

职业技术教育软件人才培养模式改革项目成果教材



计算机数学基础

叶东毅 陈昭炯 朱文兴 编



高等教育出版社

职业技术教育软件人才培养模式改革项目成果教材

计算机数学基础

叶东毅 陈昭炯 朱文兴 编

高等教育出版社

内容提要

本书是职业技术教育软件人才培养模式改革项目成果教材。本书内容主要包括一元微积分、矩阵与线性代数初步、概率论基础、数理逻辑和图论等内容,分为8章。本书重在概念和基本方法的描述以及这些数学思想和方法在计算机科学领域中的应用。

本书可作为高等职业学校、高等专科学校、成人高校、示范性软件职业技术学院、本科院校及其举办的二级职业技术学院、教育学院以及民办高校的教学用书。此外,书中某些特设的章节和所提供的数学在计算机科学领域中的应用实例对于计算机专业的本科学子来说也是具有参考价值的。

图书在版编目(CIP)数据

计算机数学基础 / 叶东毅等编. —北京:高等教育出版社,
2004.7

ISBN 7-04-014748-3

I. 计... II. 叶... III. 电子计算机 - 数学基础 -
高等学校:技术学校 - 教材 IV. TP301.6

中国版本图书馆 CIP 数据核字(2004)第 032050 号

策划编辑 冯 英 责任编辑 关 旭 封面设计 王凌波 责任绘图 黄建英
版式设计 胡志萍 责任校对 康晓燕 责任印制 陈伟光

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

购书热线 010-64054588
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

经 销 新华书店北京发行所
排 版 北京高等教育出版社照排中心
印 刷 涿州市星河印刷有限公司

开 本 787×1092 1/16
印 张 14.75
字 数 350 000

版 次 2004 年 7 月第 1 版
印 次 2004 年 9 月第 3 次印刷
定 价 18.80 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号: 14748-00

职业技术教育软件人才培养模式改革项目 成果教材编审委员会

主 任 朱之文

委 员 (按姓氏笔画为序)

马肖风	王 珊	田本和	叶东毅	冯伟国
刘志鹏	李堂秋	郑祖宪	高 林	黄旭明

在软件类高等职业技术教育建设和改革过程中,福建省坚持教育创新,把改革教学内容和课程体系,加强专业建设、教材建设和教学队伍建设作为工作的重点。目前,根据软件行业发展趋势、就业环境和软件高等职业技术教育的办学特点,经组织专家论证和审定,福建省高校首批开设了可视化编程、Web 应用程序设计、软件测试、网络系统管理员、网络构建技术、数据库管理员、图形/图像制作、多媒体制作、计算机办公应用等 9 个软件高职专业,制订了较为科学合理的人才培养方案。为配合支持软件类高职教育的建设和改革,福建省教育厅聘请软件教育有关专家、学者和著名软件企业的高级工程技术人员成立了“职业技术教育软件人才培养模式改革项目成果教材编审委员会”,以“抓好试点规划,实施精品战略”为指导方针,认真吸取国内外软件技术发展成果,根据软件企业对人才培养提出的新要求和软件高职的办学特点,认真处理好教材的统一性与多样化、基本教材与辅助教材、学历教育教材与认证培训教材的关系,以组织开展软件高职公共基础课、专业基础课和专业主干课教材的建设为重点,同时扩大品种,实现教材系列配套,在此基础上形成特色鲜明、优化配套的软件高等职业技术教育教材体系。

本软件系列教材适用于本科院校、高职高专院校、成人高校及继续教育学院的软件高职类专业及相关专业使用。

由章仁寿、陈章仁、章仁强、章仁勇、章仁平、中共河南财经学院党委等编写。二〇〇三年五月

前 言

一般来说,在大学计算机科学与技术专业的教学计划中,微积分、线性代数、概率论和离散数学是作为独立的几门必修数学基础课程分若干个学期先后开设的。因此,每门课程都有相应的理论教科书和为数众多的教学参考书。

本书则是为软件高职专业学生而编写的“计算机数学基础”课程的教材。考虑到软件高职专业的特点和培养面向应用软件开发人才的目标定位,同时基于“够用、实用”的原则并考虑到课程教学学时(80学时)的限制,本书不可能也没有必要像标准的大学数学理论教科书那样详细地介绍微积分、线性代数、概率论和离散数学的内容,而只是也只能有选择地介绍这些理论中最基本的概念和方法。因此,本书的编写思想在于将上述微积分、线性代数、概率论和离散数学这几门学科的内容进行简化并整合在一起做一个导引性的介绍,除了必要且简单的证明之外,一般不给出定理的数学证明。与此同时,考虑到软件高职专业的特点,有必要介绍这些数学思想和方法在计算机科学领域中的若干应用,使学生对计算机科学和软件开发的数学基础与这些数学思想和方法可能的应用有一个总体的了解和把握,达到学习的目的。因此,本书在微积分部分,只介绍一元微积分的基本内容,对于多元函数和多元微积分则没有介绍;在线性代数部分,主要介绍矩阵的思想和方法以及求解线性方程组的基本思路;在概率论部分,着重介绍基本的概率计算方法、随机变量及其数字特征;在离散数学部分,只介绍集合论、简单数理逻辑和简单的图论方法。带*号的章节供学习时选学。

