Matemática Discreta para Cursos de Computação

Jackson Gomes \ jgomes@ceulp.edu.br

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Códigos-fontes

Prefácio

Este é um texto de apoio à disciplina Matemática Discreta para os cursos de computação do Centro Universitário Luterano de Palmas. Sempre que possível serão apresentadas referências a conceitos da computação e como eles se relacionam com os conceitos matemáticos apresentados. A principal referência do conteúdo utilizado aqui é (MENEZES, 2010).

0.1 Convenções

Os trechos de código apresentados no livro seguem o seguinte padrão:

- **comandos**: devem ser executados no prompt; começam com o símbolo `$`
- **códigos-fontes**: trechos de códigos-fontes de arquivos

A seguir, um exemplo de comando:

```
$ mkdir hello-world
```

O exemplo indica que o comando `mkdir`, com a opção `hello-world`, deve ser executado no prompt para criar uma pasta com o nome `hello-world`.

A seguir, um exemplo de código-fonte:

```
class Pessoa:
    pass
```

O exemplo apresenta o código-fonte da classe `Pessoa`. Em algumas situações, trechos de código podem ser omitidos ou serem apresentados de forma incompleta, usando os símbolos ... e `#`, como no exemplo a seguir:

```
class Pessoa:
    def __init__(self, nome):
        self.nome = nome

    def salvar(self):
        # executa validação dos dados
```

```
        ...
        # salva
        return ModelManager.save(self)
```

0.2 Conhecimentos necessários e desejáveis

Este texto aborda conceitos matemáticos com aplicações em computação. Portanto, conhecimentos básicos dos cursos de computação são necessários, como noções de lógica, algoritmos e programação e estruturas de dados. Além disso, são desejáveis conhecimentos de bancos de dados e orientação a objetos e também podem ser recursos úteis:

- Git (GIT COMMUNITY, [s.d.])
- Visual Studio Code (MICROSOFT, [s.d.])
- TypeScript (MICROSOFT, [s.d.])
- Node.js (NODE.JS FOUNDATION, [s.d.])
- npm (NPM, INC., [s.d.])

Capítulo 1

Introdução

1.1 Matemática Discreta

Conforme (MENEZES, 2010) as Diretrizes Curriculares do MEC para os cursos de computação e informática definem que:

> A matemática, para a área de computação, deve ser vista como uma ferramenta a ser usada na definição formal de conceitos computacionais (linguagens, autômatos, métodos etc.). Os modelos formais permitem definir suas propriedades e dimensionar suas instâncias, dadas suas condições de contorno.

Além disso, afirmam:

> Considerando que a maioria dos conceitos computacionais pertencem ao domínio discreto, a **matemática discreta** (ou também chamada álgebra abstrata) é fortemente empregada.

Desta forma, a **Matemática Discreta** preocupa-se com o emprego de técnicas e abordagens da matemática para o entendimento de problemas a serem resolvidos com computação. Mas o que significa ser **discreto**? A matemática, por si, trata também do domínio **contínuo**. Assim, estes domínios são opostos: contínuo e discreto. Para entender isso melhor, observe a Figura 1.1:

Figura 1.1 representa a função $y = x^2$, com $0 \leq x \leq 5$ em dois gráficos, sendo que o da direita destaca pontos selecionados, que representam 6 amostras (0, 1, 2, 3, 4 e 5).

Aumentando-se o número de amostras em dois instantes, para 10, 100 e 1000 teríamos, como mostra a Figura 1.2:

O que se pode perceber pela Figura 1.2 é que quanto mais se aumenta o número de amostras, mais se aproxima de uma curva perfeita. Entretanto, há um certo limite de percepção da perfeição dessa curva, por assim dizer. Por exemplo, embora a quantidade de amostras do gráfico da esquerda seja menor, a diferença para o gráfico da direita, visualmente falando, é pouco perceptível.

