Capítulo 1

As Bases

Toda a teoria das linguagens formais tem como base os conceitos de letra, alfabeto, palavra e linguagem que no fundo são uma formalização daquilo que todos conhecemos sobre esses mesmos conceitos.

Vamos recordar também o importante princípio da indução matemática dada a sua relevância na formalização e utilização destes conceitos.

1.1 Alfabetos e palavras

🐝🐝🔄🚫🐝🐝➡️❓

Intuitivamente, um alfabeto é o conjunto de todos os símbolos que utilizamos num determinado contexto para formar as palavras de uma linguagem. Consideramos esses símbolos como entidades atómicas, indivisíveis e sem qualquer estrutura.

Definição 1.1.1 Alfabeto.

Um **alfabeto** é um qualquer conjunto finito, não vazio, cujos elementos se denominam **símbolos** ou **letras**.

Usamos geralmente a letra grega maiúscula Σ (sigma) para denotar o alfabeto em estudo. Sempre que necessário, recorremos a outras letras gregas maiúsculas, como a Γ (gama), ou a índices, como Σ_1, Σ_2, etc.

Exemplo 1.1.2

- O alfabeto binário é $\Sigma = \{0, 1\}$.

- O alfabeto das letras minúsculas é $\Sigma = \{a, b, c, \ldots, z\}$.
- O alfabeto dos símbolos de uma máquina de calcular muito elementar é

$$\Sigma = \{0, 1, 2, \ldots, 9\} \cup \{+, -, \times, \div, =\}.$$

- A linguagem de programação Pascal foi inventada por Niklaus Wirth (Suíça, 1934 -) no início da década de 70 com a intenção explícita de "ensinar a programar". O seu livro "Algorithms + Data Structures = Programs", [7], foi intensivamente utilizado no ensino da programação estruturada, dos algoritmos e das estruturas de dados.

  O alfabeto da linguagem de programação Pascal é

$$\begin{aligned}\Sigma =& \{\textbf{program}, \textbf{begin}, \textbf{end}, \textbf{if}, \ldots\} \cup \cdots \\ & \{a, b, \ldots, A, B, \ldots\} \cup \{0, 1, 2, \ldots\} \cup \{., ;, ', (,), \ldots\}.\end{aligned}$$

  Aqui, consideramos que as "palavras reservadas" **program**, **begin**, **end**, etc, são símbolos.

Uma palavra é essencialmente uma sequência ordenada de símbolos de um alfabeto (que admite repetições de símbolos). Desde que não seja ambíguo, representamos a sequência que define uma palavra colocando os símbolos uns a seguir aos outros, sendo a ordem definida da esquerda para a direita.

Definição 1.1.3 Palavra.

Uma **palavra** é uma qualquer sequência finita de símbolos de um alfabeto Σ.

Usamos a notação $w = \sigma_1\sigma_2 \cdots \sigma_n$ para designar a palavra que corresponde à sequência de símbolos $w = (\sigma_1, \sigma_2, \cdots, \sigma_n)$, onde $\sigma_i \in \Sigma$, $i = 1, \ldots, n$, $n \in \mathbb{N}$.

Exemplo 1.1.4

- Um exemplo de uma palavra sobre o alfabeto binário é 0101110, a qual corresponde à sequência $(0, 1, 0, 1, 1, 1, 0)$.
- As palavras $abba$ e zzz estão definidas sobre o alfabeto das letras minúsculas, $\Sigma = \{a, b, c, \ldots, z\}$.
- Qualquer programa escrito na linguagem Pascal pode ser visto como uma palavra sobre o alfabeto da linguagem Pascal, por exemplo:

```
program hello(output)
begin
writeln('hello world')
end.
```

Duas *palavras* são *iguais* se as sequências de símbolos associadas forem iguais. Chamamos palavra vazia à sequência sem símbolos.

Definição 1.1.5 Palavra Vazia.

A **palavra vazia**, denotada por ε, é aquela que contém zero ocorrências de símbolos.

O tamanho de uma palavra é naturalmente dado pelo número de símbolos na sequência que a define.

Definição 1.1.6 Tamanho de uma Palavra.

O **tamanho de uma palavra** w, denotado por $|w|$, é o número de símbolos (contando as repetições) na palavra:

1. A palavra vazia tem tamanho zero, $|\varepsilon| = 0$.
2. A palavra $w = \sigma_1\sigma_2 \cdots \sigma_n$, com $\sigma_i \in \Sigma$, $i = 1, 2, \ldots, n$ e $n \in \mathbb{N}$, tem tamanho $|w| = n$.

Duas palavras de tamanhos diferentes não podem ser iguais. Uma palavra u é menor (maior) do que uma palavra v sempre que $|u| < |v|$ ($|u| > |v|$).

Associamos ao conjunto de todas palavras que têm o mesmo tamanho uma potência do alfabeto subjacente.

Definição 1.1.7 Potências de um Alfabeto.

Para $k \in \mathbb{N}_0$, **a potência de ordem** k **de um alfabeto** Σ, denotada por Σ^k, é o conjunto de todas as palavras sobre Σ com tamanho igual a k.

Salientamos que a potência de ordem 0 de um qualquer alfabeto Σ é $\Sigma^0 = \{\varepsilon\}$.

Exemplo 1.1.8

São exemplos de potências do alfabeto $\Sigma = \{a, b\}$,

- $\Sigma^0 = \{\varepsilon\}$;

- $\Sigma^1 = \{a, b\}$
- $\Sigma^2 = \{aa, ab, ba, bb\}$;
- $\Sigma^3 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$.

Notamos que o conjunto Σ^1 é um ***conjunto de palavras***, enquanto o conjunto Σ é um ***conjunto de símbolos***.

Qualquer palavra sobre um alfabeto Σ tem tamanho finito e pertence a um e um só dos conjuntos Σ^k, para algum $k \in \mathbb{N}_0$. Para representar o conjunto de todas as palavras sobre um alfabeto vamos utilizar o símbolo $\star$, conhecido por *estrela de Kleene* (Stephen Kleene, Estados Unidos da América, 1909 - 1994).

Definição 1.1.9 Conjunto de todas as palavras.

O **conjunto de todas as palavras** sobre um alfabeto Σ, denotado por $\Sigma^\star$, é definido por

$$\Sigma^\star = \bigcup_{k \in \mathbb{N}_0} \Sigma^k = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots$$

Notamos que palavra vazia é um elemento do conjunto estrela de Kleene (i.e., $\varepsilon \in \Sigma^\star$). Utilizamos a notação

$$\Sigma^+ = \bigcup_{k \in \mathbb{N}} \Sigma^k = \Sigma^1 \cup \Sigma^2 \cup \cdots$$

para representar o conjunto de todas as palavras com pelo menos um símbolo. Assim, $\Sigma^\star = \{\varepsilon\} \cup \Sigma^+$.

A concatenação de palavras é o processo de juntar palavras, umas a seguir às outras, construindo dessa forma novas palavras.

Definição 1.1.10 Concatenação de Palavras.

Se $v = \sigma_1\sigma_2 \cdots \sigma_i$ é uma palavra de tamanho i e $w = \eta_1\eta_2 \cdots \eta_j$ é uma palavra de tamanho j então a **concatenação** (ou justaposição) de v com w é a palavra

$$vw = \sigma_1\sigma_2 \cdots \sigma_i\eta_1\eta_2 \cdots \eta_j,$$

de tamanho $i + j$.