本书的主要特点如下:

1. 专门设置了一章内容(第1章)介绍了离散数学、微积分、线性代数和概率论对于计算机科学的重要性,特别是对于计算机专业学生容易忽视的连续数学理论(如微积分),用实例阐述了它们在计算机科学中的作用,使学生认识到不掌握一定的微积分知识就可能编写出错误的计算机程序。

2. 增加了一般高等数学教科书所没有的“递归函数与递归程序设计”、“矩阵乘法顺序与计算量分析”、“贝叶斯智能决策”等与计算机软件开发密切相关的内容。

3. 强调所学知识同计算机科学领域中问题的关联性。特别是在无穷级数、矩阵、随机变量、数理逻辑和图论等章节,给出了不少应用实例,如函数近似计算、数据库系统、计算机网络、计算机图形和图像处理、软件可靠性评估、人工智能等计算机科学的分支领域。

本书由福建省软件高职教学指导委员会组织编写,其中,第1章、第2章、第3章和第4章由福州大学计算机系叶东毅编写,第5章和第6章由福州大学计算机系陈昭炯编写,第7章和第8章由福州大学计算机系朱文兴编写,叶东毅负责全书的统稿。

本书是在软件高职人才培养改革实践的过程中诞生的,限于编者的水平及教材改革新模式的初次尝试,书中难免有错漏及欠妥之处,诚恳希望使用本教材的教师与学生提出宝贵意见,以利进一步的修改与提高。

编 者

二〇〇三年十二月

目 录

第 1 章 数学——计算机科学的基础	1
1.1 概述	1
1.2 离散数学与计算机科学	1
1.3 学习微积分的重要性	1
1.4 学习线性代数和概率论的重要性	4
1.5 本书的学习方法	4
第 2 章 一元微分学初步	5
2.1 集合	5
2.1.1 集合的概念	5
2.1.2 集合的表示	6
2.1.3 集合的包含与相等关系	7
2.1.4 集合的运算	8
2.1.5 二元关系	9
2.2 函数关系	11
2.2.1 函数关系的概念	11
2.2.2 常用的函数表示法	13
2.2.3 复合函数	14
2.2.4 递归函数与递归程序设计	15
2.3 极限的概念	16
2.3.1 数列的极限	17
2.3.2 函数的极限	20
2.3.3 变量的极限	24
2.3.4 无穷大量与无穷小量	25
2.4 极限的计算	27
2.4.1 极限的运算法则	28
2.4.2 极限存在的两个准则	31
2.4.3 两个重要的极限	32
2.5 函数的连续性	35
2.6 函数的导数——一类特殊的极限	39
2.6.1 导数的概念	39
2.6.2 导数的基本公式与运算法则	45
2.6.3 高阶导数	56
2.7 函数增量的估算——微分与中值定理	57
2.7.1 函数的线性逼近和微分	57

2.7.2 微分的求法	58
2.7.3 微分在近似计算中的应用	59
2.7.4 拉格朗日中值定理	60
2.8 和式的极限——无穷级数	60
2.8.1 无穷级数的概念	61
2.8.2 无穷级数的基本性质	63
2.8.3 正项级数	65
2.8.4 交错级数与任意项级数	67
2.8.5 幂级数	70
2.8.6 泰勒公式与泰勒级数	73
2.8.7 幂级数在近似计算中的应用	77
第 2 章习题	78
第 3 章 不定积分与定积分	84
3.1 原函数与不定积分的概念	84
3.2 不定积分的计算	85
3.2.1 基本性质和基本积分公式	85
3.2.2 换元积分法	86
3.2.3 分部积分法	89
3.3 定积分	90
3.3.1 定积分的概念	90
3.3.2 积分上限的函数及其导数	92
3.3.3 定积分的换元法和分部积分法	93
3.4 广义积分	95
第 3 章习题	96
第 4 章 矩阵与线性代数初步	98
4.1 矩阵	98
4.1.1 矩阵的概念	98
4.1.2 矩阵应用在计算机科学中的例子	100
4.1.3 一些特殊的矩阵	103
4.2 矩阵的基本运算	104
4.2.1 矩阵的基本运算	104
4.2.2 矩阵的运算规则	109
4.2.3 矩阵的乘法结合顺序与计算量分析	112
4.2.4 一般线性代数方程组的矩阵	115

