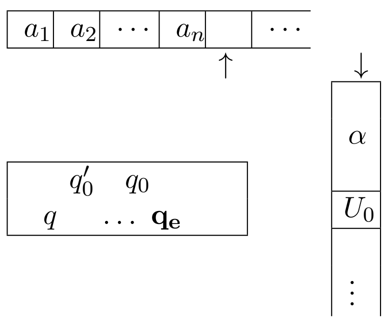
Final da primeira etapa



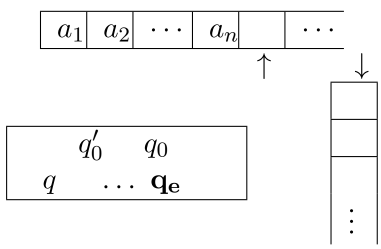
Final da etapa em que se atinge um estado $q \in F_1$ e se leu toda a palavra



Etapa seguinte



Situação final



Reciprocamente, se $u \in L_{PV}(\mathcal{M})$, então a última transição do autómato, após a leitura de u, conduz a uma configuração em que o estado é q_e e a pilha está vazia, ou seja,

$$(q_0', u, U_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q_e, \varepsilon, \varepsilon) \tag{2.8}$$

A definição de δ implica que:

1. a única configuração de $\mathcal{M}$ que engloba q_0' é a inicial,
2. depois de atingir uma configuração de estado q_e as transições do autómato têm por objetivo esvaziar a pilha e não é possível passar para outro qualquer estado,
3. as configurações de $\mathcal{M}$ que englobam q_e só ocorrem se na fita de leitura se considerar a leitura de ε.

Assim, 2.8 implica que existem $q \in F_1$ e $\alpha \in \Sigma_1^*$ tais que

$$(q_0', u, U_0) \underset{\mathcal{M}}{\vdash} (q_0, u, Z_0U_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q, \varepsilon, \alpha U_0) \underset{\mathcal{M}}{\vdash} (q_e, \varepsilon, \alpha U_0) \overset{*}{\underset{\mathcal{M}}{\vdash}} (q_e, \varepsilon, \varepsilon) \tag{2.9}$$

Consequentemente,

$$(q_0, u, Z_0) \overset{*}{\underset{\mathcal{M}_1}{\vdash}} (q, \varepsilon, \alpha)$$

ou seja, $u \in L_{EF}(\mathcal{M}_1)$. ■

Exemplo 2.14 *Recorde-se o autómato $\mathcal{M}_L$ definido no exemplo 2.10(2), que aceita as palavras do tipo $a^n b^m$, para $n \geq m$ e $n, m \in \mathbb{N}_0$, pelo critério de estados finais. Aplicando o processo descrito na demonstração da proposição 2.13, podemos construir um autómato equivalente, que aceita palavras com base no critério de pilha vazia:*

$$\mathcal{M}_{Lpv} = (\{q_0', q_0, q_1, q_e\}, \{a, b\}, \{U_0, Z_0, a\}, \delta, q_0', U_0, \emptyset)$$

em que, para $X \in \{Z_0, a\}$ e $x \in \{a, b\}$,

$$\begin{array}{lll}
\delta(q_0', \varepsilon, U_0) & = \{(q_0, Z_0U_0)\} & \} \textit{ transição inicial} \\
\delta(q_0, a, X) & = \delta_L(q_0, a, X) = \{(q_0, aX)\} & \\
\delta(q_0, \varepsilon, X) & = \delta_L(q_0, \varepsilon, X) = \{(q_1, X)\} & \} \textit{ transições equivalentes às de } \mathcal{M}_L \\
\delta(q_1, b, a) & = \delta_L(q_1, b, a) = \{(q_1, \varepsilon)\} & \\
\delta(q_1, \varepsilon, X) & = \{(q_e, X)\} & \} \textit{ transições para esvaziar a pilha} \\
\delta(q_e, \varepsilon, X) & = \{(q_e, \varepsilon)\} & \\
\delta(q, x, X) & = \emptyset & \textit{nos restantes casos}
\end{array}$$

De seguida apresentam-se algumas sequências de derivações.

$$\bullet (q_0', \varepsilon, U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, \varepsilon, Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_1, \varepsilon, Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, \varepsilon)$$

$$\bullet (q_0', b^2a^3, U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, b^2a^3, Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_1, b^2a^3, Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, b^2a^3, Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, b^2a^3, U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, b^2a^3, \varepsilon)$$

$$
\begin{aligned}
\bullet(q_0', a^4b^2, U_0) &\underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, a^4b^2, Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, a^3b^2, aZ_0U_0)\\
&\underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, a^2b^2, a^2Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, ab^2, a^3Z_0U_0)\\
&\underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, b^2, a^4Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_1, b^2, a^4Z_0U_0)\\
&\underset{\mathcal{M}_{Lpv}}{\vdash} (q_1, b, a^3Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_1, \varepsilon, a^2Z_0U_0)\\
&\underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, a^2Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, aZ_0U_0)\\
&\underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, \varepsilon)\\
\bullet(q_0', a^3, U_0) &\underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, a^3, Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, a^2, aZ_0U_0)\\
&\underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, a, a^2Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_0, \varepsilon, a^3Z_0U_0)\\
&\underset{\mathcal{M}_{Lpv}}{\vdash} (q_1, \varepsilon, a^3Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, a^3Z_0U_0)\\
&\underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, a^2Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, aZ_0U_0)\\
&\underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, Z_0U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, U_0) \underset{\mathcal{M}_{Lpv}}{\vdash} (q_e, \varepsilon, \varepsilon)
\end{aligned}
$$

Verifica-se pois que, tal como com o autómato $\mathcal{M}_L$, a^4b^2 e a^3 são aceites e que b^2a^3 não é aceite por $\mathcal{M}_{Lpv}$. E as palavras ab^2 e a^2ba são aceites? Compare com o resultado obtido em 2.10(2).

Como conclusão desta secção, e como consequência das proposições 2.11 e 2.13, pode-se afirmar que a classe das linguagens reconhecidas por autómatos de pilha segundo o critério de pilha vazia é igual à classe das linguagens reconhecidas por autómatos de pilha segundo o critério de estados finais, o que traduz a equivalência entre os dois critérios.

2.3 Autómatos de pilha e linguagens independentes de contexto

O objetivo desta secção é estabelecer a correspondência entre autómatos de pilha e linguagens independentes de contexto.