A concatenação de palavras é uma operação binária, $\cdot\colon \Sigma^\star \times \Sigma^\star \to \Sigma^\star$, sobre a qual destacamos as seguintes propriedades:

1. A concatenação é *associativa*: para quaisquer palavras u, v e w, $(uv)w = u(vw)$.

2. Em geral a concatenação *não é comutativa*: por exemplo a concatenação de $u = 10$ com $v = 000$, $uv = 10000$, é diferente da concatenação de v com u, $vu = 00010$.

3. A palavra vazia é o *elemento identidade da concatenação*: para qualquer palavra w, $w\varepsilon = \varepsilon w = w$.

4. O tamanho das palavras é *aditivo* relativamente à concatenação: para quaisquer palavras v e w, $|vw| = |v| + |w|$.

Em termos algébricos, o conjunto $\Sigma^\star$ com a operação de concatenação constitui um semigrupo multiplicativo com identidade, ou seja, é um *monóide*.

Definição 1.1.11 Potências de uma Palavra.

Para $n \in \mathbb{N}$ e uma qualquer palavra w, a **potência de ordem n da palavra w**, denotada por w^n, é a concatenação iterada n vezes de w:

$$w^n = ww \cdots w \text{ (n vezes).}$$

Por convenção, a potência de ordem zero de uma qualquer palavra w é $w^0 = \varepsilon$.

Exemplo 1.1.12

As potências da palavra $w = ab$ são:

$$w^0 = \varepsilon, \quad w^1 = ab, \quad w^2 = abab, \quad \ldots$$

Salientamos que $(ab)^3 = ababab$ não é o mesmo que $a^3b^3 = aaabbb$.

Entre outras, salientamos as seguintes propriedades das potências de palavras:

1. Se w é uma palavra e $i, j \in \mathbb{N}_0$ então $w^i w^j = w^{i+j}$.

2. Se w é uma palavra e $i, j \in \mathbb{N}_0$ então $(w^i)^j = w^{ij}$.

3. Se $n \in \mathbb{N}_0$ e w é uma palavra então $|w^n| = n|w|$.

Cada palavra é uma sequência de símbolos que se "lê" da esquerda para a direita. Quando "lemos" essa sequência da direita para a esquerda geralmente soletramos uma palavra diferente.

Definição 1.1.13 Palavra Reversa.

Sejam $\sigma_i \in \Sigma$, $i = 1, 2, \ldots, n$, $n \in \mathbb{N}$. A **palavra reversa** da palavra $w = \sigma_1\sigma_2 \cdots \sigma_n$, denotada por w^{-1}, é $\sigma_n\sigma_{n-1} \cdots \sigma_1$.

Convencionamos que $\varepsilon^{-1} = \varepsilon$.

Exemplo 1.1.14

1. A reversa da palavra abc é $(abc)^{-1} = c(ab)^{-1} = cba$, ou, de outra forma, $(abc)^{-1} = (bc)^{-1}a = cba$. Estas igualdades sugerem uma definição indutiva de palavra reversa, a qual veremos mais à frente.
2. A reversa da palavra $abcba$ é $(abcba)^{-1} = abcba$. Palavras como esta, idênticas à sua reversa, são designadas *palíndromos*.

São válidas as seguintes propriedades da operação de reversão:

1. Para qualquer palavra w, $(w^{-1})^{-1} = w$.
2. Para $n \in \mathbb{N}_0$ e uma qualquer palavra w, $(w^n)^{-1} = (w^{-1})^n$.
3. Dadas duas quaisquer palavras u e v, $(uv)^{-1} = v^{-1}u^{-1}$.
4. Para qualquer palavra w, $|w^{-1}| = |w|$.

Uma vez que uma palavra é uma sequência de símbolos, são ainda palavras as subsequências de uma palavra. Pela sua importância, atribuímos a algumas dessas palavras designações específicas.

Definição 1.1.15 Partes, Prefixos e Sufixos.

Uma palavra v é **parte** (ou uma **sub-palavra**) de uma palavra w quando existirem palavras x e y, tais que $w = xvy$.

Se $x = \varepsilon$ então $w = vy$ e a sub-palavra v é um **prefixo** da palavra w. Se além disso $y \neq \varepsilon$ então v é um **prefixo próprio** da palavra w.

Se $y = \varepsilon$ então $w = xv$ e a sub-palavra v é um **sufixo** da palavra w. Se além disso $x \neq \varepsilon$ então v é um **sufixo próprio** da palavra w.

Exemplo 1.1.16

Para o alfabeto $\Sigma = \{a, b, c, \ldots\}$,

- a palavra tal é parte da palavra $totalmente$;
- a palavra $total$ é um prefixo (próprio) da palavra $totalmente$;
- a palavra $mente$ é um sufixo (próprio) da palavra $totalmente$.

Usando a definição anterior, constatamos que:

- A palavra vazia não tem prefixos próprios nem sufixos próprios.
- Toda a palavra de tamanho $n \in \mathbb{N}_0$ tem exactamente n prefixos próprios e n sufixos próprios;

- Duas palavras do mesmo tamanho são diferentes se e só se existe um prefixo de uma que é diferente do prefixo com o mesmo tamanho da outra. Caso contrário, as duas palavras são iguais.
- Duas palavras do mesmo tamanho são diferentes se e só se existe um sufixo de uma que é diferente do sufixo com o mesmo tamanho da outra. Caso contrário, as duas palavras são iguais.

Seja $v = xy$ uma partição de uma palavra v num prefixo x e no correspondente sufixo y. À palavra $w = yx$ obtida de v por troca do prefixo x com o sufixo y chamamos uma conjugada de v.

Definição 1.1.17 Palavras Conjugadas.

Duas palavras v e w são **conjugadas** se existirem palavras x e y tais que $v = xy$ e $w = yx$.

Toda a palavra v é conjugada de si mesma, uma vez que $v = \varepsilon v = v\varepsilon$. Mais, se duas palavras v e w são conjugadas então w é conjugada de v e v é conjugada de w. Ainda, se w é conjugada de v e z é conjugada de w então z é também conjugada de v. Assim, a relação de conjugação é uma relação de equivalência.

Exemplo 1.1.18

Existem quatro palavras conjugadas da palavra $abba$: $abba$, $bbaa$, $baab$ e $aabb$. Já a palavra $abab$ tem apenas duas conjugadas: $abab$ e $baba$.

Definição 1.1.19 Palavra Primitiva.

Uma palavra $w \neq \varepsilon$ é **primitiva** se não for uma potência própria de uma qualquer outra palavra, ou seja, se para qualquer palavra u e para qualquer inteiro $k \geq 2$, $w \neq u^k$.

Definição 1.1.20 Raiz.

Uma palavra u é uma raiz índice k de uma palavra w se $w = u^k$, para algum $k \geq 1$.

Definição 1.1.21 Raiz primitiva.

A **raiz primitiva** de uma palavra w é a menor palavra u tal que $w = u^k$, para algum inteiro $k \geq 1$.

Uma palavra, diferente da palavra vazia, é primitiva se e só se é igual à sua raiz primitiva.

Exemplo 1.1.22

A palavra $u = abab$ é uma raiz da palavra $w = abababab$, uma vez que $w = u^2$. No entanto, a raiz primitiva da palavra w é ab uma vez que $w = (ab)^4$. A palavra w não é primitiva, enquanto as palavras ab e $abba$ já são primitivas.

