

# Apostila de Apoio — Aula 1

## Linguagens Formais

Teoria da Computação

Fev/2026

### Como usar

Esta apostila foi escrita para quem está começando do zero. Você não precisa saber programar. Aqui, a palavra **string** significa apenas “um texto” (uma sequência de símbolos). Nos exemplos, vamos usar alfabetos pequenos como {0,1} e {a,b} para facilitar.

### Sumário

- Como estudar este conteúdo
- Linguagens Formais: conceitos-base
- Exemplos resolvidos
- Exercícios graduais
- Erros comuns
- Glossário rápido

## 1. Como estudar este conteúdo

A disciplina começa com ideias simples (conjuntos de textos) e vai ganhando poder ao longo do semestre. Para aprender com segurança:

- Leia a definição e faça 2 ou 3 exemplos na hora, sem pular etapas.
- Quando aparecer um símbolo novo ( $\Sigma$ ,  $\epsilon$ ,  $\Sigma^*$ ), pare e explique com suas palavras.
- Nos exercícios, escreva também 2 exemplos que não pertencem à linguagem (isso fixa muito).
- Se travar, volte para: alfabeto  $\rightarrow$  cadeia  $\rightarrow$  linguagem  $\rightarrow$  operações (a base de tudo).

### Regra de ouro

Regra de ouro: sempre pergunte “**quais strings são aceitas?**”. Se você consegue listar exemplos aceitos e rejeitados, você está entendendo.

## 2. Linguagens Formais: conceitos-base

Nesta aula, a gente transforma uma ideia do dia a dia (“textos válidos”) em uma linguagem matemática precisa. Isso serve para entender validação, análise de tokens e, mais adiante, limites do que é computável.

### 2.1 String: o que é isso?

Uma string é um texto: uma sequência de símbolos. Exemplos de strings: "A", "AB", "0101", "casa". Neste curso, normalmente olhamos para a forma da string (quais símbolos e qual ordem), não para o significado da palavra.

#### Dica

No curso, vamos usar alfabetos pequenos (como {0,1}) para enxergar padrões com clareza. Isso não limita o conteúdo: as ideias valem para alfabetos maiores também.

### 2.2 Alfabeto ( $\Sigma$ ): o “teclado permitido”

Um alfabeto, escrito como  $\Sigma$  (sigma), é um conjunto finito de símbolos permitidos. Pense como o conjunto de teclas que você aceita usar.

Exemplos:  $\Sigma = \{0,1\}$  (binário),  $\Sigma = \{a,b\}$ ,  $\Sigma = \{x,y,z,0,1,2\}$ .

Se  $\Sigma = \{0,1\}$ , então a string "0101" é válida como cadeia sobre  $\Sigma$ , mas "012" não é (o símbolo 2 não está no alfabeto).

### 2.3 Cadeia ( $w$ ) e cadeia vazia ( $\epsilon$ )

Uma cadeia (ou palavra), geralmente chamada de  $w$ , é uma sequência finita de símbolos de  $\Sigma$ . A cadeia vazia, escrita  $\epsilon$  (épsilon), é a cadeia com zero símbolos.

Importante:  $\epsilon$  não é um símbolo do alfabeto.  $\epsilon$  é uma string inteira, só que vazia.

### 2.4 Comprimento ( $|w|$ )

O comprimento de uma cadeia  $w$ , escrito  $|w|$ , é a quantidade de símbolos na string.

Exemplos: se  $w = "abba"$ , então  $|w| = 4$ . E  $|\epsilon| = 0$ .

### 2.5 $\Sigma^*$ (sigma estrela): todas as strings possíveis

$\Sigma^*$  é o conjunto de todas as cadeias finitas que podem ser formadas com  $\Sigma$ , incluindo  $\epsilon$ . Se  $\Sigma = \{0,1\}$ , então  $\Sigma^*$  contém  $\epsilon, 0, 1, 00, 01, 10, 11, 000, \dots$

## Atenção

$\Sigma$  é um conjunto de símbolos.  $\Sigma^*$  é um conjunto de strings. Todo símbolo vira uma string de tamanho 1 dentro de  $\Sigma^*$ .

## 2.6 Linguagem (L): o conjunto das strings aceitas

Uma linguagem é um subconjunto de  $\Sigma^*$ :  $L \subseteq \Sigma^*$ . Ou seja: você pega todas as strings possíveis e escolhe quais são válidas para aquele contexto.

Exemplo ( $\Sigma = \{0,1\}$ ):  $L = \{ w \subseteq \Sigma^* \mid w \text{ termina com } 1 \}$ . Então "1", "01" e "101" pertencem a L, mas "0", "10" e  $\epsilon$  não pertencem.

## 2.7 Operações com linguagens (pensando como “listas de textos”)

Como linguagens são conjuntos, podemos combinar linguagens com operações.

Nesta parte, pense que cada linguagem é uma “lista” de strings (sem repetição).

