Linguagens Formais e Autômatos

Aula 4: Conceitos de *Linguagem*

Prof. Dr. Rodrigo Xavier de Almeida Leão

Cientista de Dados e Big Data

Linguagens

Uma **linguagem formal** é um conjunto, finito ou infinito, de cadeias de comprimento finito, formadas pela concatenação de elementos de um alfabeto finito e não-vazio. Além das operações previamente definidas para conjuntos, como união, diferença, intersecção etc., outras operações, tais como a concatenação e os fechamentos, também são fundamentais para a definição e o estudo das linguagens formais.

Antes de apresentá-las, convém notar a distinção que há entre os seguintes conceitos: cadeia vazia ϵ, conjunto vazio $\emptyset$ e o conjunto que contém apenas a cadeia vazia $\{\epsilon\}$.

Linguagens

O primeiro deles, ϵ, denota a **cadeia** vazia, ou seja, uma cadeia de comprimento zero, ao passo que os dois seguintes são casos particulares de **linguagens** (que por sua vez são conjuntos): $\varnothing$ denota uma linguagem vazia, ou seja, uma linguagem que não contém nenhuma cadeia, e $\{\epsilon\}$ denota uma linguagem que contém uma única cadeia, a cadeia vazia. Observe-se que $|\varnothing| = 0$ e $|\{\epsilon\}| = 1$.

Linguagens

Note-se a diferença conceitual que há entre alfabetos, linguagens e cadeias. Alfabetos são conjuntos, finitos e não-vazios, de símbolos, através de cuja concatenação são obtidas as **cadeias**. Linguagens, por sua vez, são conjuntos, finitos (eventualmente vazios) ou infinitos, de cadeias. Uma cadeia é também denominada **sentença** de uma linguagem, ou simplesmente sentença, no caso de ela pertencer à linguagem em questão. Linguagens são, portanto, coleções de sentenças sobre um dado alfabeto.

A Figura 2.1 ilustra a relação entre os conceitos de símbolo, alfabeto, cadeia e linguagem:

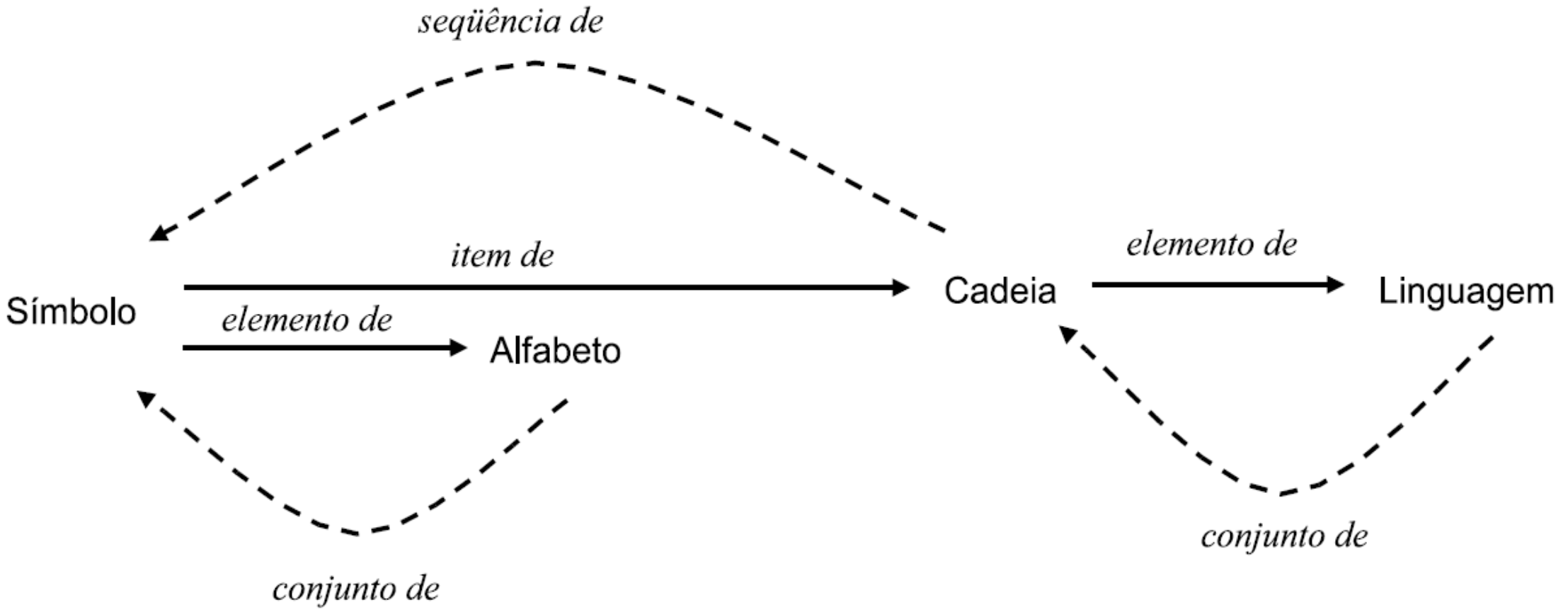

Figura 2.1: Símbolo, alfabeto, cadeia e linguagem

Linguagens

As várias leituras contidas na Figura 2.1 são: “símbolo é elemento de alfabeto”; “alfabeto é conjunto de símbolos”; “símbolo é item de cadeia”; “cadeia é seqüência de símbolos”; “cadeia é elemento de linguagem” e “linguagem é conjunto de cadeias”.

Outra maneira de associar significados aos termos “símbolo”, “alfabeto”, “cadeia” e “linguagem” é apresentada na Figura 2.2, que também ilustra o conceito de “sentença”.