Na parte final da Secção 1.3, p. 15, veremos que as palavras primitivas são as únicas que possuem tantas conjugadas como prefixos próprios.

1.2 Linguagens

> O desejo exprime-se por uma carícia, tal como o pensamento pela linguagem.
>
> —Jean-Paul Sartre

Podemos encontrar em múltiplos dicionários da Língua Portuguesa diversos significados para a palavra "linguagem". Um dos mais comuns, sugere que uma linguagem é um "sistema ou conjunto de sinais convencionais, fonéticos ou visuais, que servem para a expressão dos pensamentos e sentimentos"[1].

A linguagem portuguesa ou o conjunto de todos os programas que usam o alfabeto da linguagem de programação C e que são sintacticamente correctos, são dois exemplos de linguagens.

Definição 1.2.1 Linguagem Formal.

Uma **linguagem formal** sobre um alfabeto Σ é um qualquer subconjunto de palavras de $\Sigma^\star$.

Uma linguagem pode não conter (e geralmente não contém) todas as palavras em $\Sigma^\star$. Para além disso, as palavras de uma linguagem não precisam de utilizar todos os símbolos do alfabeto subjacente.

A **linguagem vazia**, denotada por $\emptyset$, é aquela à qual não pertencem quaisquer palavras. Salientamos que a linguagem à qual pertence apenas a palavra vazia é diferente da linguagem vazia, i.e., $\{\varepsilon\} \neq \emptyset$. A **linguagem completa** sobre um alfabeto Σ é $\Sigma^\star$.

[1] in Dicionário da Lingua Portuguesa, Porto Editora

Exemplo 1.2.2

Podemos definir várias linguagens sobre o alfabeto binário, $\Sigma = \{0, 1\}$. Por exemplo:

- A linguagem das palavras com tantos "zeros" como "uns",

$$\{w \in \Sigma^\star : \#_0(w) = \#_1(w)\},$$

  onde, para $s \in \{0, 1\}$ e $w \in \Sigma^\star$, $\#_s(w)$ designa o número de símbolos s que existem na palavra w;

- A linguagem das palavras binárias que representam números primos,

$$\{w \in \Sigma^\star : \mathrm{d}(w) \text{ é um número primo}\},$$

  onde $\mathrm{d}(w)$ é a representação decimal da sequência binária w (ver Subsecção 1.3.2, p. 20);

- A linguagem dos palíndromos, $\{w \in \Sigma^\star : w = w^{-1}\}$;

- A linguagem dos palíndromos pares, $\{ww^{-1} : w \in \Sigma^\star\}$;

- A linguagem em que cada palavra começa com uma sequência de 0´s, é seguida por uma sequência de 1´s e tem tantos 0´s como 1´s,

$$\{0^n 1^n : n \geq 0\},$$

  isto é, $\{\varepsilon, 01, 0011, 000111, \ldots\}$;

- A linguagem em que cada palavra começa com uma sequência de 0´s, é seguida por uma sequência de 1´s e tem pelo menos tantos 1´s como 0´s,

$$\{0^i 1^j : 0 \leq i \leq j\}.$$

Exemplo 1.2.3

Sobre o alfabeto $\Sigma = \{a, b\}$:

- o conjunto $\{w \in \Sigma^\star : |w| \leq 3\}$ é uma **linguagem finita**;
- o conjunto $\{awb : w \in \Sigma^\star\}$ é uma **linguagem infinita**.

1.2.1 Operações com Linguagens

Uma vez que as linguagens são conjuntos (de palavras), obtemos novas linguagens a partir de outras aplicando as operações usuais entre conjuntos.

Definição 1.2.4 União de Linguagens.

Se L e M são linguagens sobre um mesmo alfabeto Σ então a **união** de L e M é a linguagem

$$L \cup M = \{w \in \Sigma^\star : \ w \in L \text{ ou } w \in M\}.$$

Definição 1.2.5 Intersecção de Linguagens.

Se L e M são linguagens sobre um mesmo alfabeto Σ então a **intersecção** de L e M é a linguagem

$$L \cap M = \{w \in \Sigma^\star : \ w \in L \text{ e } w \in M\}.$$

Definição 1.2.6 Linguagem Complementar.

Se L é uma linguagem sobre um alfabeto Σ então a **linguagem complementar** de L é

$$\overline{L} = \{w \in \Sigma^\star : \ w \notin L\}.$$

Definição 1.2.7 Concatenação de Linguagens.

A **concatenação** de uma linguagem L com uma linguagem M é a linguagem

$$LM = \{xy : \ x \in L \text{ e } y \in M\}.$$

Exemplo 1.2.8

Para $L_1 = \{a, ab, ba\}$ e $L_2 = \{a, b\}$ temos que:

- $L_1 \cup L_2 = \{a, b, ab, ba\}$;
- $L_1 \cap L_2 = \{a\}$;
- $L_1 L_2 = \{aa, ab, aba, abb, baa, bab\}$.

Exemplo 1.2.9

Sobre o alfabeto $\Sigma = \{a, b\}$ definimos as linguagens

$$L_1 = \{w \in \Sigma^\star : \ w \text{ termina em } b\}$$

e

$$L_2 = \{w \in \Sigma^\star : \ w \text{ começa com } ab\}.$$

São ainda linguagens sobre Σ:

$$\begin{aligned} L_1 \cup L_2 &= \{w \in \Sigma^\star : \ w \text{ termina em } b \text{ ou começa com } ab\} \\ L_1 \cap L_2 &= \{w \in \Sigma^\star : \ w \text{ termina em } b \text{ e começa com } ab\} \\ L_1 L_2 &= \{w \in \Sigma^\star : \ w \text{ contém } bab\}. \end{aligned}$$

Ao concatenar iteradamente uma linguagem L, tantas vezes quanto se queira, obtemos as chamadas **potências de uma linguagem**:

$$L^n = \underbrace{LL \cdots L}_{n \text{ vezes}}.$$

Por definição, consideramos que $L^0 = \{\varepsilon\}$, qualquer que seja a linguagem L. Assim, $\emptyset^0 = \{\varepsilon\}$, mas $\emptyset^i = \emptyset$, para $i \geq 1$.

O fecho de Kleene de uma linguagem é a linguagem constituída pelas palavras que são a concatenação de um qualquer número de palavras (com eventuais repetições).

Definição 1.2.10 Fecho de Kleene de uma Linguagem.

O **fecho de Kleene** de uma linguagem L é

$$L^\star = \bigcup_{i \geq 0} L^i.$$

Alertamos para o facto de que o fecho de Kleene da linguagem vazia não é a linguagem vazia mas sim a linguagem constituída pela palavra vazia, ou seja, $\emptyset^\star = \{\varepsilon\}$.

Qualquer linguagem sobre um alfabeto Σ_1 é também uma linguagem sobre um qualquer alfabeto $\Sigma_2 \supset \Sigma_1$ (simplesmente não "usa" todos os símbolos do alfabeto). Assim, se L e M são linguagens sobre os alfabetos Σ_1 e Σ_2, respectivamente, então o alfabeto da linguagem união é (um subconjunto de) $\Sigma_1 \cup \Sigma_2$ e o alfabeto da linguagem intersecção é (um subconjunto de) $\Sigma_1 \cap \Sigma_2$.