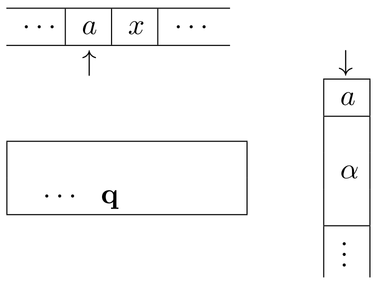
Será possível construir um autómato de pilha que reconheça uma dada linguagem L independente de contexto? E será possível construir uma gramática que gere a linguagem aceite por um dado autómato de pilha?

Seja $\mathcal{G} = (V, A, \mathcal{S}, P)$ uma GIC que gera L, isto é, $L = L(\mathcal{G})$. Recorde-se que as produções de P são do tipo

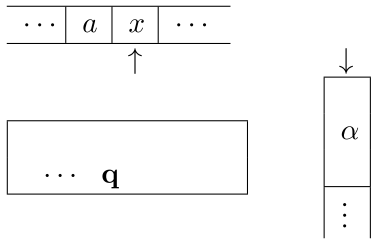
$$\mathcal{B} \longrightarrow \gamma$$

onde $\mathcal{B} \in V$ e $\gamma \in (V \cup A)^*$. Pretende-se definir um autómato em que as transições correspondam a derivações em $\mathcal{G}$, sendo que a pilha deve conter a informação sobre as produções. Deste modo, devem existir dois tipos de transições, que se descrevem abaixo:

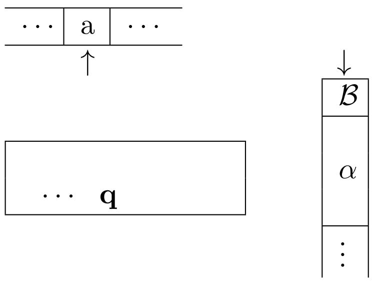
1. tirar um símbolo terminal do topo da pilha e se for igual ao símbolo lido na fita de leitura, avançar na fita de leitura, ou seja, de uma configuração do tipo



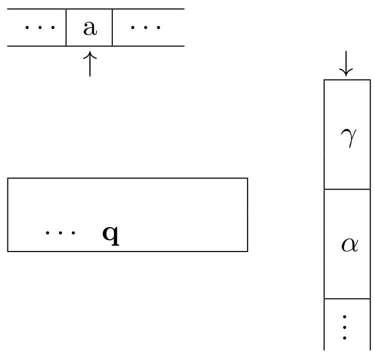
o autómato transita para



2. tirar uma variável $\mathcal{B}$ do topo da pilha e, se existir uma produção do tipo $\mathcal{B} \to \gamma$, substituir pelo membro direito da produção sem avançar na fita de leitura, ou seja, de uma configuração do tipo



o autómato transita para



Proposição 2.15 *Seja $\mathcal{G} = (V, A, \mathcal{S}, P)$ uma GIC. Então existe um autómato de pilha $\mathcal{M}$ tal que $L_{PV}(\mathcal{M}) = L(\mathcal{G})$.*

Demonstração:
Considere-se $\mathcal{M} = (Q, A, \Sigma, \delta, q_0, Z_0, \emptyset)$ um autómato de pilha tal que:

- $Q = \{q_0\}$;
- $\Sigma = V \cup A$;
- $Z_0 = \mathcal{S}$;
- $\delta : Q \times (A \cup \{\varepsilon\}) \times \Sigma \longrightarrow \mathcal{P}_{fin}(Q \times \Sigma^*)$ é definida por:

$$\begin{array}{l}\delta(q_0, \varepsilon, \mathcal{B}) = \{(q_0, \gamma) \mid \mathcal{B} \to \gamma \in P\} \\ \delta(q_0, a, a) = \{(q_0, \varepsilon)\} \text{ para qualquer } a \in A \\ \delta(q_0, a, Z) = \emptyset \text{ nos restantes casos}\end{array}$$

Para qualquer $\mathcal{B} \in V$ e $w \in A^*$, usando o Princípio de Indução Matemática, prova-se que:

i) se $\mathcal{B} \overset{m}{\underset{\mathcal{G}}{\Rightarrow}} w$, para $m \in \mathbb{N}$, então existe $n \in \mathbb{N}$ tal que $(q_0, w, \mathcal{B}) \overset{n}{\underset{\mathcal{M}}{\vdash}} (q_0, \varepsilon, \varepsilon)$;

ii) se $(q_0, w, \mathcal{B}) \overset{n}{\underset{\mathcal{M}}{\vdash}} (q_0, \varepsilon, \varepsilon)$, para $n \in \mathbb{N}$, então existe $m \in \mathbb{N}$ tal que $\mathcal{B} \overset{m}{\underset{\mathcal{G}}{\Rightarrow}} w$.

Em particular, $\mathcal{S} \overset{+}{\underset{\mathcal{G}}{\Rightarrow}} w$ se e só se $(q_0, w, \mathcal{S}) \overset{+}{\underset{\mathcal{M}}{\vdash}} (q_0, \varepsilon, \varepsilon)$, ou seja, $L_{PV}(\mathcal{M}) = L(\mathcal{G})$. ■

Exemplos 2.16

1. *Considere-se a gramática*

$$\mathcal{G}_o = (V, A, \mathcal{S}, P)$$

em que $V = \{\mathcal{S}, \mathcal{B}, \mathcal{C}, \mathcal{D}\}$, $A = \{0, \ldots, 9, (,), +, \times\}$ *e* P *é o seguinte conjunto de produções:*

$$\begin{array}{ll} \mathcal{S} & \longrightarrow \mathcal{S} + \mathcal{B} \mid \mathcal{B} \\ \mathcal{B} & \longrightarrow \mathcal{B} \times \mathcal{C} \mid \mathcal{C} \\ \mathcal{C} & \longrightarrow (\mathcal{S}) \mid \mathcal{D} \\ \mathcal{D} & \longrightarrow 0 \mid 1 \mid \ \ldots \ \mid 9 \end{array}$$

Um autómato de pilha que reconhece $L(\mathcal{G}_o)$ *pelo critério de pilha vazia é*

$$\mathcal{M}_o = (\{q_0\}, A, \Sigma, \delta_o, q_0, \mathcal{S}, \emptyset)$$

tal que:

- $\Sigma = \{\mathcal{S}, \mathcal{B}, \mathcal{C}, \mathcal{D}, 0, \ldots, 9, (,), +, \times\}$;
- $\delta_o : \{q_0\} \times (A \cup \{\varepsilon\}) \times \Sigma \longrightarrow \mathcal{P}_{fin}(Q \times \Sigma^*)$ *é definida por:*

$$\begin{array}{l} \delta_o(q_0, \varepsilon, \mathcal{S}) = \{(q_0, \mathcal{S} + \mathcal{B}),\ (q_0, \mathcal{B})\} \\ \delta_o(q_0, \varepsilon, \mathcal{B}) = \{(q_0, \mathcal{B} \times \mathcal{C}),\ (q_0, \mathcal{C})\} \\ \delta_o(q_0, \varepsilon, \mathcal{C}) = \{(q_0, (\mathcal{S})),\ (q_0, \mathcal{D})\} \\ \delta_o(q_0, \varepsilon, \mathcal{D}) = \{(q_0, 0),\ \ldots\ , (q_0, 9)\} \\ \delta_o(q_0, x, x) = \{(q_0, \varepsilon)\} \ \ \textit{para qualquer } x \in A \\ \delta_o(q_0, x, Z) = \emptyset \ \ \textit{nos restantes casos} \end{array}$$

A palavra 3×8 *é gerada por esta gramática?*

$$\mathcal{S} \underset{\mathcal{G}_o}{\Rightarrow} \mathcal{B} \underset{\mathcal{G}_o}{\Rightarrow} \mathcal{B} \times \mathcal{C} \underset{\mathcal{G}_o}{\Rightarrow} \mathcal{C} \times \mathcal{C} \underset{\mathcal{G}_o}{\Rightarrow} \mathcal{D} \times \mathcal{C} \underset{\mathcal{G}_o}{\Rightarrow} 3 \times \mathcal{C} \underset{\mathcal{G}_o}{\Rightarrow} 3 \times \mathcal{D} \underset{\mathcal{G}_o}{\Rightarrow} 3 \times 8$$

A resposta é afirmativa e consequentemente o autómato $\mathcal{M}_o$ *aceita a palavra. A sequência de transições a efetuar até aceitar a palavra* 3×8 *é a seguinte:*

$$\begin{aligned}(q_0, 3\times 8, \mathcal{S}) &\underset{\mathcal{M}_o}{\vdash} (q_0, 3\times 8, \mathcal{B}) \underset{\mathcal{M}_o}{\vdash} (q_0, 3\times 8, \mathcal{B}\times\mathcal{C}) \underset{\mathcal{M}_o}{\vdash} (q_0, 3\times 8, \mathcal{C}\times\mathcal{C})\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, 3\times 8, \mathcal{D}\times\mathcal{C}) \underset{\mathcal{M}_o}{\vdash} (q_0, 3\times 8, 3\times\mathcal{C}) \underset{\mathcal{M}_o}{\vdash} (q_0, \times 8, \times\mathcal{C})\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, 8, \mathcal{C}) \underset{\mathcal{M}_o}{\vdash} (q_0, 8, \mathcal{D}) \underset{\mathcal{M}_o}{\vdash} (q_0, 8, 8)\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, \varepsilon, \varepsilon)\end{aligned}$$

Analisemos agora uma derivação da palavra $(1+(2+3)\times 5)$

$$\begin{aligned}\mathcal{S} &\underset{\mathcal{G}_o}{\Rightarrow} \mathcal{B} \underset{\mathcal{G}_o}{\Rightarrow} \mathcal{C} \underset{\mathcal{G}_o}{\Rightarrow} (\mathcal{S}) \underset{\mathcal{G}_o}{\Rightarrow} (\mathcal{S}+\mathcal{B}) \underset{\mathcal{G}_o}{\Rightarrow} (\mathcal{B}+\mathcal{B}) \underset{\mathcal{G}_o}{\Rightarrow} (\mathcal{C}+\mathcal{B})\\ &\underset{\mathcal{G}_o}{\Rightarrow} (\mathcal{D}+\mathcal{B}) \underset{\mathcal{G}_o}{\Rightarrow} (1+\mathcal{B}) \underset{\mathcal{G}_o}{\Rightarrow} (1+\mathcal{B}\times\mathcal{C}) \underset{\mathcal{G}_o}{\Rightarrow} (1+\mathcal{C}\times\mathcal{C}) \underset{\mathcal{G}_o}{\Rightarrow} (1+(\mathcal{S})\times\mathcal{C})\\ &\underset{\mathcal{G}_o}{\Rightarrow} (1+(\mathcal{S}+\mathcal{B})\times\mathcal{C}) \underset{\mathcal{G}_o}{\Rightarrow} (1+(\mathcal{B}+\mathcal{B})\times\mathcal{C}) \underset{\mathcal{G}_o}{\Rightarrow} (1+(\mathcal{C}+\mathcal{B})\times\mathcal{C})\\ &\underset{\mathcal{G}_o}{\Rightarrow} (1+(\mathcal{D}+\mathcal{B})\times\mathcal{C}) \underset{\mathcal{G}_o}{\Rightarrow} (1+(2+\mathcal{B})\times\mathcal{C}) \underset{\mathcal{G}_o}{\Rightarrow} (1+(2+\mathcal{C})\times\mathcal{C})\\ &\underset{\mathcal{G}_o}{\Rightarrow} (1+(2+\mathcal{D})\times\mathcal{C}) \underset{\mathcal{G}_o}{\Rightarrow} (1+(2+3)\times\mathcal{C}) \underset{\mathcal{G}_o}{\Rightarrow} (1+(2+3)\times\mathcal{D})\\ &\underset{\mathcal{G}_o}{\Rightarrow} (1+(2+3)\times 5)\end{aligned}$$

A sequência de transições do autómato $\mathcal{M}_o$ *até aceitar a palavra* $(1+(2+3)\times 5)$ *é a seguinte:*