形式	112	7.1.1 命题概念	170
4.3 矩阵的逆	113	7.1.2 命题符号化	171
4.4 消元法与矩阵的初等变换	115	7.2 命题公式与公式等值	174
4.4.1 消元法	116	7.2.1 命题公式	174
4.4.2 矩阵的初等变换	117	7.2.2 真值表	174
4.4.3 利用初等变换求逆矩阵	119	7.2.3 公式等值	176
第4章习题	120	7.2.4 等值演算	179
第5章 概率论基础	123	7.3 命题逻辑基本推理	180
5.1 概率及其相关概念	123	7.4 谓词逻辑及其应用	183
5.1.1 随机事件	123	7.4.1 谓词和量词	183
5.1.2 概率	124	7.4.2 谓词逻辑命题的否定形式	184
5.1.3 随机试验与样本空间	125	7.4.3 与量词有关的推理	186
5.1.4 事件的表示和关系	126	*7.5 数理逻辑应用举例	187
5.2 古典概率问题及计算方法	127	7.5.1 计算机信息检索	187
5.3 概率的加法性质及应用	130	7.5.2 程序的简化	190
5.4 条件概率与乘法定理	131	第7章习题	191
5.5 事件的独立性	132	第8章 图论初步	195
5.6 全概率公式	134	8.1 图的基本概念	195
5.7 贝叶斯公式与智能决策	136	8.1.1 图的定义	195
5.7.1 贝叶斯公式	136	8.1.2 相邻	196
*5.7.2 贝叶斯公式在智能决策中的		8.1.3 顶点的度数	197
应用	137	8.1.4 多重图、简单图和完全图	198
第5章习题	139	8.2 图的矩阵表示	200
第6章 随机变量的分布与数字特征	141	8.2.1 无向图的相邻矩阵	200
6.1 随机变量的分布	141	8.2.2 有向图的邻接矩阵	200
6.1.1 随机变量	141	8.2.3 无向图的关联矩阵	201
6.1.2 离散型随机变量及其典型分布	143	8.2.4 有向图的关联矩阵	202
6.1.3 连续型随机变量及其典型分布	146	8.3 通路、回路和图的连通性	203
6.1.4 随机变量的分布函数	151	8.3.1 通路与回路	203
*6.1.5 随机变量函数的分布	153	8.3.2 图的连通性	204
6.2 随机变量的数字特征	156	8.3.3 欧拉通路与哈密顿通路	206
6.2.1 数学期望及其性质	156	8.3.4 赋权图与最短路问题	208
6.2.2 方差及其性质	160	8.4 树及其应用	209
6.2.3 几种重要随机变量的数学期望与		8.4.1 无向树及其性质	209
方差	162	8.4.2 生成树与最小生成树	211
*6.2.4 随机变量函数的数学期望	163	8.4.3 根树及其应用	213
*6.2.5 切比雪夫不等式及其应用	165	*8.5 应用举例——网络路由选择	218
第6章习题	167	第8章习题	221
第7章 数理逻辑初步	170	附表 标准正态分布表	225
7.1 命题及其符号化	170	参考文献	226

第1章 数学——计算机科学的基础

本章学习目标

学习完本章后,对数学作为计算机科学的必备基础应有一个基本的认识,特别是对微积分等连续数学方法在计算机科学领域中的应用有初步的了解。

1.1 概 述

作为自然科学、社会科学、经济管理科学和工程技术科学的基础,数学的重要性是众所周知的。可以说,现代任何一门学科的发展都离不开数学。计算机科学更是如此,它的诞生和发展是同数学密不可分的。在计算机科学与技术的众多分支领域中,如计算机软件与理论、计算机应用技术、计算机体系结构、计算机通信和网络技术以及人工智能等领域中,数学理论的应用随处可见。这些数学理论不仅包括本书要介绍的微积分、线性代数、概率统计等常见和普遍使用的数学工具,而且还包括应用于计算机网络安全中的数论以及代数几何中的椭圆曲线理论这些相当“纯粹”的纯数学理论。另外,计算机科学中的一些基础理论本身就归属于数学学科的范畴,而不仅仅只是数学理论的应用成果,例如,布尔代数、关系代数、数理逻辑、图论、组合数学与密码学等。因此,对于计算机专业(计算机软件专业、计算机应用技术专业)的学生来说,学习和掌握好数学知识,不仅仅是像人们从一般意义上所说的能锻炼和提高人的逻辑思维能力、想像能力、推理能力和分析问题的能力,更是从事专业工作必备的一种基本素质。

1.2 离散数学与计算机科学

数据和信息在计算机中是以二进制代码的形式存储的,即计算机处理的是离散的对象。因此,研究离散对象的离散数学很自然地成为了计算机科学基础理论的一个重要组成部分。离散数学的研究范围十分广泛,作为计算机科学基础的布尔代数、关系代数、数理逻辑、组合数学和图论等都属于离散数学研究的内容。掌握离散数学知识对我们学好计算机科学和技术是至关重要的。

1.3 学习微积分的重要性

自从牛顿和莱布尼茨创立了微积分以来,微积分作为一个基本的处理连续数学的工具,不仅自身的理论得到迅猛的发展,而且被广泛地应用于自然科学、工程技术和经济管理等领域,即使是以处理离散对象为主的计算机科学,也离不开微积分这个经典的数学工具。对于这点,学习计算机专业的学生经常会有疑惑,下面来看几个问题。

问题 1 无理数计算问题。

实数是由有理数和无理数组成的,而且无理数要比有理数多得多。在计算机中是用二进制来表示数据的。由于二进制能表示的只是有理数,因此,无理数的表示只能靠近似的方法。另外,受字长的限制,计算机中能表示的数的范围是有限的,换句话说,用有理数近似无理数的精度是有限的,它只能精确到小数点后面有限个位数。那么,如何在精度许可的范围内尽可能精确地计算一个无理数呢?或者说如何在精度许可的范围内用一个有理数来尽可能准确地表示一个无理数呢?例如,如何在计算机中表示无理数 $\sqrt{2}$,圆周率 π 和自然对数的底 e 呢?这是一个需要利用微积分的方法加以解决的问题。具体地说,可以利用无穷级数展开的方法将这些无理数表示成无穷多项有理数的和。以 e 的计算为例(参见本书第2章关于无穷级数的介绍),有

$$e = 1 + 1 + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!} + \cdots$$

可取前 $n+1$ 项作 e 的近似值(n 的确定依赖于计算机的机器精度或者根据实际的需要确定)

$$e \approx 1 + 1 + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!}$$

例如,如果取 $n=7$,即级数的前 $7+1=8$ 项作近似计算,则和式中的每一项都是有理数,最终得

$$e \approx 1 + 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \frac{1}{6!} + \frac{1}{7!} = 2.718\ 26$$

