MAC 414
Autômatos, Computabilidade e Complexidade
aula 2 — 16/09/2020

Recordando I

Recordando I

Alfabeto é um conjunto finito de letras.

Recordando I

Alfabeto é um conjunto finito de letras.

Palavra é uma sequência finita de letras.

Recordando I

Alfabeto é um conjunto finito de letras.

Palavra é uma sequência finita de letras.

Σ^* é o conjunto de todas as palavras sobre Σ

Recordando I

Alfabeto é um conjunto finito de letras.

Palavra é uma sequência finita de letras.

Σ^* é o conjunto de todas as palavras sobre Σ

É um monóide em que o produto é concatenação e o neutro é λ.

Recordando II

Recordando II

Uma linguagem é um conjunto de palavras.

Recordando II

Uma linguagem é um conjunto de palavras.

Produto de linguagens $AB = \{xy \mid x \in A, y \in B\}$

Recordando II

Uma linguagem é um conjunto de palavras.

Produto de linguagens $AB = \{xy \mid x \in A, y \in B\}$

Associativo: $A(BC) = (AB)C$

Recordando II

Uma linguagem é um conjunto de palavras.

Produto de linguagens $AB = \{xy \mid x \in A, y \in B\}$

Associativo: $A(BC) = (AB)C$

Distributivo com relação à união

$$A(B \cup C) = AB \cup AC \qquad (B \cup C)A = BA \cup CA$$

Ex:

$\{aa, aab, ab\}\{\lambda, b, a\} =$

$\{aa, \underline{aab}, ab, baa, baab, bab, aaa, aaab\}$

Vale com uniões infinitas

Vale com uniões infinitas

Se $\{A_i\}_{i \in I}$ é uma coleção de conjuntos,

$$\bigcup_{i \in I} A_i = \{x \mid \text{existe } i \in I \text{ tal que } x \in A_i\}$$

Vale com uniões infinitas

Se $\{A_i\}_{i \in I}$ é uma coleção de conjuntos,

$$\bigcup_{i \in I} A_i = \{x \mid \text{existe } i \in I \text{ tal que } x \in A_i\}$$

Fato: Se $A_i \subseteq B_i$ para todo $i \in I$, então $\bigcup_{i \in I} A_i \subseteq \bigcup_{i \in I} B_i$.

Vale com uniões infinitas

Se $\{A_i\}_{i \in I}$ é uma coleção de conjuntos,

$$\bigcup_{i \in I} A_i = \{x \mid \text{existe } i \in I \text{ tal que } x \in A_i\}$$

Fato: Se $A_i \subseteq B_i$ para todo $i \in I$, então $\bigcup_{i \in I} A_i \subseteq \bigcup_{i \in I} B_i$.

Proposição: $B \bigcup_{i \in I} A_i = \bigcup_{i \in I} B A_i$

Vale com uniões infinitas

Se $\{A_i\}_{i \in I}$ é uma coleção de conjuntos,

$$\bigcup_{i \in I} A_i = \{x \,|\, \text{existe } i \in I \text{ tal que } x \in A_i\}$$

Fato: Se $A_i \subseteq B_i$ para todo $i \in I$, então $\bigcup_{i \in I} A_i \subseteq \bigcup_{i \in I} B_i$.

Proposição: $B \bigcup_{i \in I} A_i = \bigcup_{i \in I} BA_i$

Dem: Sejam E e D os lados direito e esquerdo da igualdade.

$E \subseteq D$:

Vale com uniões infinitas

Se $\{A_i\}_{i \in I}$ é uma coleção de conjuntos,

$$\bigcup_{i \in I} A_i = \{x \mid \text{existe } i \in I \text{ tal que } x \in A_i\}$$

Fato: Se $A_i \subseteq B_i$ para todo $i \in I$, então $\bigcup_{i \in I} A_i \subseteq \bigcup_{i \in I} B_i$.

Proposição: $B \bigcup_{i \in I} A_i = \bigcup_{i \in I} BA_i$

Dem: Sejam E e D os lados direito e esquerdo da igualdade.

$E \subseteq D$: Se $x \in E$, existem $y \in B$, $z \in \bigcup_{i \in I} A_i$ tais que $x = yz$. Pela definição de união, existe $i \in I$ tal que $z \in A_i$. Logo, $x \in BA_i$, portanto $x \in D$.

