

Introdução à Teoria da Computação

Expressões Regulares

Professor Murilo V. G. da Silva

Departamento de Informática
Universidade Federal do Paraná

2026/2

Expressões Regulares

- São expressões matemáticas que representam linguagens.

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1’s ou começam com 1 e tem em seguida um número arbitrário de 0’s”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
 - **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais

Expressões Regulares

- São expressões matemáticas que representam linguagens.
 - $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
 - **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais
- Exemplo:** Comece com as linguagens $\{0\}$ e $\{1\}$,

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
- **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais

Exemplo: Comece com as linguagens $\{0\}$ e $\{1\}$,

$\Rightarrow$ a partir de $\{0\}$, obtenha $\{\epsilon, 0, 00, 000, \dots\}$

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
- **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais

Exemplo: Comece com as linguagens $\{0\}$ e $\{1\}$,

$\Rightarrow$ a partir de $\{0\}$, obtenha $\{\epsilon, 0, 00, 000, \dots\}$ note: obtivemos $\{0\}^*$

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
- **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais

Exemplo: Comece com as linguagens $\{0\}$ e $\{1\}$,

$\Rightarrow$ a partir de $\{0\}$, obtenha $\{\epsilon, 0, 00, 000, \dots\}$ note: obtivemos $\{0\}^*$

$\Rightarrow$ a partir de $\{1\}$, obtenha $\{\epsilon, 1, 11, 111, \dots\}$

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
- **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais

Exemplo: Comece com as linguagens $\{0\}$ e $\{1\}$,

$\Rightarrow$ a partir de $\{0\}$, obtenha $\{\epsilon, 0, 00, 000, \dots\}$ note: obtivemos $\{0\}^*$

$\Rightarrow$ a partir de $\{1\}$, obtenha $\{\epsilon, 1, 11, 111, \dots\}$ note: obtivemos $\{1\}^*$

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1’s ou começam com 1 e tem em seguida um número arbitrário de 0’s”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
- **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais

Exemplo: Comece com as linguagens $\{0\}$ e $\{1\}$,

$\Rightarrow$ a partir de $\{0\}$, obtenha $\{\epsilon, 0, 00, 000, \dots\}$ note: obtivemos $\{0\}^*$

$\Rightarrow$ a partir de $\{1\}$, obtenha $\{\epsilon, 1, 11, 111, \dots\}$ note: obtivemos $\{1\}^*$

$\Rightarrow$ de $\{0\}$ e $\{1\}^*$, obtenha $\{0, 01, 011, \dots\}$

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
- **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais

Exemplo: Comece com as linguagens $\{0\}$ e $\{1\}$,

$\Rightarrow$ a partir de $\{0\}$, obtenha $\{\epsilon, 0, 00, 000, \dots\}$ note: obtivemos $\{0\}^*$

$\Rightarrow$ a partir de $\{1\}$, obtenha $\{\epsilon, 1, 11, 111, \dots\}$ note: obtivemos $\{1\}^*$

$\Rightarrow$ de $\{0\}$ e $\{1\}^*$, obtenha $\{0, 01, 011, \dots\}$ concatenação $\{0\} \cdot \{1\}^*$

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
- **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais

Exemplo: Comece com as linguagens $\{0\}$ e $\{1\}$,

$\Rightarrow$ a partir de $\{0\}$, obtenha $\{\epsilon, 0, 00, 000, \dots\}$ note: obtivemos $\{0\}^*$

$\Rightarrow$ a partir de $\{1\}$, obtenha $\{\epsilon, 1, 11, 111, \dots\}$ note: obtivemos $\{1\}^*$

$\Rightarrow$ de $\{0\}$ e $\{1\}^*$, obtenha $\{0, 01, 011, \dots\}$ concatenação $\{0\} \cdot \{1\}^*$

$\Rightarrow$ de $\{1\}$ e $\{0\}^*$, obtenha $\{1, 10, 100, \dots\}$

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
- **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais

Exemplo: Comece com as linguagens $\{0\}$ e $\{1\}$,

$\Rightarrow$ a partir de $\{0\}$, obtenha $\{\epsilon, 0, 00, 000, \dots\}$ note: obtivemos $\{0\}^*$

$\Rightarrow$ a partir de $\{1\}$, obtenha $\{\epsilon, 1, 11, 111, \dots\}$ note: obtivemos $\{1\}^*$

$\Rightarrow$ de $\{0\}$ e $\{1\}^*$, obtenha $\{0, 01, 011, \dots\}$ concatenação $\{0\} \cdot \{1\}^*$