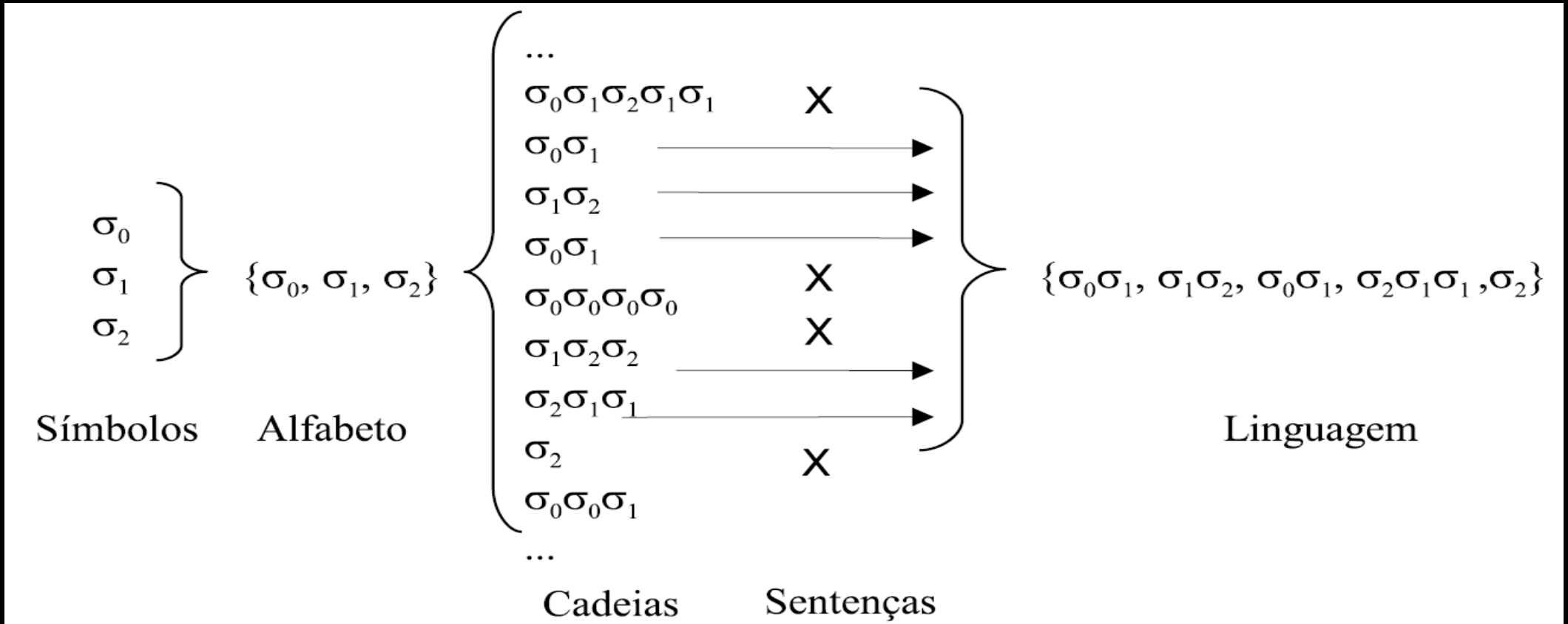

Figura 2.2: Símbolo, alfabeto, cadeia, sentença e linguagem

A Figura 2.2 facilita o entendimento das relações entre os conceitos: (i) um conjunto de símbolos forma um alfabeto, (ii) a partir de um alfabeto (finito) formam-se (infinitas) cadeias; (iii) determinadas cadeias são escolhidas para fazer parte de uma linguagem; (iv) uma linguagem é um conjunto de cadeias, que por isso são também denominadas sentenças.

Linguagens

Exemplo 2.6 O símbolo a é elemento do alfabeto $\{a\}$ e também um item da cadeia aaa, que por sua vez é elemento da linguagem $\{aaa\}$. Por outro lado, a linguagem $\{aaa\}$ é um conjunto que contém a cadeia aaa, a cadeia aaa é uma seqüência de símbolos a e o alfabeto $\{a\}$ contém o símbolo a. A Figura 2.3 ilustra esses conceitos, conforme a Figura 2.1.

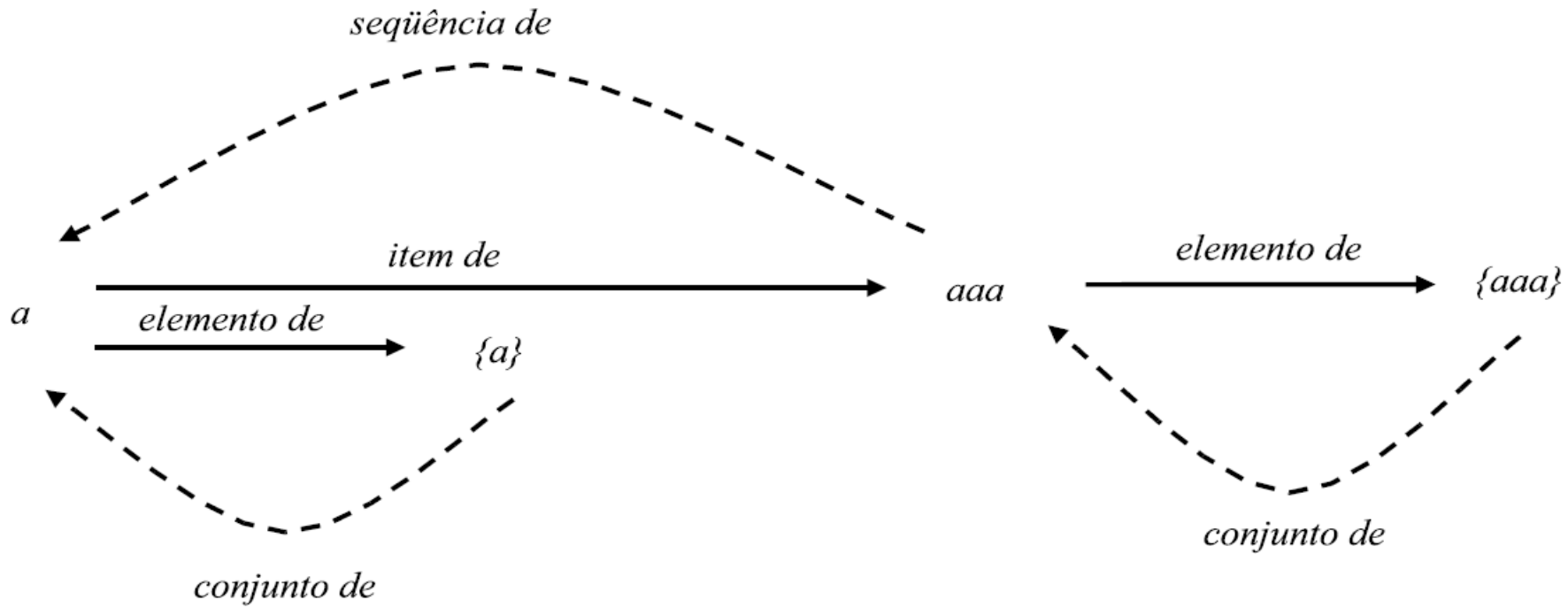

Figura 2.3: $a, \{a\}, aaa, \{aaa\}$

Linguagens

Exemplo 2.7 A Figura 2.4 ilustra uma aplicação dos conceitos da Figura 2.2 ao alfabeto $\{a, b\}$. A linguagem apresentada é, naturalmente, apenas uma das inúmeras que podem ser criadas a partir desse alfabeto.

