

原书第6版

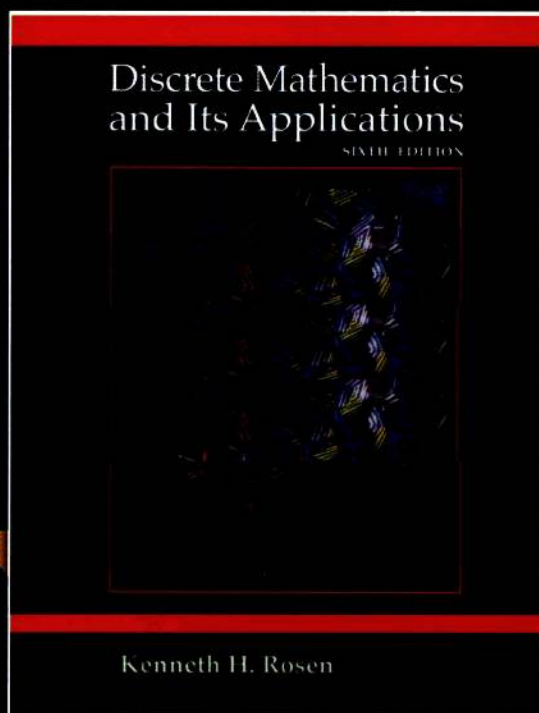
Mc
Graw
Hill Education

太科教学版

离散数学及其应用

(美) Kenneth H. Rosen 著 袁崇义 屈婉玲 张桂芸 等译 陈琼 改编

Discrete Mathematics and Its Applications
Sixth Edition



离散数学及其应用 (原书第6版·本科教学版)

Discrete Mathematics and Its Applications Sixth Edition

《离散数学及其应用》一书是介绍离散数学理论和方法的经典教材,已经成为采用率最高的离散数学教材,仅在美国就被600多所高校用作教材,并获得了极大的成功。第6版在前5版的基础上做了大量的改进,使其成为更有效的教学工具。

本书基于该书第6版进行改编,保留了国内离散数学课程涉及的基本内容,更加适合作为国内高校计算机及相关专业本科生的离散数学课程教材。

本书的具体改编情况如下:

- 第1章中补充了关于范式和标准型的基础内容
- 删去了在其他课程中讲授的内容,如数论、离散概率、归纳和递归等
- 对于保留章节,删去了编号为偶数的练习题
- 删去了相关的历史资料

作者简介:

Kenneth H. Rosen 密歇根大学数学学士,麻省理工学院数学博士。曾就职于科罗拉多大学、俄亥俄州立大学、缅因大学,后加盟贝尔实验室,现为AT&T实验室特别成员。除本书外,他还著有《初等数论及其应用》等书,并担任CRC离散数学丛书的主编。



中文版
书号: 978-7-111-23935-2
定价: 89.00元



英文精编版
书号: 978-7-111-31329-8
定价: 55.00元



英文版
书号: 978-7-111-35950-0
定价: 99.00元

客服热线: (010) 88378991, 88361066
购书热线: (010) 68326294, 88379649, 68995259
投稿热线: (010) 88379604
读者信箱: hzjsj@hzbook.com



Mc
Graw
Hill Education

www.mheducation.com

全国网站: <http://www.hzbook.com>

网上购书: www.china-pub.com

印刷设计: 陈明

上架指南: 计算机·离散数学

ISBN 978-7-111-35950-0



9 787111 359500

定价: 49.00元

计 算 机 科 学 丛 书

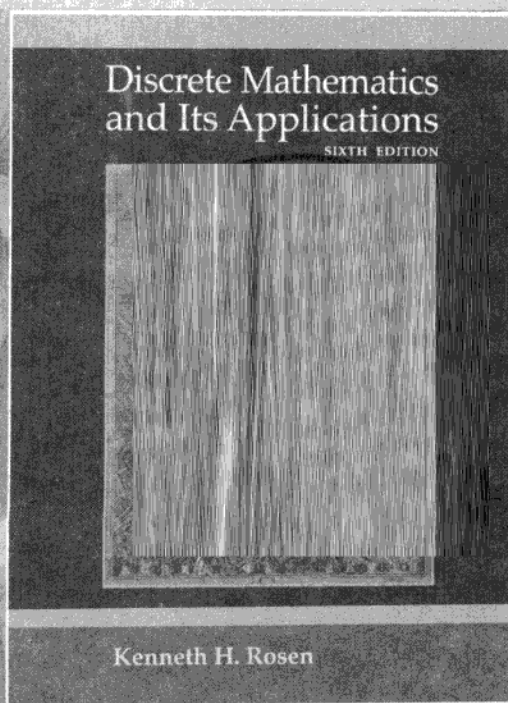
原书第6版



离散数学及其应用

(美) Kenneth H. Rosen 著 袁崇义 屈婉玲 张桂芸 等译 陈琼 改编

Discrete Mathematics and Its Applications
Sixth Edition



机械工业出版社
China Machine Press

本书是经典的离散数学教材,为全球多所大学广为采用。本书全面而系统地介绍了离散数学的理论和方法,除包括定义、定理的严密陈述外,还配备大量的实例和图表的说明、各种练习和题目。第6版在前五版的基础上做了大量的改进,使其成为更有效的教学工具。

本书可作为高等院校数学、计算机科学和计算机工程等专业的教材或参考书。

Kenneth H. Rosen; Discrete Mathematics and Its Applications, Sixth Edition (ISBN 0-07-322972-5).

Copyright © 2007 by Kenneth H. Rosen.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese adaptation is jointly published by McGraw-Hill Education (Asia) and China Machine Press. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong SAR, Macao SAR and Taiwan.

Copyright © 2011 by McGraw-Hill Education (Asia), a division of the Singapore Branch of The McGraw-Hill Companies, Inc. and China Machine Press.

版权所有。未经出版人事先书面许可,对本出版物的任何部分不得以任何方式或途径复制或传播,包括但不限于复印、录制、录音,或通过任何数据库、信息或可检索的系统。

本授权中文简体字改编版由麦格劳-希尔(亚洲)教育出版公司和机械工业出版社合作出版。此版本经授权仅限在中华人民共和国境内(不包括香港特别行政区、澳门特别行政区和台湾)销售。

版权 © 2011 由麦格劳-希尔(亚洲)教育出版公司与机械工业出版社所有。

本书封面贴有 McGraw-Hill 公司防伪标签,无标签者不得销售。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2010-3825

图书在版编目(CIP)数据

离散数学及其应用(原书第6版·本科教学版)/(美)罗森(Rosen, K. H.)著;袁崇义等译. —北京:机械工业出版社,2011.11

(计算机科学丛书)

书名原文:Discrete Mathematics and Its Applications, Sixth Edition

ISBN 978-7-111-35950-0

I. 离… II. ①罗… ②袁… III. 离散数学—高等学校—教材 IV. O158

中国版本图书馆CIP数据核字(2011)第192847号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:迟振春

北京荣盛彩色印刷有限公司印刷

2011年11月第1版第1次印刷

185mm×260mm·23.5印张

标准书号:ISBN 978-7-111-35950-0

定价:49.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010)88378991;88361066

购书热线:(010)68326294;88379649;68995259

投稿热线:(010)88379604

读者信箱:hjsj@hzbook.com

文艺复兴以降,源远流长的科学精神和逐步形成的学术规范,使西方国家在自然科学的各个领域取得了垄断性的优势;也正是这样的传统,使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中,美国的产业界与教育界越来越紧密地结合,计算机学科中的许多泰山北斗同时身处科研和教学的最前线,由此而产生的经典科学著作,不仅擘划了研究的范畴,还揭示了学术的源变,既遵循学术规范,又自有学者个性,其价值并不会因年月的流逝而减退。

近年,在全球信息化大潮的推动下,我国的计算机产业发展迅猛,对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇,也是挑战;而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下,美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此,引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用,也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始,我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力,我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系,从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品,以“计算机科学丛书”为总称出版,供读者学习、研究及珍藏。大理石纹理的封面,也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助,国内的专家不仅提供了中肯的选题指导,还不辞劳苦地担任了翻译和审校的工作;而原书的作者也相当关注其作品在中国的传播,有的还专程为其书的中译本作序。迄今,“计算机科学丛书”已经出版了近两百个品种,这些书籍在读者中树立了良好的口碑,并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑,这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化,教育界对国外计算机教材的需求和应用都将步入一个新的阶段,我们的目标是尽善尽美,而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方式如下:

华章网站: www.hzbook.com

电子邮件: hzsj@hzbook.com

联系电话: (010)88379604

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037



华章科技图书出版中心

改编者序

Discrete Mathematics and Its Applications, 6E

原版《离散数学及其应用》是一本优秀的教材，由 Kenneth H. Rosen 编著，已被全球超过 600 所大学使用。《离散数学及其应用》介绍了离散数学的主要内容以及离散数学在现实世界中的应用，书中内容全面，一些知识点给出了相关的历史背景知识，并配有大量的应用实例、参考资料和练习题，对学习该课程提供了极大的帮助。

近年来，国内的许多高校都在开展双语教学，并且采用国外的优秀原版书作为教材。而《离散数学及其应用》原版书超过 800 页，使得国内高校难以采用。为了将这本优秀教材介绍给更多的中国学生，在保留原作者写作风格的前提下，我们删减了部分章节内容编成此精编版，以适合中国学生的使用和阅读。精编版符合本科离散数学教学大纲的要求，可用于离散数学教学和课外学习。

原书中的一些内容，如数论、离散概率、归纳和递归、布尔代数以及有限状态机等，在其他课程中已有讲授或单独作为一门课程讲授，我们在精编版中将这些内容删除，保留了逻辑和证明、集合、函数和关系、图、树、计数和高级计数技术等内容。

原书有 3800 多道各种类型的练习题，其中包括帮助学生掌握基本概念的习题、提高应用所学知识分析和解决问题能力的习题等，分较为容易、难度适中和难度较大各种类型。为了保持原书的特点，我们保留了原书中奇数练习题，这样可以使得原书的各种类型、不同难度的习题都得以保留。关于一些知识点的相关历史背景知识未作保留，以减轻学生的阅读负担，减少本书的篇幅。

原书把一些概念放在练习题部分介绍，有些概念很重要，如数理逻辑中的范式，而原书对范式的描述较为简单，学生较难理解，所以我们将范式的相关内容补充到精编版的第 1 章中。

感谢我的同事杨捷在本书改编过程中提出的宝贵建议。精编版力求符合本科离散数学教学大纲的要求并将原书的精华体现出来，但难免存在不足之处，欢迎广大教师和学生指正。

陈琼

2011 年 9 月于华南理工大学

离散数学被 IEEE&ACM 确定为计算机专业核心课程,也是《中国计算机科学与技术学科教程 2002》中界定的计算机科学与技术专业的核心基础课程。离散数学为许多专业课程提供理论基础,尤其是大多数算法的基础,可以说,不理解算法背后的数学含义,就不能理解算法的本质,而这种表面的一知半解很可能会给你带来很大的麻烦。本书的写作目标就是向读者展示离散数学的实用性:为计算机专业学生提供一切必要的数学基础,使数学专业学生理解数学概念的重要性以及这些概念为什么对应用而言是重要的。

本书译自 Kenneth H. Rosen 所著的《Discrete Mathematics and Its Applications, Sixth Edition》,是介绍离散数学理论和方法的经典教材,已经成为世界上使用率最高的离散数学教材之一,目前已用五种语言出版,发行 30 多万册,仅在美国就被 600 多所高校用作教材,并获得了极大的成功。国内很多院校也选择这本书作为教材。

这本书是计算机理论中一本很好的入门教材,几乎涉及计算机理论的所有方面。该书有如下几个特点:

1) 通俗易懂,深入浅出。问题的引入、概念的描述和定理的阐述等通俗而不失科学性。

2) 书中例子很多,而且非常实用,还包含了世界名题和一些趣味性很强的问题。

3) 信息量虽然很大,但安排灵活,读者可以根据自己的兴趣选择性地学习。

4) 涉及面特别广,可以使读者大致了解计算机理论的方方面面,有助于读者理解计算机科学与技术中蕴含的数学思想、数学思维,培养读者对问题的数学建模能力,激发读者学习其他课程的兴趣。

5) 习题相当丰富,形容为“触目惊心”也不为过。题量大、题型多,有基础的也有探究的,有理论的也有实践的,正像作者在前言中强调的那样,他相当重视练习。

整本书阅读顺畅,行文活泼,作为参考书查阅也很方便。但书中多数章节还仅仅是相关的基础知识,也就是说,对这本书中的每一个知识点,比如逻辑、函数、组合数学、概率、算法等,都可以进行延伸阅读。有兴趣的话,读者不妨顺着书后的推荐阅读列表,继续深入下去。最后建议大家花些时间仔细阅读前言,这对你用好本书、更好地学习离散数学等非常有帮助。

本书第 5 版主要由袁崇义、屈婉玲、王捍贫和刘田翻译完成,而第 6 版的翻译工作是在第 5 版译稿的基础上进行的,主要由张桂芸、赵子平、裴伟东、徐计和王向云翻译完成,张桂芸做了全书的统稿和定稿工作。由于翻译水平和时间有限,书中难免有不妥的地方,敬请读者不吝赐教。

译者

2011 年 5 月于天津

本书是根据我多年讲授离散数学的经验和兴趣写成的。对学生而言,我的目的是为他们提供准确而可读的教材,清晰地介绍离散数学的概念和技术。我的目标是向爱怀疑的学生们展示离散数学的实用性,希望为学习计算机科学的学生提供一切必需的数学基础,也希望学数学的学生理解数学概念的重要性,以及这些概念为什么对应用是重要的,还希望本书既能达到这些目标,又不含太多的水分。

对教师(指导者)而言,我的目的是使用成熟的数学教学技术设计一个灵活而全面的教学工具,希望为教师们提供能够以最适合特定学生特点的方式高效地讲授离散数学的教材。希望本书能够达到这些目标。

我为本教材在过去已经取得的巨大成功而分外高兴。根据成功使用本书的600多所学校的大批师生的反馈和建议,此次第6版进行了许多改进。很多内容有所提高,辅助材料更加丰富,配套网站提供的材料更有帮助性,使师生更容易达到他们的目标。

本教材是为1至2个学期的离散数学入门课程而设计的,适用于数学、计算机科学、工程等许多专业的学生。虽然唯一的先修课要求是大学代数,但是要想学习好离散数学还需要掌握更多的数学知识。

离散数学课的目标

离散数学课有多个目标。学生不仅要学会一些特定的数学知识并知道怎样应用,更重要的是,这样一门课应教会学生怎样进行数学逻辑思维。为此,本教材特别强调数学推理及用不同的方法解题。本教材有5个重要的主题交织在一起:数学推理、组合分析、离散结构、算法思维、应用与建模。成功的离散数学课程应该努力使这五部分内容相互融合、平衡。

1. **数学推理**: 学生必须理解数学推理,以便阅读、理解和构造数学证明。本教材以数理逻辑开篇,在后面证明方法的讨论中,数理逻辑是基础。同时,本书描述了构造性证明的方法和技巧。本书还十分强调数学归纳法,不仅用许多证明的实例进行介绍,还详细地解释了数学归纳法为什么是有效的证明方法。

2. **组合分析**: 解题的一项重要技巧是计数或枚举对象。本书中,对枚举的讨论从基本的计数方法着手,重点是用组合分析方法来解决计数问题,而不直接使用公式。

3. **离散结构**: 离散数学课应该教会学生如何使用离散结构,即学会如何使用表示离散对象及其之间的关系的抽象数学结构。离散结构包括:集合、置换、关系、图、树和有限状态机等。

4. **算法思维**: 有些问题是通过详细说明其算法来求解的。算法在描述后就可构造计算机程序来实现。这一过程中用到的数学部分包括:算法描述、正确性证明以及执行算法所需要的计算机内存和时间的分析。这些内容在本书中均有介绍。算法是用英语[⊖]和一种易于理解的伪码描述的。

5. **应用与建模**: 离散数学几乎应用在所有研究领域。本书既介绍了其在计算机科学和数据网络中的许多应用,也介绍了在其他各种领域中的应用,如化学、植物学、动物学、语言学、地理、商业以及互联网等。这些均是离散数学的实际而重要的应用,而不是编造的。用离散数学来建模是十分重要的问题求解技巧。本书中的一些练习让学生有机会通过自己构造模型来掌握这一技巧。

⊖ 译著中采用汉语。——译者注

本书特点

易入门 实践证明：此课本对初学者来说易读易懂。它的大部分内容只要求学生学过大学代数，不需要其他的预备知识，少数几个涉及微积分的地方也有明确说明。大部分学生应该很容易理解课本中用于表示算法的伪码，不管他们是否学过程序设计语言。本书不要求正规计算机科学方面的预备知识。

每章都是从易理解的水平开始。本书在仔细研究基本的数学概念之后，就介绍了其他领域中更难的部分和应用。

灵活性 课本为灵活使用作了精心设计。各章对其前面内容的依赖都降到最低程度。每一章分成长度大致相等的若干节，每一节又根据内容划分成小节以方便教学。教师可以根据这些分块灵活地安排进度。

写作风格 本书的写作风格是直接和实用。使用准确的数学语言，但没有过分的形式化与过分的抽象。在记号和数学命题的词汇间作了精心的平衡。

数学严谨性和准确性 本书中所有定义和定理的陈述都十分详细，所以学生可以欣赏其语言的准确以及数学所需的严谨。但证明则是缓慢引入并展开的，每一步都经过了详细论证。本书解释并大量使用了递归定义。