$$\begin{aligned}(q_0,\ &(1+(2+3)\times 5), \mathcal{S}) \underset{\mathcal{M}_o}{\vdash} (q_0, (1+(2+3)\times 5), \mathcal{B}) \underset{\mathcal{M}_o}{\vdash} ((q_0, (1+(2+3)\times 5), \mathcal{C})\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, (1+(2+3)\times 5), (\mathcal{S})) \underset{\mathcal{M}_o}{\vdash} (q_0, 1+(2+3)\times 5), \mathcal{S}))\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, 1+(2+3)\times 5), \mathcal{S}+\mathcal{B})) \underset{\mathcal{M}_o}{\vdash} (q_0, 1+(2+3)\times 5), \mathcal{B}+\mathcal{B}))\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, 1+(2+3)\times 5), \mathcal{C}+\mathcal{B})) \underset{\mathcal{M}_o}{\vdash} (q_0, 1+(2+3)\times 5), \mathcal{D}+\mathcal{B}))\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, 1+(2+3)\times 5), 1+\mathcal{B}) \underset{\mathcal{M}_o}{\vdash} (q_0, +(2+3)\times 5), +\mathcal{B}))\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, (2+3)\times 5), \mathcal{B}) \underset{\mathcal{M}_o}{\vdash} (q_0, (2+3)\times 5), \mathcal{B}\times\mathcal{C}) \underset{\mathcal{M}_o}{\vdash} (q_0, (2+3)\times 5), \mathcal{C}\times\mathcal{C})\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, (2+3)\times 5), (\mathcal{S})\times\mathcal{C}) \underset{\mathcal{M}_o}{\vdash} (q_0, 2+3)\times 5), \mathcal{S})\times\mathcal{C})\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, 2+3)\times 5), \mathcal{S}+\mathcal{B})\times\mathcal{C}) \underset{\mathcal{M}_o}{\vdash} (q_0, 2+3)\times 5), \mathcal{B}+\mathcal{B})\times\mathcal{C}))\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, 2+3)\times 5), \mathcal{C}+\mathcal{B})\times\mathcal{C})) \underset{\mathcal{M}_o}{\vdash} (q_0, 2+3)\times 5), \mathcal{D}+\mathcal{B})\times\mathcal{C}))\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, 2+3)\times 5), 2+\mathcal{B})\times\mathcal{C})) \underset{\mathcal{M}_o}{\vdash} (q_0, +3)\times 5), +\mathcal{B})\times\mathcal{C}))\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, 3)\times 5), \mathcal{B})\times\mathcal{C})) \underset{\mathcal{M}_o}{\vdash} (q_0, 3)\times 5), \mathcal{C})\times\mathcal{C})) \underset{\mathcal{M}_o}{\vdash} (q_0, 3)\times 5), \mathcal{D})\times\mathcal{C}))\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, 3)\times 5), 3)\times\mathcal{C})) \underset{\mathcal{M}_o}{\vdash} (q_0,)\times 5),)\times\mathcal{C}))\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, \times 5), \times\mathcal{C})) \underset{\mathcal{M}_o}{\vdash} (q_0, 5), \mathcal{C})) \underset{\mathcal{M}_o}{\vdash} (q_0, 5), \mathcal{D})) \underset{\mathcal{M}_o}{\vdash} (q_0, 5), 5)) \underset{\mathcal{M}_o}{\vdash} (q_0,),))\\ &\underset{\mathcal{M}_o}{\vdash} (q_0, \varepsilon, \varepsilon)\end{aligned}$$

2. *Considere-se a linguagem* $L_c = \{a^i b^j c^i \mid i, j \in \mathbb{N}_0\}$ *estudada em 2.12. Uma gramática que gera esta linguagem é*

$$\mathcal{G}_\diamond = (\{\mathcal{S}, \mathcal{B}\}, \{a, b, c\}, \mathcal{S}, P)$$

em que P *é o seguinte conjunto de produções:*

$$\begin{array}{ll} \mathcal{S} & \longrightarrow a\mathcal{S}c \mid \mathcal{B} \mid \varepsilon \\ \mathcal{B} & \longrightarrow b\mathcal{B} \mid b \end{array}$$

Recorrendo à proposição 2.15, calcula-se um autómato de pilha que reconhece $L(\mathcal{G}_\diamond)$ *pelo critério de pilha vazia:*

$$\mathcal{M}_\diamond = (\{q_0\}, \{a, b, c\}, \{\mathcal{S}, \mathcal{B}, a, a\}, \delta_\diamond, q_0, \mathcal{S}, \emptyset)$$

tal que:

$$\begin{array}{l} \delta_\diamond(q_0, \varepsilon, \mathcal{S}) = \{(q_0, a\mathcal{S}c),\ (q_0, \mathcal{B}),\ (q_0, \varepsilon)\} \\ \delta_\diamond(q_0, \varepsilon, \mathcal{B}) = \{(q_0, b\mathcal{B}),\ (q_0, b)\} \\ \delta_\diamond(q_0, x, x) = \{(q_0, \varepsilon)\} \ \ \textit{para qualquer } x \in \{a, b, c\} \\ \delta_\diamond(q_0, x, X) = \emptyset \ \ \textit{nos restantes casos} \end{array}$$

Considere-se a palavra a^3bc^3*.* $\mathcal{M}_\diamond$ *aceita esta palavra:*

$$\begin{array}{ll} (q_0, a^3bc^3, \mathcal{S}) & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, a^3bc^3, a\mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, a^2bc^3, \mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, a^2bc^3, a\mathcal{S}c^2) \\ & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^3, \mathcal{S}c^2) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^3, a\mathcal{S}c^3) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^3, \mathcal{S}c) \\ & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^3, \mathcal{B}c^3) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^3, bc^3) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, c^3, c^3) \\ & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, c^2, c^2) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, c, c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, \varepsilon, \varepsilon) \end{array}$$

A palavra abc^2 *não é aceite, pois não existe uma sequência de derivações que conduza a* $(q_0, \varepsilon, \varepsilon)$*.*

$$\begin{array}{ll} (q_0, abc^2, \mathcal{S}) & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^2, a\mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, \mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, \mathcal{B}c) \\ & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, bc) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, c^2, c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, c, \varepsilon) \\ (q_0, abc^2, \mathcal{S}) & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^2, a\mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, \mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, \mathcal{B}c) \\ & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, b\mathcal{B}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, c^2, \mathcal{B}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, c^2, b\mathcal{B}c) \\ (q_0, abc^2, \mathcal{S}) & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^2, a\mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, \mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, \mathcal{B}c) \\ & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, b\mathcal{B}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, c^2, \mathcal{B}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, c^2, bc) \\ (q_0, abc^2, \mathcal{S}) & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^2, a\mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, \mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, \varepsilon c) \\ (q_0, abc^2, \mathcal{S}) & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^2, a\mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, \mathcal{S}c) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, bc^2, a\mathcal{S}c^2) \\ (q_0, abc^2, \mathcal{S}) & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^2, \varepsilon) \\ (q_0, abc^2, \mathcal{S}) & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^2, \mathcal{B}) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^2, b) \\ (q_0, abc^2, \mathcal{S}) & \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^2, \mathcal{B}) \underset{\mathcal{M}_\diamond}{\vdash} (q_0, abc^2, b\mathcal{B}) \end{array}$$

Sugere-se que compare o comportamento de $\mathcal{M}_\diamond$ *com o comportamento do autómato de pilha* $\mathcal{M}_c$ *definido em 2.12.*

Reciprocamente, pretende-se a partir de um autómato de pilha $\mathcal{M}$ definir uma GIC $\mathcal{G}$ que gere a linguagem reconhecida pelo autómato. Admitindo que o critério utilizado pelo autómato era o de pilha vazia, poderíamos por simplicidade pensar num processo inverso ao processo descrito na proposição 2.15. Desse modo o conjunto das variáveis corresponderia ao alfabeto da pilha, e as produções seriam obtidas a partir das transições do autómato. No entanto tal abordagem não é correta.