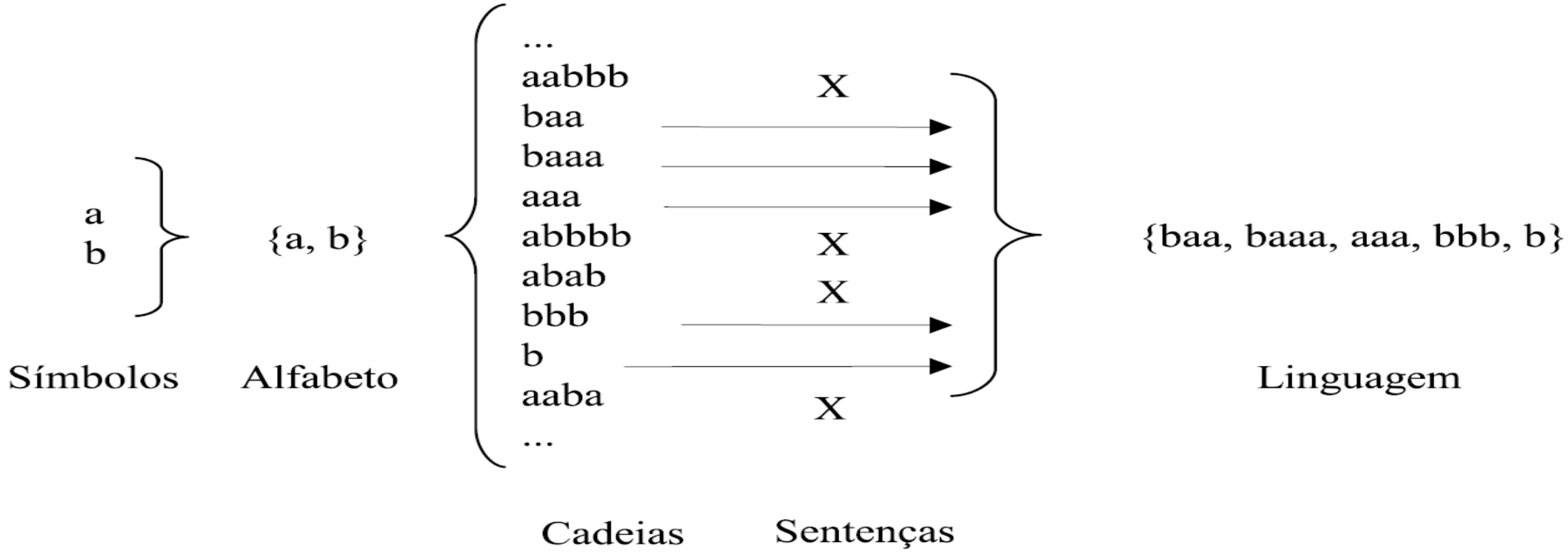

Figura 2.4: a, b, cadeias, sentenças e linguagem

Linguagens

Exemplo 2.8 Considere-se $\Sigma = \{a, b, c\}$. Então,

$$
\begin{aligned}
\Sigma^0 &= \{\epsilon\} \\
\Sigma^1 &= \{a, b, c\} \\
\Sigma^2 &= \{aa, ab, ac, ba, bb, bc, ca, cb, cc\} \\
\Sigma^3 &= \{aaa, aab, aac, aba, abb, abc, \ldots, ccc\}
\end{aligned}
$$

etc.

Linguagens

O **fechamento reflexivo e transitivo** (às vezes chamado **fechamento recursivo e transitivo**) de um alfabeto Σ é definido como o conjunto (infinito) que contém todas as possíveis cadeias que podem ser construídas sobre o alfabeto dado, incluindo a cadeia vazia. Esse conjunto, denotado por Σ^*, contém, naturalmente, todas as cadeias que podem ser definidas sobre o alfabeto Σ. Formalmente, o fechamento reflexivo e transitivo de um conjunto Σ é definido como:

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

as possíveis cadeias sobre Σ, então toda e qualquer linguagem L sobre um alfabeto Σ sempre poderá ser definida como sendo um subconjunto de Σ^*, ou seja, $L \subseteq \Sigma^*$.

Linguagens

- $\varnothing$ é o conjunto constituído por zero cadeias e corresponde à menor linguagem que se pode definir sobre um alfabeto Σ qualquer;
- Σ^* é o conjunto de todas as cadeias possíveis de serem construídas sobre Σ e corresponde à maior de todas as linguagens que pode ser definida sobre Σ;
- 2^{Σ^*} é o conjunto de todos os subconjuntos possíveis de serem obtidos a partir de Σ^*, e corresponde ao conjunto formado por todas as possíveis linguagens que podem ser definidas sobre Σ. Observe-se que $\varnothing \in 2^{\Sigma^*}$, e também que $\Sigma^* \in 2^{\Sigma^*}$.

Linguagens

Exemplo 2.9 Seja $\Sigma = \{a, b, c\}$ e P o conjunto formado pela única propriedade "todas as cadeias são iniciadas com o símbolo a". Então:

- A linguagem $L_0 = \varnothing$ é a menor linguagem que pode ser definida sobre Σ;
- A linguagem $L_1 = \{a, ab, ac, abc, acb\}$ é finita e observa P;
- A linguagem $L_2 = \{a\}\{a\}^*\{b\}^*\{c\}^*$ é infinita e observa P;
- A linguagem $L_3 = \{a\}\{a, b, c\}^*$ é infinita, observa P e, dentre todas as que observam P, trata-se da maior linguagem, pois não existe nenhuma outra cadeia em Σ^* que satisfaça a P e não pertença a L_3;
- $L_0 \subseteq \Sigma^*$, $L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Sigma^*$, $L_3 \subseteq \Sigma^*$;
- $L_0 \in 2^{\Sigma^*}$, $L_1 \in 2^{\Sigma^*}$, $L_2 \in 2^{\Sigma^*}$, $L_3 \in 2^{\Sigma^*}$;
- Além de L_0, L_1, L_2 e L_3, existem inúmeras outras linguagens que podem ser definidas sobre Σ.