Considere outro exemplo: um computador possui uma capacidade de armazenamento virtualmente infinita. “Virtualmente” porque embora se aceite um limite, ele não é conhecido, já que a quan-

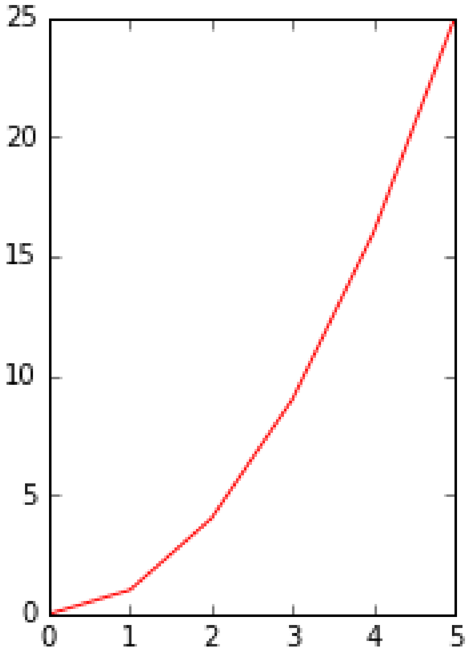

Figura 1.1: Gráfico da $y = x^2$, com $0 \leq x \leq 5$

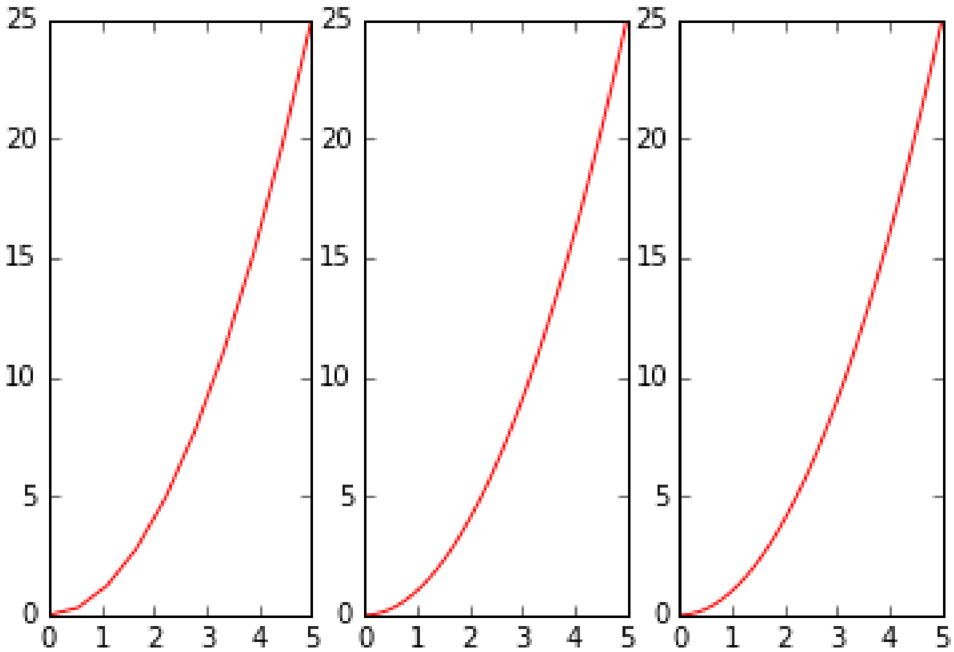

Figura 1.2: Gráfico da $y = x^2$ com mais amostras

tidade de unidades de armazemamento pode ser bastante grande, mas é **contável**. Assim, no contexto da computação, embora algo possa ser considerado finito ou infinito, ele é *contável* ou *discreto* no sentido de que pode ser enumerado ou sequenciado, de forma que não existe um elemento entre quaisquer dois elementos consecutivos da enumeração.

No exemplo do computador, embora a quantidade de unidades de armazenamento não seja conhecida, ela é contável e enumerável e não se pode afirmar que exista, por um exemplo, um disco rídigo desconhecido entre os array de discos composto por D1 e D2. Outro exemplo: na matemática, o conjunto dos números naturais é contável (ou enumerável), equanto o conjunto dos números reais não é contável.

Assim, a matemática discreta possui como ênfase os estudos matemáticos baseados em conjuntos contáveis, sejam eles finitos ou infinitos. De forma oposta, a *matemática do continuum* possui ênfase nos conjuntos não contáveis. Um exemplo disso são o cálculo diferencial e integral.

1.2 Teoria dos Conjuntos

Os **conjuntos** são a base da forma de representação de enumerações de elementos em matemática discreta. Por definição um conjunto é:

> uma estrutura que agrupa objetos e constitui uma base para construir estruturas mais complexas.