实例 本书使用了很多例子来阐述概念、建立不同内容之间的关系或导入应用。在大部分例子中，先提出问题，再适当给出它的解。

应用 书中叙述的应用展示了离散数学在解决现实问题中的使用价值，所涉及的范围很广，包括计算机科学、数据网络、心理学、化学、工程、语言学、生物学、商业和互联网。

算法 离散数学的结论常常要用算法来表示，因此，本书每一章都介绍了一些关键算法。书中还简要分析了所有算法的计算复杂性。

关键术语和结果 每一章后面都列出了本章的关键术语和结果，但只列出了学生必须学会的那些最重要的关键术语，而不是在该章中定义的所有术语。

练习 书中包含许多练习，代表了大量不同类型的问题。本书不仅提供了足够多的简单问题用于练习基本技巧，还提供了大量的中等难度的练习和许多有挑战性的练习。练习的叙述清晰而无歧义，并按难易程度进行了分级。有些特殊的论题还设计成系列练习。这些练习不仅给出了正文中没有介绍的新概念，还使学生可以通过自己的努力发现新思想。

比平均水平稍难的练习用单个星号作标记，相当有挑战性的问题则用两个星号来标记。必须用微积分来解的练习也作了明确说明。导出正文要用到的结果的练习则用符号“ $\square$ ”识别。

复习题 每章最后都有一组复习题。设计这些问题的目的是帮助学生重点学习该章最重要的概念和技术。要回答这些复习题，学生必须写出较长的答案，而只做点计算或简答是不够的。

计算机题目 每一章后面还有一组计算机题目。这些题目把学生可能已经学到的有关计算和离散数学的内容联系起来。如果从数学角度或程序设计角度来看，其难度超过平均水平的计算机题目用一个星号标记，特别有挑战性的则用两个星号标记。

计算和研究 每一章的结论部分都有一组计算和研究性问题。要完成这些练习(大约有 100 道)需要软件工具的帮助，例如学生或教师自己编写的程序，或数学计算软件包(如 Maple 或 Mathematica)。大部分这样的练习为学生提供了通过计算发现新事实或新思想的机会。

写作题目 每一章后面都有一组应该书面完成的题目。要完成这类题目，学生需要参考数学文献。有些题目在过去的历史上是很重要的，学生需要查找原始资料，其他的题目则是通往新内容和新思想的途径。所有此类练习均向学生展示正文中没有深入探讨的思想。这些题目把数学概念和书面写作的过程结合在一起，以帮助学生探索未来的研究领域。

推荐读物 在正文的最后，专门有一节用来为各章推荐参考读物，其中有不超出本书水平的书籍，

也有较难的书籍；有综述性文章，也有发表离散数学新发现的原始文章。其中一些是许多年前出版的经典读物，而其他一些是在最近几年内出版的。

配套网站

为本书开发了一个内容广泛的配套网站，该网站将不断地维护和改进。可以以多种方式使用该网站进一步学习离散数学，网站地址是：

www.mhhe.com/rosen

由此地址进入该网站的主页。该网页有下列链接：

- 信息中心
- 学生中心
- 教师中心

信息中心：该中心包含本书及其辅助资料的基本信息。教师在这个网页中可以利用网页抽取系统“Page Out”，建立自己的课程网页，还能学习定制出版信息。从该信息中心出发，沿着适当的链接，可以看到本书网站的一个概观。

学生中心：该中心包含了可供学生使用的丰富资源，包括下列与课本紧密相关的资源（在课本中用相应的图标加以标记）：



- **网络资源指南：**该指南提供了数百个含相关资料的外部网站的链接。可以通过关键词浏览或访问这些链接，它们将把你带到包含下列信息的网站：历史及传记信息、谜题及问题、讨论、Java 小程序、程序以及其他类型的资源。



- **额外的例子：**这个网站包含了大量额外的例子。这些例子主要集中在学生经常需要的课外资料的领域。虽然大部分例子只是扩充了基本概念，但还有一部分十分具有挑战性。



- **额外的步骤：**对一些困难的知识点，提供了更深入的解释帮助理解，尤其是一些特殊的证明和例子。



- **评估：**提供对七个关键概念理解程度的评估。每个评估都提供了一个题库，其中的每个试题由两部分构成：先是一段简短的复习，然后是一个多选题。如果你选择的答案不正确，该系统还能提供建议，帮助你理解错在什么地方。这样的评估系统应该能诊断出你学习中的问题，从而把精力集中在寻找纠正办法上。



- **交互式演示：**已经开发了八个交互式演示系统，你可以用它们考查一些重要算法是怎么工作的。这些演示都与课本中的相应材料相对应。

从该学生中心你可以访问网络教学系统“Net Tutor”。它提供了在线教学帮助。当你提问与课本内容相关的问题时，如果是在规定教学时间内，你会收到实时回答；否则稍后才会收到回答。

学生中心还支持能发布消息的公告板系统。使用该系统，你可以提出问题，还可以回答其他学生提出的问题。

除此之外，学生中心还包括下列资源：

- 证明书写指南
- 离散数学的常见错误
- 写作题目的建议
- Maple 软件

教师中心：除了包含学生中心和信息中心提供的所有链接外，网站中的教师中心还包含下列链接：

- 教学大纲样本
- 教学建议
- 《Applications of Discrete Mathematics》一书中的某些章节

写给学生

什么是离散数学？离散数学是数学中研究离散对象的部分。（这里“离散”的含义是“由不同的或不相连的元素组成”。）离散数学能解决的问题包括：

- 在计算机系统中，有多少种方式可以选择一个合法口令？
- 赢彩票的概率是多少？
- 两台计算机之间在网络上是否有通路？
- 怎样鉴别 E-mail 信息中的垃圾邮件？
- 怎样加密信息以避免不该收到的人读取该信息？
- 在某一交通系统下，两个城市之间的最短路径是什么？
- 怎样把整数序列按递增序排列？
- 完成上述排序需要多少步骤？
- 如何证明一个排序方法能正确地排序？
- 怎样设计两个整数相加的电路？
- 有多少合法的因特网网址？

你们将学习解决诸如以上问题要用到的离散结构和技术。

更一般地，在对对象进行计数时要用到离散数学，研究两个有限(或可数)集合之间的关系时要用到离散数学，分析只含有限步的进程时也要用到离散数学。离散数学的重要性还在不断增加，一个关键原因就是计算机以离散的方式存储和处理信息。

为什么要学离散数学？有几条重要的理由来说明需要学习离散数学。首先，通过这个课程你们可以发展自己的数学素质，即理解和创造数学证明的能力。没有这些技巧，你们在学习数学时不可能太深入。

其次，离散数学是学习所有更高级数学课程的必经之路。离散数学为学习计算机科学课程提供必要的数学基础，这些课程包括：数据结构、算法、数据库理论、自动机理论、形式语言、编译理论、计算机安全以及操作系统。学生如果没有适当的离散数学基础，在学习上述课程时会感到很困难。有个学生给我发电子邮件说，在她学习的每一门计算机科学的课中都用到了本书的知识。

以离散数学为基础的数学课程包括逻辑、集合论、数论、线性代数、抽象代数、组合论、图论及概率论(其离散部分)。

此外，离散数学还包括了解运筹学(包括许多离散优化技术)、化学、工程及生物学等领域问题的必要的数学背景。从本书中你将学到在上述某些领域中的应用。

许多学生都感到，与他们以前学过的课程相比，离散数学入门课程的挑战性要大得多。这是因为，本课程的一个主要目的是教你进行数学推理和问题求解，而不只是一些分散的技巧。从课本中练习的设计可以看出这个目的。课本中虽然有大量与重点例子类似的练习，但还是有相当比例的练习需要创造性思维。这是有意设计的。虽然课本中的材料提供了解这些问题的工具，但你的任务是创造性地使用这些工具并取得成功。本课程的另一个主要目的是学会处理你以前没有见过的问题。然而，只学会解特殊类型的练习还无法保证能学会足够多的解题技巧，也不能保证在后继课程的学习中或在将来的职业生涯中取得成功。虽然课本讨论了许多主题，但离散数学是一个极为广泛且充满变化的研究领域。作为作者，我的任务之一是帮助你开发学习新知识的能力，在将来的奋斗中你十分需要新的知识。

练习 我愿意就如何学好离散数学(或数学的其他学科和计算机科学)给同学们提点有益的建议。积极地做练习能使你最大地获益。我建议你尽可能地多做练习。在做完老师布置的练习后，我鼓励你做更多的练习，包括本书每节后面的练习和每章后面的补充练习。(注意练习前面的分级标记。)

练习标记	含 义
无标记	常规练习
*	较难的练习
**	富有挑战性的练习
□	正文中要用到该练习的某个结论
(需要用到微积分)	解题时要用到极限或微积分的概念

解题时，最好在查阅答案以前，自己先进行尝试。华章网站(www.hzbook.com)提供了练习的答案。注意，只是答案，而不是完整解答。特别地，答案中省略了解的推导过程。

网络资源 我强烈推荐你利用网络提供的新资源，特别是专为本书设计的网站 <http://www.mhhe.com/rosen> 中的资源。其中为澄清关键的概念设计了许多额外的例子，自我评测是考查你对关键问题的掌握程度；交互演示小程序帮你探索关键算法和其他概念；网络资源向导包含了广泛的与离散数学有关的外部链接；额外的解释和实践会帮助你掌握关键的概念，增加的书写证明可以避免离散数学中的一般错误；重要应用的深入探索指导应用 Maple 软件探索离散数学中的计算问题。在文中这些额外的在线资源尽可能在边缘处用特定图标标识。

本书的价值 我希望你对本书的投资能得到优质的回报。我们花了多年的努力来开发和优化本书及其配套的辅助读物和网站。对大多数人来说，我很自信本书及其配套资料对掌握离散数学大有帮助。即使你现在的课程可能略去了其中某些章节，但当你学习新课程时会发现，阅读本书中的相关章节对新课程仍然十分有益，许多学生都有这样的感觉。许多人会在将来的研究工作时又找到本书，把它当做一本有用的工具书，特别是那些继续学习计算机科学、数学或工程的人。我设计本书作为你将来研究和探索的入门，我希望你能幸运地开始你的征程。

致谢

我要感谢很多学校中使用本书的大批师生们，他们向我提出了宝贵的反馈和有益的建议。没有他们的反馈和建议，本书不可能有很大的改进。我要特别感谢 Jerrold Grossman、John Michaels 和 George Bergman，感谢他们对第 6 版的技术审阅，他们敏锐的目光确保了本书的准确性。我也要感谢通过网站提交评论的人们提供的帮助。

我感谢本书第 6 版及前面 5 版的许多评阅人，他们对我提出了许多有益的批评和鼓励，我希望本版不辜负他们的期望。

第 6 版评阅人名单

Charles Ashbacher,
Mount Mercy College

Ian Barland,
Rice University

George Bergman,
University of California, Berkeley

David Berman,
University of North Carolina,
Wilmington

Miklós Bóna,
University of Florida

Prosenjit Bose,
Carleton University

Kirby Brown,
Polytechnic University

Michael Button,
The Master's College

Greg Cameron,
Brigham Young University

John Carter,
University of Toronto

Greg Chapman,
Cosumnes River College

Chao-Kun Cheng,
Virginia Commonwealth University

Thomas Cooper,
Georgia Perimeter College,
Lawrenceville

Barbara Cortzen,
DePaul University

Daniel Cunningham,
Buffalo State College

George Davis,
University of Georgia

Beverly Diamond,
College of Charleston

Thomas Dunion,
Atlantic Union College

Bruce Elenbogen,
University of Michigan, Dearborn

Herbert Enderton,
University of California,
Los Angeles

Anthony Evans,
Wright State University

Kim Factor,
Marquette University

William Farmer,
McMaster University

Li Feng,
Albany State University

- Stephanie Fitchett,
Florida Atlantic University
- Jerry Fluharty,
Eastern Shore Community College
- Joel Fowler,
Southern Polytechnic State University
- William Gasarch,
University of Maryland
- Sudipto Ghosh,
Colorado State University
- Jerrold Grossman,
Oakland University
- Ruth Haas,
Smith College
- Patricia Hammer,
Hollins University
- Keith Harrow,
Brooklyn College
- Bert Hartnell,
St. Mary's University
- Julia Hassett,
Oakton College
- Kavita Hatwal,
Portland Community College
- John Helm,
Radford University
- Arthur Hobbs,
Texas A&M University
- Fannie Howell,
Roanoke High School
- Wei Hu,
Houghton College
- Nan Jiang,
University of South Dakota
- Margaret Johnson,
Stanford University
- Martin Jones,
College of Charleston
- Tim Kearns,
California Polytechnic State University, San Luis Obispo
- Jonathan Knappenberger,
LaSalle University
- Ed Korntved,
Northwest Nazarene University
- Przemo Kranz,
University of Mississippi
- Loredana Lanzani,
University of Arkansas
- Yoonjin Lee,
Smith College
- Miguel Lerma,
Northwestern University
- Jason Levy,
University of Hawaii
- Lauren Lilly,
Tabor College
- Ekaterina Lioutikova,
Saint Joseph College
- Vladimir Logvinenko,
De Anza College
- Joan Lukas,
University of Massachusetts, Boston
- Lester McCann,
University of Arizona
- Jennifer McNulty,
University of Montana
- John G. Michaels,
SUNY Brockport
- Michael Oppedisano,
Morrisville State College
- Michael O'Sullivan,
San Diego State University
- Charles Parry,
Virginia Polytechnic Institute
- Linda Powers,
Virginia Polytechnic Institute
- Dan Pritikin,
Miami University
- Anthony Quas,
University of Memphis
- Eric Rawdon,
Duquesne University
- Henry Ricardo,
Medgar Evers College
- Oskars Rieksts,
Kutztown University
- Stefan Robila,
Montclair State University
- Chris Rodger,
Auburn University
- Robert Rodman,
North Carolina State University
- Shai Simonson,
Stonehill College
- Barbara Smith,
Cochise College
- Wasin So,
San Jose State University
- Diana Staats,
Dutchess Community College
- Lorna Stewart,
University of Alberta
- Bogdan Suceava,
California State University, Fullerton
- Kathleen Sullivan,
Seattle University
- Laszlo Szekely,
University of South Carolina
- Daniel Tauritz,
University of Missouri, Rolla
- Don VanderJagt,
Grand Valley State University
- Fran Vasko,
Kutztown University
- Susan Wallace,
University of North Florida
- Zhenyuan Wang,
University of Nebraska, Omaha
- Tom Wineinger,
University of Wisconsin, Eau Claire
- Charlotte Young,
South Plains College
- 前 5 版评阅人名单
- Eric Allender,
Rutgers University
- Stephen Andrilli,
La Salle University
- Kendall Atkinson,
University of Iowa, Iowa City
- Scott Buffett,
University of New Brunswick
- Russell Campbell,
University of Northern Iowa
- Zhaojun Bai,
University of California, Davis
- Jack R. Barone,
Baruch College
- Klaus Bichteler,
University of Texas, Austin
- David Jonah,
Wayne State University
- Akihiro Kanamori,
Boston University
- Alfred E. Borm,
Southwest Texas State University
- Ken W. Bosworth,
University of Maryland
- Lois Brady,
California Polytechnic State University, San Luis Obispo
- Ho Kuen Ng,
San Jose State University
- Timothy S. Norfolk, Sr.
University of Akron