- a) União —  $L_1 \cup L_2$ : strings que estão em  $L_1$  ou em  $L_2$ .
- b) Concatenação —  $L_1 L_2$ : grudar uma string de  $L_1$  com uma de  $L_2$ .
- c) Estrela de Kleene —  $L^*$ : repetir strings de L zero ou mais vezes (inclui  $\epsilon$ ).

### 3. Exemplos resolvidos (passo a passo)

#### Exemplo 1: listar elementos de $\Sigma^*$ por tamanho

Se  $\Sigma = \{a, b\}$ , então:

Tamanho	Strings
0	$\epsilon$
1	a, b
2	aa, ab, ba, bb

$\Sigma^*$  continua com strings de tamanho 3, 4, 5... por isso  $\Sigma^*$  é infinito.

#### Exemplo 2: concatenação de linguagens

Se  $L_1 = \{"a", "ab"\}$  e  $L_2 = \{"b", "bb"\}$ , então  $L_1 L_2$  é formado assim:

Pegue cada elemento de  $L_1$  e cole com cada elemento de  $L_2$ :

- "a" + "b" = "ab"
- "a" + "bb" = "abb"
- "ab" + "b" = "abb" (repetiu)
- "ab" + "bb" = "abbb"

Como linguagem é conjunto (não repete), o resultado é: {"ab", "abb", "abbb"}.

#### Exemplo 3: estrela de Kleene ( $L^*$ )

Se  $L = \{"ab"\}$ , então  $L^*$  é o conjunto de repetir "ab" várias vezes:

0 vezes:  $\epsilon$ ; 1 vez: "ab"; 2 vezes: "abab"; 3 vezes: "ababab"; ...

#### Atenção

A estrela sempre inclui  $\epsilon$ , porque repetir zero vezes é permitido.

## 4. Exercícios graduais

Faça do Nível 1 ao Nível 3. Se você estiver começando, não pule o Nível 1.

### Nível 1 (bem básico)

- $\Sigma=\{0,1\}$ . Dê 8 exemplos de strings de  $\Sigma^*$ .
- $\Sigma=\{a,b\}$ .  $w="abba"$ . Calcule  $|w|$  e escreva  $w^R$  (string ao contrário).
- Marque V/F: (a)  $\epsilon \in \Sigma$  (b)  $\epsilon \in \Sigma^*$  (c)  $\Sigma \subseteq \Sigma^*$ .

### Nível 2 (operações)

- 4)  $L1=\{"a","ab","b"\}$ ,  $L2=\{"b","bb"\}$ . Calcule  $L1 \cup L2$ .
- 5) Com as mesmas  $L1$  e  $L2$ , calcule  $L1L2$  (mostre as combinações).
- 6) Se  $L=\{"ab"\}$ , escreva 6 elementos de  $L^*$ .

### Nível 3 (definir linguagens)

- 7) Em português, defina uma linguagem sobre  $\Sigma=\{0,1\}$  que represente “strings que terminam com 1”.
- 8) Em português, defina uma linguagem sobre  $\Sigma=\{a,b\}$  que represente “strings com número par de a”.
- 9) Dê 5 exemplos e 3 contraexemplos para a linguagem do exercício 8.

## 5. Erros comuns

Leia este bloco sempre que errar um exercício. Ele resolve 80% das dúvidas iniciais.

- $\epsilon$  não é símbolo do alfabeto:  $\epsilon$  é uma string vazia. Ela pertence a  $\Sigma^*$ , mas normalmente não pertence a  $\Sigma$ .
- Não confundir  $\emptyset$  e  $\epsilon$ :  $\emptyset$  aceita nada;  $\epsilon$  aceita exatamente a string vazia.
- $\Sigma$  não é  $\Sigma^*$ :  $\Sigma$  é conjunto de símbolos;  $\Sigma^*$  é conjunto de strings (inclui  $\epsilon$ ).
- Concatenação não é comutativa: "ab"  $\neq$  "ba". Em geral  $L_1L_2 \neq L_2L_1$ .
- $01^*$  não é  $(01)^*$ :  $01^*$  = 0 seguido de vários 1;  $(01)^*$  = repetições do bloco 01.
- inclui zero repetições: por isso  $r^*$  sempre aceita  $\epsilon$ .

## 6. Glossário rápido

- $\Sigma$  (alfabeto): conjunto finito de símbolos permitidos.
- símbolo: um elemento de  $\Sigma$  (por exemplo 0 ou 1).
- string/cadeia ( $w$ ): sequência finita de símbolos de  $\Sigma$ .
- $\epsilon$  (épsilon): string vazia (comprimento 0).
- $|w|$ : comprimento da string  $w$ .
- $\Sigma^*$ : conjunto de todas as strings finitas sobre  $\Sigma$  (inclui  $\epsilon$ ).
- $L$ : linguagem (um conjunto de strings),  $L \subseteq \Sigma^*$ .
- ER: expressão regular (notação para descrever linguagens regulares).
- $L(r)$ : linguagem descrita (denotada) pela ER  $r$ .
- $|$ : operador “ou” (união).
- concatenação: colocar uma string após a outra (sequência).
- $*$ : estrela de Kleene (repetição zero ou mais vezes).