

# **Lista de Exercícios**

## Expressões Regulares (ER)

### Resoluções Comentadas

Linguagens Formais e Autômatos

9 de dezembro de 2025

## **Sumário**

<b>1</b>	<b>Exercício 1: Linguagens sobre <math>\{0,1\}</math></b>	<b>2</b>
1.1	(a) Palavras com, no mínimo, um 0 . . . . .	2
1.2	(b) Palavras de tamanho ímpar . . . . .	3
1.3	(c) Palavras com tamanho múltiplo de 5 . . . . .	4
1.4	(d) Prefixo de 0's seguido de sufixo de 1's . . . . .	6
1.5	(e) Palíndromos . . . . .	7
1.6	(f) Primeira metade idêntica à segunda . . . . .	9
<b>2</b>	<b>Exercício 2: ERs sobre <math>\Sigma = \{a, b, c\}</math></b>	<b>11</b>
2.1	(a) Sufixo abe ou cba . . . . .	11
2.2	(b) Pelo menos 3 ocorrências de abc . . . . .	12
2.3	(c) Último símbolo igual ao primeiro . . . . .	13
2.4	(d) 2 a's ou 2 b's consecutivos . . . . .	14
2.5	(e) aa ou bb é subpalavra e cccc é sufixo . . . . .	15
2.6	(f) Contém ab e ba em qualquer ordem . . . . .	16
<b>3</b>	<b>Observações Finais - Parte 1</b>	<b>19</b>
3.1	Resumo dos Exercícios 1-2f . . . . .	19
3.2	Conceitos-Chave Abordados . . . . .	19
3.3	Próximos Exercícios . . . . .	20

# 1 Exercício 1: Linguagens sobre $\{0,1\}$

Descreva mais formalmente as seguintes linguagens sobre o alfabeto  $\Sigma = \{0, 1\}$ :

## 1.1 (a) Palavras com, no mínimo, um 0

**Descrição:** O conjunto das palavras com, no mínimo, um 0.

### Expressão Regular

#### Expressão Regular:

$$(0|1)^* \cdot 0 \cdot (0|1)^*$$

Ou de forma equivalente:

$$\Sigma^* \cdot 0 \cdot \Sigma^*$$

Onde  $\Sigma = \{0, 1\}$ .

### Explicação

#### Explicação:

A ER aceita qualquer sequência que contenha pelo menos um 0 em qualquer posição:

- $(0|1)^*$ : qualquer sequência de 0's e 1's antes do 0 obrigatório
- 0: pelo menos um 0 (obrigatório)
- $(0|1)^*$ : qualquer sequência de 0's e 1's depois

**Linguagem:**  $L = \{w \in \{0, 1\}^* \mid w \text{ contém pelo menos um } 0\}$

## Exemplos

### Palavras que pertencem:

- 0 (apenas um 0)
- 10 (1 seguido de 0)
- 01 (0 seguido de 1)
- 100 (1, depois 0, depois 0)
- 010 (0 no meio)
- 11011 (contém 0's)

### Palavras que NÃO pertencem:

- $\varepsilon$  (vazia - não tem 0)
- 1 (apenas 1's)
- 11 (apenas 1's)
- 1111 (apenas 1's)

## 1.2 (b) Palavras de tamanho ímpar

**Descrição:** O conjunto das palavras de tamanho ímpar.

### Expressão Regular

#### Expressão Regular:

$$(0|1) \cdot ((0|1)(0|1))^*$$

Ou de forma mais compacta:

$$(0|1) \cdot (\Sigma\Sigma)^*$$

## Explicação

### Explicação:

A ER garante tamanho ímpar:

- $(0|1)$ : primeiro símbolo (contribui com 1 ao tamanho)
- $((0|1)(0|1))^*$ : pares de símbolos repetidos 0 ou mais vezes
  - Cada iteração adiciona 2 ao tamanho
  - 0 iterações: tamanho = 1 (ímpar)
  - 1 iteração: tamanho =  $1 + 2 = 3$  (ímpar)
  - 2 iterações: tamanho =  $1 + 4 = 5$  (ímpar)
  - $k$  iterações: tamanho =  $1 + 2k$  (sempre ímpar)

Linguagem:  $L = \{w \in \{0, 1\}^* \mid |w| \equiv 1 \pmod{2}\}$

## Exemplos

### Palavras que pertencem (tamanho ímpar):

- 0 (tamanho 1)
- 1 (tamanho 1)
- 010 (tamanho 3)
- 111 (tamanho 3)
- 10101 (tamanho 5)
- 0000000 (tamanho 7)

### Palavras que NÃO pertencem (tamanho par ou vazia):

- $\varepsilon$  (tamanho 0 - par)
- 01 (tamanho 2 - par)
- 1010 (tamanho 4 - par)
- 111111 (tamanho 6 - par)

## 1.3 (c) Palavras com tamanho múltiplo de 5

Descrição: O conjunto das palavras com tamanho múltiplo de 5.

## Expressão Regular

### Expressão Regular:

$$((0|1)(0|1)(0|1)(0|1)(0|1))^*$$

Ou de forma mais compacta:

$$(\Sigma^5)^* \text{ onde } \Sigma = \{0, 1\}$$

Expandindo:

$$(00000|00001|00010|\dots|11111)^*$$

## Explicação

### Explicação:

A ER aceita apenas palavras cujo tamanho é múltiplo de 5:

- $(0|1)(0|1)(0|1)(0|1)(0|1)$ : exatamente 5 símbolos quaisquer
- $(\dots)^*$ : repete o bloco de 5 símbolos 0 ou mais vezes
  - 0 repetições:  $\varepsilon$  (tamanho 0 =  $5 \times 0$ )
  - 1 repetição: tamanho 5 =  $5 \times 1$
  - 2 repetições: tamanho 10 =  $5 \times 2$
  - $k$  repetições: tamanho  $5k$