目 录

Discrete Mathematics and Its Applications, 6E

出版者的话

改编者序

译者序

前言

第 1 章 基础：逻辑和证明 1

1.1 命题逻辑 1

1.1.1 引言 1

1.1.2 命题 1

1.1.3 条件语句 3

1.1.4 复合命题的真值表 6

1.1.5 逻辑运算符的优先级 7

1.1.6 翻译语句 7

1.1.7 系统规范说明 8

1.1.8 布尔检索 8

1.1.9 逻辑难题 9

1.1.10 逻辑运算和位运算 9

练习 10

1.2 命题等价 13

1.2.1 引言 13

1.2.2 逻辑等价 14

1.2.3 德摩根律的运用 16

1.2.4 构建新的逻辑等价式 16

练习 18

1.3 谓词和量词 20

1.3.1 引言 20

1.3.2 谓词 20

1.3.3 量词 21

1.3.4 其他量词 24

1.3.5 约束论域量词 24

1.3.6 量词的优先级 24

1.3.7 绑定变量 24

1.3.8 涉及量词的逻辑等价 24

1.3.9 否定量化表达式 25

1.3.10 翻译语句为逻辑表达式 26

1.3.11 在系统说明中运用量词 28

1.3.12 选自 Lewis Carroll 的例子 28

1.3.13 逻辑程序设计 29

练习 30

1.4 嵌套量词 32

1.4.1 引言 32

1.4.2 量词的顺序 33

1.4.3 将数学语句翻译成涉及嵌套
量词的语句 34

1.4.4 将嵌套量词翻译为汉语 35

1.4.5 将汉语语句翻译成逻辑
表达式 35

1.4.6 否定嵌套量词 36

练习 36

1.5 推理规则 39

1.5.1 引言 39

1.5.2 命题逻辑的有效论证 39

1.5.3 命题逻辑的推理规则 40

1.5.4 用推理规则建立论证 42

1.5.5 消解 43

1.5.6 谬误 43

1.5.7 带量词命题的推理规则 44

1.5.8 命题推理和量化语句推理规则
的结合 45

练习 45

1.6 证明导论 47

1.6.1 引言 47

1.6.2 一些专用术语 47

1.6.3 定理陈述的理解 47

1.6.4 证明定理的方法 48

1.6.5 直接证明 48

1.6.6 反证法 49

1.6.7 归谬证明	50	2.3.2 一对一函数和映上函数	85
1.6.8 证明中的错误	52	2.3.3 反函数和函数组合	86
1.6.9 仅仅是开始	53	2.3.4 函数的图像	88
练习	53	2.3.5 几个重要的函数	89
1.7 证明的方法和策略	54	练习	91
1.7.1 引言	54	2.4 序列与求和	93
1.7.2 穷举证明和分情形证明	54	2.4.1 引言	93
1.7.3 存在性证明	57	2.4.2 序列	93
1.7.4 唯一性证明	58	2.4.3 特殊的整数序列	94
1.7.5 证明策略	58	2.4.4 求和	96
1.7.6 寻找反例	60	2.4.5 基数	98
1.7.7 行动证明策略	60	练习	100
1.7.8 填充	60	关键术语和结果	101
1.7.9 未解决问题的作用	63	复习题	103
1.7.10 其他证明方法	63	补充练习	103
练习	64	计算机题目	105
关键术语和结果	65	计算和研究	105
复习题	66	写作题目	105
补充练习	67		
计算机题目	68	第3章 计数	106
计算和研究	69	3.1 计数基础	106
写作题目	69	3.1.1 引言	106
第2章 基本结构：集合、函数、 数列与求和	70	3.1.2 基本的计数原则	106
2.1 集合	70	3.1.3 比较复杂的计数问题	109
2.1.1 引言	70	3.1.4 容斥原理	110
2.1.2 幂集合	72	3.1.5 树图	111
2.1.3 笛卡儿积	73	练习	112
2.1.4 使用带量词的集合符号	74	3.2 鸽巢原理	114
2.1.5 量词的真值集合	74	3.2.1 引言	114
练习	74	3.2.2 广义鸽巢原理	115
2.2 集合运算	75	3.2.3 巧妙使用鸽巢原理	116
2.2.1 引言	75	练习	117
2.2.2 集合恒等式	77	3.3 排列与组合	118
2.2.3 扩展的并集和交集	79	3.3.1 引言	118
2.2.4 计算机表示集合的方式	80	3.3.2 排列	118
练习	81	3.3.3 组合	120
2.3 函数	83	练习	122
2.3.1 引言	83	3.4 二项式系数	123
		3.4.1 二项式定理	123
		3.4.2 帕斯卡恒等式和三角形	125

3.4.3 其他的二项式系数恒等式	126	4.4.1 引言	166
练习	127	4.4.2 关于幂级数的有用事实	167
3.5 排列与组合的推广	129	4.4.3 计数问题与生成函数	170
3.5.1 引言	129	4.4.4 使用生成函数求解递推关系	172
3.5.2 有重复的排列	129	4.4.5 使用生成函数证明恒等式	174
3.5.3 有重复的组合	129	练习	174
3.5.4 具有不可区别物体的集合的排列	132	4.5 容斥	176
3.5.5 把物体放入盒子	132	4.5.1 引言	176
练习	135	4.5.2 容斥原理	176
3.6 生成排列和组合	136	练习	179
3.6.1 引言	136	4.6 容斥原理的应用	180
3.6.2 生成排列	137	4.6.1 引言	180
3.6.3 生成组合	138	4.6.2 容斥原理的另一种形式	180
练习	139	4.6.3 埃拉托色尼筛	181
关键术语和结果	139	4.6.4 映上函数的个数	182
复习题	140	4.6.5 错位排列	183
补充练习	141	练习	184
计算机题目	144	关键术语和结果	185
计算和研究	144	复习题	186
写作题目	144	补充练习	186
第4章 高级计数技术	145	计算机题目	188
4.1 递推关系基础	145	计算和研究	189
4.1.1 引言	145	写作题目	189
4.1.2 递推关系	145	第5章 关系	190
4.1.3 用递推关系构造模型	146	5.1 关系及其性质	190
练习	150	5.1.1 引言	190
4.2 求解线性递推关系	152	5.1.2 函数作为关系	191
4.2.1 引言	152	5.1.3 集合的关系	191
4.2.2 求解常系数线性齐次递推关系	153	5.1.4 关系的性质	192
4.2.3 常系数线性非齐次的递推关系	156	5.1.5 关系的组合	194
练习	159	练习	195
4.3 分治算法和递推关系	160	5.2 n 元关系及其应用	197
4.3.1 引言	160	5.2.1 引言	197
4.3.2 分治递推关系	160	5.2.2 n 元关系	197
练习	165	5.2.3 数据库和关系	197
4.4 生成函数	166	5.2.4 n 元关系的运算	199
		5.2.5 SQL	200
		练习	201

5.3 关系的表示	202	6.2.4 偶图	247
5.3.1 引言	202	6.2.5 特殊类型的图的一些应用 ...	249
5.3.2 用矩阵表示关系	202	6.2.6 从旧图到新图	251
5.3.3 用图表示关系	204	练习	252
练习	206	6.3 图的表示和图的同构	253
5.4 关系的闭包	207	6.3.1 引言	253
5.4.1 引言	207	6.3.2 图的表示	253
5.4.2 闭包	207	6.3.3 邻接矩阵	254
5.4.3 有向图的路径	208	6.3.4 关联矩阵	256
5.4.4 传递闭包	209	6.3.5 图的同构	256
5.4.5 沃舍尔算法	211	练习	259
练习	213	6.4 连通性	261
5.5 等价关系基础	214	6.4.1 引言	261
5.5.1 引言	214	6.4.2 通路	261
5.5.2 等价关系	214	6.4.3 无向图的连通性	263
5.5.3 等价类	216	6.4.4 有向图的连通性	264
5.5.4 等价类与划分	217	6.4.5 通路与同构	265
练习	219	6.4.6 计算顶点之间的通路数	265
5.6 偏序	221	练习	266
5.6.1 引言	221	6.5 欧拉通路与哈密顿通路	268
5.6.2 字典顺序	222	6.5.1 引言	268
5.6.3 哈塞图	224	6.5.2 欧拉通路与欧拉回路	268
5.6.4 极大元素与极小元素	225	6.5.3 哈密顿通路与哈密顿回路 ...	272
5.6.5 格	226	练习	274
5.6.6 拓扑排序	227	6.6 最短通路问题	276
练习	229	6.6.1 引言	276
关键术语和结果	231	6.6.2 最短通路算法	278
复习题	233	6.6.3 旅行商问题	281
补充练习	234	练习	283
计算机题目	236	6.7 可平面图	284
计算和研究	237	6.7.1 引言	284
写作题目	237	6.7.2 欧拉公式	286
第6章 图	238	6.7.3 库拉图斯基定理	288
6.1 图和图模型	238	练习	289
练习	243	6.8 图着色	291
6.2 图的术语和几种特殊的图	244	6.8.1 引言	291
6.2.1 引言	244	6.8.2 图着色的应用	294
6.2.2 基本术语	244	练习	295
6.2.3 一些特殊的简单图	246	关键术语和结果	297
		复习题	298

补充练习	299	7.3.4 中缀、前缀和后缀记法	331
计算机题目	303	练习	333
计算和研究	303	7.4 生成树	334
写作题目	304	7.4.1 引言	334
第7章 树	305	7.4.2 深度优先搜索	337
7.1 概述	305	7.4.3 宽度优先搜索	338
7.1.1 树作为模型	309	7.4.4 回溯	340
7.1.2 树的性质	310	7.4.5 有向图中的深度优先搜索	341
练习	312	练习	342
7.2 树的应用	313	7.5 最小生成树	344
7.2.1 引言	313	7.5.1 引言	344
7.2.2 二叉搜索树	313	7.5.2 最小生成树算法	344
7.2.3 决策树	316	练习	347
7.2.4 前缀码	317	关键术语和结果	348
7.2.5 博弈树	320	复习题	349
练习	323	补充练习	350
7.3 树的遍历	324	计算机题目	353
7.3.1 引言	324	计算和研究	353
7.3.2 通用地址系统	325	写作题目	353
7.3.3 遍历算法	325		

练习题答案[⊖]

⊖ 见华章网站 www.hzbook.com。

基础：逻辑和证明

逻辑规则给出数学语句的准确含义。例如，逻辑规则有助于我们理解下列语句及其推理：“存在一个不是两个整数的平方和的整数”，以及“对每个正整数 n ，小于等于 n 的正整数之和是 $n(n+1)/2$ ”。逻辑是所有数学推理及所有自动推理的基础。对计算机的设计、系统规范说明、人工智能、计算机程序设计、程序设计语言以及计算机科学的其他许多领域，逻辑都有实际的应用。

为了理解数学，我们必须理解正确的数学论证（即证明）是由什么组成的。只要证明一个数学语句是真的，我们就称之为一个定理。围绕一个论题的定理的集合就将我们对这个论题的认知组织起来了。为了学习一个数学论题，我们需要积极地构造关于此论题的数学论证，而不仅仅是阅读说明。此外，因为一个定理的证明常常可能需要修改结论以适应新情境，所以证明在新思想的发展中扮演了极其重要的角色。计算机科学专业的学生常常发现，在计算机科学中证明的重要程度令人惊讶。事实上，当我们证实计算机程序对所有可能的输入值产生正确输出值的时候，当我们证明算法总是产生正确结果的时候，当我们建立系统安全，以及创造人工智能的时候，证明都扮演了极其重要的角色。自动推理系统已经被构造出来，它允许计算机构造自己的证明。

本章将解释一个正确的数学论证的组成，并介绍构造这些论证的工具。我们将发展一系列不同的证明方法以证明许多不同类型的结果。介绍完证明的多种不同方法后，我们将介绍一些构造证明的策略。我们将介绍猜想的概念，并解释用学习猜想来发展数学的过程。

1.1 命题逻辑

1.1.1 引言

逻辑规则给出数学语句的准确含义，这些规则用来区分有效和无效的数学论证。由于本书的一个主要目的是教会读者如何理解和如何构造正确的数学论证，所以我们从介绍逻辑开始离散数学的学习。

逻辑不仅对理解数学推理十分重要，而且在计算机科学中有许多应用。这些逻辑规则用于计算机电路设计、计算机程序构造、程序正确性证明以及许多其他方面。在随后的几章中将逐一讨论这些应用。

1.1.2 命题

我们首先介绍逻辑的基本成分——命题。命题是一个或真或假的陈述语句，即一个陈述事实的句子，但不能既真又假。



例 1 下面的陈述句均为命题。

1. 华盛顿是美国的首都。
2. 多伦多是加拿大的首都。
3. $1+1=2$ 。
4. $2+2=3$ 。

命题 1 和 3 为真，命题 2 和 4 为假。

例 2 给出了不是命题的若干语句。

例 2 考虑下述语句：

1. 几点了？
2. 仔细读这个。

3. $x+1=2$ 。

4. $x+y=z$ 。

语句 1 和 2 不是命题，因为它们不是陈述语句。语句 3 和 4 不是命题，因为它们既不为真，也不为假。注意，如果我们给语句 3 和 4 中的变量赋值，那么语句 3 和 4 可以变成命题。1.3 节将讨论把这一类语句改成命题的多种方法。

我们用字母来表示命题变元，代表命题的变量，就像用字母表示数字变量那样。习惯上用字母 $p, q, r, s, \dots$ 表示命题。如果一个命题是真命题，它的真值为真，用 T 表示；如果它是假命题，其真值为假，用 F 表示。

涉及命题的逻辑领域称为命题演算或命题逻辑。它最初是 2300 多年前由古希腊哲学家亚里士多德系统地创建的。

现在我们转而注意从那些已有的命题产生新命题的方法，这些方法在 1854 年曾由英国数学家布尔在他的题为 *The Laws of Thought* 的书中讨论过。许多数学陈述都是由一个或多个命题组合而来。称为复合命题的新命题由已有的命题用逻辑运算符组合而来。

定义 1 令 p 为一命题，则 p 的否定表示为 $\neg p$ (也可表示成 $\bar{p}$)，指“不是 p 所说的情形”。

命题 $\neg p$ 读作“非 p ”。 $\neg p$ 的真值和 p 的真值相反。

例 3 找出命题

“今天是星期五”

的否定，并用中文表示。

解 否定为

“并非今天是星期五”

也可以更简单地表达为

“今天不是星期五”

例 4 找出命题

“迈阿密今天至少下了 10 英寸的雨”

的否定并用中文表示。

解 否定为

“并非迈阿密今天至少下了 10 英寸的雨”

或者可以更简单地表达为

“迈阿密今天下了不足 10 英寸的雨”

注意 严格地说，像例 3 和例 4 这种含有可变时间的语句不是命题，除非假定了一个确定的时间。同样，除非假定了确定的地点，否则含有可变地点的语句不是命题；除非假定了确定的人，否则含有可变代词的语句不是命题。我们总是假定命题中有确定的时间、地点和特定的人，除非另有说明。

表 1-1 是命题 p 及其否定的真值表。此表列出命题 p 的两种可能真值。每一行显示的 $\neg p$ 的真值对应 p 的真值。

命题的否定也可以看做非运算符作用在命题上的结果。非运算符从一个已有的命题构造出一个新命题。现在将引入从两个或多个已有命题构造新命题的逻辑运算符，这些逻辑运算符也称为联接词。

定义 2 令 p 和 q 为命题。 p 、 q 的合取用 $p \wedge q$ 表示，即命题“ p 并且 q ”。当 p 和 q 都是真时， $p \wedge q$ 命题为真，否则为假。

表 1-2 是 $p \wedge q$ 的真值表。此表每一行有是 p 和 q 真值变化的 4 种可能组合中的一种。4 行分别对应真值对 TT、TF、FT 和 FF，这里只有第一对真值为真，即 p 的真值为真，对应 q 的真值也为真。

表 1-1 命题之否定的真值

p	$\neg p$
T	F
F	T

注意在逻辑中，在一个合取的“但是”一词有时可以用“并且”代替。比如，语句“阳光灿烂，但是在下雨”，还可以表示为“阳光灿烂并且在下雨”。（在自然语言中，“并且”和“但是”在意思上有微妙的不同，这里我们不关心这个细微差别。）

例5 找出命题 p 和 q 的合取，其中 p 为命题“今天是星期五”， q 为命题“今天下雨”。

解 这两个命题的合取 $p \wedge q$ 是命题“今天是星期五并且下雨”。这一命题在下雨的星期五为真，不是星期五的日子为假，不下雨的星期五也为假。

定义3 令 p 和 q 为命题。 p 、 q 的析取用 $p \vee q$ 表示，即命题“ p 或 q ”。当 p 和 q 均为假时， $p \vee q$ 命题为假，否则为真。

$p \vee q$ 的真值表如表 1-3 所示。联接词“或”在析取中的使用对应于词“或”(or)包含的两种情况之一，即“同或”(inclusive or)。析取所含两命题之一为真或两者均为真时，析取的真值为真。例如，下面这句话中，“或”表达的即是“同或”：

“选修过微积分或计算机科学的学生可以选修本课。”

这里我们指的是，选修过微积分和计算机科学两门课的学生，以及只选修过其中一门课的学生都可以选修本课。另一方面，当我们说：

“学过微积分或学过计算机科学，但不是两者都学过的学生，可以选修本课。”

使用的“或”是“异或”。这里我们的意思是既学过微积分，又学过计算机科学的学生不能选修本课；只有那些恰好在这两门课中选修过一门的的学生可以选修本课。

表 1-2 两命题合取的真值表

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

表 1-3 两命题析取的真值表

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

同样，若餐馆的菜单上写着“汤或沙拉，加一道小菜”，则一般这都表示顾客可以选择汤，也可以选择沙拉，但不是既选汤又选沙拉时，才加一道小菜。因此，这里是“异或”而不是“同或”。

例6 若 p 和 q 就是例5中的两个命题，它们的析取是什么？

Extra Examples

解 p 和 q 的析取 $p \vee q$ 是命题

“今天是星期五，或今天下雨”

这一命题在星期五或下雨天(包括下雨的星期五)的任何一天都成真。只有在既不是星期五，又不下雨的日子，此命题为假。

前面已经说过，在析取中使用的联接词“或”(or)对应于两种情形之一，即“同或”。所以当析取中的两个命题之一为真或两者均为真时，析取成真。有时我们也按“异或”的含义使用“或”。用异或来联结命题 p 和 q 时，就得到命题“ p 或 q (但非两者)”。这一命题当 p 为真且 q 为假时成真，或反过来当 p 为假且 q 为真时也为真。 p 和 q 两者均为假或均为真时，这一命题为假。

定义4 令 p 和 q 为命题。 p 和 q 的异或(用 $p \oplus q$ 表示)是一个命题。当 p 和 q 中只有一个为真时命题为真，否则为假。

两个命题异或的真值表如表 1-4 所示。

1.1.3 条件语句

下面讨论其他几个重要的命题组合方式。

定义5 令 p 和 q 为命题。条件语句 $p \rightarrow q$ 是命题“若 p ，则 q ”。当 p 为真而 q 为假时，条件语句 $p \rightarrow q$ 为假，否则为真。在条件语句 $p \rightarrow q$ 中， p 称为假设(或前项、前提)， q 称为结论(或推论)。