Em geral, a escolha do alfabeto sobre o qual se define uma linguagem não influencia as noções e resultados tratados neste livro. Temos no entanto de ter algum cuidado em situações específicas como as destacadas no exemplo seguinte.

Exemplo 1.2.11

Seja $L = \{a^n : \ n > 0\}$. Se consideramos L como uma linguagem sobre o alfabeto:

- $\Sigma = \{a\}$ então $\overline{L} = \{\varepsilon\}$;

- $\Sigma = \{a, b\}$ então

$$\overline{L} = \{\varepsilon\} \cup \{w \in \{a, b\}^\star : \ w \text{ tem pelo menos um } b\};$$

- $\Sigma = \{a, b, c\}$ então

$$\overline{L} = \{\varepsilon\} \cup \{w \in \{a, b, c\}^\star : \ w \text{ tem pelo menos um } b \text{ ou um } c\}.$$

Quando revertemos as palavras de uma linguagem obtemos uma nova linguagem, chamada a linguagem reversa (ou inversa).

Definição 1.2.12 Reversa de uma Linguagem.

Se L é uma linguagem então a *linguagem reversa* de L é

$$L^{-1} = \{w^{-1} : \ w \in L\}.$$

A reversa da reversa de uma linguagem é a própria linguagem, ou seja, para qualquer linguagem L,

$$\left(L^{-1}\right)^{-1} = L.$$

1.2.2 Homomorfismos e Substituições

Um homomorfismo de palavras é uma função que transforma palavras (sobre um alfabeto) em palavras (sobre um outro alfabeto) e que preserva a concatenação.

Definição 1.2.13 Homomorfismo de Palavras.

Sejam Σ e Γ dois quaisquer alfabetos. Um **homomorfismo** é uma função $h \colon \Sigma^\star \to \Gamma^\star$ tal que:

1. $h(\varepsilon) = \varepsilon$;
2. $\forall x, y \in \Sigma^\star, h(xy) = h(x)h(y)$.

Se L é uma linguagem sobre um alfabeto Σ e $h \colon \Sigma^\star \to \Gamma^\star$ é um homomorfismo então

$$h(L) = \{h(w) : \ w \in L\}$$

é uma linguagem sobre o alfabeto Γ, designada *imagem* de L por h.

Para definir um homomorfismo é suficiente especificar a regra de transformação dos símbolos do alfabeto Σ. Mais precisamente, dois homomorfismos $h_1, h_2 \colon \Sigma^\star \to \Gamma^\star$ são iguais se e só se $\forall a \in \Sigma$, $h_1(a) = h_2(a)$.

Os homomorfismos preservam a união e concatenação de linguagens, bem como o fecho de Kleene de uma linguagem. Propomos como exercícios as demonstrações dos resultados seguintes.

Teorema 1.2.14

Se L_1 e L_2 são linguagens sobre um alfabeto Σ e se $h\colon \Sigma^\star \to \Gamma^\star$ é um homomorfismo então $h(L_1 \cup L_2) = h(L_1) \cup h(L_2)$ e $h(L_1L_2) = h(L_1)h(L_2)$.

Teorema 1.2.15

Se L é uma linguagem sobre um alfabeto Σ e $h\colon \Sigma^\star \to \Gamma^\star$ é um homomorfismo então $h(L^\star) = h(L)^\star$.

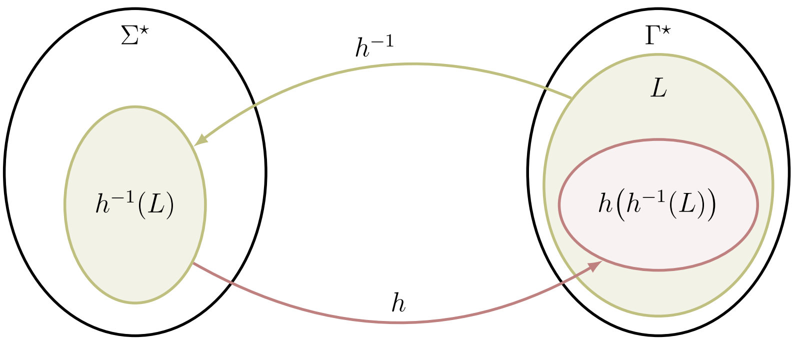
Exemplo 1.2.16

Consideremos o homomorfismo $h\colon \{a,b\}^\star \to \{0,1\}^\star$ definido por $h(a) = 01$ e $h(b) = \varepsilon$. Este homomorfismo transforma a linguagem $L = \{a^nb^n : n \geq 0\}$ na linguagem $h(L) = \{(01)^n : n > 0\}$.

Consideremos um homomorfismo $h\colon \Sigma^\star \to \Gamma^\star$ e uma linguagem L sobre o alfabeto Γ. A *imagem inversa* de L pelo homomorfismo h é a linguagem sobre o alfabeto Σ definida por

$$h^{-1}(L) = \{w \in \Sigma^\star : h(w) \in L\}.$$

Conforme ilustrado na figura seguinte, em geral $h(h^{-1}(L)) \neq L$. É no entanto verdade que $h(h^{-1}(L)) \subseteq L$.



Exemplo 1.2.17

Consideremos o homomorfismo $h\colon \{a,b\}^\star \to \{0,1\}^\star$ definido por $h(a) = 1$ e $h(b) = 00$. A imagem inversa da linguagem $L = \{0^n1^n : n \geq 0\}$ por intermédio deste homomorfismo é a linguagem $h^{-1}(L) = \{b^n a^{2n} : \ n \geq 0\}$. Note-se que $h(h^{-1}(L)) = \{0^{2n}1^{2n} : \ n \geq 0\}$.

Num homomorfismo de palavras substituímos cada símbolo do alfabeto por uma palavra. Chamamos *substituição* à generalização do conceito de homomorfismo que consiste em substituir cada símbolo do alfabeto por uma linguagem.

Definição 1.2.18 Substituição.

Uma **substituição** é uma aplicação $s : \Sigma \to \mathcal{P}(\Sigma^\star)$ que associa a cada símbolo $a \in \Sigma$ uma linguagem $s(a) = L_a$.

A extensão de uma substituição s a palavras $w = a_1 a_2 \cdots a_n \in \Sigma^\star$ é definida por

$$s(a_1 a_2 \cdots a_n) = s(a_1)s(a_2) \cdots s(a_n) = L_{a_1} L_{a_2} \cdots L_{a_n}.$$

A extensão da substituição s a uma qualquer linguagem L sobre Σ é

$$s(L) = \bigcup_{w \in L} s(w).$$

Os conceitos de homomorfismo e substituição serão utilizados nos capítulos seguintes quando estudarmos as propriedades de certas classes de linguagens. Notamos que são conceitos diferentes, embora um homomorfismo possa ser visto como um caso particular de uma substituição. Formalmente, um homomorfismo é um morfismo do monóide $(\Sigma, \cdot)$ no monóide $(\Sigma^\star, \cdot)$, onde a operação produto é a concatenação, enquanto uma substituição é um morfismo do monóide $(\Sigma^\star, \cdot)$ no monóide $(\mathcal{P}(\Sigma^\star), \cdot)$, onde $\mathcal{P}(\Sigma^\star)$ denota o conjunto de todos os subconjuntos de $\Sigma^\star$.

Nesta primeira secção, apresentámos alguns exemplos concretos de linguagens formais. No entanto, a Teoria das Linguagens Formais prende-se sobretudo com o estudo das propriedades de classes[2] de linguagens e da sua caracterização. Nos próximos capítulos veremos a importância desta abordagem, a qual, para além da sua importância teórica, possui diversas aplicações práticas.