问题2 初等函数的计算问题。

计算机的一个重要作用是能够进行大规模和快速的科学计算。而科学计算中最常遇到的是初等函数的计算。例如,对于给定的实数 x ,需要计算 $\sin x$ 、 $\ln x$ 、 $(1+x)^\alpha$ (α 是实数)等。同样,需要利用微积分中的函数幂级数展开方法加以解决,在本书的第2章中将介绍这方面的内容。

问题3 一个求和式的编程计算问题。

在中学的信息学课程中,已经初步学习了计算机编程语言,如Pascal程序设计语言或C语言,并且应用编程语言在计算机上实现一些简单的计算。现在,假设利用计算机计算下面求和式的结果

$$S(n) = \sum_{k=1}^n \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}.$$

其中, n 可以是一个很大的正整数。

一个自然的编程设计思想是:对于给定的 n ,先计算 $S(1)=1$,然后计算 $S(2)=S(1)+\frac{1}{2}$,

依次类推,最后通过 $S(n)=S(n-1)+\frac{1}{n}$ 计算出求和的结果 $S(n)$ 。这种思想可称为“由前向后加法”。

用伪Pascal代码的形式可以将上述编程思想写成如下的程序段(在中学没有学习过计算机编程语言的同学可以结合上述的编程思想进行理解,也可以待学习计算机编程语言之后阅读本段程序):

$S=0$;

```
begin
```

```
for k=1 to n do
```

```
  S=S+1/k
```

```
end;
```

```
write(S)
```

从理论上来说,上述程序是正确的,它能获得正确的答案。但是,当 n 比较大时,该程序在计算机上运行获得的结果离正确的结果会有很大的偏差,偏差的程度甚至可能非常大。

这是为什么呢?有两个原因。首先,在计算机中,两个数的相加是按照浮点运算方法进行的,存在着“大吃小”的问题,即当两个数的绝对值相差的倍数超过一定范围时,两个数之和等于其中大的那个数,小的那个数不起作用,即相当于小的那个数被吃掉了。有关这方面的内容在计算机基础课程中会专门介绍,这里,以十进制的浮点计算做个粗略和示意性的说明。假设某台计算机能表达的数的精度达到小数点后 10 位(为了说明起见所设,实际精度要比这高)。现在,要计算 $a+b$,其中 $a=10\,000\,000$, $b=0.000\,001$ 。按照浮点运算就高位的规则,因为 $a=1.0\times 10^7$,故应将 b 表示为 $b=0.000\,000\,000\,001\times 10^7$ 后进行相加运算。由于表示精度只能达到小数点后 10 位,因此, $b=0.000\,000\,000\,001\times 10^7$ 中的 0.000 000 000 001 超过了精度的范围,因而被置为 0,也就相当于在进行加法运算时, b 被 a “吃掉”了,导致 $a+b=a$ 的实际计算结果。

其次,考察一下编程计算的过程。对于 $1\leq m\leq n$, $S(m)=S(m-1)+\frac{1}{m}$,这里涉及两个数的加法,一个是 $S(m-1)$,另外一个为 $\frac{1}{m}$ 。当 m 很大时, $\frac{1}{m}$ 是一个很小的数,如果此时 $S(m-1)$ 是个很大的数,就可能出现上述的“大吃小”的问题,致使从这个 m 开始,无论 $n(>m)$ 有多大,都有 $S(m-1)=S(m)=S(m+1)=\cdots=S(n)$,换句话说,要计算的是 $S(n)$,但最终结果却是 $S(m-1)$ 。如果 $S(m-1)$ 与 $S(n)$ 相差很小,则这种近似还是可以接受的,如果相差甚远,则该程序无法得到正确(或近似正确)的结果。那么,实际情况是什么样的呢?在第 2 章无穷级数一节中,将会说明, $S(n)$ 随着 n 的增大而增大,并且逐渐趋向于“无穷大”(即“级数不收敛”),即当 m 很大时, $S(m-1)$ 是个很大的数,因此,确实会出现“大吃小”的问题;而且,当 $n>m$ 且 n 比 m 大得多时, $S(m-1)$ 与 $S(n)$ 相差是很大的。因此,上述程序无法得到正确的结果。产生编程设计错误的原因是不了解计算机浮点运算的规则,同时不了解级数的收敛性。

在计算机精度范围内正确编程计算 $S(n)$ 的方法应该是采用从后往前加的办法。对于给定的 n ,先计算 $F(1)=\frac{1}{n}$,然后计算 $F(2)=F(1)+\frac{1}{n-1}$,依此类推,最后计算 $F(n)=F(n-1)+1$,而 $S(n)=F(n)$,由此完成计算。读者可以分析一下为什么这种方法优于前面这个方法?