$\Rightarrow$ de $\{1\}$ e $\{0\}^*$, obtenha $\{1, 10, 100, \dots\}$ concatenação $\{1\} \cdot \{0\}^*$

Expressões Regulares

- São expressões matemáticas que representam linguagens.
- $L =$ “strings que começam com 0 e tem em seguida um número arbitrário de 1's ou começam com 1 e tem em seguida um número arbitrário de 0's”
 - A frase acima (uma expressão não matemática) corresponde a uma linguagem .
 - Queremos uma expressão matemática do tipo $01^* + 10^*$ para a linguagem
- **Idéia chave:** Expressão que descreve como construir L a partir de linguagens triviais

Exemplo: Comece com as linguagens $\{0\}$ e $\{1\}$,

$\Rightarrow$ a partir de $\{0\}$, obtenha $\{\epsilon, 0, 00, 000, \dots\}$ note: obtivemos $\{0\}^*$

$\Rightarrow$ a partir de $\{1\}$, obtenha $\{\epsilon, 1, 11, 111, \dots\}$ note: obtivemos $\{1\}^*$

$\Rightarrow$ de $\{0\}$ e $\{1\}^*$, obtenha $\{0, 01, 011, \dots\}$ concatenação $\{0\} \cdot \{1\}^*$

$\Rightarrow$ de $\{1\}$ e $\{0\}^*$, obtenha $\{1, 10, 100, \dots\}$ concatenação $\{1\} \cdot \{0\}^*$

$\Rightarrow$ agora faça a união de $\{1\} \cdot \{0\}^*$ com $\{0\} \cdot \{1\}^*$

Como dito, isso corresponde à expressão $01^* + 10^*$

Expressões Regulares

Seja Σ um alfabeto qualquer. Uma expressão regular R , correspondente a uma linguagem $L(R) \subseteq \Sigma^*$, é definida indutivamente:

Expressões Regulares

Seja Σ um alfabeto qualquer. Uma expressão regular R , correspondente a uma linguagem $L(R) \subseteq \Sigma^*$, é definida indutivamente:

Base:

Expressões Regulares

Seja Σ um alfabeto qualquer. Uma expressão regular R , correspondente a uma linguagem $L(R) \subseteq \Sigma^*$, é definida indutivamente:

Base:

- 1 Tanto ϵ quanto $\emptyset$ são expressões regulares e $L(\epsilon) = \{\epsilon\}$ e $L(\emptyset) = \emptyset$.

Expressões Regulares

Seja Σ um alfabeto qualquer. Uma expressão regular R , correspondente a uma linguagem $L(R) \subseteq \Sigma^*$, é definida indutivamente:

Base:

- 1 Tanto ϵ quanto $\emptyset$ são expressões regulares e $L(\epsilon) = \{\epsilon\}$ e $L(\emptyset) = \emptyset$.
- 2 Seja $a \in \Sigma$. Então a é uma expressão regular e $L(a) = \{a\}$.

Expressões Regulares

Seja Σ um alfabeto qualquer. Uma expressão regular R , correspondente a uma linguagem $L(R) \subseteq \Sigma^*$, é definida indutivamente:

Base:

- 1 Tanto ϵ quanto $\emptyset$ são expressões regulares e $L(\epsilon) = \{\epsilon\}$ e $L(\emptyset) = \emptyset$.
- 2 Seja $a \in \Sigma$. Então a é uma expressão regular e $L(a) = \{a\}$.

Indução:

Expressões Regulares

Seja Σ um alfabeto qualquer. Uma expressão regular R , correspondente a uma linguagem $L(R) \subseteq \Sigma^*$, é definida indutivamente:

Base:

- 1 Tanto ϵ quanto $\emptyset$ são expressões regulares e $L(\epsilon) = \{\epsilon\}$ e $L(\emptyset) = \emptyset$.
- 2 Seja $a \in \Sigma$. Então a é uma expressão regular e $L(a) = \{a\}$.

Indução:

- 1 Se E e F são expressões regulares, então $E + F$ é uma expressão regular e $L(E + F) = L(E) \cup L(F)$.

Expressões Regulares

Seja Σ um alfabeto qualquer. Uma expressão regular R , correspondente a uma linguagem $L(R) \subseteq \Sigma^*$, é definida indutivamente:

Base:

- 1 Tanto ϵ quanto $\emptyset$ são expressões regulares e $L(\epsilon) = \{\epsilon\}$ e $L(\emptyset) = \emptyset$.
- 2 Seja $a \in \Sigma$. Então a é uma expressão regular e $L(a) = \{a\}$.

