MAC 414
Autômatos, Computabilidade e Complexidade
aula 1 — 01/09/2020

O professor

Arnaldo Mandel
sala 110, bl. C
am@ime.usp.br

O professor

Arnaldo Mandel

sala 110, bl. C 

am@ime.usp.br

MAC0414

`www.ime.usp.br/~am/414`

Ligado ao *edisciplinas*
Avaliação por listas de exercícios.
Critério de aprovação ainda a ser definido
(ponderação e número de listas).

Sobre o que?

Problemas algorítmicos

Sobre o que?

Problemas algorítmicos

O que dá para fazer?

Sobre o que?

Problemas algorítmicos

O que dá para fazer?

Visto em quase todos os cursos
que envolvem algoritmos.

Sobre o que?

Problemas algorítmicos

O que dá para fazer?

Visto em quase todos os cursos
que envolvem algoritmos.

O que não dá para fazer?

Sobre o que?

Problemas algorítmicos

O que dá para fazer?

Visto em quase todos os cursos
que envolvem algoritmos.

O que não dá para fazer?

É O tema deste curso.
Limitações para a Computação.

Conteúdo

1. *Teoria dos Autômatos*
   Expressões regulares, Teorema de Kleene, minimização de autômatos.

Conteúdo

1. *Teoria dos Autômatos*
   Expressões regulares, Teorema de Kleene, minimização de autômatos.
2. *Computabilidade*
   Máquinas de Turing, problemas decidíveis e indecidíveis, o Problema da Parada.

Conteúdo

1. *Teoria dos Autômatos* 195x-1970 Viva
   Expressões regulares, Teorema de Kleene, minimização de autômatos.
2. *Computabilidade* 193x-1970
   Máquinas de Turing, problemas decidíveis e indecidíveis, o Problema da Parada.
3. *Complexidade* 1965
   P, NP, Teorema de Cook, existência de problemas NP-completos.

Chuva de definições

Apertem
os
cintos!

Chuva de definições

Apertem
os
cintos!

Vocês conhecem quase tudo, talvez com outros nomes.

Alfabeto

Alfabeto

Um alfabeto é um conjunto finito e não-vazio cujos elementos são chamados de letras ou *caracteres*.

Alfabeto

Um alfabeto é um conjunto finito e não-vazio cujos elementos são chamados de letras ou *caracteres*.

Um alfabeto genérico quase sempre será designado pela letra Σ.
Mas Γ ou A também têm chance.

Alfabeto

Um alfabeto é um conjunto finito e não-vazio cujos elementos são chamados de letras ou *caracteres*.

Um alfabeto genérico quase sempre será designado pela letra Σ.
Mas Γ ou A também têm chance.

As letras mais comuns são mesmo as letras e os dígitos no sentido usual.

Palavras

Palavras

Uma palavra (*string, cadeia*) sobre um alfabeto Σ é uma sequência finita de letras.

Palavras

Uma palavra (*string, cadeia*) sobre um alfabeto Σ é uma sequência finita de letras.

Se x é uma palavra, costumamos escrever $x = x_1 x_2 \ldots x_n = \mathsf{x}[1]\mathsf{x}[2]\cdots\mathsf{x}[\mathsf{n}]$.

Palavras

Uma palavra (*string, cadeia*) sobre um alfabeto Σ é uma sequência finita de letras.

Se x é uma palavra, costumamos escrever $x = x_1 x_2 \ldots x_n = \mathsf{x}[1]\mathsf{x}[2]\cdots\mathsf{x}[\mathsf{n}]$.

Ao escrever exemplos cujas letras sejam letras usuais, elas são mostradas da forma usual: MAC414, Arnaldo, 00011011100111.

Palavras

Uma palavra (*string, cadeia*) sobre um alfabeto Σ é uma sequência finita de letras.

Se x é uma palavra, costumamos escrever $x = x_1 x_2 \ldots x_n = \mathsf{x}[1]\mathsf{x}[2]\cdots\mathsf{x}[\mathsf{n}]$.

Ao escrever exemplos cujas letras sejam letras usuais, elas são mostradas da forma usual: MAC414, Arnaldo, 00011011000111.

O n que apareceu acima é o comprimento da palavra.

Palavras

Uma palavra (*string, cadeia*) sobre um alfabeto Σ é uma sequência finita de letras.

Se x é uma palavra, costumamos escrever $x = x_1 x_2 \ldots x_n = \mathrm{x}[1]\mathrm{x}[2]\cdots\mathrm{x}[\mathrm{n}]$.

Ao escrever exemplos cujas letras sejam letras usuais, elas são mostradas da forma usual: MAC414, Arnaldo, 00011011100011 1.

O n que apareceu acima é o comprimento da palavra.

Notação: $|x| = n$.

Todas as palavras

Todas as palavras

O conjunto de todas as palavras sobre Σ é denotado por Σ^*.

Todas as palavras

O conjunto de todas as palavras sobre Σ é denotado por Σ^*.

Ele sempre contém a palavra vazia, que foge um pouco da definição de palavra. Ela tem comprimento 0.