Annotation: 称语句 $p \rightarrow q$ 为条件语句，是因为 $p \rightarrow q$ 可以断定在条件 p 成立的时候 q 为真。条件语句也称为蕴含。

条件语句 $p \rightarrow q$ 的真值表如表 1-5 所示。注意，当 p 和 q 都为真，或者当 p 为假（与 q 的真值无关）时， $p \rightarrow q$ 为真。

表 1-4 两命题异或的真值表

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

表 1-5 条件命题 $p \rightarrow q$ 的真值表

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

由于在数学推理中许多地方会用到条件语句，因此表示 $p \rightarrow q$ 的术语很多，下面是常用的几个：

- “如果 p ，那么 q ”
- “如果 p ，则 q ”
- “ p 是 q 的充分条件”
- “ q 如果 p ”
- “当 p 则 q ”
- “ p 的必要条件是 q ”
- “ q ，除非 $\neg p$ ”
- “ p 蕴含 q ”
- “ q 仅当 p ”
- “ q 的充分条件是 p ”
- “ q 每当 p ”
- “ q 是 p 的必要条件”
- “ q 跟随 p ”

为了便于理解条件语句的真值表，可以将条件语句想象为合同或义务。例如，许多政治家在竞选时都许诺：

“如果我当选了，那么我将会减税。”

如果这个政治家当选了，选民将期望他能减税。进一步地，如果这个政治家没有当选，那么选民就无法指望他能减税，尽管这个人也许有足够的影响力可令当权者减税。只有在该政治家当选但却没有减税的情况下，选民才能说政治家违背了竞选诺言。这种情形对应于在 $p \rightarrow q$ 中 p 为真但 q 为假的情况。

类似地，考虑教授可能作出的如下陈述：

“如果你在期末考试得了满分，那么你的成绩将被评定为 A。”

如果你设法在期末考试得了满分，那么你可以期望得到 A。如果你没得到满分，那么你是否能得到 A 将取决于其他因素。然而，如果你得到满分，但教授没有给你 A，你会有受骗的感觉。

许多人对“ p 仅当 q ”与“如果 p ，那么 q ”表示同一意思感到有些不解。要理解这一点，请记住，“ p 仅当 q ”说的是当 q 不为真时 p 不能为真。也就是说，如果 p 为真但 q 为假，则这个语句为假。当 p 为假时， q 或者为真，或者为假，因为语句并没有谈及 q 的真值。人们常犯的一个错误是用“ q 仅当 p ”来表达 $p \rightarrow q$ 。然而，当 p 和 q 有不同的真值时，这两个语句有不同的真值。

“除非”这个词经常用于表达条件语句。例如“ q ，除非 $\neg p$ ”，意思是如果 $\neg p$ 是假的，则 q 必是真的。也就是说，当 p 为真， q 为假时，语句“ q ，除非 $\neg p$ ”是假的，否则是真的。因此，“ q ，除非 $\neg p$ ”与 $p \rightarrow q$ 总是有相同的真值。

例 7 说明了条件语句与中文语句之间的转换。

例 7 令 p 为语句“玛利亚学离散数学”， q 为语句“玛利亚将找到好工作”。用中文表达语句 $p \rightarrow q$ 。

Extra Examples

解 从条件语句的定义，我们得知，当 p 为语句“玛利亚学离散数学”， q 为语句“玛利亚将找到好工作”时， $p \rightarrow q$ 代表了语句

“如果玛利亚学习离散数学，那么她将找到好工作。”

还有许多其他表述方法来表达这个条件语句。其中大多数自然的是：

“玛利亚学离散数学时，她将找到一份好工作。”

“玛利亚为了得到一份好工作，她需要学离散数学。”

以及

“玛利亚将找到一份好工作，除非她不学离散数学。”

注意我们定义条件语句的方法比其汉语陈述更加通用。比如，在例7中的条件语句，又如语句

“如果今天天晴，那么我们将去海滩。”

中有假设和结论之间的联系，这是一般语言中的语句。而且除非玛利亚学离散数学但没有找到好工作，以及除非今天的晴天但我们不去海滩，否则上述语句总是成立。另一方面，语句

“如果今天是星期五，那么 $2+3=5$ 。”

总是成立的，因为它的结论是真的（于是假设部分的真值不起作用）。条件语句

“如果今天是星期五，那么 $2+3=6$ 。”

是除星期五以外天天成真，尽管 $2+3=6$ 为假。

在自然语言中，我们不会使用最后这两个条件语句（除非偶尔在讽刺中用到），因为其中的假设和结论之间没有什么联系。在数学推理中我们考虑的条件语句比语言中使用的要广泛一些。条件语句作为一个数学概念不依赖于假设和结论之间的因果关系。我们关于条件语句的定义规定了它的真值，而这一定义不是以语言的用法为基础的。命题语言是一种人工语言，这里为了便于使用和记忆，才将其类比为语言的用法。

许多程序设计语言中使用的 if-then（如果-那么）结构与逻辑中使用的不同。大部分程序设计语言中都有 if p then S 这样的语句，其中 p 是命题而 S 是一个程序段（待执行的一条或多条语句）。当程序的运行遇到这样一条语句时，如果 p 为真，就执行 S ；但若 p 为假，则 S 不执行。下面的例子说明这一点。

例8 若执行语句

if $2+2=4$ then $x := x+1$

之前， $x=0$ ，执行以后 x 的值是什么？（符号 $:=$ 代表赋值，语句 $x := x+1$ 表示将 $x+1$ 的值赋给 x 。）

解 因为 $2+2=4$ 为真，所以赋值语句 $x := x+1$ 被执行。因此，在执行此语句之后， x 的值是 $0+1=1$ 。

逆、倒置与反

由条件语句 $p \rightarrow q$ 可以构成一些新的条件语句。特别地，产生有3个相关的条件语句，通常它们拥有特殊的名称。命题 $q \rightarrow p$ 称为 $p \rightarrow q$ 的逆蕴含，而 $p \rightarrow q$ 的倒置蕴含是命题 $\neg q \rightarrow \neg p$ 。命题 $\neg p \rightarrow \neg q$ 称为 $p \rightarrow q$ 的反蕴含。我们会发现，3个由 $p \rightarrow q$ 衍生的条件语句中，只有倒置蕴含和 $p \rightarrow q$ 总是有相同的真值。

我们首先看到，与 $p \rightarrow q$ 有相同真值的是倒置蕴含 $\neg q \rightarrow \neg p$ 。注意，只有当 $\neg p$ 为假且 $\neg q$ 为真，即 p 为真且 q 为假时，该倒置蕴含为假。现在我们确定，无论 p 和 q 的真值如何，逆 $q \rightarrow p$ 和反蕴含 $\neg q \rightarrow \neg p$ 与原蕴含 $p \rightarrow q$ 都没有相同的真值。注意，当 p 为真 q 为假时，原蕴含命题为假，而逆命题和反命题都为真。

当两个复合命题总是具有相同真值时，我们称之为等价。因此一个条件语句与它的倒置等价，一个蕴含的逆蕴含与反蕴含也是等价的，读者可以验证这一点，但它们都不与原条件语句等价（1.2节将介绍等价命题）。一个常见的逻辑错误是假设一个条件语句的逆蕴含和反蕴含等价于这个条件语句。

例9 说明蕴含的用法。

Example 例9 找出如下语句的倒置蕴含、逆蕴含和反蕴含：

“每当下雨时，主队就能获胜。”

解 因为“ q 每当 p ”是表达语句 $p \rightarrow q$ 的一种方式，原始语句可以被改写为：

“如果下雨，那么主队就能获胜。”

因此，这个蕴含的倒置蕴含是

“如果主队没有获胜，那么没有下雨。”

逆蕴含是

“如果主队获胜，那么下雨了。”

反蕴含是

“如果没有下雨，那么主队没有获胜。”

其中只有倒置蕴含等价于原始语句。

双条件语句

下面我们介绍两个命题具有相同真值的复合命题的另外一种表达方式。

定义 6 令 p 和 q 为命题。双条件语句 $p \leftrightarrow q$ 是命题“ p 当且仅当 q ”。当 p 和 q 有同样的真值时，双条件语句为真，否则为假。双条件语句也称为双蕴含。

$p \leftrightarrow q$ 的真值表如表 1-6 所示。注意，双蕴含 $p \leftrightarrow q$ 恰在 $p \rightarrow q$ 和 $q \rightarrow p$ 两个蕴含均为真，以及均为假时为真。因此，我们用“当且仅当”来表示这一逻辑联接词，并且在符号书写时把符号 $\leftarrow$ 和 $\rightarrow$ 结合起来。一般还有一些其他方式表达 $p \leftrightarrow q$ ：

“ p 是 q 的充分必要条件”，

“如果 p 那么 q ；反之亦然”，

“ p 当且仅当 q ”。

表 1-6 双蕴含 $p \leftrightarrow q$ 的真值表

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

双条件语句的最后一种表示方式可以用缩写符号“iff”代替“当且仅当”。注意， $p \leftrightarrow q$ 与 $(p \rightarrow q) \wedge (q \rightarrow p)$ 有完全相同的真值。

例 10 令 p 为语句“你可以坐飞机”，令 q 为语句“你买机票了”。则 $p \leftrightarrow q$ 为语句

“你可以坐飞机当且仅当你买机票了。”

若 p 和 q 均为真或均为假，此语句为真。也就是说，如果你买机票了就能坐飞机，或是如果你没买机票就不能坐飞机。当 p 和 q 有对立真值时，此命题为假，也就是说，当你没买机票但却能坐飞机时（比如你获得一次免费旅行）或当你买了机票却不能坐飞机时（比如航空公司拒绝你登机）。

双条件的隐含应用

你应该意识到，在自然语言中，双条件并不总是显示的。特别是在双条件中使用的“当且仅当”结构在普通语言中很少用。通常用“如果，那么”或“仅当”结构来表示双蕴含，“当且仅当”的另一部分是隐含的。例如，考虑这个自然语句“你吃完饭才可以吃餐后甜点”。它意味着“你可以吃餐后甜点当且仅当你吃完饭”，这个语句逻辑等价于“如果你吃完饭，那么你可以吃餐后甜点”和“你可以吃餐后甜点，仅当你吃完饭”。后面这个语句在逻辑上等价于两个语句，一个是“如果你吃完饭，那么你可以吃甜点”，另一个是“仅当你吃完了饭，你才能吃甜点。”由于自然语言的这种不精确性，我们需要对自然语言中的条件语句是否隐含它的逆做出假设。因为数学和逻辑注重精确，所以我们总是区分条件语句 $p \rightarrow q$ 和双条件语句 $p \leftrightarrow q$ 。

1.1.4 复合命题的真值表

现在已经介绍了 4 个重要的逻辑联接词——合取，析取，条件，双条件，还有否定。我们可以用这些联接词包括一些命题变元构造结构复杂的复合命题。我们可以按照例 11 中阐明的方法，用真值表来决定这些复合命题的真值，用一个分栏来决定在这个复合命题中出现的每个复合表达式的真值。由每个命题变元的真值组合成的复合命题的真值在最后一栏中表示。

例 11 构造复合命题 $(p \vee \neg q) \rightarrow (p \wedge q)$ 的真值表。

解 因为真值表涉及 2 个命题变元 p 和 q ，因而此表有 4 行，对应 4 种真值组合 TT、TF、FT 和 FF。前两列分别表示 p 和 q 的真值。第三列为 $\neg q$ 的真值，以此决定在第 4 列中 $p \vee \neg q$ 的真值。

$p \wedge q$ 的真值在第 5 列。最后, $(p \vee \neg q) \rightarrow (p \wedge q)$ 的真值在最后一列。真值表的结果如表 1-7 所示。■

表 1-7 复合命题 $(p \vee \neg q) \rightarrow (p \wedge q)$ 的真值表

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

1.1.5 逻辑运算符的优先级

使用目前为止学到的否定运算符和逻辑运算符, 可以构造复合命题。一般地, 我们使用括号来指明复合命题中的逻辑运算符的顺序。例如, $(p \vee q) \wedge (\neg r)$ 是 $p \vee q$ 和 $\neg r$ 的合取。然而, 为了减少括号的数量, 我们指定否定运算符在其他所有逻辑运算符前执行。这意味着 $\neg p \wedge q$ 是 $\neg p$ 和 q 的合取, 即 $(\neg p) \wedge q$, 而不是 p 和 q 的合取的否定, 即 $\neg(p \wedge q)$ 。

另一个常用的优先级规则是合取运算符优先于析取运算符, 因此 $p \wedge q \vee r$ 意味着 $(p \wedge q) \vee r$, 而不是 $p \wedge (q \vee r)$ 。因为这个规则不太好记, 所以我们将继续使用括号来区别析取运算符和合取运算符的顺序。

最后, 一个已被接受的规则是条件运算符和双条件运算符的优先级低于合取和析取运算符。因此, $p \vee q \rightarrow r$ 等同于 $(p \vee q) \rightarrow r$ 。当涉及条件运算符和双条件运算符时, 我们也将使用括号, 尽管条件运算的优先级高于双条件运算。表 1-8 显示出各种逻辑运算符的优先级。

表 1-8 逻辑运算符的优先级

运算符	优先级
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5

1.1.6 翻译语句

有许多理由需把语言翻译成由命题变量和逻辑联接词组成的表达式, 特别是因为语言(包括一切人类语言)常有二义性, 而把句子译成逻辑表达式可以消除歧义。注意, 做这种翻译也许要在句子含义的基础上做些合理的假设。此外, 一旦我们完成了从句子到逻辑表达式的翻译, 我们就可以分析这些逻辑表达式以决定它们的真值。我们还可以对它们进行处理, 并用推理规则(参见 1.5 节)对它们进行推理。

为说明把语句翻译成逻辑表达式的过程, 考虑下面两个例子。

例 12 怎样把下面的句子翻译成逻辑表达式?

“只有你主修计算机科学或不是新生, 才可以从校园网访问因特网。”

解 翻译这一句子为逻辑表达式有许多方法。可以用一个命题变元, 例如 p 来表示这一句子, 但在分析其含义或用它做推理时, 这种表示不会有什么作用。我们的办法是用命题变元表示其中的每一个句子成分, 并找出其间合适的逻辑联接词。具体地说, 令 a 、 c 和 f 分别表示“你可以从校园网访问因特网”、“你主修计算机科学”和“你是个新生”。注意到“只有……才……”是表达条件语句的一种方式, 上述句子可以译为

$$a \rightarrow (c \vee \neg f)$$

例 13 怎样把下面的句子翻译成逻辑表达式?

“除非你已满 16 周岁, 否则只要你身高不足 4 英尺[⊖]就不能乘公园滑行铁道游乐车。”

解 令 q 、 r 和 s 分别表示“你能乘公园滑行铁道游乐车”、“你身高不足 4 英尺”和“你已满 16 周岁”, 则上述句子可以翻译为

$$(r \wedge \neg s) \rightarrow \neg q$$

⊖ 1 英尺 = 30.48 厘米。——译者注

当然，还有其他方式可以把上述句子表示为逻辑表达式，但上面使用的这一表达式已满足我们的需要。 ■

1.1.7 系统规范说明

在说明硬件系统和软件系统时，将自然语言语句翻译成逻辑表达式是很重要的一部分。系统和软件工程师从自然语言中提取需求，生成精确、无二义性的规范说明，这些规范说明可作为系统开发的基础。例14说明如何在这一过程中使用复合命题。

例14 使用逻辑联接词表示规范说明“当文件系统满时，自动应答不能够发出”。

 **解** 翻译这个规范说明的方法之一是令 p 表示“自动应答能够发出”，令 q 表示“文件系统满了”，则 $\neg p$ 表示“自动应答不能够发出”。因此，题中的规范说明可以用条件语句 $q \rightarrow \neg p$ 来表示。 ■

这些系统规范说明应该是一致的，也就是说，系统规范说明不应该包含有冲突的需求。当规范说明不一致时，无法开发出满足所有规范说明的系统。

例15 确定下列系统规范说明是否一致：

“诊断消息存储在缓冲区中或者被重传。”

“诊断消息没有存储在缓冲区中。”

“如果诊断消息存储在缓冲区中，那么它被重传。”

解 要判断这些规范说明是否一致，我们首先用逻辑表达式表示它们。令 p 为“诊断消息存储在缓冲区中”，令 q 表示“诊断消息被重传”。则上面几个规范说明可以写为 $p \vee q$ ， $\neg p$ 和 $p \rightarrow q$ 。使所有三个规范说明为真的一个真值赋值必须有 p 为假使 $\neg p$ 为真。因为我们要使 $p \vee q$ 为真，但 p 必须为假， q 必须为真。由于当 p 为假且 q 为真时， $p \rightarrow q$ 为真，我们得出结论这些规范说明是一致的，因为当 p 为假且 q 为真时它们都是真的。使用一个真值表检验有关 p 和 q 的真值的四种可能赋值，我们可以得出同样的结论。 ■

例16 如果在例15中加上一个系统规范说明“诊断消息不被重传”，它们还能保持一致吗？

解 由例15可知，只有当 p 为假且 q 为真时那三个规范说明才为真。然而，本例中的新规范说明是 $\neg q$ ，当 q 为真时它为假。因此，这四个规范说明不是一致的。 ■