As variáveis da GIC a construir serão definidas com base nos símbolos da pilha e nos estados. Sendo $\mathcal{M} = (Q, A, \Sigma, \delta, q_0, Z_0, F)$ uma variável de $\mathcal{G} = (V, A, \mathcal{S}, P)$, diferente de $\mathcal{S}$, será do tipo:

$$[p, X, q]$$

para $p, q \in Q$ e $X \in \Sigma$. Neste contexto, existem produções dos seguintes tipos:

1. $[p, X, q] \longrightarrow a$, para $a \in A \cup \{\varepsilon\}$, se $(q, \varepsilon) \in \delta(p, a, X)$, ou seja, se o autómato admite uma transição em que passa do estado p para o estado q, lê a e retira o símbolo X;

2. $[p, X, q] \longrightarrow a[p_1, X_1, p_2][p_2, X_2, p_3]\ldots[p_m, X_m, q]$, com $a \in A \cup \{\varepsilon\}$, $p_1, p_2, \ldots, p_m \in Q$ se $(p_1, X_1 \ldots X_m) \in \delta(p, a, X)$, ou seja, se o autómato admite uma transição em que passa do estado p para o estado p_1, lê a e substitui o símbolo X do topo da pilha por $X_1 \ldots X_m$;

3. $\mathcal{S} \longrightarrow [q_0, Z_0, q]$, para todo o estado q (estas produções permitem iniciar qualquer derivação).

Proposição 2.17 *Seja $\mathcal{M} = (Q, A, \Sigma, \delta, q_0, Z_0, \emptyset)$ um autómato de pilha. Então existe uma GIC $\mathcal{G}$ tal que $L_{PV}(\mathcal{M}) = L(\mathcal{G})$.*

Demonstração:
Considere-se $\mathcal{G} = (V, A, \mathcal{S}, P)$ uma gramática definida por:

1. $\mathcal{S}$ é um novo símbolo;

2. $V = \{\mathcal{S}\} \cup \{[p, X, q] \mid X \in \Sigma,\ p, q \in Q\}$;

3. P é constituído pelas produções :

$$\begin{aligned}
\mathcal{S} &\longrightarrow [q_0, Z_0, q], \quad \text{para qualquer } q \in Q \\
[p, X, q] &\longrightarrow a, \quad p, q \in Q,\ X \in \Sigma,\ a \in A \cup \{\varepsilon\},\ (q, \varepsilon) \in \delta(p, a, X) \\
[p, X, q] &\longrightarrow a[p_1, X_1, p_2][p_2, X_2, p_3]\ldots[p_m, X_m, q], \quad m \geq 1,\ p, q \in Q, \\
&\qquad a \in A \cup \{\varepsilon\}, X_1, \ldots X_m \in \Sigma, p_1, p_2, \ldots, p_m \in Q \\
&\qquad \text{tal que } (p_1, X_1 \ldots X_m) \in \delta(p, a, X)
\end{aligned}$$

A demonstração ficaria concluída com a verificação de que $\mathcal{G}$ gera a linguagem reconhecida pelo autómato. ■

Exemplo 2.18 *Recorde-se a linguagem $L = \{u \in A^* \mid u = xcx^{\mathrm{I}},\ x \in A^*\}$ onde $A = \{a, b, c\}$, introduzida em 2.1. Um autómato que reconhece L pelo critério de pilha vazia é:*

$$\mathcal{M} = (\{q_0, q_1\}, \{a, b, c\}, \{Z_0, A_0, B_0\}, \delta, q_0, Z_0, \emptyset)$$

em que δ é definida por:

$$\begin{aligned}
\delta(q_0, a, Z_0) &= \{(q_0, A_0Z_0)\}\\
\delta(q_0, b, Z_0) &= \{(q_0, B_0Z_0)\}\\
\delta(q_0, c, Z_0) &= \{(q_1, Z_0)\}\\
\delta(q_0, a, A_0) &= \{(q_0, A_0A_0)\}\\
\delta(q_0, b, A_0) &= \{(q_0, B_0A_0)\}\\
\delta(q_0, a, B_0) &= \{(q_0, A_0B_0)\}\\
\delta(q_0, b, B_0) &= \{(q_0, B_0B_0)\}\\
\delta(q_0, c, A_0) &= \{(q_1, A_0)\}\\
\delta(q_0, c, B_0) &= \{(q_1, B_0)\}\\
\delta(q_1, a, A_0) &= \{(q_1, \varepsilon)\}\\
\delta(q_1, b, B_0) &= \{(q_1, \varepsilon)\}\\
\delta(q_1, \varepsilon, Z_0) &= \{(q_1, \varepsilon)\}
\end{aligned}$$

ou seja, $L = L_{PV}(\mathcal{M})$. Escolheu-se para alfabeto de pilha o conjunto $\{Z_0, A_0, B_0\}$ (em vez de $\{Z_0, a, b\}$) de modo a ser disjunto do alfabeto A, para tornar o texto mais fácil de ler.

De acordo com o processo descrito em 2.17 uma gramática que gera L é $\mathcal{G} = (V, A, \mathcal{S}, P)$ definida por:

1. *$\mathcal{S}$ é um novo símbolo;*
2. *$V = \{\mathcal{S}\} \cup \{[p, X, q] \mid X \in \{Z_0, A_0, B_0\},\ p, q \in \{q_0, q_1\}\}$;*
3. *P é constituído pelas produções :*

$$\begin{array}{ll}
\mathcal{S} & \longrightarrow [q_0, Z_0, q_0]\\
\mathcal{S} & \longrightarrow [q_0, Z_0, q_1]\\
[q_1, A_0, q_1] & \longrightarrow a\\
[q_1, B_0, q_1] & \longrightarrow b\\
[q_1, Z_0, q_1] & \longrightarrow \varepsilon\\
[q_0, Z_0, q] & \longrightarrow a[q_0, A_0, p][p, Z_0, q]\\
[q_0, Z_0, q] & \longrightarrow b[q_0, B_0, p][p, Z_0, q]\\
[q_0, Z_0, q] & \longrightarrow c[q_1, Z_0, q]\\
[q_0, A_0, q] & \longrightarrow a[q_0, A_0, p][p, A_0, q]\\
[q_0, A_0, q] & \longrightarrow b[q_0, B_0, p][p, A_0, q]\\
[q_0, B_0, q] & \longrightarrow a[q_0, A_0, p][p, B_0, q]\\
[q_0, B_0, q] & \longrightarrow b[q_0, B_0, p][p, B_0, q]\\
[q_0, A_0, q] & \longrightarrow c[q_1, A_0, q]\\
[q_0, B_0, q] & \longrightarrow c[q_1, B_0, q]
\end{array}$$

   para quaisquer $p, q \in \{q_0, q_1\}$.