Todas as palavras

O conjunto de todas as palavras sobre Σ é denotado por Σ^*.

Ele sempre contém a palavra vazia, que foge um pouco da definição de palavra. Ela tem comprimento 0.

Denotada por λ (ϵ em muitos textos).

Todas as palavras

O conjunto de todas as palavras sobre Σ é denotado por Σ^*.

Ele sempre contém a palavra vazia, que foge um pouco da definição de palavra. Ela tem comprimento 0.

Denotada por λ (ϵ em muitos textos).

Para muitos usos, $\Sigma \subset \Sigma^*$.
Letra $\equiv$ palavra de comprimento 1.

Linguagens

Linguagens

Uma linguagem sobre Σ é um subconjunto de Σ^*.

Linguagens

Uma linguagem sobre Σ é um subconjunto de Σ^*.

Exemplos:

$\emptyset$, $\{\lambda\}$ e Σ^*.

Linguagens

Uma linguagem sobre Σ é um subconjunto de Σ^*.

Exemplos:

$\emptyset$, $\{\lambda\}$ e Σ^*.

$\{x \in \{a,b\}^* \mid |x| \equiv 3 \mod 7\}$

Linguagens

Uma linguagem sobre Σ é um subconjunto de Σ^*.

Exemplos:

$\emptyset$, $\{\lambda\}$ e Σ^*.

$\{x \in \{a,b\}^* \mid |x| \equiv 3 \mod 7\}$

$\{x \in \{0,1,\ldots,9\}^* \mid x$ é representação decimal de um primo$\}$

Linguagens

Uma linguagem sobre Σ é um subconjunto de Σ^*.

Exemplos:

$\emptyset$, $\{\lambda\}$ e Σ^*.

$\{x \in \{a, b\}^* \mid |x| \equiv 3 \mod 7\}$

$\{x \in \{0, 1, \ldots, 9\}^* \mid x$ é representação decimal de um primo$\}$

Com Σ =ASCII, os programas <insira sua linguagem> "sintaticamente corretos".

Linguagens

Uma linguagem sobre Σ é um subconjunto de Σ^*.

Exemplos:

$\emptyset$, $\{\lambda\}$ e Σ^*.

$\{x \in \{a,b\}^* \mid |x| \equiv 3 \mod 7\}$

$\{x \in \{0,1,\ldots,9\}^* \mid x$ é representação decimal de um primo$\}$

Com Σ =ASCII, os programas <insira sua linguagem> "sintaticamente corretos".

Codificações binárias de listas de adjacência de grafos.

Problemas fundamentais

Problemas fundamentais

alf de descrição $\Gamma \supseteq \Sigma$

Interpretação: $\Gamma^* \to 2^{\Sigma^*}$ (bestia)

$\mathcal{P}(\Sigma^*)$

Descrição

como descrever linguagens?

Problemas fundamentais

Descrição

como descrever linguagens?

Pertinência

dada a descrição de uma linguagem L e uma palavra x,

$$x \in L?$$

Concatenação

Concatenação

Dada palavras x, y de comprimentos n, m, seu produto (*concatenação*), denotado xy, é definido por:

$$(xy)_i = \begin{cases} x_i & \text{se } i \leq n \\ y_{i-n} & \text{se } n < i \leq n + m. \end{cases}$$

Concatenação

Dada palavras x, y de comprimentos n, m, seu produto (*concatenação*), denotado xy, é definido por:

$$(xy)_i = \begin{cases} x_i & \text{se } i \leq n \\ y_{i-n} & \text{se } n < i \leq n+m. \end{cases}$$

$$|xy| = |x| + |y|$$

Concatenação

Dada palavras x, y de comprimentos n, m, seu produto (*concatenação*), denotado xy, é definido por:

$$(xy)_i = \begin{cases} x_i & \text{se } i \leq n \\ y_{i-n} & \text{se } n < i \leq n + m. \end{cases}$$

$$|xy| = |x| + |y|$$

produto não comutativo!

$$x\lambda = x = \lambda x$$

Concatenação

Dada palavras x, y de comprimentos n, m, seu produto (*concatenação*), denotado xy, é definido por:

$$(xy)_i = \begin{cases} x_i & \text{se } i \leq n \\ y_{i-n} & \text{se } n < i \leq n+m. \end{cases}$$

$$|xy| = |x| + |y|$$

$$x\lambda = x = \lambda x$$

$$x(yz) = (xy)z \quad \text{(associatividade)}$$

Monóide

Monóide

Um monóide é um conjunto com uma operação binária associativa, e que tem um elemento neutro.

Monóide

Um monóide é um conjunto com uma operação binária associativa, e que tem um elemento neutro.

Σ^* é um monóide com neutro λ.

Monóide

Um monóide é um conjunto com uma operação binária associativa, e que tem um elemento neutro.

Σ^* é um monóide com neutro λ.

Num monóide:

Produtos podem ser escritos sem parênteses.

Monóide

Um monóide é um conjunto com uma operação binária associativa, e que tem um elemento neutro.

Σ^* é um monóide com neutro λ.

Num monóide:

Produtos podem ser escritos sem parênteses.

Podem se definir potências naturalmente.