Vale com uniões infinitas

Se $\{A_i\}_{i \in I}$ é uma coleção de conjuntos,

$$\bigcup_{i \in I} A_i = \{x \,|\, \text{existe } i \in I \text{ tal que } x \in A_i\}$$

Fato: Se $A_i \subseteq B_i$ para todo $i \in I$, então $\bigcup_{i \in I} A_i \subseteq \bigcup_{i \in I} B_i$.

Proposição: $B \bigcup_{i \in I} A_i = \bigcup_{i \in I} BA_i$

Dem: Sejam E e D os lados direito e esquerdo da igualdade.

$E \subseteq D$: Se $x \in E$, existem $y \in B$, $z \in \bigcup_{i \in I} A_i$ tais que $x = yz$. Pela definição de união, existe $i \in I$ tal que $z \in A_i$. Logo, $x \in BA_i$, portanto $x \in D$.

$E \supset D$:

Vale com uniões infinitas

Se $\{A_i\}_{i \in I}$ é uma coleção de conjuntos,

$$\bigcup_{i \in I} A_i = \{x \mid \text{existe } i \in I \text{ tal que } x \in A_i\}$$

Fato: Se $A_i \subseteq B_i$ para todo $i \in I$, então $\bigcup_{i \in I} A_i \subseteq \bigcup_{i \in I} B_i$.

Proposição: $B \bigcup_{i \in I} A_i = \bigcup_{i \in I} BA_i$

Dem: Sejam E e D os lados direito e esquerdo da igualdade.

$E \subseteq D$: Se $x \in E$, existem $y \in B$, $z \in \bigcup_{i \in I} A_i$ tais que $x = yz$. Pela definição de união, existe $i \in I$ tal que $z \in A_i$. Logo, $x \in BA_i$, portanto $x \in D$.

$E \supset D$: $A_i \subseteq \bigcup_{i \in I} A_i$, logo $BA_i \subseteq B \bigcup_{i \in I} A_i$, e usando o Fato acima, segue o resultado.

Potências

Potências

$$L^n = \{x \in \Sigma^* \mid x \text{ tem uma fatoração em } n \text{ palavras de } L\}$$

Potências

$L^n = \{x \in \Sigma^* \mid x$ tem uma fatoração em n palavras de $L\}$

Isso faz algum sentido com $n = 0$, e $L^0 = \{\lambda\}$.

Potências

$L^n = \{x \in \Sigma^* \mid x$ tem uma fatoração em n palavras de $L\}$

Isso faz algum sentido com $n = 0$, e $L^0 = \{\lambda\}$.

Ex: tome $L = \Sigma$, visto como conjunto de palavras de comprimento 1.
Σ^n é o conjunto de palavras de comprimento n.

Potências

$L^n = \{x \in \Sigma^* \mid x \text{ tem uma fatoração em } n \text{ palavras de } L\}$

Isso faz algum sentido com $n = 0$, e $L^0 = \{\lambda\}$.

Ex: tome $L = \Sigma$, visto como conjunto de palavras de comprimento 1.
Σ^n é o conjunto de palavras de comprimento n.

$(\{\lambda\} \cup \Sigma)^n = \lambda \cup \Sigma \cup \Sigma^2 \cup \cdots \cup \Sigma^n$

Potências

$L^n = \{x \in \Sigma^* \mid x$ tem uma fatoração em n palavras de $L\}$

Isso faz algum sentido com $n = 0$, e $L^0 = \{\lambda\}$.

Ex: tome $L = \Sigma$, visto como conjunto de palavras de comprimento 1.
Σ^n é o conjunto de palavras de comprimento n.

$(\{\lambda\} \cup \Sigma)^n = \lambda \cup \Sigma \cup \Sigma^2 \cup \cdots \cup \Sigma^n = \Sigma^{\leq n}$

Estrela

Estrela

Se $L \subseteq \Sigma^*$,

$$\begin{aligned} L^* &= \{x \in \Sigma^* \mid x \text{ tem uma fatoração em palavras de } L\} \\ &= \bigcup_{i \in \mathbb{N}} L^i = L^0 \cup L \cup L^2 \cup L^3 \cup \dots \end{aligned}$$

Estrela

Se $L \subseteq \Sigma^*$,

$$
\begin{aligned}
L^* &= \{x \in \Sigma^* \mid x \text{ tem uma fatoração em palavras de } L\} \\
&= \bigcup_{i \in \mathbb{N}} L^i = L^0 \cup L \cup L^2 \cup L^3 \cup \ldots
\end{aligned}
$$

Note que $\lambda \in L^*$. Mesmo $\emptyset^* = \{\lambda\}$.