Linguagens

A **complementação** de uma linguagem X definida sobre um alfabeto Σ é definida como:

$$\overline{X} = \Sigma^* - X$$

Diz-se que uma linguagem exibe a propriedade do **prefixo (sufixo) próprio** sempre que não houver nenhuma cadeia a ela pertencente que seja prefixo (sufixo) próprio de outra cadeia dessa mesma linguagem. Formalmente:

Linguagens

- Prefixo próprio: não existe $\alpha \in L \mid \beta \neq \epsilon$ e $\alpha\beta \in L$
- Sufixo próprio: não existe $\alpha \in L \mid \beta \neq \epsilon$ e $\beta\alpha \in L$

Exemplo 2.11 Considere as seguintes linguagens:

$$L_1 = \{a^i b^i \mid i \geq 1\} = \{ab, aabb, aaabbb, ...\}$$
$$L_2 = \{ab^i \mid i \geq 1\} = \{ab, abb, abbb, abbbb, ...\}$$

Neste exemplo, a linguagem L_1 exibe a propriedade do prefixo próprio, ao passo que a linguagem L_2 não a exibe. A propriedade do sufixo próprio é exibida por ambas as linguagens. □

Linguagens

Diz-se que uma linguagem L_1 é o **reverso** de uma linguagem L_2, denotando-se o fato por $L_1 = L_2^R$ (ou $L_2 = L_1^R$), quando as sentenças de L_1 corresponderem ao reverso das sentenças de L_2. Formalmente:

$$L_1 = L_2^R = \{x^R \mid x \in L_2\}$$

Exemplo 2.12 Seja $L_2 = \{\epsilon, a, ab, abc\}$. Então, $L_1 = L_2^R = \{\epsilon, a, ba, cba\}$. □

Linguagens

Exemplo 2.13 Considerem-se as linguagens:

$$
\begin{aligned}
L_1 &= \{a^i b \mid i \geq 0\} \\
L_2 &= \{a^i b c^i \mid i \geq 0\} \\
L_3 &= \{b\} \\
L_4 &= \{a^i b \mid i \geq 1\} \\
L_5 &= \{b c^i \mid i \geq 0\} \\
L_6 &= \{c^i b \mid i \geq 0\} \\
L_7 &= \{a^i \mid i \geq 0\}
\end{aligned}
$$

- $L_1 / L_3 = L_7$:

  $L_1 = \{b, ab, aab, aaab...\}$

  $L_3 = \{b\}$

- $L_1 / L_4 = L_7$:

  $L_1 = \{b, ab, aab, aaab...\}$

  $L_4 = \{ab, aab, aaab...\}$

- $L_5 / L_7 = \varnothing$

  $L_5 = \{b, bc, bcc, bccc...\}$

  $L_7 = \{ab, aab, aaab...\}$

Linguagens

Exemplo 2.13 Considerem-se as linguagens:

$$
\begin{aligned}
L_1 &= \{a^i b \mid i \geq 0\} \\
L_2 &= \{a^i b c^i \mid i \geq 0\} \\
L_3 &= \{b\} \\
L_4 &= \{a^i b \mid i \geq 1\} \\
L_5 &= \{b c^i \mid i \geq 0\} \\
L_6 &= \{c^i b \mid i \geq 0\} \\
L_7 &= \{a^i \mid i \geq 0\}
\end{aligned}
$$

- $L_2 / L_1 = L_7$

  $L_2 = \{b, ab, bc, abc, aab, bcc, aabc, abcc, aabcc...\}$

  $L_1 = \{b, ab, aab, aaab...\}$

- $L_2 / L_6 = L_7$

  $L_2 = \{b, ab, bc, abc, aab, bcc, aabc, abcc, aabcc...\}$

  $L_6 = \{b, cb, ccb, cccb...\}$

Linguagens

Exemplo 2.13 Considerem-se as linguagens:

$$
\begin{aligned}
L_1 &= \{a^i b \mid i \geq 0\} \\
L_2 &= \{a^i b c^i \mid i \geq 0\} \\
L_3 &= \{b\} \\
L_4 &= \{a^i b \mid i \geq 1\} \\
L_5 &= \{b c^i \mid i \geq 0\} \\
L_6 &= \{c^i b \mid i \geq 0\} \\
L_7 &= \{a^i \mid i \geq 0\}
\end{aligned}
$$

- $L_5 / L_2 = \{\epsilon\}$

  $L_5 = \{b, bc, bcc, bccc...\}$

  $L_2 = \{b, ab, bc, abc, aab, bcc, aabc, abcc, aabcc...\}$

Linguagens

Uma **substituição** é uma função que mapeia os elementos de um alfabeto Σ_1 em linguagens sobre um alfabeto Σ_2. Formalmente:

$$s : \Sigma_1 \rightarrow 2^{\Sigma_2^*}$$

Linguagens

Exemplo 2.14 Considere-se a linguagem $L = \{a^i b^i c^i \mid i \geq 1\}$ e a substituição s sobre $\Sigma = \{x, y, z\}$:

- $s(a) = \{x\}$;
- $s(b) = \{y, yy\}$;
- $s(c) = \{z, zz, zzz\}$.