**Linguagem:**  $L = \{w \in \{0, 1\}^* \mid |w| \equiv 0 \pmod{5}\}$

## Exemplos

**Palavras que pertencem (tamanho múltiplo de 5):**

- $\varepsilon$  (tamanho  $0 = 5 \times 0$ )
- 00000 (tamanho 5)
- 10101 (tamanho 5)
- 0000000000 (tamanho 10 =  $5 \times 2$ )
- 11110000011111 (tamanho 15 =  $5 \times 3$ )

**Palavras que NÃO pertencem:**

- 0 (tamanho 1)
- 01 (tamanho 2)
- 010 (tamanho 3)
- 0101 (tamanho 4)
- 010101 (tamanho 6)

## 1.4 (d) Prefixo de 0's seguido de sufixo de 1's

**Descrição:** O conjunto das palavras com um prefixo de um ou mais 0's seguido de um sufixo de zero ou mais 1's.

### Expressão Regular

**Expressão Regular:**

$$0^+ \cdot 1^*$$

Ou expandindo o +:

$$0 \cdot 0^* \cdot 1^*$$

## Explicação

### Explicação:

A ER define o padrão específico:

- $0^+$ : um ou mais 0's (prefixo obrigatório)
  - Pelo menos um 0 deve aparecer
  - Pode ter vários 0's consecutivos
- $1^*$ : zero ou mais 1's (sufixo opcional)
  - Pode não ter nenhum 1
  - Pode ter vários 1's consecutivos

**Padrão aceito:**  $\underbrace{00 \cdots 0}_{\geq 1} \underbrace{11 \cdots 1}_{\geq 0}$

**Linguagem:**  $L = \{0^m 1^n \mid m \geq 1, n \geq 0\}$

## Exemplos

### Palavras que pertencem:

- 0 (um 0, zero 1's)
- 01 (um 0, um 1)
- 00 (dois 0's, zero 1's)
- 001 (dois 0's, um 1)
- 0011 (dois 0's, dois 1's)
- 000111 (três 0's, três 1's)

### Palavras que NÃO pertencem:

- $\varepsilon$  (não tem nenhum 0)
- 1 (começa com 1, não com 0)
- 10 (começa com 1)
- 010 (0 e 1 intercalados)
- 0101 (não segue padrão  $0^+ 1^*$ )

## 1.5 (e) Palíndromos

**Descrição:** O conjunto de palavras em que a palavra lida é igual quando lida da esquerda para direita ou vice-versa (palíndromos).

## Expressão Regular

**Resposta:**

**NÃO É POSSÍVEL** descrever linguagem de palíndromos com Expressão Regular!  
Palíndromos sobre  $\{0, 1\}$  NÃO formam linguagem regular.

## Explicação

**Explicação:**

A linguagem de palíndromos  $L_{pal} = \{w \mid w = w^R\}$  **não é regular**.

**Prova (Lema do Bombeamento):**

Suponha que  $L_{pal}$  é regular com constante  $p$ .

Considere  $s = 0^p 1 0^p \in L_{pal}$  (é palíndromo).

Pelo lema:  $s = xyz$  onde  $|xy| \leq p$ ,  $|y| > 0$ .

Logo  $xy$  está contido nos primeiros  $p$  símbolos (todos 0's), então  $y = 0^k$  com  $k > 0$ .

Para  $i = 2$ :  $xy^2z = 0^p 0^k 1 0^p = 0^{p+k} 1 0^p$

Mas  $0^{p+k} 1 0^p$  NÃO é palíndromo (mais 0's à esquerda que à direita).

Contradição! Logo palíndromos não são regulares.

**Classificação:**

- Palíndromos são LLC (Livres de Contexto)
- Podem ser descritos por GLC
- Não podem ser descritos por ER

**GLC para palíndromos:**

$$S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \mid \epsilon$$

## Exemplos

**Exemplos de palíndromos (mas não há ER):**

- $\varepsilon$  (vazia)
- 0, 1 (tamanho 1)
- 00, 11 (tamanho 2)
- 010, 101 (tamanho 3)
- 0110, 1001 (tamanho 4)
- 01010, 10101 (tamanho 5)

**Não-palíndromos:**

- 01 (reverso é 10)
- 10 (reverso é 01)
- 001 (reverso é 100)
- 0101 (reverso é 1010)

## 1.6 (f) Primeira metade idêntica à segunda

**Descrição:** O conjunto das palavras de tamanho par cuja primeira metade é idêntica à segunda.

### Expressão Regular

**Resposta:**

**NÃO É POSSÍVEL** descrever esta linguagem com Expressão Regular!  
A linguagem  $L = \{ww \mid w \in \{0,1\}^*\}$  NÃO é regular.

## Explicação

### Explicação:

A linguagem  $L = \{ww \mid w \in \{0,1\}^*\}$  não é regular.

### Prova (Lema do Bombeamento):

Suponha que  $L$  é regular com constante  $p$ .

Considere  $s = 0^p 1^p 0^p 1^p \in L$  (é  $ww$  onde  $w = 0^p 1^p$ ).

Temos  $|s| = 4p \geq p$ . Pelo lema:  $s = xyz$  onde  $|xy| \leq p$ ,  $|y| > 0$ .

Logo  $xy$  está nos primeiros  $p$  símbolos (todos 0's da primeira metade), então  $y = 0^k$  com  $k > 0$ .

Para  $i = 2$ :  $xy^2z = 0^{p+k} 1^p 0^p 1^p$

Mas isso não é da forma  $ww$  para nenhum  $w$  (primeira metade tem  $p + k$  zeros, segunda tem  $p$  zeros).

Contradição! Logo  $L$  não é regular.