Estrela

Se $L \subseteq \Sigma^*$,

$$\begin{aligned} L^* &= \{x \in \Sigma^* \mid x \text{ tem uma fatoração em palavras de } L\} \\ &= \bigcup_{i \in \mathbb{N}} L^i = L^0 \cup L \cup L^2 \cup L^3 \cup \ldots \end{aligned}$$

Note que $\lambda \in L^*$. Mesmo $\emptyset^* = \{\lambda\}$.

L^* é o menor submonóide de Σ^* contendo L.

Estrela

Se $L \subseteq \Sigma^*$,

$$\begin{aligned} L^* &= \{x \in \Sigma^* \mid x \text{ tem uma fatoração em palavras de } L\} \\ &= \bigcup_{i \in \mathbb{N}} L^i = L^0 \cup L \cup L^2 \cup L^3 \cup \ldots \end{aligned}$$

Note que $\lambda \in L^*$. Mesmo $\emptyset^* = \{\lambda\}$.

L^* é o menor submonóide de Σ^* contendo L.

Submonóide: $L^* L^* = L^*$.

Expressões regulares

São fórmulas envolvendo letras, λ, e símbolos para produto, união e estrela.

Expressões regulares

São fórmulas envolvendo letras, λ, e símbolos para produto, união e estrela.

Uma expressão regular sobre Σ é uma palavra sobre $\Sigma \cup \{\lambda, \emptyset, +, {}^*, (,)\}$. $\mathscr{E}_\Sigma$ ou só $\mathscr{E}$ é o conjunto de expressões regulares sobre Σ.

1. $\emptyset, \lambda, a \in \mathscr{E}$ para todo $a \in \Sigma$.

Expressões regulares

São fórmulas envolvendo letras, λ, e símbolos para produto, união e estrela.

Uma expressão regular sobre Σ é uma palavra sobre $\Sigma \cup \{\lambda, \emptyset, +, ^*, (,)\}$. $\mathscr{E}_\Sigma$ ou só $\mathscr{E}$ é o conjunto de expressões regulares sobre Σ.

1. $\emptyset, \lambda, a \in \mathscr{E}$ para todo $a \in \Sigma$.
2. Se $x, y \in \mathscr{E}$, então $(xy), (x+y) \in \mathscr{E}$.

Expressões regulares

São fórmulas envolvendo letras, λ, e símbolos para produto, união e estrela.

Uma expressão regular sobre Σ é uma palavra sobre $\Sigma \cup \{\lambda, \emptyset, +, ^*, (,)\}$. $\mathscr{E}_\Sigma$ ou só $\mathscr{E}$ é o conjunto de expressões regulares sobre Σ.

1. $\emptyset, \lambda, a \in \mathscr{E}$ para todo $a \in \Sigma$.
2. Se $x, y \in \mathscr{E}$, então $(xy), (x + y) \in \mathscr{E}$.
3. Se $x \in \mathscr{E}$ então $x^*, (x) \in \mathscr{E}$

Expressões regulares

São fórmulas envolvendo letras, λ, e símbolos para produto, união e estrela.

Uma expressão regular sobre Σ é uma palavra sobre $\Sigma \cup \{\lambda, \emptyset, +, ^*, (,)\}$. $\mathscr{E}_\Sigma$ ou só $\mathscr{E}$ é o conjunto de expressões regulares sobre Σ.

1. $\emptyset, \lambda, a \in \mathscr{E}$ para todo $a \in \Sigma$.
2. Se $x, y \in \mathscr{E}$, então $(xy), (x + y) \in \mathscr{E}$.
3. Se $x \in \mathscr{E}$ então $x^*, (x) \in \mathscr{E}$
4. Nada mais está em $\mathscr{E}$.

Expressões descrevem linguagens

$\mathcal{L}(x)$ = ling. denotada pela ER x

$$\mathscr{L}(\emptyset) = \emptyset, \mathscr{L}(\lambda) = \{\lambda\}, \mathscr{L}(a) = \{a\}$$
$$\mathscr{L}((xy)) = \mathscr{L}(x)\mathscr{L}(y)$$
$$\mathscr{L}((x+y)) = \mathscr{L}(x) \cup \mathscr{L}(y)$$
$$\mathscr{L}(x^*) = \mathscr{L}(x)^*$$
$$\mathscr{L}((x)) = \mathscr{L}(x)$$

Expressões descrevem linguagens

$$
\begin{aligned}
\mathscr{L}(\emptyset) = \emptyset, \mathscr{L}(\lambda) = \{\lambda\}, \mathscr{L}(a) = \{a\} \\
\mathscr{L}((xy)) = \mathscr{L}(x)\mathscr{L}(y) \\
\mathscr{L}((x+y)) = \mathscr{L}(x) \cup \mathscr{L}(y) \\
\mathscr{L}(x^*) = \mathscr{L}(x)^* \\
\mathscr{L}((x)) = \mathscr{L}(x)
\end{aligned}
$$

Uma linguagem é regular se existe uma ER que a descreve.