A aplicação de s em L define a linguagem $s(L) = \{x^i y^j z^k \mid i \geq 1, i \leq j \leq 2i, i \leq k \leq 3i\}$.
A título de ilustração, serão apresentadas a seguir todas as transliterações possíveis para a cadeia abc obtidas através da substituição s acima definida:

$$s(abc) = \{xyz, xyzz, xyzzz, xyyz, xyyzzz\}$$

Linguagens

Note-se que, neste exemplo, $s(abc) = s(a)s(b)s(c)$, onde:

$s(a)$ pode ser substituído apenas por x
$s(b)$ pode ser substituído por y ou yy
$s(c)$ pode ser substituído por z, zz ou zzz

O conjunto $s(abc)$ é obtido a partir de todas as combinações possíveis de substituições para os símbolos a, b e c na cadeia abc conforme s. □

Linguagens

Diz-se que uma substituição é um **homomorfismo** se $2^{\Sigma_2^*}$ contiver apenas um elemento para cada $\sigma \in \Sigma_1$. Neste caso, costuma-se considerar os elementos do conjunto imagem como simples cadeias e não como conjuntos compostos por um único elemento:

$$h : \Sigma_1 \rightarrow \Sigma_2^*$$

Assim, um homomorfismo é uma função que mapeia cada símbolo de um alfabeto Σ_1 em uma cadeia única contida em uma linguagem Σ_2^*.

Linguagens

Exemplo 2.15 Considere-se agora a mesma linguagem L do Exemplo 2.14 e o homomorfismo h sobre o respectivo alfabeto Σ:

- $h(a) = x$;
- $h(b) = x$;
- $h(c) = z$.

A linguagem definida por esse homomorfismo é $h(L) = \{x^{2i} z^i \mid i \geq 1\}$. □

Linguagens

Nos casos em que h é um homomorfismo, diz-se que $h(L)$ é a **imagem homomórfica** de L. A **imagem homomórfica inversa** de uma cadeia y é definida através de um **homomorfismo inverso** h^{-1}:

$$h^{-1}(y) = \{x \mid y = h(x)\}$$

Observe-se que a imagem homomórfica inversa de uma cadeia é definida como um conjunto de cadeias, e não como uma cadeia única, uma vez que a função h não é necessariamente injetora. A imagem homomórfica inversa de uma linguagem L é definida como:

$$h^{-1}(L) = \{x \mid h(x) \in L\}$$

Exemplo 2.16 Ainda no Exemplo 2.3, a imagem homomórfica inversa de $h(L)$ pode ser definida através do homomorfismo inverso h^{-1}:

- $h^{-1}(x) = \{a, b\}$;
- $h^{-1}(z) = \{c\}$

Pelo fato de a função h originalmente adotada não ser usualmente injetora, obtém-se:

$$h^{-1}(h(L)) = \{(a \mid b)^{2i} c^i \mid i \geq 1\}$$

e, portanto, $h^{-1}(h(L)) \neq L$. □

Linguagens

Um homomorfismo é denominado **isomorfismo** sempre que a função h for injetora. Neste caso, $h(L)$ é denominada **imagem isomórfica** de L. Isomorfismos viabilizam a definição de imagens homomórficas inversas através de funções injetoras denominadas **isomorfismos inversos**. Neste caso, a linguagem $h^{-1}(L)$ é denominada **imagem isomórfica inversa** de L.

Exemplo 2.17 Considere-se a linguagem $L_1 = \{a^i b^i \mid i \geq 1\}$ sobre o alfabeto $\Sigma_1 = \{a, b\}$ e o isomorfismo j, definido sobre $\Sigma_2 = \{c, d\}$:

- $j(a) = c$;
- $j(b) = dd$.

Linguagens

Neste caso, a imagem isomórfica de L_1, ou seja, a linguagem L_2, resultante da aplicação do isomorfismo j sobre o alfabeto da linguagem L_1, é a seguinte:

$$L_2 = \{c^i d^{2i} \mid i \geq 1\}$$

A imagem isomórfica inversa de L_2, ou seja, a linguagem L_1, pode ser obtida a partir do isomorfismo inverso:

- $j^{-1}(c) = a$;
- $j^{-1}(dd) = b$.

Observe, neste caso, que $j^{-1}(j(L_1)) = L_1$. □

Linguagens

Isomorfismos e isomorfismos inversos são muito úteis, pois permitem mudar de Σ_1 para Σ_2 o domínio na resolução de problemas, visando com isso facilitar a sua resolução no domínio Σ_2. Por se tratar de um mapeamento feito através de uma função injetora, torna-se sempre possível retornar ao domínio original Σ_1, preservando-se a associação unívoca entre os elementos dos domínios considerados.

Linguagens

Exemplo 2.18 Considere-se $L = \mathbb{N}$ e o isomorfismo j, sobre $\Sigma = \{0, 1\}$:

- $j(0) = 0000$
- $j(1) = 0001$
- $j(2) = 0010$
- $j(3) = 0011$
- $j(4) = 0100$
- $j(5) = 0101$
- $j(6) = 0110$
- $j(7) = 0111$
- $j(8) = 1000$
- $j(9) = 1001$

- $j^{-1}(0000) = 0$
- $j^{-1}(0001) = 1$
- $j^{-1}(0010) = 2$
- $j^{-1}(0011) = 3$
- $j^{-1}(0100) = 4$
- $j^{-1}(0101) = 5$
- $j^{-1}(0110) = 6$
- $j^{-1}(0111) = 7$
- $j^{-1}(1000) = 8$
- $j^{-1}(1001) = 9$

Linguagens

$j(L)$ mapeia os números naturais na representação equivalente em BCD (*Binary Coded Decimal*). O isomorfismo inverso j^{-1}, abaixo, permite retornar ao domínio dos números naturais:

As linguagens de interesse prático, como é o caso das linguagens de programação, normalmente correspondem a um subconjunto próprio do fechamento reflexivo e transitivo do alfabeto sobre o qual as suas cadeias são construídas. Assim, tornam-se necessários métodos e notações que permitam, dentro do conjunto fechamento, identificar as cadeias que efetivamente pertencem à linguagem que estiver sendo definida, descartando-se as demais

Linguagens

Um outro aspecto importante, referente à definição rigorosa das linguagens formais, diz respeito ao fato de que, em sua maioria, as linguagens de interesse contêm, se não uma quantidade infinita, ao menos um número finito, porém muito grande, de cadeias.

Por esses dois motivos, há um interesse muito grande em relação a métodos que permitam especificar linguagens, sejam elas finitas ou não, através de representações finitas. Por outro lado, nem todas as linguagens podem ser representadas por meio de uma especificação finita. Apesar do reduzido interesse prático que recai sobre tais linguagens, é possível comprovar a existência de linguagens para as quais é impossível se obter uma representação finita.