Segue uma definição mais formal:

> Um *conjunto* é uma coleção de zero ou mais objetos distintos, chamados *elementos* do conjunto, os quais não possuem qualquer ordem associada.

O fato de não haver uma *ordem associada* não significa que os elementos não possam estar ordenados, num dado contexto, conforme algum critério. Apenas indica que, no geral, isso não é obrigatório.

Há duas formas (notações) de representar conjuntos: notação por extensão e notação por compreensão.

Notação por extensão é quando todos os elementos do conjunto estão enumerados, representados entre chaves e separados por vírgula. Exemplo:

Vogais = $\{a, e, i, o, u\}$.

Entende-se que se um conjunto pode ser representado por extensão, então ele é *finito*. Caso contrário, é *infinito*.

Notação por compreensão representa conjuntos usando propriedades. Os exemplos a seguir usam uma pequena diferença de notação, mas representam a mesma coisa:

- Pares = $\{n \mid n$ é um número par$\}$
- Pares = $\{n : n$ é um número par$\}$

Este conjunto é interpretado como: o conjunto de todos os elementos n tal que n é um número par. A forma geral de representar um conjunto por propriedades é:

$X = \{x : p(x)\}$

Isso quer dizer que x é um elemento de X se a propriedade $p(x)$ for verdadeira.

A notação por propriedades é uma boa forma de representar conjuntos *infinitos*.

Há ainda uma outra forma aceitável de representar conjuntos usando uma representação semelhante à de por extensão. Exemplos:

- Digitos $= \{0, 1, 2, ..., 9\}$
- Pares $= \{0, 2, 4, 6, ...\}$

Embora haja elementos ausentes, substituídos por reticências (...) é completamente aceitável e entendível o que se quer informar com a descrição do conjunto.

O número de elementos de um conjunto A é representado por $|A|$ (isso também é chamado "cardinalidade"). Portanto, se $A = \{1, 2, 3\}$, $|A| = 3$ e $|\emptyset| = 0$.

A seguir, revemos conceitos de algumas relações entre e com conjuntos ou elementos.

1.2.1 Pertinência

Se um elemento a pertence ao conjunto A isso é representado como: $a \in A$. Caso contrário, se a não pertence a A, então representa-se como: $a \notin A$.

Exemplos: Pertence, não pertence

Quanto ao conjunto Vogais $= \{a, e, i, o, u\}$:
a) $a \in$ Vogais
b) $h \notin$ Vogais

Quanto ao conjunto $B = \{x : x$ é brasileiro$\}$:
a) Pele $\in B$
b) Bill Gates $\notin B$

1.2.2 Conjuntos importantes

O **conjunto vazio** é um conjunto sem elementos, representado como $\{\}$ ou $\emptyset$. Exemplos:

- o conjunto de todos os brasileiros com mais de 300 anos;
- o conjunto dos números que são, simultaneamente, ímpares e pares.

O **conjunto unitário** é um conjunto constituído por um único elemento. Exemplos:

- o conjunto constituído pelo jogador de futebol Pelé;
- o conjunto de todos os números que são, simultaneamente, pares e primos, ou seja: $P = \{2\}$;
- um conjunto unitário cujo elemento é irrelevante: $1 = \{*\}$.

O **conjunto universo**, normalmente denotado por U, contém todos os conjuntos considerados em um dado contexto.

Outros conjuntos importantes:

- N: o conjunto dos números naturais (inteiros positivos e o zero)
- Z: o conjunto dos números inteiros (inteiros negativos, positivos e o zero)
- Q: o conjunto dos números racionais (os que podem ser representados na forma de fração)
- I: o conjunto dos números irracionais
- R: o conjunto dos números reais

1.2.3 Alfabetos, palavras e linguagens

Em computação, e mais especificamente em linguagens de programação, um conceito importante é o que define o conjunto de elementos ou termos-chave da linguagem.

Um **alfabeto** um conjunto finito cujos elementos são denominados *símbolos* ou *caracteres*.

Uma **palavra** (cadeia de caracteres ou sentença) sobre um alfabeto é uma sequência finita de símbolos justapostos.

Uma **linguagem** [**formal**] é um conjunto de palavras sobre um alfabeto.