Indução:

- 1 Se E e F são expressões regulares, então $E + F$ é uma expressão regular e $L(E + F) = L(E) \cup L(F)$.
- 2 Se E e F são expressões regulares, então EF é uma expressão e $L(EF) = L(E)L(F)$.

Expressões Regulares

Seja Σ um alfabeto qualquer. Uma expressão regular R , correspondente a uma linguagem $L(R) \subseteq \Sigma^*$, é definida indutivamente:

Base:

- 1 Tanto ϵ quanto $\emptyset$ são expressões regulares e $L(\epsilon) = \{\epsilon\}$ e $L(\emptyset) = \emptyset$.
- 2 Seja $a \in \Sigma$. Então a é uma expressão regular e $L(a) = \{a\}$.

Indução:

- 1 Se E e F são expressões regulares, então $E + F$ é uma expressão regular e $L(E + F) = L(E) \cup L(F)$.
- 2 Se E e F são expressões regulares, então EF é uma expressão e $L(EF) = L(E)L(F)$.
- 3 Se E é uma expressão regular, então E^* é uma expressão regular e $L(E^*) = (L(E))^*$.

Expressões Regulares

Seja Σ um alfabeto qualquer. Uma expressão regular R , correspondente a uma linguagem $L(R) \subseteq \Sigma^*$, é definida indutivamente:

Base:

- 1 Tanto ϵ quanto $\emptyset$ são expressões regulares e $L(\epsilon) = \{\epsilon\}$ e $L(\emptyset) = \emptyset$.
- 2 Seja $a \in \Sigma$. Então a é uma expressão regular e $L(a) = \{a\}$.

Indução:

- 1 Se E e F são expressões regulares, então $E + F$ é uma expressão regular e $L(E + F) = L(E) \cup L(F)$.
- 2 Se E e F são expressões regulares, então EF é uma expressão e $L(EF) = L(E)L(F)$.
- 3 Se E é uma expressão regular, então E^* é uma expressão regular e $L(E^*) = (L(E))^*$.
- 4 Se E é uma expressão regular, então (E) é uma expressão regular e $L((E)) = L(E)$.

Exercícios:

1. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem de todas as strings sobre Σ .

Exercícios:

1. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem de todas as strings sobre Σ .
2. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem das strings sobre Σ que começam e terminam com a e têm pelo menos um b .

Exercícios:

1. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem de todas as strings sobre Σ .
2. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem das strings sobre Σ que começam e terminam com a e têm pelo menos um b .
3. Considere o alfabeto $\Sigma = \{a, b, \dots, z, \cdot\}$ (ou seja, alfabeto (minúsculo) adicionado do símbolo “ponto”). Apresente uma expressão regular para:

Exercícios:

1. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem de todas as strings sobre Σ .
2. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem das strings sobre Σ que começam e terminam com a e têm pelo menos um b .
3. Considere o alfabeto $\Sigma = \{a, b, \dots, z, \cdot\}$ (ou seja, alfabeto (minúsculo) adicionado do símbolo “ponto”). Apresente uma expressão regular para:
 - Conjunto das palavras terminadas com “.pdf”

Exercícios:

1. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem de todas as strings sobre Σ .
2. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem das strings sobre Σ que começam e terminam com a e têm pelo menos um b .
3. Considere o alfabeto $\Sigma = \{a, b, \dots, z, \cdot\}$ (ou seja, alfabeto (minúsculo) adicionado do símbolo “ponto”). Apresente uma expressão regular para:
 - Conjunto das palavras terminadas com “.pdf”
 - Conjunto com as duas palavras {factorization, factorisation}

Exercícios:

1. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem de todas as strings sobre Σ .
2. Seja $\Sigma = \{a, b, c\}$. Apresente a expressão regular para a linguagem das strings sobre Σ que começam e terminam com a e têm pelo menos um b .
3. Considere o alfabeto $\Sigma = \{a, b, \dots, z, \cdot\}$ (ou seja, alfabeto (minúsculo) adicionado do símbolo “ponto”). Apresente uma expressão regular para:
 - Conjunto das palavras terminadas com “.pdf”
 - Conjunto com as duas palavras {factorization, factorisation}
4. Apresente a ER para a linguagem binária das strings com 0's e 1's alternados.