Neste texto serão considerados três métodos dentre os mais empregados para a representação finita de linguagens:

1. **Gramáticas**: correspondem a especificações finitas de dispositivos de geração de cadeias. Um dispositivo desse tipo deve ser capaz de gerar toda e qualquer cadeia

Gramáticas e Reconhecedores

Gramáticas: correspondem a especificações finitas de dispositivos de geração de cadeias. Um dispositivo desse tipo deve ser capaz de gerar toda e qualquer cadeia pertencente à linguagem definida pela gramática, e nada mais. Assim, as cadeias não pertencentes à linguagem não devem poder ser geradas pela gramática em questão. Essa forma de especificação é aplicável para linguagens finitas e infinitas.

Reconhecedores: correspondem a especificações finitas de dispositivos de aceitação de cadeias. Um dispositivo desse tipo deverá aceitar toda e qualquer cadeia pertencente à linguagem por ele definido, e rejeitar todas as cadeias não-pertencentes à linguagem. O método é aplicável para a especificação formal de linguagens finitas e infinitas.

Gramáticas e Reconhecedores

Enumerações: relacionam, de forma explícita e exaustiva, todas as cadeias pertencentes à particular linguagem a ser especificada. Toda e qualquer cadeia pertencente à linguagem deve constar desta relação. As cadeias não pertencentes à linguagem não fazem parte dessa relação. Aplicam-se apenas para a especificação de linguagens finitas e preferencialmente não muito extensas.

Na prática, as gramáticas e os reconhecedores, além de oferecerem uma grande concisão na representação de linguagens de cardinalidade elevada, também podem ser empregados na definição de linguagens infinitas, ao contrário das enumerações. Em contraste com o que ocorre com as enumerações, as gramáticas e os reconhecedores geralmente possibilitam uma percepção melhor da estrutura sintática inerente às sentenças das linguagens por eles definidas.

Gramáticas e Reconhecedores

Diz-se que uma gramática é **equivalente** a um reconhecedor se as duas seguintes condições forem simultaneamente verificadas (lembrar que os formalismos devem definir todas as sentenças da linguagem desejada, e nenhuma outra cadeia):

1. Toda cadeia gerada pela gramática é também aceita pelo reconhecedor.

2. Toda cadeia aceita pelo reconhecedor é também gerada pela gramática.

Gramáticas e Reconhecedores

Exemplo 2.20 Considerem-se a gramática G e o reconhecedor M, respectivamente definidos através das linguagens gerada e aceita:

- $G \mid L_1(G) = \{\alpha \in \{a, b\}^* \mid$ o primeiro símbolo de α é "a"$\}$
- $M \mid L_2(M) = \{\beta \in \{a, b\}^* \mid$ o primeiro símbolo de β é "a" e o último símbolo é "b"$\}$

Portanto:

- $L_1 = \{a, aa, ab, aaa, aab, aba, abb, aaa...\}$
- $L_2 = \{ab, aab, abb, aaab, aabb, abab, abbb...\}$

É fácil perceber que a condição (2) acima é verificada, mas a condição (1) não. Por exemplo, a cadeia $ab \in L_2$ e $ab \in L_1$. Por outro lado, a cadeia $aba \in L_1$, porém $aba \notin L_2$. Logo, $L_1 \subset L_2$ e não se pode dizer que G e M sejam equivalentes. □

Gramáticas e Reconhecedores

Conforme discutido mais adiante, as gramáticas e os reconhecedores são formas duais de representação de linguagens, ou seja, para cada gramática é possível obter um reconhecedor que aceite a linguagem correspondente e vice-versa. À particular notação utilizada para representar uma linguagem, seja através de gramáticas ou de reconhecedores, dá-se o nome de **metalinguagem**.

Gramática

Como um primeiro exemplo de gramática, suponha que estamos interessados em especificar a linguagem L_D dos números, numerais para sermos mais precisos, não negativos na base decimal. Estamos interessados não só nos inteiros, como também nos números com ponto decimal. Assim como nas principais linguagens de programação, usaremos ponto decimal (e não vírgula) para separar a parte decimal do número. Exemplos de números nesta linguagem são $L_D = \{0,1,2,\ldots 9,10,11,\ldots,99,100,\ldots,0.1,0.2,\ldots\}$. Neste exemplo fica clara a dificuldade de especificação de uma linguagem infinita. O alfabeto sobre o qual L_D está definida é $\Sigma = \{0,1,2,3,4,5,6,7,8,9,.\}$, ou seja, os 10 dígitos decimais mais o ponto. Uma forma de especificarmos isso é através das regras a seguir:

Gramática

- $N \rightarrow L$ (um número N pode ser uma lista de dígitos L)
- $N \rightarrow L.L$ (N pode ser uma lista de dígitos L seguida de outra lista L)
- $L \rightarrow D$ (uma lista de dígitos L pode ser um dígito D)
- $L \rightarrow LD$ (uma lista de dígitos L pode ser outra lista L seguida de um dígito D)

Gramática

- $D \rightarrow 0$ (uma dígito D pode ser o dígito 0)
- $D \rightarrow 1$ (uma dígito D pode ser o dígito 1)
- $D \rightarrow 2$ (uma dígito D pode ser o dígito 2)
- $D \rightarrow 3$ (uma dígito D pode ser o dígito 3)
- $D \rightarrow 4$ (uma dígito D pode ser o dígito 4)
- $D \rightarrow 5$ (uma dígito D pode ser o dígito 5)
- $D \rightarrow 6$ (uma dígito D pode ser o dígito 6)
- $D \rightarrow 7$ (uma dígito D pode ser o dígito 6)
- $D \rightarrow 8$ (uma dígito D pode ser o dígito 8)
- $D \rightarrow 9$ (uma dígito D pode ser o dígito 9)

Gramática

Neste caso, podemos concluir que 1.23 é uma cadeia especificada por esta gramática, porque: N é um número (numeral); devido à regra $N \rightarrow L.L$, $L.L$ é um número (numeral decimal); mas devido à regra $L \rightarrow D$, $D.L$ é um numeral decimal. Seguimos a sequência, sem indicar as regras: Se $D.L$ é um numeral, $1.L$ é um numeral, portanto $1.LD$ é um numeral, assim $1.DD$ é um numeral, portanto $1.2D$ é um numeral, logo 1.23 também é um numeral.