Expressões descrevem linguagens

$$
\begin{aligned}
\mathscr{L}(\emptyset) = \emptyset, \mathscr{L}(\lambda) &= \{\lambda\}, \mathscr{L}(a) = \{a\} \\
\mathscr{L}((xy)) &= \mathscr{L}(x)\mathscr{L}(y) \\
\mathscr{L}((x+y)) &= \mathscr{L}(x) \cup \mathscr{L}(y) \\
\mathscr{L}(x^*) &= \mathscr{L}(x)^* \\
\mathscr{L}((x)) &= \mathscr{L}(x)
\end{aligned}
$$

Uma linguagem é regular se existe uma ER que a descreve.

O problema da pertinência dada uma ER pode ser resolvido em tempo polinomial!

Equivalência

Equivalência

Consideramos duas ERs equivalentes se elas descrevem a mesma lingugem. Notação:

$$x = y \iff \mathscr{L}(x) = \mathscr{L}(y).$$

Equivalência

Consideramos duas ERs equivalentes se elas descrevem a mesma lingugem. Notação:

$$x = y \iff \mathscr{L}(x) = \mathscr{L}(y).$$

$a^{i_0}\, b a^{i_1}\, b a^{i_2}\, b \ldots$

$a^*(ba^*)^* = (a+b)^*$

Como conferir se é o caso, dadas x, y?

Equivalência

Consideramos duas ERs equivalentes se elas descrevem a mesma lingugem. Notação:

$$x = y \iff \mathscr{L}(x) = \mathscr{L}(y).$$

Como conferir se é o caso, dadas x, y?

$$(b + a((a+b)b)(a+b)a)^* a((a+b)b)^* = b^*((a+b)(ab^* + b))^*$$

Equivalência

Consideramos duas ERs equivalentes se elas descrevem a mesma lingugem. Notação:

$$x = y \iff \mathscr{L}(x) = \mathscr{L}(y).$$

Como conferir se é o caso, dadas x, y?

$$(b + a((a+b)b)(a+b)a)^* a((a+b)b)^* = b^*((a+b)(ab^* + b))^*$$

Vamos ver um algoritmo — só que é exponencial :-(.

Equivalência

Consideramos duas ERs equivalentes se elas descrevem a mesma lingugem. Notação:

$$x = y \iff \mathscr{L}(x) = \mathscr{L}(y).$$

Como conferir se é o caso, dadas x, y?

$$(b + a((a+b)b)(a+b)a)^* a((a+b)b)^* = b^*((a+b)(ab^* + b))^*$$

Vamos ver um algoritmo — só que é exponencial :-(.

E não tem jeito.

Açúcar sintático

Açúcar sintático

Parênteses: eliminar, com as prioridades naturais.

Açúcar sintático

Parênteses: eliminar, com as prioridades naturais.

Σ para soma de todas as letras

Açúcar sintático

Parênteses: eliminar, com as prioridades naturais.

Σ para soma de todas as letras

$L^+ = \bigcup_{i \geq 1} L^i$

Açúcar sintático

Parênteses: eliminar, com as prioridades naturais.

Σ para soma de todas as letras

$L^+ = \bigcup_{i \geq 1} L^i$

potências, intervalos de potências.

Açúcar sintático

Parênteses: eliminar, com as prioridades naturais.

Σ para soma de todas as letras

$L^+ = \bigcup_{i \geq 1} L^i$ $\qquad$ $\mathcal{E}$

potências, intervalos de potências.

E no Linux?

Açúcar sintático

Parênteses: eliminar, com as prioridades naturais.

Σ para soma de todas as letras

^E$ ER curso

$L^{+} = \bigcup_{i \geq 1} L^{i}$

ER grep

potências, intervalos de potências.

Σ^{n} 6 Σ^{n}

E no Linux?

| por $\cup$, escapes, abreviações para vários conjuntos de caracteres e potências.

