Apostila de Apoio — Aula 2

Expressões Regulares

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

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 (Σ, ε, Σ*), 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 → cadeia → linguagem → 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. Expressões Regulares: ideia, definição e leitura

Na Aula 1, você aprendeu que uma linguagem é um conjunto de strings. O problema é que muitas linguagens são grandes (às vezes infinitas). Então, precisamos de uma forma compacta de descrever linguagens: as Expressões Regulares (ER).

2.1 Intuição: ER como uma "receita de strings"

Uma ER é como uma receita que descreve quais strings são aceitas. Ela não produz apenas uma string; ela descreve um conjunto inteiro.

Ideia central

Na teoria, ER usa apenas três ideias: OU (|), sequência (concatenação) e repetição (*). Essas três ideias já bastam para descrever todas as linguagens regulares.

2.2 Definição formal (indutiva)

Seja Σ um alfabeto. As ER são definidas por regras:

- Casos base: ∅ é ER; ε é ER; e para cada a ∈ Σ, o símbolo a é ER.
- Construção: se r e s são ER, então (r|s), (rs) e (r*) também são ER.
- Nada além disso é ER.

2.3 Linguagem denotada por uma ER: L(r)

Cada ER r descreve uma linguagem, escrita como L(r). As regras de significado são:

- $L(\emptyset)=\emptyset$ (não aceita nenhuma string).
- $L(\varepsilon)=\{\varepsilon\}$ (aceita apenas a string vazia).
- $L(a)=\{a\}$ (aceita apenas a string com um símbolo a).
- $L(r|s)=L(r) \cup L(s)$.
- $L(rs)=L(r)L(s)$ (concatenação de linguagens).
- $L(r^*)=(L(r))^*$.

2.4 Exemplos explicados

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

Exemplo A: r = 0|1

$L(0) = \{0\}$, $L(1) = \{1\} \Rightarrow L(0|1) = \{0,1\}$.

Exemplo B: r = 01

$L(01) = L(0)L(1) = \{0\}\{1\} = \{01\}$.

Exemplo C: r = 1*

L(1*) = {ε,1,11,111,...}. (Inclui ε porque pode repetir zero vezes.)

Exemplo D: r = (0|1)*

Descreve todas as strings binárias: ε,0,1,00,01,10,11,...

2.5 Como ler ER sem errar: precedência

Para evitar parênteses demais, usamos uma prioridade de operadores:

1) * (mais forte)
2) concatenação (sequência)
3) | (mais fraco)

Exemplo: 01* significa 0 seguido de vários 1 (0, 01, 011, ...). Já (01)* significa repetir o bloco 01 (ε, 01, 0101, ...).

3. Construção de ER a partir de enunciados

Agora vamos transformar frases em português em ER. A ideia é pensar em blocos:

“qualquer coisa” + “padrão obrigatório” + “qualquer coisa”.

3.1 Receitas úteis (muito usadas)

- “zero ou mais repetições de r”: r*
- “uma ou mais repetições de r”: rr*
- “r é opcional”: (r|ε)
- “começa com r”: r(Σ)*
- “termina com r”: (Σ)*r
- “contém r”: (Σ)* r (Σ)*

Quando Σ={0,1}, então (Σ) pode ser escrito como (0|1).

3.2 Exemplos guiados (com teste)

1. Termina com 1 (Σ={0,1})

ER: (0|1)*1

Aceita: 1, 01, 101, 0001. Rejeita: ε, 0, 10, 1110.

2. Começa com 1

ER: 1(0|1)*

Aceita: 1, 10, 101, 1110. Rejeita: ε, 0, 011.

3. Contém 101

ER: (0|1)*101(0|1)*

Aceita: 101, 01010? (não), 1101011 (sim).

Como testar

Dica de teste: pegue 3 exemplos que você acha que deveriam passar e 3 que deveriam falhar. Se sua ER está aceitando algo absurdo, revise os parênteses e o uso de *.

4. Exercícios graduais

Assuma Σ = {0,1} nos exercícios 1 a 8. Depois Σ={a,b} nos exercícios 9 a 12.

Nível 1 — interpretar (liste aceitas e rejeitadas)

1) r=1*. Liste 5 strings aceitas e 3 rejeitadas.
2) r=(01)*. Liste 4 aceitas e 3 rejeitadas.
3) r=0|11. Escreva exatamente a linguagem L(r).

Nível 2 — construir (padrões diretos)

4) ER para “todas as strings binárias”.
5) ER para “termina com 0”.
6) ER para “começa com 1”.

Nível 3 — construir (padrões mais específicos)

7) ER para “contém 11”.
8) ER para “tem exatamente um 1”.

Agora Σ={a,b}

9) ER para “strings de comprimento 2”.
10) ER para “strings só com a” (nenhum b).
11) ER para “começa com a”.
12) ER para “termina em bb”.

5. Erros comuns

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

- ε não é símbolo do alfabeto: ε é uma string vazia. Ela pertence a Σ*, mas normalmente não pertence a Σ.
- Não confundir Ø e ε: Ø aceita nada; ε aceita exatamente a string vazia.
- Σ não é Σ*: Σ é conjunto de símbolos; Σ* é conjunto de strings (inclui ε).
- Concatenação não é comutativa: "ab" ≠ "ba". Em geral L1L2 ≠ L2L1.
- 01* não é (01)*: 01* = 0 seguido de vários 1; (01)* = repetições do bloco 01.
- inclui zero repetições: por isso r* sempre aceita ε.
- ER não “conta” com memória infinita: não dá para impor “mesma quantidade” (como a^n b^n) só com ER.

6. Glossário rápido

- Σ (alfabeto): conjunto finito de símbolos permitidos.
- símbolo: um elemento de Σ (por exemplo 0 ou 1).
- string/cadeia (w): sequência finita de símbolos de Σ.
- ε (épsilon): string vazia (comprimento 0).
- |w|: comprimento da string w.
- Σ*: conjunto de todas as strings finitas sobre Σ (inclui ε).
- L: linguagem (um conjunto de strings), L ⊆ Σ*.
- 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).