Capítulo 1: Alfabetos, cadeias, linguagens

Símbolos e alfabetos. Um *alfabeto* é, para os nossos fins, um conjunto finito não vazio cujos elementos são chamados de símbolos. Dessa maneira, os conceitos de símbolo e alfabeto são introduzidos de forma interdependente: um alfabeto é um conjunto de símbolos, e um símbolo é um elemento qualquer de um alfabeto.

Note que consideramos aqui apenas *alfabetos finitos*: isso é feito por simplicidade, naturalmente, e também porque são raros os casos em que a consideração de um alfabeto infinito seria desejável.[1]

Qual o alfabeto que devemos considerar, ou seja, quais são os símbolos do alfabeto considerado depende do contexto em que pretendemos trabalhar. Como exemplos de alfabetos, citamos {0, 1} ou {*a*, *b*}, o alfabeto da língua portuguesa { a, b, c, ç, … z}, o conjunto de caracteres ASCII, etc.

Até certo ponto, podemos arbitrar os símbolos que nos interessam, e incluir apenas esses símbolos no alfabeto. Para cada aplicação específica, o usuário deve escolher o alfabeto que pretende utilizar: para exemplos no quadro negro, alfabetos como {0, 1}, e {a, b} são boas escolhas; para ensinar a linguagem Pascal, o alfabeto escolhido deverá conter símbolos como *program*, *begin*, *end*, *if*, *then*, *else*; para implementar a linguagem Pascal, provavelmente o alfabeto adequado não conterá símbolos como os vistos acima, mas sim caracteres como os do conjunto ASCII (letras, dígitos, +, *, etc.), uma vez que são estes os componentes básicos de um arquivo fonte.

Cadeias. Formalmente, uma *cadeia de símbolos* em um alfabeto Σ pode ser definida como uma função: uma sequência s de comprimento n no alfabeto Σ, é uma função $s{:}[n] \rightarrow \Sigma$, com domínio $[n]$, e com contradomínio Σ. O número natural n é o *comprimento* de s, e é representado por $|s|$.

Por exemplo, se o alfabeto considerado for Σ = {a, b, c}, a sequência de comprimento 4 (composta por quatro ocorrências de símbolos) s = cbba pode ser vista como a função $s{:}[4] \rightarrow \Sigma$, definida por $s(1)$ = c, $s(2)$ = b, $s(3)$ = b, $s(4)$ = a.

Concatenação. A principal operação sobre sequências é a operação de concatenação. Informalmente, o resultado da concatenação das sequências x e y é a sequência xy, ou $x \circ y$, composta pelos símbolos de x, seguidos pelos símbolos de y, nessa ordem. Mais formalmente, dadas duas sequências (funções) $x{:}[m] \rightarrow \Sigma$ e $y{:}[n] \rightarrow \Sigma$, de comprimentos m e n, no mesmo alfabeto Σ, definimos a sequência (função) $x \circ y{:}[m+n] \rightarrow \Sigma$, de comprimento $m+n$, por

$$x \circ y(i) = \begin{cases} x(i), \text{ se } i \leq m \\ y(i-m), \text{ se } i > m \end{cases}$$

[1] Em geral, é possível usar alguma forma de *codificação*, e representar cada símbolo de um alfabeto infinito *enumerável* através de uma sequência de símbolos de um alfabeto finito.

Assim, se tivermos Σ = {a, b, c}, x = cbba, e y = ac, teremos $x \circ y$ = cbbaac. Representando as funções através de tabelas, temos:

i	$x(i)$
1	c
2	b
3	b
4	a

i	$y(i)$
1	a
2	c

i	$x \circ y(i)$
1	c
2	b
3	b
4	a
5	a
6	c

Naturalmente, $|x \circ y| = |x| + |y|$.

No que se segue, em geral não faremos referência ao fato de que sequências são funções. Se considerarmos símbolos x_1 , x_2 , … x_n , de um alfabeto Σ, representaremos a sequência formada por ocorrências desses símbolos, nessa ordem, por $x_1x_2\ldots x_n$. Note que no caso especial n = 1, a notação acima confunde a sequência a de comprimento 1 com o símbolo a. Esta ambiguidade não causa maiores problemas.