以上 3 个例子说明,无穷级数展开和级数收敛的知识对于计算机科学而言是十分重要的,甚至对于保障程序设计的正确性来说也是必不可少的。而要掌握有关无穷级数的知识,就必须学习极限、收敛性、函数的导数等知识,这些正是微积分的基本内容。况且,无穷级数本身就是微积分的一个重要研究内容。因此,学习微积分的重要性不言而喻。

1.4 学习线性代数和概率论的重要性

在大量的工程设计、经济管理以及计算机应用等领域中,常常需要求解多个未知数(简称多元)的线性方程组。研究多元线性方程组解的存在性、解的结构以及如何有效求解多元线性方程组,是线性代数的一个主要研究内容。矩阵是研究多元线性代数方程组求解的一个基本的工具,也是研究线性代数的一个基本工具。特别是矩阵的概念和描述问题的方式可以应用于诸如数据结构、计算机算法设计、图论、计算机图形学、计算机网络分析、图像处理等计算机科学的诸多分支领域。因此,掌握矩阵的方法及其应用不仅是学好线性代数的基础,也可为今后学习计算机科学理论和程序设计奠定基础。

概率论在计算机领域的应用非常广泛,学习和掌握概率论思想方法对今后学习计算机知识和从事相关工作有着重要的意义。例如,下列问题都同概率论的思想方法密切相关:如何根据机房遭受病毒攻击的记录分析病毒发作的规律,计算机系统各部件发生故障的可能性的计算,一台安装了两组密码的系统比只安装一组密码的系统安全系数大了多少,电脑公司如何分配用于不同媒体的广告费以获得更高的顾客购买率,一幅图像传输时的误码率有多大,一个软件工程师如何利用概率分析的方法进行故障诊断,等等。

除了基本的概率计算和分析方法外,概率论中的另一个重要概念“随机变量”及其分布模型在计算机科学领域的应用也是十分广泛的。例如,网站的点击次数用什么模型描述;服务器每分钟接到的服务请求次数服从什么分布;销售过程中售后服务人员安排多少合适;一个班级某课程的考试成绩尽管是个随机变量,但总体上都呈现出“中间高,两头低”的形式,这种现象如何描述;当急于访问某热门站点,但又不愿意等待太长时间时,访问到的可能性有多大;计算机图像处理中的直方图均衡化问题该如何有效地处理等。

本书将对上述这些概率论在计算机科学中的应用问题加以描述。

1.5 本书的学习方法

前面简要地介绍了离散数学、微积分、线性代数和概率论对于计算机科学的重要性,特别是对于那些不容易引起计算机专业学生重视的连续数学理论,如微积分和概率论并特别阐述了它们在计算机科学中的作用。

一般来说,在本科计算机的教学计划中,微积分、线性代数、概率论和离散数学是作为独立的几门必修数学基础课程分若干个学期先后开设的。因此,每门课程都有相应的理论教科书。而本书是为计算机软件高职专业学生而编写的,它旨在将微积分、线性代数、概率论和离散数学这几门课程的内容进行简化并结合在一起做引导性的介绍,使学生对计算机的数学基础有一个总体的了解和把握。因此,学生在学习本书时应着重理解本书所介绍的数学知识在计算机科学中的应用,做到学以致用。

第 2 章 一元微分学初步

本章学习目标

学习完本章后应掌握:

- 集合的概念、表示方法和基本运算以及二元关系的定义。
- 函数的概念和基本表示方法并了解函数的递归表示形式。
- 极限的概念、性质和运算规则并能进行较为简单的极限计算。
- 函数连续性的概念和判定连续性的方法。
- 函数导数的概念、基本公式和运算法则,并掌握隐函数求导方法和对数求导方法。
- 函数微分的概念及其在近似计算中的应用,了解拉格朗日中值定理的几何意义。
- 无穷级数的概念和级数收敛的必要条件,能用比较判别法和比值判别法等判定级数的敛散性,了解函数展开成泰勒级数的方法以及幂级数在近似计算中的应用。

2.1 集 合

2.1.1 集合的概念

集合可以说是数学中最基本的概念,以集合概念为基础发展起来的集合论自 19 世纪以来,不仅为整个经典数学的各个分支提供了共同的理论基础,而且,作为一个重要的抽象数据类型,在现代计算机科学中,特别是在关系代数、数据结构、算法与程序设计中被广泛的应用。

集合是一个原始的概念,但这个概念对于大家来说并不陌生。人们常用“物以类聚,人以群分”来表达具有相同属性的群体,这其实就表达了集合的概念。通俗地讲,集合是具有某种属性的事物的全体,或者是一些确定对象的汇总。例如,全体中国人,天上的星星,银行中所有的储户账号,一个图书馆中所有的藏书以及会用 C 语言编写程序的人等,这些对象的汇总事实上都是集合。

某个集合的元素是指属于该集合的任何对象。为书写统一起见,本书约定用大写的英文字母表示集合,小写英文字母表示集合的元素,用 $\in$ 表示“属于”, $\notin$ 表示“不属于”,它们反映了集合元素对集合的隶属关系。例如,元素 a 属于集合 T 可表示为 $a \in T$,元素 b 不属于集合 T 记为 $b \notin T$ 。