### Classificação:

- Esta linguagem é LLC (Livre de Contexto)
- Pode ser descrita por GLC
- Não pode ser descrita por ER

### GLC para $ww$ :

$$S \rightarrow 0S0 \mid 1S1 \mid \varepsilon$$

## Exemplos

### Exemplos de $ww$ (mas não há ER):

- $\varepsilon$  (vazia =  $\varepsilon\varepsilon$ )
- 00 ( $w = 0$ , então  $ww = 00$ )
- 11 ( $w = 1$ , então  $ww = 11$ )
- 0101 ( $w = 01$ , então  $ww = 0101$ )
- 1010 ( $w = 10$ , então  $ww = 1010$ )
- 001001 ( $w = 001$ , então  $ww = 001001$ )

### Não é $ww$ :

- 0 (tamanho ímpar)
- 01 (não é repetição)
- 010 (tamanho ímpar)
- 0011 (não é  $ww$ )
- 0110 (não é  $ww$ )

## 2 Exercício 2: ERs sobre $\Sigma = \{a, b, c\}$

Construa ERs para as seguintes linguagens sobre  $\Sigma = \{a, b, c\}$ :

### 2.1 (a) Sufixo abe ou cba

Linguagem:  $L_a = \{w \mid w \text{ contém o sufixo } abe \text{ ou } cba\}$

#### Expressão Regular

##### Expressão Regular:

$$(a|b|c)^* \cdot (abe|cba)$$

Ou expandindo:

$$\Sigma^* \cdot (abe|cba)$$

#### Explicação

##### Explicação:

A ER garante que a palavra termina com *abe* ou *cba*:

- $(a|b|c)^*$ : qualquer sequência de símbolos antes do sufixo
- $(abe|cba)$ : palavra deve terminar com uma dessas duas sequências

#### Exemplos

##### Palavras que pertencem:

- *abe* (apenas sufixo)
- *cba* (apenas sufixo)
- *aabe* (*a* + *abe*)
- *bcba* (*b* + *cba*)
- *abcabe* (*abc* + *abe*)
- *aacba* (*aa* + *cba*)

##### Palavras que NÃO pertencem:

- $\varepsilon$  (vazia)
- *ab* (incompleto)
- *abc* (termina com *c*, não com sufixo correto)
- *abea* (sufixo não está no final)
- *cbab* (termina com *b*, não com sufixo correto)

## 2.2 (b) Pelo menos 3 ocorrências de abc

Linguagem:  $L_b = \{w \mid w \text{ contém pelo menos 3 ocorrências de } abc\}$

### Expressão Regular

Expressão Regular:

$$\Sigma^* \cdot abc \cdot \Sigma^* \cdot abc \cdot \Sigma^* \cdot abc \cdot \Sigma^*$$

Expandindo  $\Sigma$ :

$$(a|b|c)^* \cdot abc \cdot (a|b|c)^* \cdot abc \cdot (a|b|c)^* \cdot abc \cdot (a|b|c)^*$$

### Explicação

Explicação:

A ER força a presença de pelo menos 3 ocorrências de  $abc$ :

- $\Sigma^*$ : símbolos arbitrários antes do primeiro  $abc$
- $abc$ : primeira ocorrência obrigatória
- $\Sigma^*$ : símbolos arbitrários entre primeira e segunda ocorrências
- $abc$ : segunda ocorrência obrigatória
- $\Sigma^*$ : símbolos arbitrários entre segunda e terceira ocorrências
- $abc$ : terceira ocorrência obrigatória
- $\Sigma^*$ : símbolos arbitrários após a terceira ocorrência

Pode ter mais de 3 ocorrências - a ER garante **pelo menos 3**.

## Exemplos

### Palavras que pertencem:

- $abcabcbcabc$  (exatamente 3 ocorrências)
- $aabcbabccabc$  (3 ocorrências com símbolos entre elas)
- $abcaabcbabc$  (3 ocorrências)
- $abcabcbcabc$  (4 ocorrências)
- $xabcyabczabc$  (onde x,y,z são sequências quaisquer)

### Palavras que NÃO pertencem:

- $\varepsilon$  (nenhuma ocorrência)
- $abc$  (apenas 1 ocorrência)
- $abcbc$  (apenas 2 ocorrências)
- $aabbcc$  (não contém  $abc$ )
- $abab$  (não contém  $abc$ )

## 2.3 (c) Último símbolo igual ao primeiro

Linguagem:  $L_c = \{w \mid \text{o último símbolo de } w \text{ seja igual ao primeiro}\}$

### Expressão Regular

#### Expressão Regular:

$$a \cdot \Sigma^* \cdot a \mid b \cdot \Sigma^* \cdot b \mid c \cdot \Sigma^* \cdot c \mid a \mid b \mid c$$

Ou de forma mais compacta:

$$a(\Sigma^* a | \varepsilon) \mid b(\Sigma^* b | \varepsilon) \mid c(\Sigma^* c | \varepsilon)$$

### Explicação

#### Explicação:

A ER cobre todos os casos onde primeiro = último:

- $a \cdot \Sigma^* \cdot a$ : começa com  $a$ , termina com  $a$  (tamanho  $\geq 2$ )
- $b \cdot \Sigma^* \cdot b$ : começa com  $b$ , termina com  $b$  (tamanho  $\geq 2$ )
- $c \cdot \Sigma^* \cdot c$ : começa com  $c$ , termina com  $c$  (tamanho  $\geq 2$ )
- $a \mid b \mid c$ : palavras de tamanho 1 (primeiro = último trivialmente)

O  $\Sigma^*$  no meio permite qualquer sequência entre primeiro e último símbolo.

## Exemplos

### Palavras que pertencem:

- $a$  (tamanho 1)
- $b, c$  (tamanho 1)
- $aa$  (começa e termina com  $a$ )
- $aba$  (começa e termina com  $a$ )
- $bab$  (começa e termina com  $b$ )
- $cabac$  (começa e termina com  $c$ )
- $abcba$  (começa e termina com  $a$ )

### Palavras que NÃO pertencem:

- $\varepsilon$  (sem símbolos)
- $ab$  (começa com  $a$ , termina com  $b$ )
- $ba$  (começa com  $b$ , termina com  $a$ )
- $abc$  (começa com  $a$ , termina com  $c$ )
- $cab$  (começa com  $c$ , termina com  $b$ )

## 2.4 (d) 2 a's ou 2 b's consecutivos

Linguagem:  $L_d = \{w \mid w \text{ tem } 2 \text{ a's consecutivos ou } 2 \text{ b's consecutivos}\}$

### Expressão Regular

#### Expressão Regular:

$$\Sigma^* \cdot (aa|bb) \cdot \Sigma^*$$

Expandindo:

$$(a|b|c)^* \cdot (aa|bb) \cdot (a|b|c)^*$$

## Explicação

### Explicação:

A ER garante a presença de pelo menos uma das duas condições:

- $\Sigma^*$ : qualquer sequência antes dos consecutivos
- $(aa|bb)$ : deve conter  $aa$  OU  $bb$  em algum lugar
- $\Sigma^*$ : qualquer sequência depois dos consecutivos

Basta ter  $aa$  ou  $bb$  em qualquer posição da palavra.