1.1.8 布尔检索

 逻辑联接词广泛用于大量信息检索中，例如，检索网页索引。由于这些检索使用来自命题逻辑的技术，所以称为布尔检索。

在布尔检索中，联接词 AND 由于匹配包含两个检索项的记录，联接词 OR 用于匹配两个检索项之一或两项均匹配的记录，而联接词 NOT （有时写作 $AND NOT$ ）用于排除某个特定的检索项。当用布尔检索为有潜在价值的信息定位时，常需要细心安排逻辑联接词的使用。下面的例子说明布尔检索是怎样执行的。

 **例17 网页检索。**大部分网上搜索引擎支持布尔检索技术，它有助于找到有关特定主题的网页。例如，要用布尔检索找出关于新墨西哥州(New Mexico)各大学的网页，我们可以寻找与 $NEW AND MEXICO AND UNIVERSITIES$ 匹配的网页，检索的结果将包括含 NEW （新）、 $MEXICO$ （墨西哥）和 $UNIVERSITIES$ （大学）三个词的那些网页。这里面包含了所有我们感兴趣的网页，还包括其他网页，例如有关墨西哥的新大学的网页。另一例子是，要找出与新墨西哥州或亚利桑那州(Arizona)的大学有关的网页，我们可以检索与 $(NEW AND MEXICO OR ARIZONA) AND UNIVERSITIES$ 匹配的网页。（注意，其中联接词 AND 优先于联接词 OR 。）这一检索的结果将包括含 $UNIVERSITIES$ 一词和 NEW 与 $MEXICO$ 两词的所有网页及含 $UNIVERSITIES$ 一词和 $ARIZONA$ 一词的所有网页。同样除这两类我们感兴趣的网页外还会列出其他网页。最后，要想找出有关墨西哥（不是新墨西哥州）的大学的网页，可以先找与 $MEXICO AND UNIVERSITIES$ 匹配的网页，但由于这一检索的结果将会包括有关新墨西哥州的大学的网页以及墨西哥的大学的网页，所以更好的办法是检索与 $(MEXICO AND UNIVERSITIES) NOT$

NEW 匹配的网页。这一检索的结果将包括含 MEXICO 和 UNIVERSITIES 两个词但不含词 NEW 的所有网页。在一些搜索引擎中，词 NOT 会被减号所代替，因此最后搜索的短语会是 MEXICO UNIVERSITIES-NEW。

1.1.9 逻辑难题

可以用逻辑推理解决的难题称为逻辑难题。求解逻辑难题是实践逻辑规则的一种好方法。同样，设计用于执行逻辑推理的计算机程序通常也使用著名的逻辑难题来演示它们的能力。许多人对求解逻辑难题感兴趣，有许多书和杂志也登载逻辑难题以供娱乐。

在此我们将讨论两个逻辑难题。首先介绍一个由逻辑学家和难题制作大师 Raymond Smullyan 提出的难题，Smullyan 已经出版了十多本与逻辑难题(涉及逻辑推理)有关的书。

例 18 Smullyan[Sm78]中提出了许多与如下情形有关的难题：一个岛上居住着两类人——骑士和流氓。骑士说的都是实话，而流氓只会说谎。你碰到两个人 A 和 B，如果 A 说“B 是骑士”，B 说“我们两人不是一类人”。请判断 A、B 两人到底是流氓还是骑士。

解 令 p 和 q 分别表示语句“A 是骑士”和“B 是骑士”，则 $\neg p$ 和 $\neg q$ 分别表示“A 是流氓”和“B 是流氓”。

我们首先考虑 A 是骑士的情形，这就是说 p 是真的。如果 A 是骑士，那他说“B 是骑士”就是真话，因此 q 为真，A 和 B 就是一类人。但是，如果 B 是骑士，那么 B 说的话应该为真，即 $(p \wedge \neg q) \vee (\neg p \wedge q)$ 为真，然而却并非如此，因为 A 和 B 都是骑士。因此，我们可以得出 A 不是骑士，即 p 为假。

如果 A 是流氓，则据题目可知 A 所说的“B 是骑士”就不是真话，这意味着 q 为假且 B 也是流氓。而且，如果 B 是流氓，那么 B 说的话也是假的，这与 A 和 B 都是流氓是一致的。所以，我们得出结论 A 和 B 都是流氓。

在本节末的练习 28~30 中，我们会进一步讨论 Smullyan 关于骑士和流氓的难题。接下来，我们介绍一个与两个孩子有关的“泥巴孩子难题”(muddy children puzzle)。

例 19 父亲让两个孩子(一个男孩，一个女孩)在后院玩耍，并让他们不要把身上搞脏。然而，在玩的过程中，两个孩子都在额头上沾了泥。当孩子们回来后，父亲说“你们当中至少有一个人额头上有泥”，然后他问每个孩子“你知道你额头上有没有泥吗？”同样的问题对每个孩子都问了两遍。假设每个孩子都可以看到对方的额头上是否有泥，但不能看见自己的额头，孩子们将会怎样回答呢？假设两个孩子都很诚实并且都同时回答每一次提问。

解 令 s 和 d 分别表示语句“儿子的额头上有泥”和“女儿的额头上有泥”。当父亲说“你们当中至少有一个人额头上有泥”时，表示的是 $s \vee d$ 为真。当父亲第一次问那个问题时两个孩子都将回答“不知道”，因为他们都看到对方的额头上有泥。也就是说，儿子知道 d 为真，但不知道 s 是否为真。而女儿知道 s 为真，但不知道 d 是否为真。

在儿子对第一次询问回答“不知道”后，女儿可以判断出 d 必为真。这是因为问第一次问题时，儿子知道 $s \vee d$ 为真，但不能判断 s 是否为真。利用这个信息，女儿能够得出结论 d 必定为真，因为如果 d 为假，儿子有理由推出，由于 $s \vee d$ 为真，那么 s 必定为真，因此他对第一个问题的回答应为“知道”。儿子也可以类似推断出 s 必为真。因此，第二次两个孩子都将回答“知道”。

1.1.10 逻辑运算和位运算

计算机用字位表示信息，每个字位有两个可能的值，即 0 和 1。字位的这一含义来自二进制数字，因为 0 和 1 是用在数的二进制表示中的数字。1946 年著名的统计学家图基(John Tukey)引入了这一术语。字位可以用于表示真值，因为只有两个真值，即真与假。习惯上用 1 表示真，用 0 表示假。也就是说，1 表示 T(真)，0 表示 F(假)。若变量的值为真或为假，则此变量称为布尔变量。于是一个布尔变量可以用一个字位表示。

真值	位
T	1
F	0

计算机的字位运算对应于逻辑联接词，只要在运算符 $\wedge$ 、 $\vee$ 和 $\oplus$ 的真值表中用 1 代替 T，用 0

代替 F，就能得到表 1-9 所示的对应的字位运算表。我们还将用符号 OR、AND 和 XOR 表示运算符 $\vee$ 、 $\wedge$ 和 $\oplus$ ，许多程序设计语言正是这样表示的。

信息一般用位串表示，也就是用 0 和 1 的序列表示。这时，对位串的运算即可用来处理信息。

表 1-9 字位运算符 OR、AND 和 XOR 的真值表

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

定义 7 位串是 0 个或多个字位的序列。位串的长度就是它所含的字位的个数。

例 20 101010011 是长度为 9 的一个位串。

我们可以把字位运算扩展到位串上。我们在长度相同的两个位串上定义它们的按位 OR、按位 AND 和按位 XOR 分别为这样的位串，其中每个字位均由被运算的两个位串对应字位经 OR、AND 和 XOR 运算而得。我们分别用符号 $\vee$ 、 $\wedge$ 和 $\oplus$ 表示按位 OR、按位 AND 和按位 XOR 运算。我们用下面的例子解释对位串的这些按位运算。

例 21 求位串 01 1011 0110 和 11 0001 1101 的按位 OR、按位 AND 和按位 XOR(这里以及本书其他地方均把位串按 4 位分块以便于阅读)。

解 这两个位串的按位 OR、按位 AND 和按位 XOR 分别由对应字位的 OR、AND 和 XOR 得到，其结果是

$$\begin{array}{r}
 01\ 1011\ 0110 \\
 11\ 0001\ 1101 \\
 \hline
 11\ 1011\ 1111 \quad \text{按位 OR} \\
 01\ 0001\ 0100 \quad \text{按位 AND} \\
 10\ 1010\ 1011 \quad \text{按位 XOR}
 \end{array}$$

练习

- 下列哪些语句是命题？这些是命题的语句的真值是什么？
 - 波士顿是马萨诸塞州首府。
 - 迈阿密是佛罗里达州首府。
 - $2+3=5$ 。
 - $5+7=10$ 。
 - $x+2=11$ 。
 - 回答这一问题。
- 下列各命题的否定是什么？
 - 今天是星期四。
 - 新泽西没有污染。
 - $2+1=3$ 。
 - 缅因州的夏天又热又晒。
- 令 p 和 q 分别表示命题“在新泽西海岸游泳是允许的”和“在海岸附近发现过鲨鱼”。把下列各复合命题表达为汉语句子：

a) $\neg q$	b) $p \wedge q$	c) $\neg p \vee q$	d) $p \rightarrow \neg q$
e) $\neg q \rightarrow p$	f) $\neg p \rightarrow \neg q$	g) $p \leftrightarrow \neg q$	h) $\neg p \wedge (p \vee \neg q)$
- 令 p 、 q 为如下命题：

p ：气温在零度以下。

q ：正在下雪。

用 p 、 q 和逻辑联接词写出下列各命题：

 - 气温在零度以下且正在下雪。
 - 气温在零度以下，但没下雪。
 - 气温不在零度以下，也不下雪。
 - 也许在下雪，也许在零度以下(也许两者在内)。

- e) 若气温在零度以下，那也就在下雪。
 f) 也许气温在零度以下，也许在下雪，但如果在零度以下，就不在下雪。
 g) 气温在零度以下是下雪的充分必要条件。
5. 令 p 、 q 为如下命题：
 p : 你的车速超过每小时 65 英里(1 英里 = 1.6 公里)。
 q : 你接到一张超速罚单。
 用 p 、 q 和逻辑联接词写出下列命题：
 a) 你的车速没有超过每小时 65 英里。
 b) 你的车速超过每小时 65 英里，但没接到超速罚单。
 c) 你的车速若超过每小时 65 英里，将接到一张超速罚单。
 d) 你的车速不超过每小时 65 英里，就不会接到超速罚单。
 e) 车速超过每小时 65 英里足以接到超速罚单。
 f) 你接到一张超速罚单，但你的车速没超过每小时 65 英里。
 g) 只要你接到一张超速罚单，你的车速就超过每小时 65 英里。
6. 令 p 、 q 、 r 为如下命题：
 p : 在这个地区发现过灰熊。
 q : 在小路上徒步旅行是安全的。
 r : 小路两旁的草莓成熟了。
 用 p 、 q 、 r 和逻辑联接词写出下列命题：
 a) 小路两旁的草莓成熟了，但在这个地区没有发现过灰熊。
 b) 在这个地区没有发现过灰熊，且在小路上徒步旅行是安全的，但小路两旁的草莓成熟了。
 c) 如果小路两旁的草莓成熟了，徒步旅行是安全的当且仅当在这个地区没有发现过灰熊。
 d) 在小路上徒步旅行不安全，但在这个地区没有发现过灰熊且小路两旁的草莓成熟了。
 e) 为了使在小路上旅行很安全，其必要条件(但并不充分)是小路两旁的草莓没有成熟且在这个地区没有发现过灰熊。
 f) 无论何时在这个地区发现过灰熊且小路两旁的草莓成熟了，在小路上徒步旅行就不安全。
7. 判断下列各条件语句是真还是假：
 a) 若 $1+1=2$ ，则 $2+2=5$ 。
 b) 若 $1+1=3$ ，则 $2+2=4$ 。
 c) 若 $1+1=3$ ，则 $2+2=5$ 。
 d) 若猴子会飞，那么 $1+1=3$ 。
8. 下列各语句，判断其中想表达的是同或还是异或，说明理由：
 a) 晚餐有咖啡或者茶。
 b) 关键词一定至少要有 3 个数字或至少 8 个字符长。
 c) 这个课程的必要条件是数论课程或者是密码课程。
 d) 你能用美元或者欧元。
9. 对下列各语句，说一说其中的或是同或(即析取)与异或时它们的含义。你认为语句想表示的是哪个或？
 a) 要选修离散数学课，你必须已经选修微积分或计算机科学的一门课。
 b) 从 Acme 汽车公司购买一部新车，你就能得到 2 000 美元现金回扣，或 2% 的汽车贷款。
 c) 两人套餐包括 A 列中的两项或 B 列中的三项。
 d) 若下雪超过两英尺或寒风低于 -100，学校就停课。
10. 把下列语句写成“如果 p ，那么 q ”的形式。[提示：参考本节列出的通常表达条件语句的方式。]
 a) 吹东北风的时候就下雪。
 b) 暖天持续一周苹果树就开花。
 c) 活塞队赢得冠军就意味着他们打败了湖人队。
 d) 必须走 8 英里才能到朗斯峰的顶峰。
 e) 要得到教授职位，世界闻名就够了。
 f) 如果你驾车超过 400 英里，就需要买汽油了。
 g) 只有你购买的 CD 机不超过 90 天，你的保修单才有效。
 h) 除非水太凉了，否则 Jan 将去游泳。

11. 把下列命题写成“ p 当且仅当 q ”的形式：

- 如果外边热你就买冰激凌；如果你买冰激凌，外边就热。
- 你赢得竞赛的充分必要条件是拥有那唯一的获胜券。
- 只有你有关系才能得到提拔，只有得到提拔你才有关系。
- 如果你看电视，记忆会衰退；反之亦然。
- 火车恰在我乘坐的那些日子晚点。

12. 给出下列各条件语句的逆命题、逆否命题和否命题：

- 如果今天下雪，我明天就去滑雪。
- 只要有测验，我就来上课。
- 只有当正整数没有1和它自己以外的因数时，它才是素数。

13. 下列各复合命题的真值表显示有多少行？

- $p \rightarrow \neg p$
- $(p \vee \neg r) \wedge (q \vee \neg s)$
- $q \vee p \vee \neg s \vee \neg r \vee \neg t \vee u$
- $(p \wedge r \wedge t) \leftrightarrow (q \wedge t)$

14. 为下列各复合命题构造真值表：

- $p \wedge \neg p$
- $p \vee \neg p$
- $(p \vee \neg q) \rightarrow q$
- $(p \vee q) \rightarrow (p \wedge q)$
- $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$
- $(p \rightarrow q) \rightarrow (q \rightarrow p)$

15. 为下列各复合命题构造真值表：

- $(p \vee q) \rightarrow (p \oplus q)$
- $(p \oplus q) \rightarrow (p \wedge q)$
- $(p \vee q) \oplus (p \wedge q)$
- $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow q)$
- $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow \neg r)$
- $(p \oplus q) \rightarrow (p \oplus \neg q)$

16. 为下列各复合命题构造真值表：

- $p \rightarrow \neg q$
- $\neg p \leftrightarrow q$
- $(p \rightarrow q) \vee (\neg p \rightarrow q)$
- $(p \rightarrow q) \wedge (\neg p \rightarrow q)$
- $(p \leftrightarrow q) \vee (\neg p \leftrightarrow q)$
- $(\neg p \leftrightarrow \neg q) \leftrightarrow (p \leftrightarrow q)$

17. 为下列各复合命题构造真值表：

- $p \rightarrow (\neg q \vee r)$
- $\neg p \rightarrow (q \rightarrow r)$
- $(p \rightarrow q) \vee (\neg p \rightarrow r)$
- $(p \rightarrow q) \wedge (\neg p \rightarrow r)$
- $(p \leftrightarrow q) \vee (\neg q \leftrightarrow r)$
- $(\neg p \leftrightarrow \neg q) \leftrightarrow (q \leftrightarrow r)$

18. 构造 $(p \leftrightarrow q) \leftrightarrow (r \leftrightarrow s)$ 的真值表。

19. 求下列各对位串的按位 OR、按位 AND 及按位 XOR：

- 101 1110, 010 0001
- 1111 0000, 1010 1010
- 00 0111 0001, 10 0100 1000
- 11 1111 1111, 00 0000 0000

模糊逻辑可用于人工智能。在模糊逻辑中命题的真值是介于0和1(包括0和1)之间的数。以0为真值的命题为假，以1为真值的命题为真。0和1之间的真值表示不同程度的真值。例如，语句“傅雷德是幸福的”的真值可以是0.8，因为傅雷德大部分时间是幸福的；“约翰是幸福的”的真值可能是0.4，因为他幸福的时间比一半稍短。

20. 模糊逻辑中命题否定的真值是1减去该命题的真值。语句“傅雷德不幸福”和“约翰不幸福”的真值是什么？

21. 模糊逻辑中两个命题的析取的真值是两个命题真值的最大值。语句“傅雷德幸福或约翰幸福”与“傅雷德不幸福或约翰不幸福”的真值是什么？

*22. 在一个100条语句的列表中，第 n 条语句是“在这个列表中，恰有 n 条语句为假。”

- 从这些语句中你可以得出什么结论？
- 若第 n 条语句是“在这个列表中，至少有 n 条语句为假”，回答问题a)。
- 假设这个列表包含99条语句，回答问题b)。