需要特别注意的是,这里讲的集合属于朴素集合论的范畴,它具有确定性的特征,即一个元素对于给定的某个集合来说,要么属于这个集合,要么不属于这个集合,二者必居其一。因此,如果说“跑步跑得比较快的人”的全体,就不是这里所说的集合的概念,因为没有确定的标准来描述构成它的对象,即没有明确什么叫跑得比较快。像这类没有确定性特征的集合可以用模糊集的概念来描述,模糊集理论在计算机科学领域有着重要的应用,由于篇幅所限,本书就不作介绍了。

2.1.2 集合的表示

给定一个集合,最常见的描述方法是列举法和特征描述法(也称谓词法)。

(1) 列举法

按照任意顺序列出集合的所有元素并用花括号 $\{\}$ 括起来。如果元素有重复,则只列一次。

例 2.1.1 由 5 个元素 a, b, c, d, e 组成的集合 A , 可以表示为: $A = \{a, b, c, d, e\}$ 。

例 2.1.2 由二元一次方程 $x^2 - 2x + 1 = 0$ 的根组成的集合 R , 可以表示为 $R = \{-1, 1\}$ 。

例 2.1.3 小于 10 的正整数集合, 可表示为 $Z = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ 。

如上述例子所示,用列举法表示一个集合时,通常要逐个列出集合的元素。当集合的元素很多时,这样做就显得十分烦琐。因此,为了简洁起见,在没有歧义的情况下,可以用延伸的方法和记号来表示集合中的大部分元素。例如小于 10 的正整数集合也可以表示为 $N = \{1, 2, 3, \dots, 9\}$, 其中的 4, 5, 6, 7, 8 这 5 个元素按照元素 1, 2, 3 逐一递增的规则延伸而得,直至元素 9 为止。

需要注意的是,采用延伸的方法时,列举元素的顺序要有规律可循,否则会造成歧义。例如,如果记 $N = \{1, 9, 2, \dots, 3\}$, 则无法确定该集合包含哪些元素。

列举法一般用于表示集合元素为有限多个的情形。对于含有无穷多个元素的集合,在没有歧义的情况下,仍然可以用列举法并结合上述延伸的方法来表示含有无穷多个元素的集合。例如,全体正整数的集合 $I = \{1, 2, 3, \dots\}$, 某个实数 a 的所有幂次方构成的集合 $S = \{1, a, a^2, a^3, \dots\}$ 。同样,为了避免二义性,列举元素要按照一定的顺序,以便确定延伸的规律。

列举法的优点是简明清楚,一般适合于表达仅包含有限个元素的有限集或在没有歧义的情况下含无限个可排列的元素的集合,但它有局限性。有一些集合是无法用列举法进行表示的,例如,由 $0 \sim 1$ 之间的全体实数组成的集合,由于元素“太多”了,无法用延伸的办法按照一定的顺序列举出来。有的集合,即便是有限集也难以用列举法表示出来。例如,六元一次方程 $x^6 + 2x^5 - 7x^4 + x^3 - 3x + 1 = 0$ 的根构成的集合有 6 个根,但是,由于无法像二元一次方程那样用求根公式表示出根来,因此,该集合难以用列举法表示出来。此时,可以采用直接描述特征的方法刻画一个集合。这就是下面要介绍的描述特征法。

(2) 描述特征法

用元素的特征来直接描述集合,并形式上表示为 $A = \{a | P(a)\}$, 其中 a 为集合的任意一个元素, $P(a)$ 为某个与 a 有关的规则或条件,更准确地说, $P(a)$ 是一个谓词,可以用谓词来概括元素的特征属性。在本书的第 7 章中,会专门进行讨论。描述特征法也叫做谓词法。

利用描述特征法,由 $0 \sim 1$ 之间的全体实数组成的集合就可以直接记为 $R = \{x | 0 \leq x \leq 1\}$, 而 $x^6 + 2x^5 - 7x^4 + x^3 - 3x + 1 = 0$ 的根构成的集合就可以表示为

$$R = \{x | x^6 + 2x^5 - 7x^4 + x^3 - 3x + 1 = 0\}$$

为表示方便起见,以后用符号“ $\wedge$ ”表示“与”或“且”的含义,用“ $\vee$ ”表示“或”的含义。在第 7 章中,还会进一步介绍这方面的内容。

需要注意的是,集合仅由它们所含有的不同的元素所决定,与元素出现的顺序或次数无关,例如

$\{a, b, c, d, e\}$ 与 $\{b, c, a, e, d\}$ 表示同一个集合。

式一民: $B \supseteq A$ 成因: $0 = 0 \cdot 1 + 1 \cdot 0$ $\{a, a, b, b, b, c, d, d, e\}$ 虽断断素元个 + 恰中 A , 而集容就是同样的集合。

但是,在实际应用中有时会用到元素可以多次出现的集合,这种集合称作多重集。在多重集中某元素出现的次数叫做该元素的重复数。在表示一个多重集时,应表明元素的重复数,例如把