Exemplo: “0’s e 1’s alternados”

A expressão regular para a linguagem $L = \{ \text{“strings contendo 0’s e 1’s alternados”} \}$ é

Exemplo: "0's e 1's alternados"

A expressão regular para a linguagem $L = \{ \text{"strings contendo 0's e 1's alternados"} \}$ é

$$\underline{(01)^* + (10)^* + 0(10)^* + 1(01)^*}.$$

Exemplo: “0’s e 1’s alternados”

A expressão regular para a linguagem $L = \{ \text{“strings contendo 0’s e 1’s alternados”} \}$ é

$$\underline{(01)^* + (10)^* + 0(10)^* + 1(01)^*}.$$

Note que:

Exemplo: “0’s e 1’s alternados”

A expressão regular para a linguagem $L = \{\text{“strings contendo 0’s e 1’s alternados”}\}$ é

$$\underline{(01)^* + (10)^* + 0(10)^* + 1(01)^*}.$$

Note que:

- Como $0 \in \Sigma$, então $\underline{0}$ é uma expressão regular que representa $\{0\}$.

Exemplo: "0's e 1's alternados"

A expressão regular para a linguagem $L = \{\text{"strings contendo 0's e 1's alternados"}\}$ é

$$\underline{(01)^* + (10)^* + 0(10)^* + 1(01)^*}.$$

Note que:

- Como $0 \in \Sigma$, então $\underline{0}$ é uma expressão regular que representa $\{0\}$.
- O mesmo vale para: $\underline{1}$ é uma expressão regular para $\{1\}$.

Exemplo: “0’s e 1’s alternados”

A expressão regular para a linguagem $L = \{\text{“strings contendo 0’s e 1’s alternados”}\}$ é

$$\underline{(01)^* + (10)^* + 0(10)^* + 1(01)^*}.$$

Note que:

- Como $0 \in \Sigma$, então $\underline{0}$ é uma expressão regular que representa $\{0\}$.
- O mesmo vale para: $\underline{1}$ é uma expressão regular para $\{1\}$.
- Portanto, $\underline{01}$ também é uma expressão regular e $L(\underline{01}) = \{0\} \cdot \{1\} = \{01\}$.

Exemplo: “0’s e 1’s alternados”

A expressão regular para a linguagem $L = \{ \text{“strings contendo 0’s e 1’s alternados”} \}$ é

$$\underline{(01)^* + (10)^* + 0(10)^* + 1(01)^*}.$$

Note que:

- Como $0 \in \Sigma$, então $\underline{0}$ é uma expressão regular que representa $\{0\}$.
- O mesmo vale para: $\underline{1}$ é uma expressão regular para $\{1\}$.
- Portanto, $\underline{01}$ também é uma expressão regular e $L(\underline{01}) = \{0\} \cdot \{1\} = \{01\}$.
- Logo, $\underline{(01)^*}$ é uma expressão regular e $L(\underline{(01)^*}) = \{\epsilon, 01, 0101, 010101, \dots\}$.

Exemplo: “0’s e 1’s alternados”

A expressão regular para a linguagem $L = \{ \text{“strings contendo 0’s e 1’s alternados”} \}$ é

$$\underline{(01)^* + (10)^* + 0(10)^* + 1(01)^*}.$$

Note que:

- Como $0 \in \Sigma$, então $\underline{0}$ é uma expressão regular que representa $\{0\}$.
- O mesmo vale para: $\underline{1}$ é uma expressão regular para $\{1\}$.
- Portanto, $\underline{01}$ também é uma expressão regular e $L(\underline{01}) = \{0\} \cdot \{1\} = \{01\}$.
- Logo, $\underline{(01)^*}$ é uma expressão regular e $L(\underline{(01)^*}) = \{\epsilon, 01, 0101, 010101, \dots\}$.
- Raciocínio análogo para os demais termos da expressão

Precedências:

- O operador $*$ tem precedência mais alta. Portanto, 01^* é equivalente a $0(1)^*$

Precedências:

- O operador $*$ tem precedência mais alta. Portanto, $\underline{01^*}$ é equivalente a $0\underline{(1)^*}$
- O operador com segunda maior precedência é o operador de concatenação. Portanto, $\underline{a + bc}$ é equivalente a $\underline{a + (bc)}$

Precedências:

- O operador $*$ tem precedência mais alta. Portanto, $\underline{01^*}$ é equivalente a $\underline{0(1)^*}$
- O operador com segunda maior precedência é o operador de concatenação. Portanto, $\underline{a + bc}$ é equivalente a $\underline{a + (bc)}$
- O operador de menor precedência é o operador $+$

Precedências:

- O operador $*$ tem precedência mais alta. Portanto, $\underline{01^*}$ é equivalente a $\underline{0(1)^*}$
- O operador com segunda maior precedência é o operador de concatenação. Portanto, $\underline{a + bc}$ é equivalente a $\underline{a + (bc)}$
- O operador de menor precedência é o operador $+$
- Como de costume usamos parentesis para alterar a precedência de operadores