[a-z]

$E? \leftarrow E + \lambda$

Açúcar sintático

Parênteses: eliminar, com as prioridades naturais.

Σ para soma de todas as letras

$L^+ = \bigcup_{i \geq 1} L^i$

potências, intervalos de potências.

E no Linux?

| por $\cup$, escapes, abreviações para vários conjuntos de caracteres e potências.

Memória

\1, \2, \3, . . .

Foge da teoria!

Exr de ER

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

$\{x \mid |x| \text{ é par}\} = (\Sigma^2)^*$

$\equiv 0 \bmod 7 \qquad (\Sigma^7)^* \Sigma^2$

$\equiv 2 \bmod 7$

$|x|_a$ = nº de vezes em que a ocorre em x

$\{x \mid |x|_1 \equiv 2 \bmod 4\}$

Existem linguagens não regulares

Existem linguagens não regulares

Montes!

- Com mais teoria dá para mostrar exemplos bem simples.

Existem linguagens não regulares

Montes!

- Com mais teoria dá para mostrar exemplos bem simples.
- Vale para qualquer forma de descrição

Teorema: ninguém descreve tudo

Teorema: ninguém descreve tudo

Esquema:

- Alfabetos $\Sigma \subseteq \Gamma$. Para simplificar, $\Sigma = \{0, 1\}$.

Teorema: ninguém descreve tudo

Esquema:

- Alfabetos $\Sigma \subseteq \Gamma$. Para simplificar, $\Sigma = \{0, 1\}$.
- Interpretação $F : \Gamma \to 2^{\Sigma^*} \cup$

Teorema: ninguém descreve tudo

Esquema:

- Alfabetos $\Sigma \subseteq \Gamma$. Para simplificar, $\Sigma = \{0, 1\}$.
- Interpretação $F : \Gamma \to 2^{\Sigma^*} \cup$

Teorema: ninguém descreve tudo

Esquema:

- Alfabetos $\Sigma \subseteq \Gamma$. Para simplificar, $\Sigma = \{0, 1\}$.
- Interpretação $F : \Gamma^* \to 2^{\Sigma^*} \cup$ 

Vamos mostrar $L \subseteq \Sigma^*$ que não está na imagem de F

Teorema: ninguém descreve tudo

Esquema:

- Alfabetos $\Sigma \subseteq \Gamma$. Para simplificar, $\Sigma = \{0, 1\}$.
- Interpretação $F : \Gamma^* \to 2^{\Sigma^*} \cup$

Vamos mostrar $L \subseteq \Sigma^*$ que não está na imagem de F

Numere as letras de Γ, codifique γ_i por 10^i, para $w \in \Gamma^*$, $C(w)$ é a concatenação das codificaçoes da letras.

Teorema: ninguém descreve tudo

Esquema:

- Alfabetos $\Sigma \subseteq \Gamma$. Para simplificar, $\Sigma = \{0, 1\}$.
- Interpretação $F : \Gamma \to 2^{\Sigma^*} \cup$

Vamos mostrar $L \subseteq \Sigma^*$ que não está na imagem de F

Numere as letras de Γ, codifique γ_i por 10^i, para $w \in \Gamma^*$, $C(w)$ é a concatenação das codificaçoes da letras.

$$L = \{C(w) | \, w \in \Gamma^*, C(w) \notin F(w)\}$$

Teorema: ninguém descreve tudo

Esquema:

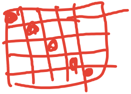
- Alfabetos $\Sigma \subseteq \Gamma$. Para simplificar, $\Sigma = \{0, 1\}$.
- Interpretação $F : \Gamma^* \to 2^{\Sigma^*} \cup$

Vamos mostrar $L \subseteq \Sigma^*$ que não está na imagem de F

Numere as letras de Γ, codifique γ_i por 10^i, para $w \in \Gamma^*$, $C(w)$ é a concatenação das codificaçoes da letras.

Diagonalização Cantor

$$L = \{C(w) \mid w \in \Gamma^*, C(w) \notin F(w)\}$$ LP



Se $L = F(w)$, $C(w) \in L$?

$C(w) \in F(w)$

$\Gamma = \{a, b, c\}$
$\quad\;\; 1 \quad 2 \quad 3$

$a \rightarrow 10$
$b \rightarrow 100$
$c \rightarrow 1000$

baba

$C(\cdot)$: 100 10 100 10

$a \rightarrow 00$
$b \rightarrow 01$
$c \rightarrow 10$