23. 边远村庄的每个人要么总说真话，要么总说谎。对旅游者的问题，村民要么回答“是”，要么回答“不”。假定你在这一地区旅游，走到了一个岔路口，一条岔路通向你想去的遗址，另一条岔路通向丛林深处。此时恰有一村民站在岔路口，问村民什么样的一个问题就能决定走哪条路？

24. 使用命题 p “扫描消息中的病毒”和 q “消息来自一个未知的系统”以及逻辑联接词来表达下列系统规范说明。

- “每当消息来自一个未知的系统时，就扫描消息中的病毒。”
- “消息来自一个未知的系统，但不扫描消息中的病毒。”
- “每当消息来自一个未知的系统时，就有必要扫描消息中的病毒。”
- “当消息不是来自一个未知的系统时，就不扫描消息中的病毒。”

25. 下列系统规范说明一致吗？

“当且仅当系统正常操作时，系统处于多用户状态。如果系统正常操作，则它的核心程序正在运行。核心程序不能正常运行，或者系统处于中断模式。如果系统不处于多用户状态，它就处于中断模式。系统不处在中断模式。”

26. 下列系统规范说明一致吗？

“路由器能向边缘系统发送分组仅当它支持新的地址空间时。要让路由器支持新的地址空间，就必须安装最新的软件发布。如果最新的软件发布被安装了，路由器就能向边缘系统发送分组。路由器不支持新的地址空间。”

27. 你会用什么样的布尔检索来寻找关于新泽西州海滩的网页？如果你想找关于泽西岛（在英吉利海峡）海滩的网页呢？

练习 28~30 是关于 Smullyan 的趣味逻辑的。一个岛上居住着两类人：骑士和流氓。骑士说的都是实话，而流氓只会说谎。根据下列不同情况判断 A、B 两人到底是流氓还是骑士，如果认为得不出最终结论的，也请说出任何可能的结论。

28. A 说：“我们之间至少有一个是流氓”，B 什么都没说。

29. A 说：“我是流氓或者 B 是骑士”，B 什么都没说。

30. A 说：“我们都是流氓”，B 什么都没说。

练习 31~33 是逻辑难题，解题时可以先把语句翻译成逻辑表达式，再用真值表从这些表达式作推理。

31. 斯蒂夫想用两个事实来判断三位工作伙伴的相对薪水。首先他知道如果傅雷德的薪水不是三人中最高的，那么杰尼斯的最高。其次他知道如果杰尼斯的薪水不是最低的，那么麦吉的最高。从以上斯蒂夫知道的事实，有可能决定傅雷德、麦吉和杰尼斯的相对薪水吗？如果能，谁的最高，谁的最低？解释你的推理。

32. 侦探调查了罪案的四位证人。从证人的话侦探得出的结论是：如果男管家说的是真话，那么厨师说的也是真话；厨师和园丁说的不可能都是真话；园丁和杂役不可能都在说谎；如果杂役说真话，那么厨师在说谎。侦探能判定这四位证人分别是在说谎还是在说真话吗？解释你的推理。

*33. 试求解下面这个由爱因斯坦提出的著名的逻辑难题（也称为斑马难题）。5 个不同国家且工作各不相同的人分别住在一条街上的 5 所房子里，每所房子的颜色不同，每个人都有自己养的不同的宠物，喜欢喝不同的饮料。根据以下提示，你能告诉我哪所房子里的人养斑马（zebra），哪所房子里的人喜欢喝矿泉水吗？

英国人住在红色的房子里。西班牙人养了一条狗。日本人是一个油漆工。意大利人喜欢喝茶。挪威人住在左边的第一个房子里。绿房子在白房子的右边。摄影师养了一只蜗牛。外交官住在黄房子里。中间那个房子里的人喜欢喝牛奶。喜欢喝咖啡的人住在绿房子里。挪威人住在蓝色的房子旁边。小提琴家喜欢喝橘子汁。养狐狸的人所住的房子与医师的房子相邻。养马的人所住的房子与外交官的房子相邻。[提示：绘一张表，其中行表示每个人，列表示他们所住的房子的颜色、他们的工作、他们养的宠物以及他们喜欢喝的饮料，用逻辑推理来判断表中正确的项。]

1.2 命题等价

1.2.1 引言

数学证明中使用的一种重要方法是用真值相同的一条语句取代另一条语句。因此，在构造数学证明时广泛使用从已知复合命题产生具有同样真值的其他命题的方法。注意，我们将用术语“复合命题”来解释由用于逻辑运算的命题变元形成的表达式，比如 $p \wedge q$ 。

我们就从根据可能的真值对复合命题进行分类开始讨论。

定义 1 复合命题称为永真式（或重言式），如果无论其中出现的命题的真值是什么，它的真值总是真。真值永远为假的复合命题称为矛盾。最后，既不是永真式又不是矛盾的命题称为可能式。

在数学推理中永真式和矛盾往往很重要，下面的例 1 解释了两类复合命题。

例 1 我们可以只用一个命题构造永真式和矛盾。 $p \vee \neg p$ 和 $p \wedge \neg p$ 的真值表如表 1-10 所示。因为 $p \vee \neg p$ 总是真，所以它是永真式。因为 $p \wedge \neg p$ 总是假，所以它是矛盾。

表 1-10 永真式和矛盾的例子

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

1.2.2 逻辑等价

在所有可能的情况下都有相同真值的两个复合命题称为逻辑等价。我们也可以如下定义这一概念。

定义 2 如果 $p \leftrightarrow q$ 是永真式，命题 p 和 q 称为是逻辑等价的。记号 $p \equiv q$ 表示 p 和 q 逻辑等价。

注意 符号 $\equiv$ 不是逻辑联接词，因为 $p \equiv q$ 不是复合命题，而语句 $p \leftrightarrow q$ 是永真式。符号 $\Leftrightarrow$ 有时用来代替 $\equiv$ 表示等价。

判定两个命题是否等价的方法之一是使用真值表。特别地，命题 p 和 q 等价当且仅当给出它们真值的两列完全一致。例 2 建立了一个非常重要且有用的逻辑等价，即 $\neg(p \vee q)$ 或 $\neg p \wedge \neg q$ ，用以说明这个方法。这个逻辑等价是两大德摩根律之一，

表 1-11 德摩根定律

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$

如表 1-11 所示。这是用 19 世纪中叶英国数学家德摩根(Augustus De Morgan)的名字命名的。

例 2 证明 $\neg(p \vee q)$ 和 $\neg p \wedge \neg q$ 逻辑等价。

解 表 1-12 给出了这些复合命题的真值。由于对 p 和 q 所有可能的真值组合，命题 $\neg(p \vee q)$ 和 $\neg p \wedge \neg q$ 的真值都一样，所以 $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ 是永真式，这两个命题逻辑等价。

表 1-12 $\neg(p \vee q)$ 和 $\neg p \wedge \neg q$ 的真值表

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

例 3 证明命题 $p \rightarrow q$ 和 $\neg p \vee q$ 逻辑等价。

解 在表 1-13 中构造了这两个复合命题的真值表。由于 $\neg p \vee q$ 和 $p \rightarrow q$ 的真值一致，因此它们是逻辑等价的。

表 1-13 $\neg p \vee q$ 和 $p \rightarrow q$ 的真值表

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

现在，我们将建立涉及 3 个不同命题变元 p 、 q 、 r 的 2 个复合命题的逻辑等价。真值表需要建立 8 行，每一行为 3 个变元真值的一种可能组合。列出 p 、 q 、 r 的真值来标记这些组合。这 8 种真值组合是 TTT, TTF, TFT, TFF, FTT, FTF, FFT, 以及 FFF；我们用这个顺序显示真值表行。注意在真值表中增加一个命题变元，我们需要用原来行数的 2 倍来展示复合命题是等价的，这样就需要 16 行来建立由 4 个命题变元组成的 2 个复合命题的逻辑等价，以此类推，如果一个复合命题由 n 个命题变元组成，则需要 2^n 行。

例 4 证明命题 $p \vee (q \wedge r)$ 和 $(p \vee q) \wedge (p \vee r)$ 逻辑等价。这是析取对合取的分配律。

解 表 1-14 中构造了这两个复合命题的真值表。因为 $p \vee (q \wedge r)$ 和 $(p \vee q) \wedge (p \vee r)$ 的真值一样，它们是逻辑等价的。

表 1-14 $p \vee (q \wedge r)$ 和 $(p \vee q) \wedge (p \vee r)$ 逻辑等价的演示

p	q	r	$q \wedge r$	$p \vee (q \wedge r)$	$p \vee q$	$p \vee r$	$(p \vee q) \wedge (p \vee r)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

表 1-15 给出了若干重要的等价关系^①。在这些等价关系中，T 表示永远为真的复合命题，F 表示永远为假的复合命题。对于涉及条件语句和双条件语句的复合命题，我们也在表 1-16 和表 1-17 中给出了一些有用的等价关系。本节末的练习中要求读者证明表 1-15~表 1-17 中的等价关系。

表 1-15 逻辑等价

等 价 关 系	名 称
$p \wedge T \equiv p$ $p \vee F \equiv p$	恒等律
$p \vee T \equiv T$ $p \wedge F \equiv F$	支配律
$p \vee p \equiv p$ $p \wedge p \equiv p$	幂等律
$\neg(\neg p) \equiv p$	双非律
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	交换律
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	结合律
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	分配律
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	德摩根定律
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	吸收律
$p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$	否定律

表 1-16 涉及条件语句的逻辑等价

$p \rightarrow q \equiv \neg p \vee q$
$p \rightarrow q \equiv \neg q \rightarrow \neg p$
$p \vee q \equiv \neg p \rightarrow q$
$p \wedge q \equiv \neg(p \rightarrow \neg q)$
$\neg(p \rightarrow q) \equiv p \wedge \neg q$
$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

表 1-17 涉及双条件的逻辑等价

$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

析取的结合律表明表达式 $p \vee q \vee r$ 是有定义的。因为先析取 p 和 q 再与 r 析取或先析取 q 和 r 再与 p 析取，其结果一样。同样， $p \wedge q \wedge r$ 也是有定义的。扩展这一推理过程可以证明： $p_1 \vee p_2 \vee \cdots \vee p_n$ 和 $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ 均有定义，只要 $p_1, p_2, \dots, p_n$ 为命题。进而可以注意到德摩根律可以扩展为

$$\neg(p_1 \vee p_2 \vee \cdots \vee p_n) \equiv (\neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n)$$

和

$$\neg(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \equiv (\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n)$$

① 这些恒等式是布尔代数中恒等式的一种特殊情况。请将这些恒等式与 2.2 节表 2-21 中的集合恒等式比较一下。

1.2.3 德摩根律的运用

德摩根律的两个逻辑等价式非常重要。它们告诉我们怎么去否定合取和析取。特别地，等价式 $\neg(p \vee q) \equiv \neg p \wedge \neg q$ 说明，析取的否定是由各分命题的否定的合取组成的。同理，等价式 $\neg(p \wedge q) \equiv \neg p \vee \neg q$ 说明，一个合取的否定是由各分命题的否定的析取组成的。例5说明了德摩根律的运用。

例5 用德摩根律分别表达“迈格尔有一部手机且有一台便携式电脑”和“希瑟或史蒂夫将去看音乐会”的否定。

解 令 p 为“迈格尔有一部手机”， q 为“迈格尔有一个便携式电脑”，那么“迈格尔有一部手机且有一台便携式电脑”可以表达为 $p \wedge q$ 。用德摩根第一定律， $\neg(p \wedge q)$ 等价于 $\neg p \vee \neg q$ 。结果，我们可以将原命题的否定表达为“迈格尔没有一部手机或没有一台便携式电脑”。

令 r 为“希瑟将去看音乐会”， s 为“史蒂夫将去看音乐会”，那么“希瑟或史蒂夫将去看音乐会”可以表达为 $r \vee s$ 。用德摩根第二定律， $\neg(r \vee s) \equiv \neg r \wedge \neg s$ 。结果，我们可以将原命题的否定表达为“希瑟和史蒂夫都将不去看音乐会”。

1.2.4 构建新的逻辑等价式

表1-15中的逻辑等价关系以及业已建立起来的其他(如表1-16和表1-17中所示的那些)等价关系，可以用于构造其他等价关系，原因是复合命题中的一个命题可以用与它逻辑等价的复合命题替换而不改变原复合命题的真值。这种方法可由例6至例8得到说明。在这些例子中，我们还使用了如下事实：如果 p 和 q 逻辑等价， q 和 r 逻辑等价，那么 p 和 r 也逻辑等价。

Extra Examples

例6 证明 $\neg(p \rightarrow q)$ 和 $p \wedge \neg q$ 逻辑等价。

解 我们可以用真值表来证明这两个复合命题是等价的(和例4中的方法相似)。事实上，不必那么复杂。然而，想解释如何用我们已经知道的逻辑证明来建立新的逻辑证明，要用到大量的变元。因此，以 $\neg(p \rightarrow q)$ 开始，以 $p \wedge \neg q$ 结束，每次用表1-15中的一个等价式，通过发展一系列逻辑等价式的方法来建立这个等价式。我们有下列等价式。

$$\begin{aligned}\neg(p \rightarrow q) &\equiv \neg(\neg p \vee q) && \text{由例3} \\ &\equiv \neg(\neg p) \wedge \neg q && \text{由德摩根第二定律} \\ &\equiv p \wedge \neg q && \text{由双非律}\end{aligned}$$

例7 证明 $\neg(p \vee (\neg p \wedge q))$ 和 $\neg p \wedge \neg q$ 逻辑等价。

解 我们可用真值表证明这两个复合命题等价。不过我们不这样做，而是每次使用表1-15中的一个等价关系，依次建立一串等价。从 $\neg(p \vee (\neg p \wedge q))$ 开始，到 $\neg p \wedge \neg q$ 结束，最终证明它们等价。我们有下列等价关系。

$$\begin{aligned}\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) && \text{由德摩根第二定律} \\ &\equiv \neg p \wedge [\neg(\neg p) \vee \neg q] && \text{由德摩根第一定律} \\ &\equiv \neg p \wedge (p \vee \neg q) && \text{由双非律} \\ &\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{由分配律} \\ &\equiv \mathbf{F} \vee (\neg p \wedge \neg q) && \text{由于 } \neg p \wedge p \equiv \mathbf{F} \\ &\equiv (\neg p \wedge \neg q) \vee \mathbf{F} && \text{由析取的交换律} \\ &\equiv \neg p \wedge \neg q && \text{由 } \mathbf{F} \text{ 的恒等律}\end{aligned}$$

于是 $\neg(p \vee (\neg p \wedge q))$ 和 $\neg p \wedge \neg q$ 逻辑等价。

例8 证明 $(p \wedge q) \rightarrow (p \vee q)$ 为永真式。

解 为证明这个命题是永真式，我们将用逻辑等价证明它逻辑上等价于 $\mathbf{T}$ 。(注意：这也可以用真值表来完成。)

$$\begin{aligned}(p \wedge q) \rightarrow (p \vee q) &\equiv \neg(p \wedge q) \vee (p \vee q) && \text{由例3} \\ &\equiv (\neg p \vee \neg q) \vee (p \vee q) && \text{由德摩根第一定律} \\ &\equiv (\neg p \vee p) \vee (\neg q \vee q) && \text{由析取的结合律和交换律} \\ &\equiv \mathbf{T} \vee \mathbf{T} && \text{由例1和析取的交换律} \\ &\equiv \mathbf{T} && \text{由支配律}\end{aligned}$$

真值表可以用于判定复合命题是否为永真式。对于只含少数变量的复合命题，可以用手工完成这一工作。但当变量数目增多时，就不可行了。例如，对于含 20 个变量的复合命题，它的真值表就有 $2^{20} = 1\,048\,576$ 行。显然，你需要一台计算机帮助你以这种方式判定含 20 个变量的复合命题是否为永真式。但是当变量数为 1000 时，一台计算机能在一个可以接受的时间内判定复合命题是否为永真式吗？要检查 2^{1000} 种（这是一个超过 300 位的十进制数）可能的真值组合中的每一种，一台计算机在几万亿年之内都不可能完成。而且迄今尚没有其他已知的计算过程能使计算机在合理的时间之内判定变量数这么大的复合命题是否为永真式。

含有两个变量的简单合取式有 $p \wedge q$, $\neg p \wedge q$, $p \wedge \neg q$ 和 $\neg p \wedge \neg q$ 。只有当 p 和 q 的真值都为真时， $p \wedge q$ 的真值为真；类似地， $\neg p \wedge q$ 为真，仅当 p 为假和 q 为真； $p \wedge \neg q$ 为真，仅当 p 为真和 q 为假； $\neg p \wedge \neg q$ 为真，仅当 p 和 q 都为假。