式一民: 集合式 $\{a, a, b, b, b, c, d, d, e\}$ 不合集个一果成 4.1.5 定义

看作多重集,则可以记做 式一民: 集合式 $\{0 = 1 + 1 \cdot a \wedge \text{数实数 } x | x\} = A$ 合集,成因

集合个一果出 $\{2 \cdot a, 3 \cdot b, 1 \cdot c, 2 \cdot d, 1 \cdot e\}$ 果,成因: 集合式 $\{x\} = A$ 合集,成因

其中,2、3、1、2 和 1 分别是 a 、 b 、 c 、 d 和 e 的重复数。 集合个合集时一果集合 1.1.5 定义

给定多重集 集合个合集个某量集合的及相果成,成因: 集合个合集个一果 2.1.5 定义

$$S = \{\lambda_1 \cdot x_1, \lambda_2 \cdot x_2, \dots, \lambda_n \cdot x_n\}$$

对于元素 x_i ,重复数的值可以是某个正整数,也可以是 0 或 ∞ ,如果 $\lambda_i = 0$,则认为元素 $x_i \notin S$;如果 $\lambda_i = \infty$,则认为 S 中有无穷多个 x_i 。容易看出,一般集合就是 x_i 只能取 0 或 1 的多重集。在本书中如果没有特别说明,所指集合均为一般集合,不是多重集。在本书的第 8 章中,将用到多重集的概念。

2.1.3 集合的包含与相等关系

一个集合可以有不同的表示方法,因此,形式上不一样的两个集合有可能是同一个集合。那么,如何判断两个集合是否相等,以及集合之间能否像实数那样区分出大小呢?为此,需要引入集合的包含和相等的概念。

定义 2.1.1 设 A, B 是两个集合,如果 A 的每一个元素都是 B 的元素,则称 A 是 B 的子集,记为 $A \subseteq B$,也可称 B 包含 A ,记为 $B \supseteq A$ 。

定义 2.1.2 设 A, B 是两个集合,如果 $A \subseteq B$,而且,至少存在一个元素 $x \in B$,但 $x \notin A$,则称 A 是 B 的真子集,记为 $A \subsetneq B$ 。

定义 2.1.3 设 A, B 是两个集合,如果它们相互包含,即 $A \subseteq B$ 且 $B \subseteq A$,则称 A 与 B 相等,记做 $A = B$ 。如果 A 与 B 不相等,则记做 $A \neq B$ 。

定义 2.1.3 实际上提供了一种判断两个集合相等的方法。即要说明两个集合相等,只要说明它们互相包含。

例 2.1.4 给定集合 $A = \{x | x \leq 10 \wedge x \text{ 是素数}\}$, $B = \{2, 3, 5\}$,试判断它们之间是否存在包含或相等的关系。

解 集合 A 是用特征描述法表示,而集合 B 则是用列举法表示。根据特征描述法可知, $A = \{2, 3, 5, 7\}$ 。因为列举法比较直观,因此,先判断 B 中的元素是否属于 A 。因为 2、3 和 5 这三个数都是小于 10 的素数,即对 B 中的任何一个元素 x ,都有 $x \in A$,故 $B \subseteq A$,即 B 是 A 的子集。另一方面,因为 A 中有一个元素 7 不属于 B ,因此, B 是 A 的真子集,即 $B \subsetneq A$ 。

例 2.1.5 考虑如下两个集合

$$A = \{x | x \leq 10 \wedge x \text{ 是素数}\}, B = \{x | x^4 - 17x^3 + 101x^2 - 247x + 210 = 0\} \quad (1)$$

试判断它们之间是否存在包含或相等的关系。

解 两个集合采用的都是用特征描述法表示的。对例 2.1.4 采用列举法,有 $A = \{2, 3, 5, 7\}$,

容易验证, A 中的 4 个元素都满足方程 $x^4 - 17x^3 + 101x^2 - 247x + 210 = 0$, 因此 $A \subseteq B$; 另一方面, 因为一元 4 次多项式方程有且仅有 4 个根, 即 2、3、5 和 7 分别是该方程的根, 故也有 $B \subseteq A$, 由定义 2.1.3, $A = B$ 。

下面介绍一个特殊的集合。

定义 2.1.4 如果一个集合不包含任何元素, 则称该集合为空集, 记做 $\emptyset$ 。

例如, 集合 $A = \{x | x \text{ 是实数} \wedge x^2 + 1 = 0\}$ 就是空集。因为方程 $x^2 + 1 = 0$ 没有实数解, 因此, 该集合不含任何元素。再如, 集合 $A = \{x | x < 0 \text{ 且 } x > 1\}$ 也是一个空集。

性质 2.1.1 空集是一切集合的子集。换句话说, 空集属于任何集合。

定义 2.1.5 在一个具体问题中, 如果所涉及的集合都是某个集合的子集, 则称这个集合为全集, 记做 U 。

全集是一个相对的概念, 它依赖于所研究的问题。问题不同, 对应的全集也不同。例如, 在实数集合中, 如果关心的只是有理数, 则全集可以是有理数集合; 如果只研究整数, 则全集就可以取整数集合, 当然, 也可以把有理数当作全集。

2.1.4 集合的运算

实数之间有加、减、乘、除和幂等基本运算, 并且满足一定的运算规律。类似地, 集合与集合之间也可以定义一些基本运算(如并、交、补和幂等)并且满足相应的一些运算规律, 如交换律、分配律等。给定若干集合, 可以利用集合的并、交、补和幂等运算来产生一系列新的集合。