Neste caso dizemos que N gera 1.23. O processo pode ser representado pelo símbolo $\Rightarrow$, da seguinte forma:

$$N \Rightarrow L.L \Rightarrow D.L \Rightarrow 1.L \Rightarrow 1.LD \Rightarrow 1.DD \Rightarrow 1.2D \Rightarrow 1.23$$

Gramática

Neste caso dizemos que N gera 1.23. O processo pode ser representado pelo símbolo $\Rightarrow$, da seguinte forma:

$$N \Rightarrow L.L \Rightarrow D.L \Rightarrow 1.L \Rightarrow 1.LD \Rightarrow 1.DD \Rightarrow 1.2D \Rightarrow 1.23$$

A essa especificação damos o nome de gramática. Observando a especificação com mais detalhes, podemos observar que ela possui os seguintes elementos:

- Um alfabeto sobre o qual a linguagem é definida, chamamos os elementos deste alfabeto de símbolos terminais: $T = \{0,1,2,3,4,5,6,7,8,9,.\}$
- Um conjunto de símbolos auxiliares, aos quais chamamos de símbolos não terminais ou variáveis: $V = \{N, L, D\}$;

Gramática

- Um conjunto de regras que indicam como os símbolos são substituídos para gerar uma cadeia da linguagem: $P = \{N \to L,\ N \to L.L,\ L \to D,\ L \to LD,\ D \to 0,\ D \to 1,\ D \to 2,\ D \to 3,\ D \to 4,\ D \to 5,\ D \to 6,\ D \to 7,\ D \to 8,\ D \to 9\}$. Cada regra da forma $L \to \omega$ indica que podemos substituir qualquer ocorrência de L por ω na palavra (cadeia) que está sendo gerada.

- Uma das variáveis de V designada por símbolo inicial, símbolo do qual se iniciam as derivações, neste caso: N.

Gramática

Uma gramática G é uma tupla (V, T, P, S) onde:

- V é um conjunto finito não vazio de variáveis.
- T é um conjunto finito não vazio de símbolos terminais.
- P é um conjunto finito de regras de produção da forma: $\alpha \rightarrow \beta$ onde $\alpha, \beta \in (V \cup T)^*$, onde α tem ao menos uma variável.
- $S \in V$ é o símbolo inicial.

Gramática

Na Seção 1.1 estudamos o conceito de relação. O funcionamento da gramática se dá através da relação $\Rightarrow_G$ ou simplesmente $\Rightarrow$. O domínio e o contradomínio desta relação é o conjunto $(V \cup T)^*$. Se $G = (V, T, P, S)$, dizemos que $\alpha \Rightarrow_G \beta$ quando $\alpha = \delta_1 \alpha_1 \delta_2$ e $\beta = \delta_1 \beta_1 \delta_2$ e a regra $\alpha_1 \rightarrow \beta_1$ está em P. Ou seja, se uma regra da gramática descreve como podemos trocar α_1, um pedaço de α igual ao lado esquerdo de uma regra, pelo lado direito da mesma regra, obtendo β. Neste caso dizemos que α deriva β. Observe que α_1 pode ocorrer em mais de uma posição em α e que a aplicação da regra $\alpha_1 \rightarrow \beta_1$ pode ser aplicada em qualquer destas posições. Devemos aplicar as regras até que não haja mais não variáveis.

Gramática

Vamos considerar uma gramática análoga à gramática vista para números decimais. Entretanto, esta gramática só possui os dígitos 0 e 1, gerando os números de ponto flutuante na base binária. $G = (V, T, P, N)$, onde:

- $T = \{0, 1\}$
- $V = \{N, L, D\}$;
- $P = \{N \to L, \quad N \to L.L, L \to D, L \to LD, D \to 0, D \to 1\}$.

Para esta gramática temos os seguintes exemplos de $\Rightarrow_G$:

Gramática

- $L.L \Rightarrow_G L.LD$
- $L.L \Rightarrow_G LD.L$

Quando a gramática G está subentendida no contexto, nós a suprimimos da notação, escrevendo simplesmente:

- $L.LD \Rightarrow L.L0$

Gramática

- $L.L \Rightarrow_G L.LD$
- $L.L \Rightarrow_G LD.L$

Quando a gramática G está subentendida no contexto, nós a suprimimos da notação, escrevendo simplesmente:

- $L.LD \Rightarrow L.L0$

É muito útil escrever mais de um passo da relação $\Rightarrow_G$ com um único símbolo $\Rightarrow_G^*$, por exemplo, se $N \Rightarrow_G L.L \Rightarrow_G D.L \Rightarrow_G 1.L$, podemos escrever $N \Rightarrow_G^* 1.L$. O * indica que $\Rightarrow_G^*$ também é reflexiva, ou seja, $\alpha \Rightarrow^* \alpha$.

Gramática

Dada uma gramática G é uma tupla (V,T,P,S), dizemos que a relação $\Rightarrow_G^* \subseteq (V \cup T)^* \times (V \cup T)^*$ é a menor relação tal que:

- Se $\alpha \Rightarrow_G \beta$ então $\alpha \Rightarrow_G^* \beta$.
- Para todo $\omega \in (V \cup T)^*$, $\omega \Rightarrow_G^* \omega$.
- Para todos $\alpha, \beta, \gamma \in (V \cup T)^*$, se $\alpha \Rightarrow_G^* \beta$ e $\beta \Rightarrow_G^* \gamma$ então $\alpha \Rightarrow_G^* \gamma$.

Esta notação é lida como "deriva em zero ou mais passos". Por exemplo, $N \Rightarrow_G^* 1.L$ é pronunciado "N deriva em zero ou mais passos $1.L$".

Para finalizar o presente item, deve-se mencionar que as linguagens apresentam duas componentes básicas: sintaxe e semântica. A **sintaxe** de uma linguagem refere-se à sua apresentação visual, à forma, à estrutura de suas cadeias, e não leva em consideração qualquer informação sobre o significado associado às mesmas. O significado que se atribui a uma cadeia, ou conjunto de cadeias de uma mesma linguagem, deriva do significado que se atribui às construções da linguagem, ou seja, decorre diretamente da sua **semântica**.