Considere-se a palavra bacab. O autómato $\mathcal{M}$ aceita a palavra pois

$$\begin{aligned}
(q_0, bacab, Z_0) &\underset{\mathcal{M}}{\vdash} (q_0, acab, B_0Z_0) \underset{\mathcal{M}}{\vdash} (q_0, cab, A_0B_0Z_0)\\
&\underset{\mathcal{M}}{\vdash} (q_1, ab, A_0B_0Z_0) \underset{\mathcal{M}}{\vdash} (q_1, b, B_0Z_0) \underset{\mathcal{M}}{\vdash} (q_1, \varepsilon, Z_0) \underset{\mathcal{M}}{\vdash} (q_1, \varepsilon, \varepsilon)
\end{aligned}$$

A correspondente derivação na gramática $\mathcal{G}$ é:

$$\begin{aligned}\mathcal{S} &\underset{\mathcal{G}}{\Rightarrow} [q_0, Z_0, q_1]\\ &\underset{\mathcal{G}}{\Rightarrow} b[q_0, B_0, q_1][q_1, Z_0, q_1]\\ &\underset{\mathcal{G}}{\Rightarrow} ba[q_0, A_0, q_1][q_1, B_0, q_1][q_1, Z_0, q_1]\\ &\underset{\mathcal{G}}{\Rightarrow} bac[q_1, A_0, q_1][q_1, B_0, q_1][q_1, Z_0, q_1]\\ &\underset{\mathcal{G}}{\Rightarrow} baca[q_1, B_0, q_1][q_1, Z_0, q_1]\\ &\underset{\mathcal{G}}{\Rightarrow} bacab[q_1, Z_0, q_1]\\ &\underset{\mathcal{G}}{\Rightarrow} bacab \qquad\qquad .\end{aligned}$$

2.4 Exercícios

1. Sejam $A = \{a, b\}$ um alfabeto e

$$\begin{aligned}L_1 &= \{xx^{\mathrm{I}} \ : \ x \in A^+\}\\ L_2 &= \{x \in A^* \ : \ x^{\mathrm{I}} = x \wedge |x| \text{ é ímpar}\}\\ L_3 &= \{x \in A^* \ : \ x^{\mathrm{I}} = x\}\end{aligned}$$

   (a) Determine um autómato de pilha $\mathcal{M}_i$ que reconheça a linguagem L_i, pelo critério de pilha vazia, para $i = 1, 2, 3$.

   (b) Calcule as sequências de movimentos possíveis

      i. para $\mathcal{M}_1$, a partir de $(q_0, aabbaa, Z_0)$, de $(q_0, ababa, Z_0)$ e de $(q_0, bbbb, Z_0)$.

      ii. para $\mathcal{M}_2$, a partir de $(q_0, aabbaa, Z_0)$, de $(q_0, aababaa, Z_0)$ e de $(q_0, aababa, Z_0)$.

      iii. para $\mathcal{M}_3$, a partir de $(q_0, aabbaa, Z_0)$, de $(q_0, aababaa, Z_0)$ e de $(q_0, ababab, Z_0)$.

   (c) Interprete os resultados obtidos na alínea anterior.

2. Seja $A = \{a, b\}$ um alfabeto e, em cada um dos casos seguintes, indique um autómato de pilha que reconheça a linguagem:

   (a) $\{x \in \{a, b\}^* \ : \ |x|_a = |x|_b\}$.

   (b) $\{a^i b^j \ : \ 0 \leq i \leq j\}$.

3. Considere a linguagem $L = \{x \in \{a, b\}^* : |u|_a > |u|_b\}$.

   (a) Indique um autómato de pilha que reconheça a linguagem L.

   (b) Calcule as sequências de movimentos possíveis a partir de:

      (i) $(q_0, baaba, Z_0)$; (ii) $(q_0, aabba, Z_0)$; (iii) $(q_0, abbba, Z_0)$.

4. Considere a linguagem $L = \{a^n b^{2n} \ : \ n \geq 0\}$.

   (a) Indique um autómato de pilha determinista que reconheça a linguagem L.

   (b) Calcule as sequências de movimentos possíveis a partir de:

      (i) $(q_0, babbba, Z_0)$; (ii) $(q_0, aabbbbbba, Z_0)$; (iii) $(q_0, aabbbb, Z_0)$.

      (iv) $(q_0, aaabbbb, Z_0)$; (v) $(q_0, aaaabb, Z_0)$; (vi) (q_0, ε, Z_0).

5. Indique tabelas de transição para autómatos de pilha que reconheçam cada uma das linguagens seguintes:

   (a) A linguagem de todas as palavras sobre $\{a, b\}$ que não são capícuas.

   (b) A linguagem $\{a^n x : n \geq 0 \text{ e } |x| \leq n\}$ sobre o alfabeto $\{a, b\}$.

6. As tabelas de transição seguintes descrevem a função de transição de autómatos de pilha com estado inicial q_0 e um único estado de aceitação q_2. Em cada caso, descreva a linguagem aceite pelo critério de estados finais.