Um outro caso especial importante é o caso n = 0, em que falamos da *sequência vazia*, indicada por ε. Usamos um nome "ε" para a sequência vazia simplesmente porque não é possível usar a mesma notação das outras sequências. Afinal, se escrevermos zero símbolos, como convencer alguém de que alguma coisa foi escrita? A sequência vazia ε é o elemento neutro (identidade) da concatenação: qualquer que seja a sequência x, temos

$$x \circ \varepsilon = \varepsilon \circ x = x$$

Linguagens. Dado um alfabeto Γ, uma *linguagem* em Γ é um conjunto de sequências de símbolos de Γ.

O conjunto de todas as sequências de símbolos de um alfabeto Γ é uma linguagem, indicada por Γ*. A linguagem Γ* inclui todas as sequências de símbolos de Γ, incluindo também a sequência vazia ε. Com essa notação, uma linguagem L em Γ é um subconjunto de Γ*, ou seja, $L \subseteq \Gamma^*$.

Note que todas essas sequências satisfazem a definição anterior, e tem como comprimento um número natural finito. Podemos assim ter linguagens infinitas, mesmo sem considerar sequências infinitas.

Exemplo: Os conjuntos a seguir são linguagens em Γ = {a, b}.

∅
{ε}
{a, aa, aaa}
{a, b}*
{x | |x| é par}
{a, b} — *notação ambígua*

Nota: Já observamos que a notação aqui usada é ambígua; essa ambiguidade se torna aparente neste último exemplo: {a, b} pode ser uma linguagem (conjunto de sequências) ou um alfabeto (conjunto de símbolos). Isto vale para qualquer alfabeto Γ. Se isso fosse necessário, a ambiguidade poderia ser evitada usando-se uma notação apropriada: em uma das notações possíveis, representamos *sequências entre aspas*, e *símbolos entre plicas*, de forma que "a" fica sendo a sequência, 'a' o símbolo, {"a", "b"} a linguagem e {'a', 'b'} o alfabeto. Para nós, entretanto, tais distinções não serão necessárias.

Operações com linguagens. Linguagens são conjuntos, de forma que as operações de conjuntos podem ser diretamente usadas com linguagens. Assim, não há necessidade de definir *união*, *interseção* ou *diferença* de linguagens; no caso do *complemento*, podemos usar como *universo* o conjunto Γ* de todas as sequências no alfabeto considerado Γ.

Se L e M são linguagens em Γ, temos:

$$\begin{aligned} \textit{união:} \quad & L \cup M = \{ x \mid x \in L \text{ ou } x \in M \} \\ \textit{interseção:} \quad & L \cap M = \{ x \mid x \in L \text{ e } x \in M \} \\ \textit{diferença:} \quad & L - M = \{ x \mid x \in L \text{ e } x \notin M \} \\ \textit{complemento:} \quad & \overline{L} = \Gamma^* - L = \{ x \in \Gamma^* \mid x \notin L \} \end{aligned}$$

Exemplo: Seja o alfabeto Γ = {a, b, c} , e sejam as linguagens L = {a, bc, cb} e M = {aa, bb, cc, bc, cb} , em Γ. Temos:

$L \cup M = \{a, bc, cb, aa, bb, cc\}$
$L \cap M = \{bc, cb\}$
$L - M = \{a\}$
$M - L = \{aa, bb, cc\}$
$\overline{L} = \{ x \in \Gamma^* \mid x \neq a, x \neq bc \text{ e } x \neq cb \} =$
$= \{\varepsilon, b, aa, ab, ac, ba, bb, ca, cb, cc, aaa, \ldots \}$

□

Concatenação de linguagens. A operação de concatenação, que foi definida para sequências, pode ser estendida a linguagens:

$$L \circ M = \{ x \circ y \mid x \in L \text{ e } y \in M \} .$$

Exemplo: Sendo L e M como no exemplo anterior,

$L \circ M = \{aaa, abb, acc, abc, acb, bcaa, bcbb, bccc, bcbc,$
$bccb, cbaa, cbbb, cbcc, cbbc, cbcb\}$
$M \circ L = \{aaa, aabc, aacb, bba, bbbc, bbcb, cca, ccbc,$
$cccb, bca, bcbc, bccb, cba, cbbc, cbcb\}$

□

Fato: A linguagem I = { ε } é o elemento neutro (identidade) da concatenação de linguagens, ou seja, para qualquer linguagem L,

$L \circ I = I \circ L = L.$

Dem.: Exercício.