Concluímos, com um exemplo concreto de uma classe de linguagens e de algumas das suas propriedades. Relembrando a noção de prefixo próprio de uma palavra (ver Definição 1.1.15, p. 6), pensemos naquelas linguagens em que nenhuma das suas palavras é prefixo próprio de uma outra palavra da mesma linguagem.

[2]Uma classe de linguagens é um conjunto de linguagens.

Definição 1.2.19 Linguagem Livre de Prefixos.

Uma linguagem é **livre de prefixos** se nenhum prefixo próprio de uma qualquer palavra da linguagem é uma palavra da linguagem.

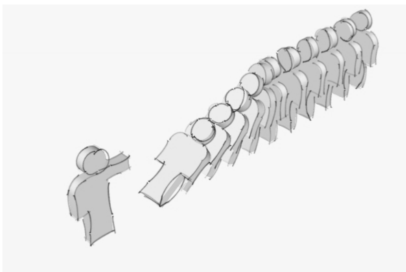
Fixo um alfabeto Σ, consideremos a classe de todas as linguagens sobre Σ que são livres de prefixos. Quais as propriedades desta classe de linguagens? Como a caracterizar?

Dadas duas quaisquer linguagens livres de prefixos, L_1 e L_2, a sua intersecção $L_1 \cap L_2$ é também uma linguagem livre de prefixos e dizemos assim, que a classe das linguagens livres de prefixos é fechada para a intersecção. Esta classe é também fechada para a concatenação, para a potência e para a diferença de linguagens.

No entanto a união de duas linguagens livres de prefixos não é necessariamente livre de prefixos, pelo que a classe das linguagens livres de prefixos não é fechada para a união. O mesmo se pode dizer relativamente ao complementar, à reversa e ao fecho de Kleene.

1.3 Indução

Como sabemos, existem muitos conceitos matemáticos que admitem naturalmente formulações ditas recorrentes ou indutivas.



As demonstrações das propriedades dos números naturais são frequentemente realizadas por indução e a própria definição do conceito número natural é recorrente segundo a axiomática de Peano (Giuseppe Peano, Itália, 1858 - 1932).

Já os objectos estudados na área das Linguagens Formais são estruturados e requerem um tratamento específico. Por exemplo, uma palavra de uma linguagem é um objecto constituído por uma sequência de símbolos que se pode definir de forma indutiva de diversas formas:

- Uma palavra ou é a palavra vazia ou é um símbolo seguido de uma palavra;
- Uma palavra ou é a palavra vazia ou é uma palavra seguida de um símbolo.

Assim, antes de provarmos quaisquer resultados sobre linguagens formais, é essencial estabelecer à priori definições adequadas para os conceitos utilizados. Vejamos dois exemplos.

Exemplo 1.3.1 Potência de uma palavra.

A **potência** de ordem $n \in \mathbb{N}_0$ de uma palavra w é

$$w^n = \begin{cases} \varepsilon & \text{se } n = 0 \\ ww^{n-1} & \text{se } n > 0 \end{cases}.$$

Exemplo 1.3.2 Potência de uma linguagem.

A **potência** de ordem $n \in \mathbb{N}_0$ de uma linguagem L é

$$L^n = \begin{cases} \{\varepsilon\} & \text{se } n = 0 \\ L^{n-1}L & \text{se } n > 0 \end{cases}.$$

Nos exemplos anteriores pudemos ver como se definem conceitos usando indução sobre o conjunto dos números naturais. Nos exemplos seguintes, usamos indução estrutural para formalizar os conceitos de tamanho e de reversa de uma palavra.

Exemplo 1.3.3 Tamanho de uma palavra.

O **tamanho** de uma palavra é definido indutivamente por:

Caso base. $|\varepsilon| = 0$;

Passo indutivo. Para $a \in \Sigma$ e $w \in \Sigma^\star$, $|aw| = 1 + |w|$.

Exemplo 1.3.4 Palavra reversa.

Uma possível definição indutiva de *palavra reversa* de uma palavra w sobre um alfabeto Σ é

$$w^{-1} = \begin{cases} \varepsilon & \text{se } w = \varepsilon \\ av^{-1} & \text{se } w = va, \text{ com } a \in \Sigma, v \in \Sigma^\star \end{cases}.$$

Usando esta definição, calculamos a reversa da palavra abc da seguinte forma:

$$\begin{aligned} (abc)^{-1} &= ((ab)c)^{-1} = c\,(ab)^{-1} \\ &= c\,((a)b)^{-1} = c\left(b(a)^{-1}\right) = cb(a)^{-1} \\ &= cb\,(\varepsilon a)^{-1} = cb\left(a(\varepsilon^{-1})\right) = cba\left(\varepsilon^{-1}\right) \\ &= cba\,(\varepsilon) = cba. \end{aligned}$$

Ao longo deste livro, vamos utilizar frequentemente a técnica de demonstração que consiste em aplicar uma "generalização" do princípio de indução matemática (definido sobre os número naturais).

Princípio 1.3.5 Indução Estrutural.

A veracidade de uma proposição $P(x)$ definida sobre todos os elementos x de um conjunto X, indutivamente definido, é estabelecida em duas etapas:

Caso(s) base. *Provar directamente para o(s) elemento(s) "pequenos" x de X (elementos iniciais ou mínimos do conjunto);*

Passo indutivo. *Para cada regra indutiva utilizada na construção de um elemento $x \in X$ a partir dos elementos $x_1, x_2, \ldots, x_k$, provar $P(x)$ assumindo que são verdadeiras $P(x_1), P(x_2), \ldots, P(x_k)$ e utilizando esses factos.*

Nos seguintes exemplos apresentamos demonstrações de uma mesma propriedade: no primeiro usamos indução matemática e no segundo indução estrutural.

Exemplo 1.3.6

Pretendemos provar que o tamanho de palavras é aditivo relativamente à concatenação, isto é, que $\forall v, w \in \Sigma^\star$, $|vw| = |v| + |w|$.

Vamos usar indução matemática sobre o tamanho da palavra v, considerando uma qualquer palavra w (fixa).

Caso base. Se $|v| = 0$ então $v = \varepsilon$ e portanto

$$
\begin{aligned}
|vw| &= |\varepsilon w| \\
&= |w| && \text{(porque } \varepsilon \text{ é o elemento identidade da concatenação)} \\
&= 0 + |w| && \text{(porque } 0 \text{ é o elemento identidade da soma)} \\
&= |v| + |w| && \text{(porque } |v| = 0\text{).}
\end{aligned}
$$

Passo indutivo. Consideramos como *hipótese de indução*

$$\forall v, |v| = n \implies |vw| = |v| + |w|.$$

Vamos demonstrar a veracidade da *tese*

$$\forall v, |v| = n + 1 \implies |vw| = |v| + |w|.$$

Seja então v tal que $|v| = n + 1$. Existem $a \in \Sigma$ e $x \in \Sigma^\star$, com $|x| = n$ tais que $v = ax$. Assim,

$$
\begin{aligned}
|vw| &= |axw| && \text{(construção } v = ax\text{)}\\
&= 1 + |xw| && \text{(definição de tamanho)}\\
&= 1 + |x| + |w| && \text{(Hipótese de Indução)}\\
&= |v| + |w| && \text{(construção de } v \text{ e definição de tamanho).}
\end{aligned}
$$

Verificados o caso base e demonstrado o passo indutivo, concluímos que a proposição é verdadeira.