Exemplos: alfabeto, palavra
a) Os conjuntos $\emptyset$ e $\{a, b, c\}$ são alfabetos
b) O conjunto N não é um alfabeto
c) ϵ é uma palavra vazia
d) Σ é geralmente usada para representar um alfabeto
e) Σ^* é o conjunto de todas as palavras possíveis sobre o alfabeto Σ
f) ϵ é uma palavra do alfabeto $\emptyset$
g) $\{a, b\}^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, ...\}$

1.2.4 Continência, subconjunto e igualdade de conjuntos

A *continência* permite introduzir os conceitos de *subconjunto* e *igualdade de conjunto*.

Se todos os elementos de um conjunto A também são elementos de um conjunto B, então A está *contido* em B, o que é representado por: $A \subseteq B$. Isso também é lido como A é *subconjunto* de B.

Se $A \subseteq B$, mas há $b \in B$ tal que $b \notin A$, então pode-se dizer que A está *contido propriamente* em B, ou que A é *subconjunto próprio* de B. Isso é denotado por: $A \subset B$.

A negação de *subconjunto* e *subconjunto próprio* é, respectivamente:

- $A \nsubseteq B$ e
- $A \not\subset B$

Exemplos: continência, subconjunto
a) $\{a, b\} \subseteq \{b, a\}$ b) $\{a, b\} \subset \{a, b, c\}$, e $\{a, b\} \subseteq \{a, b, c\}$

A *igualdade de conjuntos* é um conceito baseado em pertinência: se os elementos de A também são elementos de B e vice-versa, então $A = B$. Formalmente, uma condição para $A = B$ é que $A \subseteq B$ e $B \subseteq A$.

Exemplo
$\{1, 2, 3\} = \{3, 3, 3, 2, 2, 1\}$

É importante notar que pertinência ($\in$) é usada entre elementos e conjuntos, enquanto continência ($\subset$ e $\subseteq$) é usada entre conjuntos.

Por definição, um conjunto qualquer é subconjunto de si mesmo, e $\emptyset$ é subconjunto de qualquer conjunto.

Exemplo
Seja $A = \{1, 2\}$ então os subconjuntos de A são: $\emptyset$, $\{1\}$, $\{2\}$ e $\{1, 2\}$.

1.3 Conjuntos, Tuplas e Listas

Uma **Tupla** (ou ênupla) é uma sequência ordenada de n elementos (ou componentes). As principais diferenças para **conjunto** são:

- uma ênupla pode conter um elemento mais de uma vez; e
- os elementos são, obrigatoriamente, ordenados, ou seja, cada elemento está em uma posição diferente.

A notação utilizada é $X = (c_1, c_2, ..., c_n)$ onde:

- X é uma ênupla com n componentes
- c_i é o enésimo componente da ênupla

Exemplos:

- $X = (1, 1, 2, 3)$
- $X = (1_1, 1_2, 2_3, 3_4)$
- $Y = (3_1, 2_2, 1_3)$

As mesmas relações de pertinência entre conjuntos e elementos podem ser aplicadas entre ênuplas e seus componentes.

Uma **Lista** é uma estrutura de dados que implementa o conceito matemático de **Tupla**, então podemos afirmar para Vogais $= (a, e, i, o, u)$:

- Vogais é uma lista com 5 elementos
- a é a primeira vogal
- u é a última vogal

1.4 Exercícios

Questão 1. Para cada conjunto abaixo: a) descreva de forma alternativa (usando outra forma de notação); e b) diga se é finito ou infinito.

a) todos os números inteiros maiores que 10
b) $\{1, 3, 5, 7, 9, 11, ...\}$
c) todos os países do mundo (Terra)
d) a linguagem de programação Python

Questão 2. Para $A = \{1\}$, $B = \{1, 2\}$ e $C = \{\{1\}, 1\}$, marque as afirmações corretas:

a) $A \subset B$
b) $A \subseteq B$
c) $A \in B$
d) $A = B$
e) $A \subset C$
f) $A \subseteq C$
g) $A \in C$
h) $A = C$
i) $1 \in A$
j) $1 \in C$
k) $\{1\} \in A$
l) $\{1\} \in C$
m) $\emptyset \notin C$
n) $\emptyset \subset C$

Questão 3. Sejam $a = \{x \mid 2x = 6\}$ e $b = 3$. É correto afirmar que $a = b$? Por que?