## Exemplos

### Palavras que pertencem:

- $aa$  (dois  $a$ 's consecutivos)
- $bb$  (dois  $b$ 's consecutivos)
- $aab$  (começa com  $aa$ )
- $bba$  (começa com  $bb$ )
- $caa$  (termina com  $aa$ )
- $cbb$  (termina com  $bb$ )
- $caac$  ( $aa$  no meio)
- $abbc$  ( $bb$  no meio)
- $aabb$  (tem ambos  $aa$  e  $bb$ )

### Palavras que NÃO pertencem:

- $\varepsilon$  (vazia)
- $a, b, c$  (apenas um símbolo)
- $ab$  (não tem consecutivos)
- $abc$  (nenhum símbolo se repete)
- $abab$  ( $a$ 's e  $b$ 's alternados)
- $ababc$  (sem consecutivos)
- $cabac$  (sem  $aa$  nem  $bb$ )

## 2.5 (e) aa ou bb é subpalavra e cccc é sufixo

Linguagem:  $L_e = \{w \mid aa \text{ ou } bb \text{ é subpalavra e } cccc \text{ é sufixo}\}$

### Expressão Regular

#### Expressão Regular:

$$\Sigma^* \cdot (aa|bb) \cdot \Sigma^* \cdot cccc$$

Expandindo:

$$(a|b|c)^* \cdot (aa|bb) \cdot (a|b|c)^* \cdot cccc$$

## Explicação

### Explicação:

A ER combina duas condições obrigatórias:

1. **Contém aa ou bb como subpalavra:**

- $\Sigma^* \cdot (aa|bb) \cdot \Sigma^*$
- aa ou bb pode aparecer em qualquer posição

2. **Termina com cccc:**

- cccc no final é obrigatório
- São exatamente 4 c's consecutivos ao final

Ambas as condições devem ser satisfeitas simultaneamente.

## Exemplos

### Palavras que pertencem:

- aacccc (tem aa e termina com cccc)
- bbcccc (tem bb e termina com cccc)
- caacccc (tem aa e termina com cccc)
- cbbcccc (tem bb e termina com cccc)
- aabbcccc (tem aa e bb, termina com cccc)
- abacccc (tem aa, termina com cccc)

### Palavras que NÃO pertencem:

- aaccc (tem aa mas só 3 c's no final)
- bbccccc (tem bb mas 5 c's no final)
- abcccc (termina com cccc mas sem aa nem bb)
- aabbc (tem aa e bb mas não termina com cccc)
- cccc (termina com cccc mas sem aa nem bb)

## 2.6 (f) Contém ab e ba em qualquer ordem

**Linguagem:**  $L_f = \{w \mid w \text{ contém as substrings } ab \text{ e } ba \text{ (em qualquer ordem)}\}$

## Expressão Regular

**Expressão Regular:**

$$\Sigma^* \cdot ab \cdot \Sigma^* \cdot ba \cdot \Sigma^* \mid \Sigma^* \cdot ba \cdot \Sigma^* \cdot ab \cdot \Sigma^*$$

Ou de forma mais compacta:

$$\Sigma^*(ab\Sigma^*ba|ba\Sigma^*ab)\Sigma^*$$

## Explicação

**Explicação:**

A ER captura as duas ordens possíveis:

1. **Primeiro  $ab$ , depois  $ba$ :**

- $\Sigma^* \cdot ab \cdot \Sigma^* \cdot ba \cdot \Sigma^*$
- $ab$  aparece antes de  $ba$  (com símbolos entre eles ou não)

2. **Primeiro  $ba$ , depois  $ab$ :**

- $\Sigma^* \cdot ba \cdot \Sigma^* \cdot ab \cdot \Sigma^*$
- $ba$  aparece antes de  $ab$  (com símbolos entre eles ou não)

A união ( $|$ ) das duas alternativas garante que ambas as substrings apareçam, independentemente da ordem.

## Exemplos

### Palavras que pertencem:

- $abba$  ( $ab$  seguido de  $ba$ )
- $baab$  ( $ba$  seguido de  $ab$ )
- $abcba$  ( $ab$ , depois  $c$ , depois  $ba$ )
- $bacab$  ( $ba$ , depois  $c$ , depois  $ab$ )
- $aba$  (contém  $ab$  e  $ba$  sobrepostos)
- $bab$  (contém  $ba$  e  $ab$  sobrepostos)
- $aabba$  (contém ambos)
- $cabbac$  (contém ambos)

### Palavras que NÃO pertencem:

- $\varepsilon$  (vazia)
- $ab$  (só tem  $ab$ , falta  $ba$ )
- $ba$  (só tem  $ba$ , falta  $ab$ )
- $aaa$  (não tem  $ab$  nem  $ba$ )
- $abc$  (tem  $ab$  mas não tem  $ba$ )
- $bac$  (tem  $ba$  mas não tem  $ab$ )
- $aabbcc$  (tem  $ab$  mas não tem  $ba$ )

## 2.7 (g) Palavras xyz com $|x| = 3$ e $|z| = 3$

Linguagem:  $L_g = \{xyz \mid x, y, z \in \Sigma^* \text{ e } |x| = 3 \text{ e } |z| = 3\}$