Exemplo 1.3.7

Pretendemos provar que o tamanho de palavras é aditivo relativamente à concatenação, isto é, que

$$\forall v, w \in \Sigma^\star,\ |vw| = |v| + |w|.$$

Vamos usar indução estrutural sobre as palavras v, considerando uma qualquer palavra w (fixa).

Caso base. Se $v = \varepsilon$,

$$
\begin{aligned}
|vw| &= |\varepsilon w| &&\\
&= |w| && \text{(porque } \varepsilon \text{ é o elemento identidade da concatenação)}\\
&= 0 + |w| && \text{(porque 0 é o elemento identidade da soma)}\\
&= |v| + |w| && \text{(porque } |v| = 0\text{).}
\end{aligned}
$$

Passo indutivo. Se $v \neq \varepsilon$, então $v = ax$ com $a \in \Sigma$ e $x \in \Sigma^\star$. Consideramos como *hipótese de indução* que a propriedade é verdadeira para a palavra x:

$$|xw| = |x| + |w|.$$

Vamos demonstrar a veracidade da *tese*

$$|vw| = |v| + |w|.$$

Assim,

$$
\begin{aligned}
|vw| &= |axw| && \text{(por construção } v = ax\text{)}\\
&= 1 + |xw| && \text{(pela definição de tamanho)}\\
&= 1 + |x| + |w| && \text{(por Hipótese de Indução)}\\
&= |v| + |w| && \text{(pela construção de } v \text{ e pela definição de tamanho).}
\end{aligned}
$$

Verificados o caso base e demonstrado o passo indutivo, concluímos que a proposição é verdadeira.

Os princípios de indução marcam presença frequente em diversos contextos, tanto na formulação de definições, como instrumento essencial nas demonstrações de resultados. Seguidamente apresentamos algumas aplicações.

1.3.1 Expressões Aritméticas

Na maioria das linguagens de programação é definido um conjunto de regras lexicais vocacionadas para exprimir quais as formas admissíveis de representação de números, constantes, variáveis, operadores aritméticos, entre outros. Para além do léxico, é essencial a especificação da sintaxe, indicando quais as formas válidas de combinar os diversos elementos.

Vamos apresentar uma definição muito simples, de carácter académico, do que é uma expressão aritmética, seguida da demonstração de uma propriedade das expressões aritméticas, igualmente simples.

Exemplo 1.3.8 Expressão Aritmética.

Consideremos o alfabeto $\Sigma = \{a, b, c, \ldots\} \cup \mathbb{R} \cup \{+, \times, (,)\}$. Uma *expressão Aritmética* é uma palavra em $\Sigma^\star$ definida indutivamente por:

Casos base. Qualquer número ("real") ou qualquer variável (letra) é uma *expressão*;

Passo indutivo. Se E e F são *expressões* então também são *expressões* $E + F$, $E \times F$ e (E).

Assim, são exemplos de expressões as palavras 2, x, $x + 2$, $(x + 2)$, $2 \times (x + 2)$ ou $x \times (x + 2)$.

Exemplo 1.3.9

Vamos provar que em toda a expressão aritmética (definida indutivamente no exemplo anterior) o número de parêntesis esquerdos e direitos é igual, ou seja, que a proposição $P(G)$ definida por "G possui o mesmo número de parêntesis esquerdos e direitos", é verdadeira qualquer que seja a expressão G.

Casos base. Se G for apenas um número ou uma variável então tem zero parêntesis esquerdos e zero direitos. Assim, G possui o mesmo número de parêntesis esquerdos e direitos.

Passo indutivo. Senão, G foi construída segundo uma das 3 regras:

1. $G = E + F$;
2. $G = E \times F$;

3. $G = (E)$.

Consideremos como *hipótese de indução* que $P(E)$ e $P(F)$ são verdadeiras.

Sejam assim,

$$\begin{aligned} m = & \text{ número de parêntesis esquerdos de } E \\ = & \text{ número de parêntesis direitos de } E; \end{aligned}$$

$$\begin{aligned} n = & \text{ número de parêntesis esquerdos de } F \\ = & \text{ número de parêntesis direitos de } F. \end{aligned}$$

Vamos provar a veracidade da *tese* $P(G)$:

1. Se $G = E + F$ então

$$\begin{aligned} m + n = & \text{ número de parêntesis esquerdos de } G \\ = & \text{ número de parêntesis direitos de } G; \end{aligned}$$

2. O caso $G = E \times F$ é análogo;
3. Se $G = (E)$ então

$$\begin{aligned} m + 1 = & \text{ número de parêntesis esquerdos de } G \\ = & \text{ número de parêntesis direitos de } G. \end{aligned}$$

1.3.2 Representações de Números

São conhecidos registos babilónicos da utilização de sistemas posicionais para a representação de números. Podemos representar os números inteiros não negativos usando sistemas posicionais. Nós, humanos, estamos acostumados a utilizar o sistema decimal, de base 10, enquanto nos computadores impera o sistema binário, de base 2.

Num sistema de numeração posicional, cada uma das posições é preenchida com um dígito (símbolo numeral) de um conjunto fixo de dígitos. O valor de cada um dos dígitos depende da sua posição. Assim, o valor eem cada posição é dado pelo produto do valor base do dígito por uma certa potência da base do sistema, potência essa que varia com a posição. O número representado pelo numeral é então dado pela soma dos valores associados às posições.

Por exemplo, na base 10,

$$\begin{aligned} 2368 &= 2 \times 10^3 + 3 \times 10^2 + 6 \times 10^1 + 8 \times 10^0 \\ &= 2000 + 300 + 60 + 8. \end{aligned}$$

Exemplo 1.3.10 Conversão de binário em decimal.

Consideremos o alfabeto binário $\Sigma = \{0, 1\}$. As palavras em Σ^+ podem ser interpretadas como representações binárias dos números inteiros não negativos (numerais na base 2). Por exemplo:

$$1001 \mapsto 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9_{(10)}$$

e

$$11001010 \mapsto 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^3 + 1 \times 2^1 = 202_{(10)}.$$

De uma forma geral, cada palavra $w \in \{0, 1\}^+$ é a representação de um número inteiro não negativo, de acordo com a correspondência

$$w = b_k \cdots b_1 b_0 \mapsto \sum_{i=0}^{k} b_i 2^i = n \in \mathbb{N}_0.$$

Para calcular a representação de uma palavra binária, seguindo a fórmula anterior, é necessário calcular as sucessivas potências de 2. Em alternativa, torna-se mais expedito utilizar o método derivado da proposição seguinte.

Proposição 1.3.11 Conversão Binário $\mapsto$ Decimal.

Caso base.

$$\begin{aligned} 0 \in \Sigma^+ &\mapsto 0 \in \mathbb{N}_0 \\ 1 \in \Sigma^+ &\mapsto 1 \in \mathbb{N}_0. \end{aligned}$$

Passo indutivo. *Se* $w \in \{0, 1\}^+ \mapsto n \in \mathbb{N}_0$ *então*

$$\begin{aligned} w0 \in \Sigma^+ &\mapsto 2 \times n \\ w1 \in \Sigma^+ &\mapsto 2 \times n + 1. \end{aligned}$$

Demonstração.