Questão 4. Quais todos os subconjuntos dos seguintes conjuntos?

a) $A = \{a, b, c\}$
b) $B = \{a, \{b, c\}, D\}$, dado que $D = \{1, 2\}$

Questão 5. O conjunto vazio está contido em qualquer conjunto, inclusive nele próprio? Justifique.

Questão 6. Todo conjunto possui um subconjunto próprio? Justifique.

Questão 7. Sejam $A = \{0, 1, 2, 3, 4, 5\}$, $B = \{3, 4, 5, 6, 7, 8\}$, $C = \{1, 3, 7, 8\}$, $D = \{3, 4\}$, $E = \{1, 3\}$, $F = \{1\}$ e X um conjunto desconhecido. Para cada item abaixo, determine quais dos conjuntos A, B, C, D, E ou F podem ser iguais a X:

a) $X \subseteq A$ e $X \subseteq B$
b) $X \not\subset B$ e $X \subseteq C$
c) $X \not\subset A$ e $X \not\subset C$

d) $X \subseteq B$ e $X \not\subset C$

Questão 8. Sejam A um subconjunto de B e B um subconjunto de C. Suponha que $a \in A$, $b \in B$, $c \in C$, $d \notin A$, $e \notin B$, $f \notin C$. Quais das seguintes afirmações são verdadeiras?

a) $a \in C$
b) $b \in A$
c) $c \notin A$
d) $d \in B$
e) $e \notin A$
f) $f \notin A$

Questão 9. Bancos de dados relacionais costumam representar conjuntos de dados organizados em tabelas, colunas e registros. É possível considerar alguma semelhança entre o conceito de tabela e o conceito de conjunto? Há recursos de consulta (em linguagem SQL, por exemplo) que permitam identificar, ainda que parcialmente, relações entre tabelas e registros (ou valores) como as expressadas nesse capítulo entre elementos e conjuntos? Explique.

Questão 10. Escolha uma linguagem de programação e demonstre seus recursos para lidar com conjuntos e listas. Utilizando código-fonte (e a sintaxe da linguagem de programação) demonstre e explique as diferenças conceituais e demonstre recursos da linguagem que permitam identificar as relações de pertinência, continência, subconjunto e igualdade.

Referências

GIT COMMUNITY. **Git**, [s.d.]. Disponível em: <https://git-scm.com/>. Acesso em: 22 jul. 2018

MENEZES, P. B. **Matemática discreta para computação e informática**. Traducao. 3. ed. Porto Alegre: Bookman, 2010.

MICROSOFT. **Visual Studio Code - Code Editing. Redefined**, [s.d.]. Disponível em: <https://code.visualstudio.com/>. Acesso em: 22 jul. 2018a

MICROSOFT. **TypeScript - JavaScript that scales**, [s.d.]. Disponível em: <https://www.typescriptlang.org/index.html>. Acesso em: 24 jul. 2018b

NODE.JS FOUNDATION. **Node.js**, [s.d.]. Disponível em: <https://nodejs.org>. Acesso em: 23 jul. 2018

NPM, INC. **npm**, [s.d.]. Disponível em: <https://www.npmjs.com/>. Acesso em: 23 jul. 2018

THE JQUERY FOUNDATION. **jQuery**, [s.d.]. Disponível em: <http://jquery.com/>. Acesso em: 22 jul. 2018

W3SCHOOLS. **JavaScript Tutorial**, [s.d.]. Disponível em: <https://www.w3schools.com/js/default.asp>. Acesso em: 22 jul. 2018a

W3SCHOOLS. **JavaScript and HTML DOM Reference**, [s.d.]. Disponível em: <https://www.w3schools.com/jsref/default.asp>. Acesso em: 22 jul. 2018b

W3SCHOOLS. **HTML5 Tutorial**, [s.d.]. Disponível em: <https://www.w3schools.com/html/default.asp>. Acesso em: 22 jul. 2018c

W3SCHOOLS. **CSS Tutorial**, [s.d.]. Disponível em: <https://www.w3schools.com/css/default.asp>. Acesso em: 22 jul. 2018d

W3SCHOOLS. **CSS Reference**, [s.d.]. Disponível em: <https://www.w3schools.com/cssref/default.asp>. Acesso em: 22 jul. 2018e

W3SCHOOLS. **HTML Element Reference**, [s.d.]. Disponível em: <https://www.w3schools.com/tags/default.asp>. Acesso em: 22 jul. 2018f