   (a)

Nº movimento	Estado	Entrada	Símbolo da pilha	Movimento(s)
1	q_0	a	Z_0	(q_1, aZ_0)
2	q_0	b	Z_0	(q_1, bZ_0)
3	q_1	a	a	$(q_1, a), (q_2, a)$
4	q_1	b	a	(q_1, a)
5	q_1	a	b	(q_1, b)
6	q_1	b	b	$(q_1, b), (q_2, b)$
	(todas as outras combinações)			nenhum

   (b)

Nº movimento	Estado	Entrada	Símbolo da pilha	Movimento(s)
1	q_0	a	Z_0	(q_0, XZ_0)
2	q_0	b	Z_0	(q_0, XZ_0)
3	q_0	a	X	(q_0, XX)
4	q_0	b	X	(q_0, XX)
5	q_0	c	X	(q_1, X)
6	q_0	c	Z_0	(q_1, Z_0)
7	q_1	a	X	(q_1, ε)
8	q_1	b	X	(q_1, ε)
9	q_1	ε	Z_0	(q_2, Z_0)
	(todas as outras combinações)			nenhum

7. Considere o autómato de pilha $\mathcal{M} = (\{q_o, q_1\},\ \{a, b, c\},\ \{Z_0, b\},\ \delta,\ q_0,\ Z_0,\ \emptyset)$ tal que:

$$
\begin{array}{ll}
1 - \ \delta(q_0, a, Z_0) = \{(q_0, Z_0)\} & 2 - \ \delta(q_0, b, Z_0) = \{(q_0, bZ_0)\} \\
3 - \ \delta(q_0, c, Z_0) = \{(q_1, Z_0)\} & 4 - \ \delta(q_0, b, b) = \{(q_o, bb)\} \\
5 - \ \delta(q_0, c, b) = \{(q_1, \varepsilon)\} & 6 - \ \delta(q_1, c, Z_0) = \{(q_1, Z_0)\} \\
7 - \ \delta(q_1, c, b) = \{(q_1, \varepsilon)\} & 8 - \ \delta(q_1, \varepsilon, Z_0) = \{(q_1, \varepsilon)\}
\end{array}
$$

   que reconhece palavras utilizando o critério de pilha vazia.

   (a) Verifique que $a^2b^3c^6 \in L_{pv}(\mathcal{M})$ e que $b^4c^2 \notin L_{pv}(\mathcal{M})$.

   (b) Verifique que $L_{pv}(\mathcal{M}) = \{a^i b^j c^k \mid 0 \leq j \leq k,\ i, j, k \in \mathbb{N}_0\}$.

   (c) Determine um autómato de pilha $\mathcal{M}'$ equivalente a $\mathcal{M}$ que reconheça palavras utilizando o critério de estados finais.

(d) Verifique se os autómatos de pilha $\mathcal{M}$ ou $\mathcal{M}'$ são deterministas.

8. Em cada uma das alíneas seguintes (de 8a a 8e) é indicada uma gramática independente de contexto e uma palavra x gerada pela gramática. Para cada um dos casos determine:

   i. o autómato de pilha que reconhece $L(G)$, seguindo a demonstração da proposição 2.15;
   ii. uma sequência de movimentos do autómato construído na alínea a), que permita concluir que $x \in L(G)$, indicando, em cada passo, a configuração do autómato. Simultaneamente, indique a derivação mais à esquerda da palavra x na gramática G.

   (a) A gramática $G = (\{S\}, \{a, b\}, S, P)$ com produções

   $$S \to aSb \mid aaSb \mid \varepsilon$$

   e $x = aaabb$.

   (b) A gramática $G = (\{S\}, \{[,]\}, S, P)$ com produções

   $$S \to SS \mid [S] \mid \varepsilon$$

   e $x = [\,]\,[\,[\,]\,[\,]\,]$.

   (c) A gramática $G = (\{S, A, B\}, \{a, b\}, S, P)$ com produções

   $$\begin{array}{lcl} S & \to & A \mid B \\ A & \to & aAb \mid ab \\ B & \to & abB \mid \varepsilon \end{array}$$

   e $x = abab$.

   (d) A gramática $G = (\{S, A, B\}, \{a, b\}, S, P)$ com produções

   $$\begin{array}{lcl} S & \to & ABA \\ A & \to & aA \mid \varepsilon \\ B & \to & bB \mid \varepsilon \end{array}$$

   e $x = abaa$.

   (e) A gramática $G = (\{S, T, F\}, \{a, (,), +, *\}, S, P)$ com produções

   $$\begin{array}{lcl} S & \to & S + T \mid T \\ T & \to & T * F \mid F \\ F & \to & (S) \mid a \end{array}$$

   e $x = (a + a * a) * a$.

9. Seja $\mathcal{G} = (\{\mathcal{S}\}, \{(,), [,]\}, \mathcal{S}, P)$ a gramática com produções:

   $$\mathcal{S} \to \mathcal{S}\mathcal{S} \mid (\mathcal{S}) \mid [\mathcal{S}] \mid \varepsilon$$

   (a) Verifique se as palavras ([( )]), [ ] ([ ]) ou ([ )] ( ) são geradas pela gramática $\mathcal{G}$.

(b) Determine um autómato de pilha que reconheça $L(G)$.

(c) Calcule as sequências de movimentos possíveis a partir de:

i) $(q_0, (\ [\ (\ )\ ]\ ), Z_0)$; ii) $(q_0, [\ ]\ (\ [\ ]\ ), Z_0)$; iii) $(q_0, (\ [\ )\ ]\ (\ ), Z_0)$.

10. Calcule as gramáticas independentes de contexto correspondentes aos autómatos calculados nos exercícios 1, 2 e 3, seguindo a demonstração da proposição 2.17.

11. Considere a gramática $G = (V, \{a,\ b,\ c\},\ S,\ P)$ em que $V = \{S, B, C\}$ e P é definido por:

$$\begin{array}{lcl} S & \to & BC \\ B & \to & aBb \mid \varepsilon \\ C & \to & bCc \mid \varepsilon \end{array}$$

(a) Verifique que $a^3b^5c^2$ é uma palavra gerada por G e elabore a respetiva árvore de derivação.

(b) Determine a linguagem gerada por G, $L(G)$.

(c) Determine um autómato de pilha $\mathcal{M}$ que reconheça $L(G)$ utilizando o critério de estados finais.

(d) Determine sequências de transições que permitem verificar se $a^2b^6a^2c^2$ ou a^3b^4c são aceites por $\mathcal{M}$.

12. Considere a gramática $G = (V,\ A,\ S,\ P)$ definida por:

$$V = \{S, B, C, D\} \qquad\qquad A = \{b,\ c,\ d\}$$

$$\begin{array}{lcl} S & \to & BC \\ B & \to & bB \mid \varepsilon \\ C & \to & cCd \mid D \\ D & \to & Dd \mid d \end{array}$$

(a) Verifique se $b^3c^2d^3,\ c^2d^3b^3\ \in L(G)$.

(b) Determine a linguagem $L(G)$.

(c) Determine um autómato de pilha $\mathcal{M}$ que reconheça $L(G)$ utilizando o critério de pilha vazia.

(d) Verifique que se $u \in L_{pv}(\mathcal{M})$ então $b^n u \in L_{pv}(\mathcal{M})$, para qualquer $n \in \mathbb{N}$.