Vamos provar o caso base e, em seguida, o passo indutivo.

Caso base.

$$\begin{aligned} 0 \in \Sigma &\mapsto 0 \times 2^0 = 0 \\ 1 \in \Sigma &\mapsto 1 \times 2^0 = 1. \end{aligned}$$

Passo indutivo. Se $w = b_k \cdots b_1 b_0 \mapsto \sum_{i=0}^{k} b_i 2^i = n$ então

$$
\begin{aligned}
w0 &\mapsto \sum_{i=0}^{k} b_i 2^{i+1} + 0 \times 2^0 = 2 \times \sum_{i=0}^{k} b_i 2^i = 2 \times n \\
w1 &\mapsto \sum_{i=0}^{k} b_i 2^{i+1} + 1 \times 2^0 = 2 \times \sum_{i=0}^{k} b_i 2^i + 1 = 2 \times n + 1.
\end{aligned}
$$

Exemplo 1.3.12

Aplicando a proposição anterior obtemos a conversão da palavra binária $w = 11001$:

$$
\begin{array}{c|ll}
1 & 1 & = 1 \\
1 & 2 \times 1 + 1 & = 3 \\
0 & 2 \times 3 & = 6 \\
0 & 2 \times 6 & = 12 \\
1 & 2 \times 12 + 1 & = 25
\end{array}
$$

1.3.3 Primitivas e Conjugadas

Já sabemos que, em geral, a concatenação de palavras não é comutativa. No entanto, existem palavras que comutam entre si. Por exemplo, as palavras $x = aa$ e $y = aaa$ comutam (i.e., $xy = yx$). Também a palavra vazia comuta com qualquer palavra, uma vez que é o elemento neutro da concatenação.

Será que conseguimos caracterizar as palavras que comutam entre si? No que se segue, veremos que sim.

Começamos por enunciar uma propriedade das palavras que começam e terminam com uma mesma sub-palavra, ou seja, as palavras que possuem um sufixo próprio igual a um seu prefixo próprio.

Lema 1.3.13

Se x, y e z são palavras tais que $x \neq \varepsilon$ e $xy = yz$ então existem palavras u e v e um inteiro $k \geq 0$ tais que $x = uv$, $y = (uv)^k u = u(vu)^k$ e $z = vu$.

Demonstração.

Caso $|x| \geq |y|$. Conforme ilustrado no esquema seguinte, existe uma palavra v (sendo $v = \varepsilon$ quando $|x| = |y|$) tal que $x = yv$.

<table>
<tr><td colspan="2">x</td><td>y</td></tr>
<tr><td>y</td><td colspan="2">z</td></tr>
<tr><td>y</td><td>v</td><td>y</td></tr>
</table>

Considerando $k = 0$ e $u = y$, obtemos $x = yv = uy$, $y = (uv)^0u$ e $z = vy = vu$.

Caso $|x| < |y|$. Vamos demonstrar este caso por indução sobre o tamanho de y.

Se $|y| = 0$ então $y = \varepsilon$, bastado fixar $u = \varepsilon$, $v = x = z$ e $k = 0$.

Consideremos como hipótese de indução que a proposição é verdadeira para qualquer palavra y tal que $|y| \leq n$ e seja y tal que $|y| = n + 1$.

Uma vez que $xy = yz$ e $|x| < |y|$, $y = xw$ para alguma palavra w. Daqui resulta que $xxw = xwz$ e assim $xw = wz$, conforme ilustrado no esquema seguinte.

<table>
<tr><td>x</td><td colspan="2">y</td></tr>
<tr><td>x</td><td>w</td><td>z</td></tr>
<tr><td colspan="2">y</td><td>z</td></tr>
</table>

Uma vez que $x \neq \varepsilon$, $|w| < |y|$ e portanto $|w| \leq n$. Aplicando a hipótese de indução a w, concluímos que existem palavras u e v e um inteiro não negativo k tais que $x = uv$, $w = (uv)^k u = u(vu)^k$ e $z = vu$.

Uma vez que $y = xw$, $y = (uv)(uv)^k u = (uv)u(vu)^k$, ou seja, $y = (uv)^{k+1}u = u(vu)^{k+1}$.

Sejam x e y duas palavras que são potências de uma mesma palavra w, ou seja, existem $i, j \in \mathbb{N}_0$ tais que $x = w^i$ e $y = w^j$. Então claramente x e y comutam entre si (i.e., $xy = yx$). Vamos agora demonstrar o recíproco, concluindo que duas palavras comutam entre si se e só se são ambas potências de uma mesma palavra.

Teorema 1.3.14

Se $x \neq \varepsilon$ e $y \neq \varepsilon$ são duas palavras tais que $xy = yx$ então existe uma palavra w e existem inteiros não negativos i e j tais que $x = w^i$ e $y = w^j$.

Demonstração.

Vamos provar o teorema por indução sobre o tamanho de xy.

Se $|xy| = 2$ então x e y têm apenas uma letra e a conclusão é imediata.

Consideremos por hipótese de indução que a propriedade enunciada no teorema é verdadeira para quaisquer palavras x e y que satisfaçam $|xy| \leq n$.

Sejam x e y tais que $|xy| = n + 1$. Aplicando o Lema 1.3.13, p. 22 concluímos que existem duas palavras u e v e existe $k \geq 0$ de tal modo que $x = uv$ e $y = (uv)^k u$. Assim, $uv(uv)^k u = (uv)^k uuv$, donde $uvu = uuv$ e portanto $uv = vu$.

Uma vez que $|uv| \leq n$, podemos aplicar a hipótese de indução e concluir que existe uma palavra w bem como inteiros não negativos p e q tais que $u = w^p$ e $v = w^q$. Daqui resulta que $x = uv = w^{p+q}$ e também que $y = (uv)^k u = w^{k(p+q)+p}$.

Aplicamos agora o teorema anterior, para alcançar uma caracterização das palavras primitivas como função do número de conjugadas distintas que possuem.

Corolário 1.3.15

Uma palavra w tem $|w|$ conjugadas distintas se e só se é primitiva.

Demonstração.

Qualquer palavra w possui exactamente $|w|$ prefixos próprios distintos que correspondem no máximo a $|w|$ palavras conjugadas distintas (por troca de cada prefixo de w com o sufixo correspondente).

Se w não é primitiva então existe uma palavra u tal que $w = u^i$ para algum $i \geq 2$. Assim, $w = uu^{i-1} = u^{i-1}u$ pelo que a palavra w tem menos de $|w|$ conjugadas.

Suponhamos agora que existem duas partições distintas $w = u_1v_1$ e $w = u_2v_2$, com $u_1 \neq u_2$, cujas conjugadas $w_1 = v_1u_1$ e $w_2 = v_2u_2$ são iguais. Uma vez que w_1 e w_2 também são conjugadas entre si, existem $x \neq \varepsilon$ e $y \neq \varepsilon$ tais que $w_1 = xy = yx = w_2$.

Pelo Teorema 1.3.14, p. 23, as palavras x e y são potências de uma mesma palavra u.

Assim, para algum $i \geq 2$, $w_1 = w_2 = u^i$ e portanto w_1 não é primitiva (nem w_2). Uma vez que w é uma conjugada de w_1, w também não é primitiva. Concluímos que se w for uma palavra primitiva então possui tantos prefixos próprios distintos como conjugadas distintas, pelo que o número total de conjugadas distintas é $|w|$.