## Expressão Regular

### Expressão Regular:

$$\Sigma^3 \cdot \Sigma^* \cdot \Sigma^3$$

Expandindo:

$$(a|b|c)(a|b|c)(a|b|c) \cdot (a|b|c)^* \cdot (a|b|c)(a|b|c)(a|b|c)$$

## Explicação

### Explicação:

A ER garante tamanho mínimo de 6 com restrições específicas:

- $\Sigma^3$ : exatamente 3 símbolos no início (parte  $x$ )
- $\Sigma^*$ : zero ou mais símbolos no meio (parte  $y$ )
- $\Sigma^3$ : exatamente 3 símbolos no final (parte  $z$ )

**Tamanho mínimo:**  $|w| \geq 6$  (quando  $y = \varepsilon$ )

**Linguagem:** Todas as palavras com tamanho  $\geq 6$  sobre  $\Sigma = \{a, b, c\}$

## Exemplos

**Palavras que pertencem ( $|w| \geq 6$ ):**

- $abcabc$  (tamanho 6:  $x = abc$ ,  $y = \varepsilon$ ,  $z = abc$ )
- $aaabbb$  (tamanho 6:  $x = aaa$ ,  $y = \varepsilon$ ,  $z = bbb$ )
- $abcdefg$  (tamanho 7:  $x = abc$ ,  $y = d$ ,  $z = efg$ )
- $abcXYZdef$  (tamanho 9:  $x = abc$ ,  $y = XYZ$ ,  $z = def$ )
- $aaaaaaaaaa$  (tamanho 10:  $x = aaa$ ,  $y = aaaa$ ,  $z = aaa$ )

**Palavras que NÃO pertencem ( $|w| < 6$ ):**

- $\varepsilon$  (tamanho 0)
- $abc$  (tamanho 3)
- $abcd$  (tamanho 4)
- $abcde$  (tamanho 5)

## 2.8 (h) Exatamente um a

**Linguagem:**  $L_h = \{w \mid w \text{ contém exatamente um } a\}$

### Expressão Regular

**Expressão Regular:**

$$(b|c)^* \cdot a \cdot (b|c)^*$$

## Explicação

### Explicação:

A ER garante exatamente uma ocorrência de  $a$ :

- $(b|c)^*$ : zero ou mais símbolos  $b$  ou  $c$  antes do  $a$
- $a$ : exatamente um  $a$  (único e obrigatório)
- $(b|c)^*$ : zero ou mais símbolos  $b$  ou  $c$  depois do  $a$

**Restrição importante:** Não pode haver nenhum outro  $a$  na palavra.

## Exemplos

### Palavras que pertencem:

- $a$  (apenas o  $a$ )
- $ba$  ( $b$  seguido de  $a$ )
- $ab$  ( $a$  seguido de  $b$ )
- $bac$  ( $b, a, c$ )
- $bbabb$  (um  $a$  cercado por  $b$ 's)
- $ccaccc$  (um  $a$  cercado por  $c$ 's)
- $bcabc$  (um  $a$  no meio)

### Palavras que NÃO pertencem:

- $\varepsilon$  (nenhum  $a$ )
- $bc$  (nenhum  $a$ )
- $aa$  (dois  $a$ 's)
- $aba$  (dois  $a$ 's)
- $aaa$  (três  $a$ 's)
- $abac$  (dois  $a$ 's)

## 2.9 (i) Não contém dois $a$ 's adjacentes

**Linguagem:**  $L_i = \{w \mid w \text{ não contém dois } a\text{'s adjacentes}\}$

## Expressão Regular

### Expressão Regular:

$$(b|c)^* \cdot (a(b|c)^+)^* \cdot a^? \cdot (b|c)^*$$

Ou de forma mais simples:

$$((b|c)^*a)^*((b|c)^*)$$

## Explicação

### Explicação:

A ER garante que entre quaisquer dois  $a$ 's há pelo menos um  $b$  ou  $c$ :

- $((b|c)^*a)^*$ : repete o padrão "símbolos não- $a$  seguidos de um  $a$ "
  - Cada  $a$  é precedido por zero ou mais  $b$ 's ou  $c$ 's
  - Garante que  $a$ 's não ficam adjacentes
- $(b|c)^*$ : símbolos finais (sem  $a$  ou terminando antes de  $a$  adjacente)

**Padrão proibido:**  $aa$  não pode aparecer em nenhum lugar.

## Exemplos

### Palavras que pertencem:

- $\varepsilon$  (vazia - sem  $aa$ )
- $a$  (só um  $a$ )
- $abc$  (um  $a$  sem adjacente)
- $aba$  ( $a$ 's separados por  $b$ )
- $aca$  ( $a$ 's separados por  $c$ )
- $bacabaca$  ( $a$ 's sempre separados)
- $bbbcc$  (sem nenhum  $a$ )
- $abcababc$  ( $a$ 's separados)

### Palavras que NÃO pertencem:

- $aa$  (dois  $a$ 's adjacentes)
- $aab$  ( $aa$  no início)
- $baa$  ( $aa$  no final)
- $baab$  ( $aa$  no meio)
- $aaa$  (múltiplos  $a$ 's adjacentes)
- $abaa$  ( $aa$  no final)

## 2.10 (j) Número par de substrings ba

**Linguagem:**  $L_j = \{w \mid w \text{ contém um número par de substrings } ba\}$

### Expressão Regular

#### Expressão Regular:

Esta é uma linguagem regular complexa que requer AFD com estados para rastrear paridade.