例 9 请写出一个命题公式，使其具有如表 1-18 所示的真值表。

表 1-18 真值表

p	q	?
T	T	T
T	F	T
F	T	F
F	F	T

对于真值表中真值为真的每一行，选择变量仅在该行所列取值时真值为真的二元合取式，将它们用 $\vee$ 连接，得到的就是以这个表为真值表的命题公式。真值表中真值为真的三行对应的合取式为 $p \wedge q$, $p \wedge \neg q$ 和 $\neg p \wedge \neg q$ ，所以该真值表的命题公式为

$$(p \wedge q) \vee (p \wedge \neg q) \vee (\neg p \wedge \neg q)$$
 ■

含有三个变量的合取式有类似的情况。对于表 1-19 的每一行，合取式的真值为真仅当其中的变量的取值为该行所列的变量值，否则合取式的真值为假。

表 1-19 含有三个变量的合取式情况

	p	q	r
$p \wedge q \wedge r$	T	T	T
$p \wedge q \wedge \neg r$	T	T	F
$p \wedge \neg q \wedge r$	T	F	T
$p \wedge \neg q \wedge \neg r$	T	F	F
$\neg p \wedge q \wedge r$	F	T	T
$\neg p \wedge q \wedge \neg r$	F	T	F
$\neg p \wedge \neg q \wedge r$	F	F	T
$\neg p \wedge \neg q \wedge \neg r$	F	F	F

注意，在表 1-19 的每一行，当一个变量的取值为假时，该变量在合取式中被否定。

定义 3 一个合取式包含每个变量，每个变量（包括变量的否定）出现一次且仅一次，称这样的合取式为小项。由若干个小项的析取构成的析取式，称为析取范式。如果 $m_1, m_2, m_3, \dots, m_n$ 是小项， $m_1 \vee m_2 \vee m_3 \vee \dots \vee m_n$ 是析取范式。

要找到一个命题公式，使它的真值表是给定的真值表，可以对真值表中真值为真的每一行在表 1-19 中找到对应的合取式，将它们用 $\vee$ 连接。例如，要找到一个命题公式，使它的真值表如表 1-20 所示。表 1-20 中有三行真值为真，它对应的命题公式为

$$(p \wedge q \wedge r) \vee (p \wedge \neg q \wedge \neg r) \vee (\neg p \wedge q \wedge r)$$

这个析取式就是析取范式，其中的合取式 $p \wedge q \wedge r, p \wedge \neg q \wedge \neg r, \neg p \wedge q \wedge r$ 为小项。

表 1-20 真值表

p	q	r	$?$
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	F

类似地，析取式 $p \vee q \vee r$ 的真值为假，仅当 p, q 和 r 都为假。表 1-21 列出所有三个变量的析取式。

表 1-21 含有三个变量的析取式情况

	p	q	r
$\neg p \vee \neg q \vee \neg r$	T	T	T
$\neg p \vee \neg q \vee r$	T	T	F
$\neg p \vee q \vee \neg r$	T	F	T
$\neg p \vee q \vee r$	T	F	F
$p \vee \neg q \vee \neg r$	F	T	T
$p \vee \neg q \vee r$	F	T	F
$p \vee q \vee \neg r$	F	F	T
$p \vee q \vee r$	F	F	F

表 1-21 中每行的析取式，仅当变量取值为这一行所列变量值时，它的真值为假，否则它的真值为真。注意，当一个变量的取值为真时，该变量在对应的析取式中被否定。

定义 4 一个析取式包含每个变量，每个变量(包括变量的否定)出现一次且仅一次，称这样的析取式为大项。由若干个大项的合取构成的合取式，称为合取范式。如果 $M_1, M_2, M_3, \dots, M_n$ 是大项， $M_1 \wedge M_2 \wedge M_3 \wedge \dots \wedge M_n$ 是合取范式。

要找到一个命题公式，使它的真值表是给定的真值表，可以对真值表中真值为假的每一行在表 1-21 中找到对应的析取式，将它们用 $\wedge$ 连接。例如，要找到一个命题公式，使它的真值表如表 1-20 所示。表 1-20 中有 5 行真值为假，它对应的命题公式为

$$(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee \neg r) \wedge (p \vee q \vee r)$$

这个合取式就是合取范式，其中的析取式 $\neg p \vee \neg q \vee r, \neg p \vee q \vee \neg r, p \vee \neg q \vee r, p \vee q \vee \neg r, p \vee q \vee r$ 为大项。

练习

1. 用真值表证明下列等价关系。

- a) $p \wedge \text{F} \equiv p$
- b) $p \vee \text{F} \equiv p$
- c) $p \wedge \text{F} \equiv \text{F}$
- d) $p \vee \text{T} \equiv \text{T}$
- e) $p \vee p \equiv p$
- f) $p \wedge p \equiv p$

2. 用真值表证明交换律。

a) $p \vee q \equiv q \vee p$

b) $p \wedge q \equiv q \wedge p$

3. 用真值表证明分配律。

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

4. 用德摩根律求下列命题的否定。

- a) Jan 是富裕的, 也是快乐的。
- b) Carlos 明天骑自行车或者跑步。
- c) Mei 步行或乘公共汽车去上课。
- d) Ibrahim 既聪明又工作努力。

5. 用真值表证明下列各条件语句为永真式。

- a) $(p \wedge q) \rightarrow p$
- b) $p \rightarrow (p \vee q)$
- c) $\neg p \rightarrow (p \rightarrow q)$
- d) $(p \wedge q) \rightarrow (p \rightarrow q)$
- e) $\neg(p \rightarrow q) \rightarrow p$
- f) $\neg(p \rightarrow q) \rightarrow \neg q$

6. 不用真值表证明练习 5 中的各条件语句为永真式。

7. 用真值表证明下列称为吸收律的等价关系。

- a) $p \vee (p \wedge q) \equiv p$
- b) $p \wedge (p \vee q) \equiv p$

8. 判断 $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$ 是否为永真式。

练习 9~14 都是要求证明两个复合命题逻辑等价。做这样的题目, 你要证明表达式中命题变元的相同真值组合, 或者都为真, 或者都为假。(每题都很容易。)

9. 证明 $\neg(p \leftrightarrow q)$ 和 $p \leftrightarrow \neg q$ 逻辑等价。

10. 证明 $\neg p \leftrightarrow q$ 和 $p \leftrightarrow \neg q$ 逻辑等价。

11. 证明 $\neg(p \leftrightarrow q)$ 和 $\neg p \leftrightarrow q$ 逻辑等价。

12. 证明 $(p \rightarrow r) \wedge (q \rightarrow r)$ 和 $(p \vee q) \rightarrow r$ 逻辑等价。

13. 证明 $(p \rightarrow r) \vee (q \rightarrow r)$ 和 $(p \wedge q) \rightarrow r$ 逻辑等价。

14. 证明 $p \leftrightarrow q$ 和 $(p \rightarrow q) \wedge (q \rightarrow p)$ 逻辑等价。

15. 证明 $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$ 是永真式。

16. 证明 $(p \rightarrow q) \rightarrow r$ 和 $p \rightarrow (q \rightarrow r)$ 不等价。

17. 证明 $(p \rightarrow q) \rightarrow (r \rightarrow s)$ 和 $(p \rightarrow r) \rightarrow (q \rightarrow s)$ 不逻辑等价。

只含逻辑运算符 $\vee$ 、 $\wedge$ 和 $\neg$ 的复合命题的对偶是将该命题中的每个 $\vee$ 用 $\wedge$ 代替, 每个 $\wedge$ 用 $\vee$ 代替, 每个 T 用 F 代替, 每个 F 用 T 代替得到的命题。命题 s 的对偶用 s' 表示。

18. 求下列命题的对偶。

- a) $p \wedge \neg q \wedge \neg r$
- b) $(p \wedge q \wedge r) \vee s$
- c) $(p \vee F) \wedge (q \vee T)$

19. 当 s 是复合命题时, 证明 $(s')' = s$ 。

**20. 为什么两个只含运算符 $\wedge$ 、 $\vee$ 和 $\neg$ 的等价复合命题的对偶也等价?

21. 找一个只含命题 p 、 q 和 r 的复合命题, 在 p 、 q 和 r 中恰有两个为真时命题为真, 否则为假。[提示: 构造合取的析取。对命题成真的每一种组合包含一个合取。每个合取都应包含三个命题变元中每个或它的否定。]

22. 证明 $\neg$ 、 $\wedge$ 和 $\vee$ 构成一个功能完备的逻辑运算符集合。[提示: 利用每个复合命题都逻辑等价于一个析取范式这个事实。]

*23. 证明 $\neg$ 和 $\vee$ 构成一个功能完备的逻辑运算符集合。

下面几道题用到逻辑运算符 NAND(与非)和 NOR(或非)。命题 $p \text{ NAND } q$ 在 p 为假或 q 为假或两者均为假时为真; 当 p 和 q 均为真时为假。命题 $p \text{ NOR } q$ 只在 p 和 q 均为假时为真, 否则为假。命题 $p \text{ NAND } q$ 和 $p \text{ NOR } q$ 分别表示为 $p \mid q$ 和 $p \downarrow q$ 。(运算符 $\mid$ 和 $\downarrow$ 分别以谢佛和皮尔斯的名字命名为谢佛竖(Sheffer stroke)和皮尔斯箭头。)

24. 证明 $p \mid q$ 逻辑等价于 $\neg(p \wedge q)$ 。

25. 证明 $p \downarrow q$ 逻辑等价于 $\neg(p \vee q)$ 。

*26. 只用运算符 $\downarrow$ 构造一个等价于 $p \rightarrow q$ 的命题。

27. 证明 $p \mid q$ 和 $q \mid p$ 等价。

*28. 只用命题变元 p 和 q 能有多少不同的复合命题真值表?

29. 下面的语句取自一个电话系统的规范说明: “如果电话号码数据库是打开的, 那么监督程序被置于关闭状态, 只要系统不在初态。”这句话有两个条件语句, 使规范说明很难懂。找一个等价的易懂的规范说明, 使其只涉及析取和否定, 不涉及条件语句。

30. 通过对 p, q, r, s 赋真值, 析取表达式 $p \vee \neg q \vee s, \neg p \vee \neg r \vee s, \neg p \vee \neg r \vee \neg s, \neg p \vee q \vee \neg s, q \vee r \vee \neg s, q \vee \neg r \vee \neg s, \neg p \vee \neg q \vee \neg s, p \vee r \vee s, p \vee r \vee \neg s$ 中有多少可以同时为真?

如果对命题变量的真值赋值可使复合命题为真, 则称这个复合命题是可满足的。

31. 解释一下用于判定复合命题可满足性的算法如何能用来判断一个复合命题是否是永真式。[提示: 考虑 $\neg p$, 其中 p 是已经考察过的命题。]

1.3 谓词和量词

1.3.1 引言

在 1.1 节和 1.2 节中学习的命题逻辑, 不能充分地表达数学语言和自然语言中语句的意思。比如, 假设我们知道

“每个连接到大学网络的电脑运行正常。”

没有命题逻辑规则让我们判断

“MATH3 在正常运行”

这句话的真实性, 其中 MATH3 是连接在大学网络的一台电脑。同样, 我们不能用命题逻辑的规则判断以下语句的正确性:

“CS2 被一个入侵者攻击。”

其中 CS2 是连接在大学网络的计算机; “有一台连接在大学网络的计算机正遭受一入侵者的攻击”。

本节将介绍一种更强类型的逻辑, 即谓词逻辑。我们将明白如何用谓词逻辑表达命题在数学和计算机科学中广泛的语句意义, 用这个方法可以帮助我们推理和探索对象之间的关系。为了理解谓词逻辑, 我们首先需要介绍谓词的概念, 之后, 介绍量词的概念, 它可以让我们用能够判断语句属于某种类型的所有对象拥有某种性质的语句断言, 以及具有特定性质的语句断言。

1.3.2 谓词

含变量的语句, 如

“ $x > 3$ ”, “ $x = y + 3$ ”, “ $x + y = z$ ”

“计算机 x 被一个入侵者攻击”

以及

“计算机 x 在正常运行”

常见于数学断言、计算机程序, 以及系统规格说明。在变量值未知的时候, 这些语句既不为真也不为假。本节将讨论从这种语句产生命题的方式。

语句“ x 大于 3”有两部分, 第一部分即变量 x 是语句的主语; 第二部分谓词“大于 3”, 表明语句的主语会有的一性质。我们可以用 $P(x)$ 表示语句“ x 大于 3”, 其中 P 表示谓词“大于 3”, 而 x 是变量。也把语句 $P(x)$ 说成是命题函数 P 在 x 的值。一旦给变量 x 赋一个值, 语句 $P(x)$ 就成为命题, 因而有真值。考虑下面的例 1 和例 2。

例 1 令 $P(x)$ 表示语句“ $x > 3$ ”, $P(4)$ 和 $P(2)$ 的真值是什么?

解 在语句“ $x > 3$ ”中让 $x=4$ 即得到语句 $P(4)$ 。因此 $P(4)$ 即语句“ $4 > 3$ ”, 为真; 但语句 $P(2)$ 即“ $2 > 3$ ”, 为假。 ■

例 2 令 $A(x)$ 表示语句“计算机 x 正被一个入侵者攻击”。假设在校园的计算机中, 只有 CS2 和 MATH1 经常被入侵者攻击, 那么 $A(\text{CS1})$ 、 $A(\text{CS2})$ 和 $A(\text{MATH1})$ 的真值是什么?

解 在语句“计算机 x 正被一个入侵者攻击”中, 设 $x=\text{CS1}$ 得到语句 $A(\text{CS1})$ 。因为 CS1 不在经常受到攻击的名单中, 所以判断 $A(\text{CS1})$ 为假。同样, 因为 CS2 和 MATH1 在经常受攻击的名单中, 我们知道 $A(\text{CS2})$ 和 $A(\text{MATH1})$ 为真。 ■

还可以让语句中含不止一个变量, 例如, 考虑语句“ $x = y + 3$ ”。我们可以用 $Q(x, y)$ 表示这条语句, 其中 x, y 为变量, Q 为谓词。在赋值给 x 和 y 时, 语句 $Q(x, y)$ 就有了真值。

例 3 令 $Q(x, y)$ 表示语句“ $x = y + 3$ ”, 命题 $Q(1, 2)$ 和 $Q(3, 0)$ 的真值是什么?

解 要得到 $Q(1, 2)$, 在 $Q(x, y)$ 中令 $x=1, y=2$ 。因此, $Q(1, 2)$ 为语句“ $1 = 2 + 3$ ”。

为假。语句 $Q(3, 0)$ 为语句“ $3=0+3$ ”，为真。 ■

例 4 令 $A(c, n)$ 表示语句“计算机 c 被连接到网络 n ”，其中变量 c 代表一台计算机，变量 n 代表一网络。假设计算机 MATH1 被连接到 CAMPUS2，但没有被连接到 CAMPUS1。 $A(\text{MATH1}, \text{CAMPUS1})$ 和 $A(\text{MATH1}, \text{CAMPUS2})$ 的真值是什么？

解 因为 MATH1 没有连接到 CAMPUS1 网络，我们知道 $A(\text{MATH1}, \text{CAMPUS1})$ 为假。然而，因为 MATH1 连接到了 CAMPUS2 网络，我们知道 $A(\text{MATH1}, \text{CAMPUS2})$ 为真。 ■

同样，我们可以令 $R(x, y, z)$ 表示语句“ $x+y=z$ ”，当变量 x, y, z 被赋值时，此句就有真值。

例 5 命题 $R(1, 2, 3)$ 和 $R(0, 0, 1)$ 的真值是什么？

解 在语句 $R(x, y, z)$ 中令 $x=1, y=2, z=3$ ，即得到命题 $R(1, 2, 3)$ 。可以看出 $R(1, 2, 3)$ 就是语句“ $1+2=3$ ”，为真。 $R(0, 0, 1)$ 即语句“ $0+0=1$ ”，为假。 ■

一般涉及 n 个变量 $x_1, x_2, \dots, x_n$ 的语句可以用 $P(x_1, x_2, \dots, x_n)$ 表示。

形为 $P(x_1, x_2, \dots, x_n)$ 的语句是命题函数 P 在 n 元组 $(x_1, x_2, \dots, x_n)$ 的值， P 也称为 n 元谓词。

下面的例子说明命题函数可以出现在计算机程序中。

例 6 考虑语句

if $x > 0$ **then** $x := x + 1$

当程序中遇到这样一条语句时，变量 x 在程序运行到此刻的值即被代入 $P(x)$ ，也就是代入“ $x > 0$ ”。如果对 x 的这一值 $P(x)$ 为真，即执行赋值语句 $x := x + 1$ ，于是 x 的值增加 1。如果对 x 的这一值 $P(x)$ 为假，则不执行赋值语句，所以 x 的值不改变。 ■

也可以用谓词验证计算机程序，当有效输入时总能得到正确的输出。描述有效输入的语句叫做前置条件，当程序运行时，输出应该满足的条件称为后置条件。如例 7 所示，用谓词来表达前置条件和后置条件。

例 7 考虑下面的交换变量 x 和 y 值的程序。

```
temp := x
x := y
y := temp
```

找出能作为前置条件和后置条件的谓词，以证明此程序的正确性。然后解释如何用它们验证对所有有效的输入，程序的运行都能达到目的。

解 对于前置条件，我们需要表达在运行程序之前 x 和 y 被赋值。因此，对于这个前置条件可以用谓词 $P(x, y)$ 表示，其中 $P(x, y)$ 指语句“ $x=a, y=b$ ”， a 和 b 是在运行程序之前 x 和 y 的值。因为我们想证明对于所有输入变量，程序交换了 x 和 y 的值，对后置条件可以用 $Q(x, y)$ 表示，其中 $Q(x, y)$ 表示语句“ $x=b, y=a$ ”。