1.4 Exercícios

1. Caracterize as linguagens $L_1 \cup \overline{L_2}$, $\overline{L_1} \cap L_2$, $L_1\overline{L_2}$ e $\overline{L_2}L_1$, onde L_1 e L_2 são as linguagens definidas no Exemplo 1.2.9, p. 10.

2. É verdade que o fecho de Kleene de uma qualquer linguagem é sempre uma linguagem infinita?

3. Seja $\Sigma = \{a, b\}$.

- **(a)** Quantas palavras em $\Sigma^\star$ têm tamanho 0? E tamanho 1? E tamanho 2?
- **(b)** Para $n \in \mathbb{N}_0$, estabeleça uma fórmula para o número de palavras em $\Sigma^\star$ de tamanho n e demonstre-a.
- **(c)** O conjunto $\Sigma^\star$ é enumerável?

4. Para $\Sigma = \{0, 1\}$ e $n \in \mathbb{N}_0$, seja $L_n = \{w \in \Sigma^\star : \ |w| \leq n\}$. Qual é o cardinal de L_n?

5. Seja Σ um alfabeto de tamanho m. Para $n \in \mathbb{N}_0$, estabeleça uma fórmula para o número de palavras em $\Sigma^\star$ com tamanho n e demonstre-a.

6. Seja L uma linguagem. Mostre, usando as definições de $L^\star$ e L^+, que:

(a) $L^+ = LL^\star$;

(b) $L^\star \setminus L^+ = \{\varepsilon\}$.

7. Diga, justificando, se as seguintes afirmações são verdadeiras ou falsas:

(a) Estima-se que na Via Láctea existam cerca de $10^{12} = 1000000000000$ estrelas. Embora muitas, as estrelas na nossa galáxia não são tantas quantas as linguagens definidas sobre o alfabeto $\Sigma = \{\bigstar\}$ e que são constituídas por palavras de tamanho inferior a 48.

(b) Seja x uma qualquer palavra sobre um alfabeto $\Sigma = \{a, b\}$. Não existem 4 palavras y de tamanho $|y| = 3$ que façam com que xyx^{-1} seja um palíndromo ímpar.

8. Mostre que o conjunto de todas as linguagens sobre um qualquer alfabeto Σ não é enumerável.

9. Seja Σ um qualquer alfabeto. Diga, justificando, qual é o valor lógico das seguintes afirmações:

(a) $\Sigma^\star$ é um conjunto finito.

(b) Para $x, y \in \Sigma^\star$, se $x = x^{-1}$ e $y = y^{-1}$ então $(xy)^{-1} = yx$.

(c) Se L é uma qualquer linguagem então $L^3 = \{w^3 : \ w \in L\}$.

10. Seja x uma qualquer palavra sobre um alfabeto $\Sigma = \{a, b\}$. Será que é sempre possível definir uma palavra y, de tamanho $|y| \geq 2$, de tal modo que a palavra xyx^{-1} seja um palíndromo ímpar?

11. Apresente exemplos de linguagens L e valores de $n \in \mathbb{N}_0$ para os quais se verifique

$$L^n = \{w^n : \ w \in L\}.$$

12. Partindo da definição indutiva de potência de uma palavra do Exemplo 1.3.1, p. 16, mostre que $\forall n \in \mathbb{N}_0, \forall w \in \Sigma^\star, w^{n+1} = w^n w$.

13. Partindo da definição indutiva de palavra reversa do Exemplo 1.3.4, p. 16, mostre que $\forall a \in \Sigma, \forall v \in \Sigma^\star, (av)^{-1} = v^{-1}a$.

14. Seja Σ um qualquer alfabeto. Demonstre as propriedades seguintes.

(a) $\forall w \in \Sigma^\star, \forall n \in \mathbb{N}_0, \ |w^n| = n|w|$.

(b) $\forall x, y \in \Sigma^\star, (xy)^{-1} = y^{-1}x^{-1}$.

(c) $\forall w \in \Sigma^\star, (w^{-1})^{-1} = w$.

(d) $\forall w \in \Sigma^\star, \forall n \in \mathbb{N}_0, \ (w^n)^{-1} = (w^{-1})^n$.

(e) $\forall n \in \mathbb{N}, \forall x, y \in \Sigma^\star,\ x^n = y^n \implies x = y$.

(f) $\forall n \in \mathbb{N}_0, \forall w \in \Sigma^\star,\ w = w^{-1} \implies w^n = (w^n)^{-1}$.

(g) $\forall n \in \mathbb{N}, \forall w \in \Sigma^\star,\ w^n = (w^n)^{-1} \implies w = w^{-1}$.

Sugestão.

(a) A propriedade demonstra-se por indução, usando a definição de potência de uma palavra (Exemplo 1.3.1, p. 16) e a propriedade aditiva do tamanho de palavras relativamente à concatenação (Exemplo 1.3.7, p. 18).

(b) A propriedade demonstra-se por indução estrutural sobre x, usando a propriedade referida no Exercício 1.4.13, p. 25 ou por indução estrutural sobre y, usando a definição indutiva de palavra reversa apresentada no Exemplo 1.3.4, p. 16.

(c) A propriedade demonstra-se por indução estrutural, usando a propriedade referida no Exercício 1.4.13, p. 25 e a definição indutiva de palavra reversa apresentada no Exemplo 1.3.4, p. 16.

(d) A propriedade demonstra-se por indução sobre n, usando a propriedade referida na alínea (b, p. 25) e a definição indutiva de potência de uma palavra.

(e) Basta utilizar a definição de igualdade de palavras em termos da igualdade de prefixos.

(f) É consequência da propriedade referida na alínea (d, p. 25).

(g) É consequência das propriedades referidas nas alíneas (d, p. 25) e (e, p. 25).

15. Seja L uma linguagem com pelo menos duas palavras. Mostre que $L^2 \neq \{w^2 :\ w \in L\}$.

16. Seja L uma linguagem com pelo menos duas palavras. Mostre que para $n \geq 2$, $L^n \neq \{w^n :\ w \in L\}$.

17. Considere a definição de palavras conjugadas apresentada na Definição 1.1.17, p. 7. Mostre que a relação de conjugação entre palavras é uma relação de equivalência.

18. Uma palavra w é livre de quadrados se não contém sub-palavras da forma uu, onde $u \neq \varepsilon$.

(a) Quantas palavras binárias livres de quadrados existem?

(b) *** Mostre que existe um número infinito de palavras livres de quadrados definidas sobre o alfabeto $\Sigma = \{a, b, c\}$.

19. Mostre que dois homomorfismos de linguagens $h_1, h_2 \colon \Sigma^\star \to \Gamma^\star$ são iguais se e só se $\forall a \in \Sigma$, $h_1(a) = h_2(a)$.

20. Demonstre o Teorema 1.2.14, p. 13.

21. Demonstre o Teorema 1.2.15, p. 13.

22. Sejam $h \colon \Sigma^\star \to \Sigma^\star$ um homomorfismo de palavras e L uma linguagem sobre o alfabeto Σ. Mostre que $h(h^{-1}(L)) \subseteq L$ e que $L \subseteq h^{-1}(h(L))$.

23. Averigue se a classe das linguagens livres de prefixos é fechada para cada uma seguintes operações:

(a) união;

(b) intersecção;

(c) complementar;

(d) concatenação;

(e) diferença;

(f) potência;

(g) fecho de Kleene.