#### Abordagem simplificada (incompleta):

$$(a|b|c)^* \text{ (com restrições de paridade)}$$

**Solução via AFD:** Construir AFD com 2 estados (par/ímpar) e converter para ER.

## Explicação

### Explicação:

Esta linguagem é regular mas sua ER direta é muito complexa. A melhor abordagem é:

#### 1. Construir AFD:

- Estado  $q_{par}$ : número par de  $ba$  vistos (aceita)
- Estado  $q_{impar}$ : número ímpar de  $ba$  vistos
- Transições rastreiam quando  $ba$  é formado

#### 2. Converter AFD para ER:

- Usar algoritmo de eliminação de estados
- ER resultante será muito longa

**Desafio:** Contar ocorrências de substring requer rastrear contexto (último símbolo visto).

### Exemplo de lógica:

- Se último símbolo foi  $b$  e lemos  $a$ : incrementa contador
- Aceitamos quando contador é par ( $0, 2, 4, \dots$ )

## Exemplos

### Palavras que pertencem (par de $ba$ ):

- $\varepsilon$  (0 ocorrências de  $ba$  - par)
- $abc$  (0 ocorrências)
- $baba$  (2 ocorrências:  $ba$  e  $ba$ )
- $ababab$  (3  $ba$ 's... não, conta sobreposições)
- $bacba$  (2 ocorrências separadas)

**Observação:** A contagem exata depende se substrings podem se sobrepor.

### Palavras que NÃO pertencem (ímpar de $ba$ ):

- $ba$  (1 ocorrência)
- $bac$  (1 ocorrência)
- $aba$  (1 ocorrência)

## 2.11 (k) Não contém aa nem bb como subpalavras

**Linguagem:**  $L_k = \{w \mid w \text{ não contém } aa \text{ nem } bb \text{ como subpalavras}\}$

## Expressão Regular

### Expressão Regular:

$$(c^*ac^*b)^*c^*a^? \mid (c^*bc^*a)^*c^*b^?$$

Ou de forma mais compacta:

$$c^*(ac^*b)^*(a|b|\varepsilon)c^* \mid c^*(bc^*a)^*(a|b|\varepsilon)c^*$$

## Explicação

### Explicação:

A ER garante alternância forçada entre  $a$  e  $b$ :

- $c$ 's podem aparecer livremente em qualquer quantidade
- $a$ 's e  $b$ 's devem alternar (nunca  $aa$  ou  $bb$ )
- Padrões válidos:
  - $a$  seguido de  $b$  (com  $c$ 's opcionais entre/ao redor)
  - $b$  seguido de  $a$  (com  $c$ 's opcionais entre/ao redor)
  - Apenas  $c$ 's
  - No máximo um  $a$  ou um  $b$  sem repetição

### Restrições:

- Proibido:  $aa, bb$
- Permitido:  $ab, ba, aba, bab$ , etc. (com  $c$ 's)

## Exemplos

### Palavras que pertencem:

- $\varepsilon$  (vazia)
- $c, cc, ccc$  (apenas  $c$ 's)
- $a, b$  (um único símbolo)
- $ab, ba$  ( $a$  e  $b$  alternados)
- $aba, bab$  (alternância)
- $cab, abc, bca$  ( $c$ 's misturados)
- $cacbcac$  ( $a$  e  $b$  separados por  $c$ 's)
- $abacaba$  (espere, tem  $aa\dots$  não válido)
- $acabcac$  (válido se não tiver  $aa$  ou  $bb$ )

### Palavras que NÃO pertencem:

- $aa$  (proibido)
- $bb$  (proibido)
- $aab$  (contém  $aa$ )
- $bba$  (contém  $bb$ )
- $aabc$  (contém  $aa$ )
- $cbbc$  (contém  $bb$ )
- $abaa$  (contém  $aa$ )

## 2.12 (l) Quarto símbolo é um a

Linguagem:  $L_l = \{w \mid \text{o quarto símbolo da esquerda para a direita é um } a\}$

### Expressão Regular

#### Expressão Regular:

$$\Sigma^3 \cdot a \cdot \Sigma^*$$

Expandindo:

$$(a|b|c)(a|b|c)(a|b|c) \cdot a \cdot (a|b|c)^*$$

## Explicação

### Explicação:

A ER força o quarto símbolo a ser  $a$ :

- $\Sigma^3$ : exatamente 3 símbolos quaisquer (posições 1, 2, 3)
- $a$ : quarto símbolo deve ser  $a$  (posição 4)
- $\Sigma^*$ : zero ou mais símbolos quaisquer após a posição 4

**Tamanho mínimo:**  $|w| \geq 4$  (quatro símbolos ou mais)

**Padrão:** \_\_\_ $a$ ... (qualquer, qualquer, qualquer,  $a$ , qualquer...)

## Exemplos

### Palavras que pertencem:

- $bbba$  (quarto é  $a$ , tamanho 4)
- $ccca$  (quarto é  $a$ , tamanho 4)
- $abcab$  (quarto é  $a$ )
- $bbbabb$  (quarto é  $a$ )
- $123a567$  (notaçāo: posições 1,2,3 quaisquer, 4 é  $a$ )
- $aaaaaaaa$  (quarto é  $a$ )

### Palavras que NĀO pertencem:

- $\varepsilon$  (tamanho  $0 < 4$ )
- $a$  (tamanho  $1 < 4$ )
- $abc$  (tamanho  $3 < 4$ )
- $abcb$  (quarto é  $b$ , não  $a$ )
- $abcc$  (quarto é  $c$ , não  $a$ )
- $bbbb$  (quarto é  $b$ , não  $a$ )

### 3 Exercício 3: Análise de ERs

Dado o alfabeto  $\Sigma = \{a, b\}$ . Para cada uma das linguagens a seguir, representadas na forma de expressões regulares, apresente pelo menos duas palavras que pertençam à linguagem e duas que não pertençam:

#### 3.1 (a) $a^*b^*$

##### Expressão Regular

**Expressão Regular:**  $a^*b^*$

**Linguagem:**  $L = \{a^m b^n \mid m, n \geq 0\}$

Palavras formadas por zero ou mais  $a$ 's seguidos de zero ou mais  $b$ 's.