定义 2.1.6 设 A, B 是两个集合, 则它们的并集 $A \cup B$ 、交集 $A \cap B$ 、相对补集 $A \setminus B$ 分别为

$$A \cup B = \{x | x \in A \vee x \in B\}$$

$$A \cap B = \{x | x \in A \wedge x \in B\}$$

$$A \setminus B = \{x | x \in A \wedge x \notin B\}$$

换句话说, $A \cup B$ 由 A 和 B 中的元素构成, $A \cap B$ 由 A 和 B 中的公共元素构成, $A \setminus B$ 由属于 A 但不属于 B 的元素构成。例如

$$A = \{a, b, c, d\}, B = \{a, c, e, f\}$$

则

$$A \cup B = \{a, b, c, d, e, f\}$$

$$A \cap B = \{a, c\}$$

$$A \setminus B = \{b, d\}$$

$$B \setminus A = \{e, f\}$$

当两个集合的交集为空时, 称它们是不交的。

含有 n 个元素的集合简称为 n 元集, 它的含有 m 个元素的子集称为 m 元子集。下面讨论给定一个 n 元集, 求出它的全部子集的方法。

例如, 集合 $A = \{a, b, c\}$, 其子集有:

(1) 0 元子集, 也就是空集, 只有一个: $\emptyset$ 。

(2) 1 元子集, 有 3 个: $\{a\}, \{b\}, \{c\}$ 。

(3) 2 元子集, 有 3 个: $\{a, b\}, \{a, c\}, \{b, c\}$ 。

(4) 3元子集,就是集合本身: $\{a, b, c\}$ 。因此,集合 A 共有 8 个子集。

一般地,对于 n 元集,它的 0 元子集有 C_n^0 个,1 元子集有 C_n^1 个…… m 元子集有 C_n^m 个, n 元子集有 C_n^n 个,故子集总数为

$$C_n^0 + C_n^1 + \cdots + C_n^n = (1+1)^n = 2^n \text{ 个}$$

定义 2.1.7 设 A 为集合,把 A 的全体子集构成的集合叫做 A 的幂集,记做 2^A 。

由前面的分析可知,如果 A 有 n 个元素,则其幂集有 2^n 个元素。

定义 2.1.8 全集 U 中所有不属于 A 的元素构成的集合,称为 A 的补集,记做 $\complement_U A$,本书记做 A' 。

例如, $U = \{a, b, c, d, e, f, g\}$, $A = \{a, b, c\}$, 则 $A' = \{d, e, f, g\}$ 。

例 2.1.6 设有 3 种编程语言 Pascal、C++ 和 VB 供 2004 级软件高职 1 班的学生选修,每个学生最多可以选择其中的两种语言。已知 A 是只选修了 Pascal 语言的学生集合, B 是选修 C++ 语言的学生集合, C 是同时选修 VB 语言和 C++ 语言的学生集合,试写出选修 VB 语言的学生集合。

解 设 D 为选修 VB 语言的学生集合, E 为只选修 VB 语言的学生集合。因为 A 是只选修了 Pascal 语言的学生集合,因此, A 的补集 A' 就是所有选修 C++ 语言或者选修 VB 语言的学生集合。因为, B 是选修 C++ 语言的学生集合, C 是同时选修 VB 语言和 C++ 语言的学生集合,所以 $B \setminus C$ 就是选修 C++ 语言且不选 VB 语言的学生集合。于是, $A' \setminus (B \setminus C)$ 为 A' 中除去只选修 C++ 语言的学生后剩余的学生集合,也就是选修 VB 语言的学生集合,即

$$D = A' \setminus (B \setminus C)$$

集合的运算满足许多定律,这里选择其中最常用的:

- (1) 交换律: $A \cup B = B \cup A$, $A \cap B = B \cap A$ 。
- (2) 结合律: $(A \cup B) \cup C = A \cup (B \cup C)$, $(A \cap B) \cap C = A \cap (B \cap C)$ 。
- (3) 等幂律: $A \cup A = A$, $A \cap A = A$ 。
- (4) 分配律: $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$, $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$ 。
- (5) 德·摩根律(关于补集): $(A \cup B)' = A' \cap B'$, $(A \cap B)' = A' \cup B'$ 。
- (6) 吸收律: $A \cup (A \cap B) = A$, $A \cap (A \cup B) = A$ 。

2.1.5 二元关系

“关系”一词大家是十分熟悉的,在日常生活中,经常会谈到诸如父母与子女之间的关系,教师与学生之间的关系,朋友之间的关系,学生与选修课目之间的关系,客户与购买的货物之间的关系,数与数之间的大小关系,集合之间的包含关系等。因此,关系是一个应用面很广的概念。前面说的这些关系都是一些具体对象之间的关系,而且,即便是同样的对象,还可以有各种不同含义的关系,例如,除了集合之间的包含关系外,还有集合之间的相等关系、互补关系等。因此,有必要对“关系”的概念和性质进行研究。在这一小节中,主要介绍基于集合的二元关系的概念。二元关系以及以此为基础的关系代数是关系型数据库的基础,在数据库系统和计算机管理信息系统中有着重要的应用价值。