□

Potências. Podemos introduzir as potências L^i (para *i* natural) de uma linguagem L através de uma definição recursiva:

$L^0 = \{ \varepsilon \}$

$L^{i+1} = L \circ L^i$, para qualquer $i \in$ **Nat**.

Exemplo: Seja L = {0, 11}. Então temos:

$L^0 = \{ \varepsilon \}$
$L^1 = L \circ L^0 = \{ \varepsilon \} \circ \{0, 11\} = \{0, 11\}$
$L^2 = L \circ L^1 = \{0, 11\} \circ \{0, 11\} = \{00, 011, 110, 1111\}$
$L^3 = L \circ L^2 = \{0, 11\} \circ \{00, 011, 110, 1111\} =$
$= \{000, 0011, 0110, 01111, 1100, 11011, 11110, 111111\}$
e assim por diante.

□

Fato:

(1) Para qualquer linguagem L, $L^1 = L$.

(2) Para qualquer linguagem L, temos $L^i \circ L^j = L^{i+j}$, para *i* e *j* quaisquer.

Dem.: Exercício.

□

Fechamento. Podemos definir, para uma linguagem L qualquer, o seu fechamento L* como sendo a união de todas as potências de L:

$$L^* = \bigcup_{i=0}^{\infty} L^i = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \ldots$$

Outra notação frequentemente utilizada é L^+, que indica a união de todas as potências de L, excluída a potência 0:

$$L^+ = \bigcup_{i=1}^{\infty} L^i = L^1 \cup L^2 \cup L^3 \cup \ldots$$

Exemplo: Para a linguagem L do exemplo anterior, temos:

$L^* = \{ \varepsilon, 0, 11, 00, 011, 110, 1111, 000, 0011, 0110,$
$01111, 1100, 11011, 11110, 111111, \ldots\}$

$L^+ = \{ 0, 11, 00, 011, 110, 1111, 000, 0011, 0110,$
$01111, 1100, 11011, 11110, 111111, \ldots\}$

Exercício: Caracterize a classe das linguagens L para as quais $L^* = L^+$.

□

Fato: Para qualquer alfabeto Γ, o conjunto Γ* de todas as sequências de símbolos de Γ é enumerável. (Note que Γ pode ser considerado um alfabeto ou uma linguagem, sem que isso altere o valor de Γ*.)

Dem. Considere (por exemplo) a seguinte enumeração:

1. Escolha uma ordem qualquer (ordem alfabética) para os elementos do alfabeto Γ.
2. Enumere as sequências por ordem crescente de comprimento, e, dentro de cada comprimento, por ordem alfabética. Por exemplo, se Γ = {a, b, c}, a enumeração pode ser

   ε, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, …

Observamos que a ordem alfabética simples não seria adequada para as cadeias, uma vez que teríamos, neste caso,

ε, a, aa, aaa, aaaa, aaaaa, ….

excluindo, portanto, da enumeração, todas as sequências que não pertencessem a $\{a\}^*$ Assim, a cadeia b, por exemplo, *nunca seria atingida*.

O esquema apresentado se baseia no fato de que, para cada comprimento, o número de sequências de comprimento n é finito. Assim, se Γ pudesse ser infinito, a enumeração não seria possível, como descrita, uma vez que não teríamos um número finito de sequências para cada comprimento.

□

No que se segue, usaremos a notação x_i para representar a *i*-ésima sequência numa enumeração de Γ*, para um alfabeto Γ fixo, supondo uma ordenação fixada para Γ*.

Fato: Qualquer linguagem é enumerável.

Dem.: Toda linguagem em Γ é um subconjunto de Γ*.

□

Fato: A classe de todas as linguagens em um alfabeto Γ não é enumerável.

Dem.: A classe de todas as linguagens em Γ é P(Γ*), e já vimos que o conjunto formado por todos os subconjuntos de um conjunto enumerável infinito não é enumerável.

□

Exercício: Aplique a técnica da diagonalização diretamente, para mostrar que a classe P(Γ*) de todas as linguagens no alfabeto Γ não é enumerável.

□

Exercício: Suponha Γ = { a, b, c }.

(a) Escreva um programa que, quando recebe como entrada o natural i, determina a cadeia $x_i \in \Gamma^*$.

(b) Escreva um programa que, quando recebe como entrada uma cadeia $x \in \Gamma^*$, determina o natural i tal que $x = x_i$.

rev. 17/6/96