##### Explicação

###### Explicação:

A ER aceita palavras no padrão:  $\underbrace{aa \cdots a}_{m \geq 0} \underbrace{bb \cdots b}_{n \geq 0}$

###### Características:

- Todos os  $a$ 's (se houver) aparecem antes de todos os  $b$ 's (se houver)
- Pode ter apenas  $a$ 's, apenas  $b$ 's, ambos, ou nenhum (vazia)
- NÃO permite intercalação de  $a$ 's e  $b$ 's

##### Aceita

**Duas (ou mais) palavras que PERTENCEM à linguagem:**

1.  $\epsilon$  (cadeia vazia:  $a^0b^0$ )
2.  $a$  (apenas um  $a$ :  $a^1b^0$ )
3.  $b$  (apenas um  $b$ :  $a^0b^1$ )
4.  $ab$  ( $a^1b^1$ )
5.  $aab$  ( $a^2b^1$ )
6.  $abb$  ( $a^1b^2$ )
7.  $aaabbb$  ( $a^3b^3$ )
8.  $aaaaa$  (apenas  $a$ 's:  $a^5b^0$ )
9.  $bbbbbb$  (apenas  $b$ 's:  $a^0b^5$ )

**Rejeita**

**Duas (ou mais) palavras que NÃO PERTENCEM à linguagem:**

1.  $ba$  ( $b$  seguido de  $a$  - ordem errada)
2.  $aba$  ( $a$ 's e  $b$ 's intercalados)
3.  $abab$  (intercalação)
4.  $bba$  (tem  $b$  antes de  $a$ )
5.  $aabba$  (tem  $bb$  seguido de  $a$ )
6.  $baaab$  (começa com  $b$ , depois  $a$ 's)

**Razão:** Todas violam o padrão  $a^*b^*$  por terem  $b$  antes de  $a$  ou intercalação.

### 3.2 (b) $a(ba)^*b$

**Expressão Regular**

**Expressão Regular:**  $a(ba)^*b$

**Linguagem:**  $L = \{a(ba)^n b \mid n \geq 0\}$

Palavras que começam com  $a$ , terminam com  $b$ , e têm zero ou mais repetições de  $ba$  no meio.

**Explicação****Explicação:**

A ER define o padrão:  $a\underbrace{babab\dots ba}_n b$

**Características:**

- Sempre começa com  $a$  (obrigatório)
- Sempre termina com  $b$  (obrigatório)
- No meio: zero ou mais repetições do bloco  $ba$
- Tamanho mínimo: 2 (quando  $n = 0$ :  $ab$ )
- Tamanho geral:  $2 + 2n$  (sempre par)

**Padrão de tamanhos:**

- $n = 0$ :  $ab$  (tamanho 2)
- $n = 1$ :  $abab$  (tamanho 4)
- $n = 2$ :  $ababab$  (tamanho 6)
- $n = k$ : tamanho  $2 + 2k$

**Aceita****Duas (ou mais) palavras que PERTENCEM à linguagem:**

1.  $ab$  (caso base:  $n = 0$ )
2.  $abab$  (uma repetição:  $n = 1$ )
3.  $ababab$  (duas repetições:  $n = 2$ )
4.  $abababab$  (três repetições:  $n = 3$ )
5.  $ababababab$  (quatro repetições:  $n = 4$ )

**Rejeita****Duas (ou mais) palavras que NÃO PERTENCEM à linguagem:**

1.  $\varepsilon$  (vazia - não começa com  $a$  nem termina com  $b$ )
2.  $a$  (não termina com  $b$ )
3.  $b$  (não começa com  $a$ )
4.  $ba$  (ordem errada - começa com  $b$ )
5.  $aba$  (termina com  $a$ , não com  $b$ )
6.  $bab$  (começa com  $b$ , não com  $a$ )
7.  $aab$  (tem  $aa$  - não segue padrão  $(ba)^*$ )
8.  $abb$  (tem  $bb$  - não segue padrão  $(ba)^*$ )
9.  $aabb$  (não segue padrão  $a(ba)^*b$ )

**Razão:** Violam início com  $a$ , fim com  $b$ , ou padrão  $(ba)^*$  no meio.**3.3 (c)  $(aaa)^*$** **Expressão Regular****Expressão Regular:**  $(aaa)^*$ **Linguagem:**  $L = \{(aaa)^n \mid n \geq 0\} = \{a^{3n} \mid n \geq 0\}$ Palavras formadas por zero ou mais repetições de  $aaa$  (tamanho múltiplo de 3).

## Explicação

### Explicação:

A ER aceita apenas palavras compostas exclusivamente por  $a$ 's cujo tamanho é múltiplo de 3:

### Características:

- Apenas símbolo  $a$  (nenhum  $b$ )
- Tamanho deve ser  $3n$  para algum  $n \geq 0$
- Tamanhos válidos: 0, 3, 6, 9, 12, 15, ...

### Exemplos por tamanho:

- $n = 0$ :  $\varepsilon$  (tamanho 0)
- $n = 1$ :  $aaa$  (tamanho 3)
- $n = 2$ :  $aaaaaa$  (tamanho 6)
- $n = 3$ :  $aaaaaaaa$  (tamanho 9)

## Aceita

### Duas (ou mais) palavras que PERTENCEM à linguagem:

1.  $\varepsilon$  (vazia:  $(aaa)^0$ , tamanho  $0 = 3 \times 0$ )
2.  $aaa$  (uma repetição:  $(aaa)^1$ , tamanho 3)
3.  $aaaaaa$  (duas repetições:  $(aaa)^2$ , tamanho 6)
4.  $aaaaaaaa$  (três repetições:  $(aaa)^3$ , tamanho 9)
5.  $aaaaaaaaaaa$  (quatro repetições:  $(aaa)^4$ , tamanho 12)

**Rejeita**

**Duas (ou mais) palavras que NÃO PERTENCEM à linguagem:**

1.  $a$  (tamanho 1 - não é múltiplo de 3)
2.  $aa$  (tamanho 2 - não é múltiplo de 3)
3.  $aaaa$  (tamanho 4 - não é múltiplo de 3)
4.  $aaaaa$  (tamanho 5 - não é múltiplo de 3)
5.  $b$  (contém  $b$  - só permite  $a$ )
6.  $ab$  (contém  $b$ )
7.  $aaab$  (contém  $b$ )
8.  $aaaaaaaa$  (tamanho 7 - não é múltiplo de 3)

**Razão:** Tamanho não é múltiplo de 3 ou contém símbolo diferente de  $a$ .

### 3.4 (d) $aba \cup baba$

**Expressão Regular**

**Expressão Regular:**  $aba \cup baba$  (ou  $aba|baba$ )

**Linguagem:**  $L = \{aba, baba\}$

Linguagem finita contendo exatamente duas palavras.