Sintaxe e semântica constituem tópicos que se complementam, sendo de muito interesse e grande aplicação em computação. O presente texto trata, porém, apenas dos aspectos referentes à formalização da estrutura das linguagens, bem como do estudo de suas propriedades sintáticas. Devido à sua complexidade, muito maior que a da sintaxe, o estudo da semântica formal das linguagens está bem menos desenvolvido. No entanto, consideráveis progressos teóricos e práticos vêm sendo obtidos nas duas últimas décadas em semântica formal, configurando tema de estudos avançados na área das linguagens formais ([59], [30]). No entanto, o tema escapa ao escopo deste texto.

Gramáticas

Também conhecidas como **dispositivos generativos**, **dispositivos de síntese**, ou ainda dispositivos de geração de cadeias, as **gramáticas** constituem sistemas formais baseados em regras de substituição, através dos quais é possível sintetizar, de forma exaustiva, o conjunto das cadeias que compõem uma determinada linguagem.

Para ilustrar esse conceito, nada melhor do que a própria noção intuitiva, adquirida à época do ensino fundamental, do significado do termo "gramática": o livro através do qual são aprendidas as regras que indicam como falar e escrever corretamente um idioma.

Como se sabe, as regras assim definidas especificam combinações válidas dos símbolos que compõem o alfabeto — os diversos verbos, substantivos, adjetivos, advérbios, pronomes, artigos etc. —, e isso é feito com o auxílio de entidades abstratas denominadas classes sintáticas: sujeito, predicado etc. Assim, por exemplo, a frase "O menino atravessou a rua distraidamente" é considerada correta, do ponto de vista gramatical, pois ela obedece a uma das inúmeras regras de formação de frases, baseada no padrão sujeito + predicado + complemento.

Gramáticas

De acordo com tais regras, um sujeito pode ser composto por um *artigo* (**"O"**) seguido de um *substantivo* (**"menino"**), o *predicado* pode conter um *verbo* (**"atravessou"**) e um correspondente *objeto direto* (**"a rua"**), e o *complemento* pode modificar a ação (**"distraidamente"**). O *objeto direto*, por sua vez, pode seguir o mesmo padrão estrutural do *sujeito*: *artigo* (**"a"**) seguido de *substantivo* (**"rua"**).

Naturalmente, o conjunto das regras que formam uma "gramática" deve ser suficiente para permitir a elaboração de qualquer frase ou discurso corretamente construído em um determinado idioma, e não deve permitir a construção de qualquer cadeia que não pertença à linguagem. Convém notar, no exemplo do parágrafo acima, que os termos em *itálico* correspondem às denominadas classes sintáticas do português, e os termos em **negrito**, aos símbolos que efetivamente fazem parte do seu alfabeto.

Gramáticas

Assim como ocorre no caso das linguagens naturais, as linguagens formais também podem ser especificadas através de "gramáticas" a elas associadas. No caso das gramáticas das linguagens formais, que constituem o objeto deste estudo, a analogia com as "gramáticas" das linguagens naturais é muito grande. Tratam-se, as primeiras, de conjuntos de regras que, quando aplicadas de forma recorrente, possibilitam a geração de todas as cadeias pertencentes a uma determinada linguagem.

Diferentemente das gramáticas das linguagens naturais, que são descritas por intermédio de linguagens também naturais (muitas vezes a mesma que está sendo descrita pela gramática), as gramáticas das linguagens formais são descritas utilizando notações matemáticas rigorosas que visam, entre outros objetivos, evitar dúvidas na sua interpretação. Tais notações recebem a denominação de **metalinguagens** — linguagens que são empregadas para definir outras linguagens.

Formalmente, uma gramática G pode ser definida como sendo uma quádrupla[2]:

$$G = (V, \Sigma, P, S)$$

onde:

- V é o **vocabulário** da gramática; corresponde a um conjunto (finito e não-vazio) de símbolos;
- Σ é o conjunto (finito e não-vazio) dos símbolos **terminais** da gramática; também denominado **alfabeto**;
- P é o conjunto (finito e não-vazio) de **produções** ou **regras de substituição** da gramática;
- S é a **raiz** da gramática, $S \in V$.

De uma forma mais geral do que triplas ou quádruplas ordenadas, chamamos de tupla a $(x_1, x_2, \ldots, x_n)$ e dizemos que $(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n)$ se, e somente se, para todo o $i \in \{1, 2, \ldots,\}$, $x_i = y_i$. Com este conceito de tuplas, podemos passar a uma definição mais formal de gramática. (MENEZES, 2000)

Exemplo 2.21 Seja $G_1 = (V_1, \Sigma_1, P_1, S)$, com:

$$
\begin{aligned}
V_1 &= \{0, 1, 2, 3, S, A\} \\
\Sigma_1 &= \{0, 1, 2, 3\} \\
N_1 &= \{S, A\} \\
P_1 &= \{S \to 0S33, S \to A, A \to 12, A \to \epsilon\}
\end{aligned}
$$

É fácil verificar que G_1 está formulada de acordo com as regras gerais acima enunciadas para a especificação de gramáticas. □

Denomina-se **forma sentencial** qualquer cadeia obtida pela aplicação recorrente das seguintes regras de substituição:

1. S (a raiz da gramática) é por definição uma forma sentencial;

2. Seja $\alpha\rho\beta$ uma forma sentencial, com α e β cadeias quaisquer de terminais e/ou não-terminais da gramática, e seja $\rho \to \gamma$ uma produção da gramática. Nessas condições, a aplicação dessa produção à forma sentencial, substituindo a ocorrência de ρ por γ, produz uma nova forma sentencial $\alpha\gamma\beta$.

Denota-se a substituição acima definida, também conhecida como **derivação direta**, por:

$$\alpha\rho\beta \Rightarrow_G \alpha\gamma\beta$$

O índice "G" designa o fato de que a produção aplicada, no caso $\rho \rightarrow \gamma$, pertence ao conjunto de produções que define a gramática G. Nos casos em que a gramática em questão puder ser facilmente identificada, admite-se a eliminação de referências explícitas a ela. Note-se a distinção gráfica e de significado que se faz entre o símbolo empregado na denotação das produções da gramática ($\rightarrow$) e o símbolo utilizado na denotação das derivações ($\Rightarrow$).