$a^n = \underbrace{a \cdot a \cdot \cdots a}_{n \times}$

$a^0 =$ neutro

$a^{n+1} = a^n \cdot a$

$a^n \cdot a^m = a^{n+m}$

Fatores

Fatores

A palavra w é um fator de z se existem palavras x,y tais que

substring, subcadeia

$$z = xwy.$$

Fatores

A palavra w é um fator de z se existem palavras x,y tais que

$$z = xwy.$$

Se $z = xy$, então x é um prefixo de z e y é um sufixo de z.

Fatores

A palavra w é um fator de z se existem palavras x,y tais que

$$z = xwy.$$

Se $z = xy$, então x é um prefixo de z e y é um sufixo de z.

λ e x são prefixos e sufixos (*impróprios*) de x.

Fatores

A palavra w é um fator de z se existem palavras x,y tais que

$$z = xwy.$$

Ex: Se $x, y \in \Sigma^*$ comutam ($xy = yx$) então existe z tq x e y são potências de z.

Se $z = xy$, então x é um prefixo de z e y é um sufixo de z.

λ e x são prefixos e sufixos (*impróprios*) de x.

Fato: para $i = 0, 1, \ldots, |x|$, x tem um único prefixo e um único sufixo de comprimento i.

Operações sobre linguagens

Booleanas, claro: $\cup, \cap, \setminus$.

$\Sigma^* \setminus A$ melhor que $\overline{A}$

Só porque são conjuntos

Produto

Dadas $A, B \subseteq \Sigma^*$, seu produto é

$$\begin{aligned} AB &= \{xy \mid x \in A, y \in B\} \\ &= \{w \in \Sigma^* \mid \text{existem } x \in A, y \in B \text{ tq } w = xy\} \end{aligned}$$

Produto

Dadas $A, B \subseteq \Sigma^*$, seu produto é

$$AB = \{xy \mid x \in A, y \in B\}$$
$$= \{w \in \Sigma^* \mid \text{existem } x \in A, y \in B \text{ tq } w = xy\}$$

$\emptyset A =$

Produto

Dadas $A, B \subseteq \Sigma^*$, seu produto é

$$\begin{aligned} AB &= \{xy \mid x \in A, y \in B\} \\ &= \{w \in \Sigma^* \mid \text{existem } x \in A, y \in B \text{ tq } w = xy\} \end{aligned}$$

$\emptyset A = \emptyset = A\emptyset$

Produto

Dadas $A, B \subseteq \Sigma^*$, seu produto é

$$
\begin{aligned}
AB &= \{xy \mid x \in A, y \in B\} \\
&= \{w \in \Sigma^* \mid \text{existem } x \in A, y \in B \text{ tq } w = xy\}
\end{aligned}
$$

$$\emptyset A = \emptyset = A\emptyset$$

$$\{\lambda\}A = A = A\{\lambda\}$$

Associativo !

$A(BC) = (AB)C$

Combina com booleanas?

Combina com booleanas?

Mais ou menos. Tudo abaixo são exercícios.

Combina com booleanas?

Mais ou menos. Tudo abaixo são exercícios.

$A \subseteq B \Rightarrow CA \subseteq CB$ $\quad AC \subseteq BC$

mas a inclusão pode ser própria antes, com resultado igual depois (com $C \neq \emptyset$).

Combina com booleanas?

Mais ou menos. Tudo abaixo são exercícios.

$A \subseteq B \Rightarrow CA \subseteq CB$

mas a inclusão pode ser própria antes, com resultado igual depois (com $C \neq \emptyset$).

$CA \cup CB \subseteq C(A \cup B)$

$C(A \cup B) = CA \cup CB$

$CA \subseteq C(A \cup B)$

$CB \subseteq C(A \cup B)$

$CA \cup CB \subseteq C(A \cup B)$

Combina com booleanas?

Mais ou menos. Tudo abaixo são exercícios.

$A \subseteq B \Rightarrow CA \subseteq CB$

mas a inclusão pode ser própria antes, com resultado igual depois (com $C \neq \emptyset$).

$C(A \cup B) = CA \cup CB$

E quanto a $\cap$?

Potências

Potências

Fato: produto de linguagens é associativo.

Potências

Fato: produto de linguagens é associativo.

Podemos definir potências de uma linguagem.

Potências

Fato: produto de linguagens é associativo.

Podemos definir potências de uma linguagem.

$A^0 = \{\lambda\}, A^{n+1} = A^n A.$

Potências

Fato: produto de linguagens é associativo.

Podemos definir potências de uma linguagem.

$A^0 = \{\lambda\}, A^{n+1} = A^n A.$

A^n consiste de todas as palavras que tem uma fatoração em *exatamente* n fatores em A.

Potências

Fato: produto de linguagens é associativo.

Podemos definir potências de uma linguagem.

$A^0 = \{\lambda\}$, $A^{n+1} = A^n A$.

A^n consiste de todas as palavras que tem uma fatoração em *exatamente* n fatores em A.

$(\{\lambda\} \cup A)^n$ consiste de todas as palavras que têm uma fatoração em *até* n fatores em A.