13. Considere a linguagem $L = (abb \cup b)^*(ab)^*$ sobre o alfabeto $\{a, b\}$.

(a) Construa um autómato finito que reconheça L.

(b) Construa uma gramática regular que gere L.

(c) Determine um autómato de pilha $\mathcal{M}$ que reconhece L pelo critério de estados finais.

14. Considere a gramática $G = (V,\ A,\ S,\ P)$ definida por:

$$V = \{S, B, B_0, B_1, C, C_0, C_1\} \qquad\qquad A = \{a,\ b,\ c\}$$

$$\begin{array}{lcl} S & \to & B \mid C \\ B & \to & B_0B_1 \\ B_0 & \to & aB_0b \mid \varepsilon \\ B_1 & \to & B_1c \mid \varepsilon \\ C & \to & C_0C_1 \\ C_0 & \to & aC_0 \mid \varepsilon \\ C_1 & \to & bC_1c \mid \varepsilon \end{array}$$

(a) Verifique que $a^3b^3c^2$ e $a^3b^2c^2$ são geradas por G.

(b) Determine a linguagem $L(G)$.

(c) Determine um autómato de pilha $\mathcal{M}$ que reconheça $L(G)$ utilizando o critério de pilha vazia.

(d) Verifique se o autómato de pilha $\mathcal{M}$ é determinista.

15. Considere a linguagem $L = \{a^ib^jc^k \mid i = j + k,\ i, j, k \in \mathbb{N}_0\}$ sobre o alfabeto $\{a, b, c\}$.

(a) Determine uma gramática que gere L.

(b) Com base na gramática definida, verifique se $a^4b^3c \in L$ ou $a^3b^4c \in L$ e, nos casos de resposta afirmativa, elabore a respetiva árvore de derivação.

(c) Determine um autómato de pilha $\mathcal{M}$ que reconheça L utilizando o critério de pilha vazia.

(d) Determine sequências de transições que permitam verificar se a^2b^3c ou $a^5b^3c^2$ são aceites por $\mathcal{M}$.

16. Considere a linguagem $L = \{a^ib^jc^k \mid j \geq k \geq 0,\ i, j, k \in \mathbb{N}_0\}$ sobre o alfabeto $\{a, b, c\}$.

(a) Determine uma gramática que gere L.

(b) Com base na gramática definida, verifique se $b^3c \in L$ ou $a^3b^2c^3 \in L$ e, nos casos de resposta afirmativa, elabore a respetiva árvore de derivação.

(c) Verifique se a gramática definida é regular.

(d) Verifique se existe uma gramática regular que gere L.

(e) Determine um autómato de pilha $\mathcal{M}$ que reconheça L utilizando o critério de estados finais.

17. Considere as gramáticas $\mathcal{G}_1 = (\{\mathcal{S}_1, \mathcal{A}_1\}, \{a, b, c\}, \mathcal{S}_1, P_1)$ e $\mathcal{G}_2 = (\mathcal{S}_2, \mathcal{A}_2\}, \{a, b, c\}, \mathcal{S}_2, P_2)$ que têm, respetivamente, as seguintes produções

$$\begin{array}{lcl} \mathcal{S}_1 & \to & a\mathcal{S}_1b \mid \mathcal{A}_1 \\ \mathcal{A}_1 & \to & b\mathcal{A}_1 \mid \varepsilon \end{array} \qquad \begin{array}{lcl} \mathcal{S}_2 & \to & b\mathcal{S}_2c \mid \mathcal{A}_2 \\ \mathcal{A}_2 & \to & c\mathcal{A}_2 \mid \varepsilon \end{array}$$

Seja $\mathcal{G}_3 = (\{\mathcal{S}_3, \mathcal{S}_1, \mathcal{A}_1\}, \{a, b, c\}, \mathcal{S}_3, P_3)$ a gramática tal que

$$P_3 = P_1 \cup \{\mathcal{S}_3 \to \mathcal{S}_1\mathcal{S}_3 \mid \varepsilon\}$$

e seja $\mathcal{G} = (\{\mathcal{S}, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{A}_1, \mathcal{A}_2\}, \{a, b, c\}, \mathcal{S}, P)$ a gramática que tem como conjunto de produções

$$P = P_1 \cup P_2 \cup P_3 \cup \{\mathcal{S} \to \mathcal{S}_3\mathcal{S}_2\}$$

(a) Indique uma derivação de ab^2c^3 na gramática $\mathcal{G}$ e elabore a respetiva árvore de derivação. Diga, justificando, se a gramática é ambígua.

(b) Determine a linguagem gerada por $\mathcal{G}$.

(c) Determine um autómato de pilha $\mathcal{M}$ que reconheça $L(\mathcal{G})$ utilizando o critério de estados finais.

(d) Determine sequências de transição que permitam verificar se $ab^2c^2 \in L_{ef}(\mathcal{M})$ ou se $acb \in L_{ef}(\mathcal{M})$.

18. Considere o alfabeto $\Sigma = \{a, b\}$ e as linguagens

$$L_1 = \{a^n : n \in \mathbb{N}_0\} \text{ e } L_2 = \{b^i c^j : i, j \in \mathbb{N}_0, i < j\}.$$

(a) Determine autómatos, $\mathcal{M}_1$ e $\mathcal{M}_2$, que reconheçam L_1 e L_2, respetivamente.

(b) Determine um autómato de pilha $\mathcal{M}$ que reconheça L_1L_2 pelo critério de estados finais. Diga, justificando, se o autómato $\mathcal{M}$ é determinista.

(c) Determine sequências de transição que permitam verificar se $abc^2 \in L_{ef}(\mathcal{M})$ ou se $abc \in L_{ef}(\mathcal{M})$.

(d) Determine um autómato de pilha $\mathcal{M}^*$ que reconheça L_2^* pelo critério de estados finais.

19. Sejam $\mathcal{M}_1 = (Q_1, A, \Sigma_1, \delta_1, q_{0\ 1}, Z_{0\ 1}, F_1)$ e $\mathcal{M}_2 = (Q_2, A, \Sigma_2, \delta_2, q_{0\ 2}, Z_{0\ 2}, F_2)$ autómatos de pilha que reconhecem as linguagens L_1 e L_2, respetivamente, pelo critério de estados finais. Defina um autómato de pilha $\mathcal{M}$ que, segundo o critério de estados finais, reconhece a linguagem:

(a) L_1^*;

(b) L_1^+;

(c) L_1L_2;

(d) $L_1 \cup L_2$;

(e) $L_1^*L_2$.