**Explicação****Explicação:**

A ER define uma linguagem **finita** com apenas 2 elementos:

**Características:**

- União ( $\cup$  ou  $|$ ) de duas cadeias específicas
- Apenas duas palavras são aceitas:  $aba$  e  $baba$
- Qualquer outra palavra é rejeitada
- Tamanhos:  $aba$  tem tamanho 3,  $baba$  tem tamanho 4

**Importante:** A união não gera novas palavras, apenas aceita as duas especificadas.

**Aceita****Duas palavras que PERTENCEM à linguagem:**

1. *aba* (primeira alternativa da união)
2. *baba* (segunda alternativa da união)

**Observação:** Estas são as **únicas** duas palavras da linguagem!**Rejeita****Duas (ou mais) palavras que NÃO PERTENCEM à linguagem:**

1.  $\varepsilon$  (vazia)
2. *a* (diferente de *aba* e *baba*)
3. *b* (diferente de *aba* e *baba*)
4. *ab* (prefixo de *aba* mas não é *aba*)
5. *ba* (prefixo de *baba* mas não é *baba*)
6. *abab* (parece mistura mas não é exatamente *aba* nem *baba*)
7. *aaba* (ordem errada)
8. *abba* (diferente)
9. *bababa* (mais longo que *baba*)
10. Literalmente qualquer outra palavra diferente de *aba* ou *baba*

**Razão:** A linguagem contém APENAS as palavras *aba* e *baba*, nada mais.

## 4 Observações Finais

### 4.1 Resumo Completo dos Exercícios

Exercício	Linguagem	Regular?
1a	Contém pelo menos um 0	Sim
1b	Tamanho ímpar	Sim
1c	Tamanho múltiplo de 5	Sim
1d	Padrão $0^+1^*$	Sim
1e	Palíndromos	<b>NÃO</b> (LLC)
1f	$ww$ (repetição)	
2a	Sufixo $abe$ ou $cba$	Sim
2b	$\geq 3$ ocorrências de $abc$	Sim
2c	Primeiro = último	Sim
2d	Contém $aa$ ou $bb$	Sim
2e	( $aa$ ou $bb$ ) e sufixo $cccc$	Sim
2f	Contém $ab$ e $ba$	Sim
2g	$ x  = 3,  z  = 3$ ( $ w  \geq 6$ )	Sim
2h	Exatamente um $a$	Sim
2i	Sem $aa$ adjacente	Sim
2j	Par de substrings $ba$	Sim (complexa)
2k	Sem $aa$ nem $bb$	
2l	Quarto símbolo é $a$	
3a	$a^*b^*$	Sim
3b	$a(ba)^*b$	Sim
3c	$(aaa)^*$	Sim
3d	$aba \cup baba$	Sim (finita)

### 4.2 Conceitos-Chave Abordados

- Operadores básicos de ER:

- Concatenação:  $AB$  (A seguido de B)
- União:  $A|B$  ou  $A \cup B$  (A ou B)
- Fecho de Kleene:  $A^*$  (zero ou mais A's)
- Fecho positivo:  $A^+$  (um ou mais A's) =  $AA^*$

- Padrões comuns:

- Prefixo:  $padrão \cdot \Sigma^*$
- Sufixo:  $\Sigma^* \cdot padrão$
- Subpalavra:  $\Sigma^* \cdot padrão \cdot \Sigma^*$
- Tamanho fixo  $n$ :  $\Sigma^n$
- Tamanho ímpar:  $\Sigma(\Sigma\Sigma)^*$
- Tamanho múltiplo de  $k$ :  $(\Sigma^k)^*$
- Primeira = última:  $a\Sigma^*a|b\Sigma^*b|\dots|a|b|\dots$

- **Negações e restrições:**

- Sem símbolo  $s$ : usar apenas outros símbolos
- Sem substring  $xy$ : garantir alternância ou separação
- Exatamente  $k$  ocorrências: padrão específico

- **Limitações de ERs:**

- Não expressam palíndromos (requer LLC)
- Não expressam  $ww$  (requer LLC)
- Não expressam  $a^n b^n$  (requer LLC)
- Não conseguem contar arbitrariamente
- Limitadas a linguagens regulares

- **Lema do Bombeamento:**

- Ferramenta para provar que linguagem NÃO é regular
- Aplicado em palíndromos e  $ww$
- Contradição via bombeamento de substring

- **Tipos de linguagens regulares:**

- Finitas: conjunto fixo de palavras (ex:  $\{aba, baba\}$ )
- Infinitas: padrões repetitivos (ex:  $a^*b^*$ ,  $(aaa)^*$ )
- Com restrições: tamanho, posição, ocorrências

### 4.3 Técnicas de Construção de ERs

1. **Análise da linguagem:**

- Identificar padrões obrigatórios
- Identificar partes opcionais
- Identificar ordem e estrutura

2. **Decomposição:**

- Dividir em prefixo, meio, sufixo
- Usar união para alternativas
- Usar concatenação para sequências

3. **Casos especiais:**

- Cadeia vazia: incluir  $\varepsilon$  ou usar  $*$
- Tamanho 1: símbolos individuais
- Tamanho fixo: contar símbolos

4. **Verificação:**

- Testar com exemplos pequenos
- Verificar casos extremos (vazio, tamanho 1)
- Confirmar contra-exemplos

#### 4.4 Aplicações Práticas

- **Análise léxica:** Tokenização em compiladores
- **Validação de entrada:** Formatos de email, telefone, datas
- **Busca de texto:** grep, sed, editores de texto
- **Processamento de strings:** Parsing simples
- **Redes:** Filtros de pacotes, padrões de tráfego