为证明程序总是按照预期运行，假设前置条件 $P(x, y)$ 成立。也就是说，假设命题“ $x=a, y=b$ ”为真，这意味着 $x=a, y=b$ 。程序的第一步， $\text{temp} := x$ ，将 x 的值赋给 temp ，所以下一步是 $x=a, \text{temp}=a, y=b$ 。程序第二步， $x := y$ ，因此 $x=b, \text{temp}=a, y=b$ 。最后，第三步，我们知道 $x=b, \text{temp}=a$ ，以及 $y=a$ 。结果，程序运行后，后置条件 $Q(x, y)$ 成立，也就是说，语句“ $x=b, y=a$ ”为真。 ■

1.3.3 量词

当命题函数中所有变量均被赋值时，得到的命题有一个真值。还有另一重要方式，也可以从命题函数产生命题，这就是量化。量化表示谓词在一定范围的事物上成立的程度。在语言中，单词“所有”、“一些”、“许多”、“没有”以及“没几个”被用于量化。这里讨论两类量化：全称量化，它告诉我们一个谓词在所考虑的每一对象中都为真；存在量化，它告诉我们一个谓词对所考虑中的一个或多个对象成真。处理谓词和量词的逻辑领域称为谓词演算。

全称量词 许多数学命题对于某一性质在变量的某一特定域内的所有值均为真，这一特定域称为变量的**论域**(或**全体域**)，时常简单地称为**域**。这类语句用全称量化表示。 $P(x)$ 对特定论域的全称量化是指：它断言 $P(x)$ 对 x 在其论域中的所有值为真。注意，论域是变量 x 所有可能指定的值。当我们改变论域时， $P(x)$ 的全称量化的意义也随之改变。在用全称量词时，必须指定论域，否则语句的全称量词无意义。

定义 1 $P(x)$ 的全称量化是语句“ $P(x)$ 对 x 在其论域的所有值为真”。

符号 $\forall xP(x)$ 表示 $P(x)$ 的全称量化，其中 $\forall$ 称为**全称量词**。命题 $\forall xP(x)$ 也可表示为“对所有 x ， $P(x)$ ”或“对每个 x ， $P(x)$ ”。对于 $P(x)$ 是假的对象称为 $\forall xP(x)$ 的**反例**。

全称量词的意义在表 1-22 的第一行做了总结。我们用例 8~13 来说明全称量词的使用。

表 1-22 量词

命 题	何时为真	何时为假
$\forall xP(x)$	对每一个 x ， $P(x)$ 都为真	有一个 x ，使 $P(x)$ 为假
$\exists xP(x)$	有一个 x ，使 $P(x)$ 为真	对每一个 x ， $P(x)$ 都为假

例 8 令 $P(x)$ 为命题“ $x+1>x$ ”，量化语句 $\forall xP(x)$ 的真值是什么？其中论域是实数集合。

解 由于 $P(x)$ 对所有实数 x 均为真，量化语句 $\forall xP(x)$

的值为真。

注意 通常，一个隐含假设使量词的所有论域为非空的。如果论域为空，那么 $\forall xP(x)$ 对任何命题函数 $P(x)$ 都为真，因为论域中没有元素 x 使 $P(x)$ 为假。

除了“对所有”和对“每一个”外，全称量词还可以用其他方式表达，包括“所有的”、“对任一个”、“任何”、“对任意的”、“给定任何的”等。

注意 最好避免使用“对于任一 x ”，因为它常常引起“每一”和“一些”的歧义。某种程度上，“任一”是歧义的，就像它用于否定句时那样，如“没有任一理由忽视学习。”

当且仅当 $P(x)$ 不总为真时 $\forall xP(x)$ 为假，其中 $P(x)$ 是一个命题函数， x 在论域中。要证明当 x 在论域中的情况下 $P(x)$ 不总为真，一种方法是找到 $\forall xP(x)$ 的一个反例。注意仅仅一个反例就是我们确定 $\forall xP(x)$ 为假所需要的全部。例 9 解释了如何使用反例。

例 9 令 $Q(x)$ 表示语句“ $x<2$ ”。若论域是实数集合，量化语句 $\forall xQ(x)$ 的真值是什么？

解 $Q(x)$ 并非对所有实数都为真，例如， $Q(3)$ 就为假。也就是说， $x=3$ 是语句 $\forall xQ(x)$ 的反例。因此 $\forall xQ(x)$

为假。

例 10 假设 $P(x)$ 是“ $x^2>0$ ”。要证明语句 $\forall xP(x)$ 为假(其中论域是所有整数)，我们给出一个反例。 $x=0$ 是一个反例，因为当 $x=0$ 时， $x^2=0$ 而不大于 0。

在数学研究中，寻找全称量化语句的反例是一个重要的部分，在本书后续章节中还会介绍。当论域中的所有元素可以一一列出时，例如列成 $x_1, x_2, \dots, x_n$ ，全称量化 $\forall xP(x)$ 与合取 $P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n)$

是一回事，因为这一合取为真，当且仅当 $P(x_1), P(x_2), \dots, P(x_n)$ 全为真。

例 11 若论域是不超过 4 的正整数， $P(x)$ 是语句“ $x^2<10$ ”， $\forall xP(x)$ 的真值是什么？

解 语句 $\forall xP(x)$ 就是合取 $P(1) \wedge P(2) \wedge P(3) \wedge P(4)$

因为论域由 1、2、3 和 4 组成。由于 $P(4)$ 即语句“ $4^2<10$ ”为假， $\forall xP(x)$ 为假。

例 12 如果 $N(x)$ 指“计算机 x 被连到网络”，论域为校园里所有的计算机，那么 $\forall xN(x)$ 是什

么意思呢?

解 语句 $\forall x N(x)$ 的意思是对于校园里的每一台计算机 x , 它都被连到了网络。这句话可以用自然语言表达为“校园里每一台计算机都连到网络”。

正如我们已经指出的那样, 当使用量词时, 说明论域是很重要的。量化语句的真值通常取决于哪个对象处于论域中, 如例 13 所示。

例 13 如果论域包括所有实数, $\forall x(x^2 \geq x)$ 的真值是什么? 如果论域包括所有整数, 真值又是什么?

解 论域是所有实数时, 全称量词 $\forall x(x^2 \geq x)$ 为假。例如, $(1/2)^2 \not\geq 1/2$ 。注意 $x^2 \geq x$ 当且仅当 $x^2 - x = x(x-1) \geq 0$ 。因此, $x^2 \geq x$ 当且仅当 $x \leq 0$ 或 $x \geq 1$ 。由此得出, 如果论域包括所有实数, $\forall x(x^2 \geq x)$ 为假(因为当 $0 < x < 1$ 时, 不等式不成立)。然而, 如果论域为整数, $\forall x(x^2 \geq x)$ 为真, 因为对任一整数 x 都不存在 $0 < x < 1$ 。

存在量词 许多数学定理断定有一个具有某种性质的元素。这类语句用存在量化表示。用存在量化可以构成这样一个命题: 它成真的充分必要条件是论域中至少有一个值满足 $P(x)$ 为真。

定义 2 $P(x)$ 的存在量化是命题“论域中存在一个元素 x 满足 $P(x)$ ”。

我们用符号 $\exists x P(x)$ 表示 $P(x)$ 的存在量化, 其中 $\exists$ 称为**存在量词**。

当使用语句 $\exists x P(x)$ 时, 必须指定论域。而且当论域变化时, $\exists x P(x)$ 的意义也改变。如果没有指定论域, 那么语句 $\exists x P(x)$ 没有意义。存在量化 $\exists x P(x)$ 可读作

“有一个 x 满足 $P(x)$ ”,

“至少有一个 x 满足 $P(x)$ ”,

或

“对某个 x , $P(x)$ ”。

除了“存在”, 我们也可以用许多其他方式表达存在量化, 如“有一些”、“至少一个”或“有”。

存在量词的意义见表 1-22 的第二行。例 14~16 说明了存在量词的运用。

例 14 令 $P(x)$ 表示语句“ $x > 3$ ”, 论域为实数集合, 量化语句 $\exists x P(x)$ 的真值是什么?

解 因为“ $x > 3$ ”在如 $x=4$ 时为真, $P(x)$ 的存在量化 $\exists x P(x)$ 为真。

我们发现, 当且仅当论域中没有对象满足 $P(x)$ 为真时, 语句 $\exists x P(x)$ 为假。也就是说, $\exists x P(x)$ 为假, 当且仅当 $P(x)$ 对于论域中的每一个对象都为假。例 15 将解释这个发现。

例 15 令 $Q(x)$ 表示语句“ $x = x+1$ ”, 论域是实数集, 量化语句 $\exists x Q(x)$ 的真值是什么?

解 因为对每个实数 x , $Q(x)$ 都为假, $Q(x)$ 的存在量化 $\exists x Q(x)$ 为假。

注意 通常, 当讲到量词的论域时, 隐含假设它为**非空集**。如果论域为空, 那么无论 $Q(x)$ 是什么样的命题函数, $\exists x Q(x)$ 都为假。因为当论域为空时, 论域中没有对象使 $Q(x)$ 为真。

当论域中的所有元素可以一一列出时, 例如列成 $x_1, x_2, \dots, x_n$, 存在量化 $\exists x P(x)$ 与析取

$$P(x_1) \vee P(x_2) \vee \dots \vee P(x_n)$$

是一回事, 因为当且仅当 $P(x_1), P(x_2), \dots, P(x_n)$ 中至少一个为真时, 这一析取为真。

例 16 若 $P(x)$ 是语句“ $x^2 > 10$ ”, 论域为不超过 4 的正整数, $\exists x P(x)$ 的真值是什么?

解 由于论域为 $\{1, 2, 3, 4\}$, 命题 $\exists x P(x)$ 就是析取

$$P(1) \vee P(2) \vee P(3) \vee P(4)$$

由于 $P(4)$ 即“ $4^2 > 10$ ”为真, $\exists x P(x)$ 为真。

在决定量化语句的真值时, 借助循环与搜索来思考是有益的。假定在变量 x 的论域中有 n 个对象, 要决定 $\forall x P(x)$ 是否为真, 可以对 x 的 n 个值循环查看 $P(x)$ 是否总是真。如果遇到 x 的一个值使 $P(x)$ 为假, 那就证明 $\forall x P(x)$ 为假, 否则 $\forall x P(x)$ 为真。要决定 $\exists x P(x)$ 是否为真, 我们循环查看 x 的 n 个值, 搜索使 $P(x)$ 为真的 x 之值。如果找到一个, 那么 $\exists x P(x)$ 为真; 如果总也找不到这样的 x , 则判定 $\exists x P(x)$ 为假。(注意, 当论域有无穷多个值时, 这一搜索过程不适用。不过

以这种方式思考量化语句的真值仍是有益的。)

1.3.4 其他量词

现在已经介绍了全称量词和存在量词，它们是数学和计算机科学中最重要的量词。然而，我们定义的不同量词没有数量的限制，如“确定有2个”，“有不超过3个”，“至少有100个”等。其他量词中最常见的是唯一量词，用符号 $\exists!$ 或 $\exists_1$ 表示。符号 $\exists!xP(x)$ (或 $\exists_1xP(x)$)指“存在唯一 x 使 $P(x)$ 为真”。其他表示唯一量词的词语有“确切存在一个”，“有且只有一个”。我们可以用前边学过的量词，以及命题逻辑来表达唯一性，所以唯一量词是可以避免的。通常，最好坚持运用存在量词和全称量词，因此这些量词的推理规则可以运用。

1.3.5 约束论域量词

一个缩略符号常被用来约束量词的论域。在这个符号里，在量词后有一个变量必须满足的条件。这将在例17中解释。我们也将在此节描述这个符号涉及集合成员的其他形式。

例17 语句 $\forall x < 0 (x^2 > 0)$ ， $\forall y \neq 0 (y^3 \neq 0)$ ，以及 $\exists z > 0 (z^2 = 2)$ 的意思是什么？其中各句的论域都为实数集。

解 语句 $\forall x < 0 (x^2 > 0)$ 表示对于每一个实数 $x < 0$ ，则 $x^2 > 0$ 。也就是说，它表示“一个负实数的平方为正实数”。这个语句与 $\forall x (x < 0 \rightarrow x^2 > 0)$ 意义相同。

语句 $\forall y \neq 0 (y^3 \neq 0)$ 表示对于每一个实数 $y \neq 0$ ，则有 $y^3 \neq 0$ 。也就是说，它表示“每一个非零实数的立方不为零”。注意这个语句等价于 $\forall y (y \neq 0 \rightarrow y^3 \neq 0)$ 。

语句 $\exists z > 0 (z^2 = 2)$ 表示存在一个实数 $z > 0$ ，使 $z^2 = 2$ 。也就是说，它表示“有一个正数是2的平方根”。这个语句等价于 $\exists z (z > 0 \wedge z^2 = 2)$ 。 ■

注意，全称量化的约束和一个条件语句的全称量化等价。比如， $\forall x < 0 (x^2 > 0)$ 是表达“ $\forall x (x < 0 \rightarrow x^2 > 0)$ ”的另一种方式。另一方面，存在量化的约束和一个合取的存在量化等价。比如 $\exists z > 0 (z^2 = 2)$ 是表达 $\exists z (z > 0 \wedge z^2 = 2)$ 的另一种方式。

1.3.6 量词的优先级

量词 $\forall$ 和 $\exists$ 比所有命题演算的逻辑运算符有更高的优先级。比如， $\forall x P(x) \vee Q(x)$ 是 $\forall x P(x)$ 和 $Q(x)$ 的析取。换句话说，它表示 $(\forall x P(x)) \vee Q(x)$ ，而不是 $\forall x (P(x) \vee Q(x))$ 。

1.3.7 绑定变量

当量词作用于变量 x 时，我们说此变量的这一次出现为绑定的。没有被量词绑定或设置为与某一特定值相等的变量出现称为自由的。出现在命题函数中的所有变量必须是绑定的，才能把此命题函数转变为命题。可以用全称量词、存在量词和赋值来完成转变。

逻辑表达式中应用量词的部分称为这个量词的作用域。因此，一个变量是自由的——如果在指定这个变量的公式中，变量在所有量词的作用域之外。

例18 在语句 $\exists x (x + y = 1)$ 中，变量 x 被存在量词 $\exists x$ 绑定，但是变量 y 是自由的，因为它没有被一个量词绑定且没有值被指定到这个变量。这解释了在语句 $\exists x (x + y = 1)$ 中， x 被绑定而 y 是自由的。

在语句 $\exists x (P(x) \wedge Q(x)) \vee \forall x R(x)$ 中，所有变量都绑定了。第一个量词 $\exists x$ 的作用域是表达式 $P(x) \wedge Q(x)$ ，因为 $\exists x$ 只作用于 $P(x) \wedge Q(x)$ ，没有语句的其余部分。类似地，第二个量词 $\forall x$ 的作用域是表达式 $R(x)$ 。也就是说，存在量词在 $P(x) \wedge Q(x)$ 中绑定变量 x ，全称量词 $\forall x$ 在 $R(x)$ 中绑定变量 x 。由此，我们可以用两个不同的变量 x 和 y 将语句写为 $\exists x (P(x) \wedge Q(x)) \vee \forall y R(y)$ ，因为两个量词的作用域不重叠。读者在日常使用中应注意这一点，相同的字母经常被用于表示被不同量词绑定而其作用域不重叠的变量。 ■

1.3.8 涉及量词的逻辑等价

在1.2节介绍了复合命题逻辑等价的概念。我们可将这个概念扩展到涉及谓词和量词的表达

式中。

定义 3 涉及谓词和量词的语句是逻辑等价的，当且仅当无论什么谓词代入这些语句，也不论用哪个个体论域于这些命题函数里的变量上，它们都有相同的真值。用符号 $S \equiv T$ 来表示涉及谓词和量词的两个语句 S 和 T 逻辑等价。

例 19 说明了如何表示两个涉及谓词和量词的语句逻辑等价。

例 19 表明 $\forall x(P(x) \wedge Q(x))$ 和 $\forall xP(x) \wedge \forall xQ(x)$ 逻辑等价(论域始终相同)。这个逻辑等价式表明可以在一个合取上分配全称量词。此外，也可以在一个析取上分配存在量词。然而，不能在析取上分配全称量词，也不能在合取上分配存在量词。(见练习 26。)

解 为表示这两个语句逻辑等价，必须表示出，不论 P 和 Q 是什么谓词，也不论运用哪个个体论域，它们总有相同真值。假设有特定的谓词 P 和 Q ，论域为常用的。可以做两件事来表示出 $\forall x(P(x) \wedge Q(x))$ 和 $\forall xP(x) \wedge \forall xQ(x)$ 逻辑等价。首先，表示出如果 $\forall x(P(x) \wedge Q(x))$ 为真，那么 $\forall xP(x) \wedge \forall xQ(x)$ 为真。第二步，表示出如果 $\forall xP(x) \wedge \forall xQ(x)$ 为真，那么 $\forall x(P(x) \wedge Q(x))$ 为真。

因此，假设 $\forall x(P(x) \wedge Q(x))$ 为真。这意味着如果 a 在论域中，那么 $P(a) \wedge Q(a)$ 为真。所以， $P(a)$ 为真，且 $Q(a)$ 为真。因为对论域中每个对象 $P(a)$ 为真，且 $Q(a)$ 为真都成立，可以下结论， $\forall xP(x)$ 和 $\forall xQ(x)$ 都为真。这意味着 $\forall xP(x) \wedge \forall xQ(x)$ 为真。

下一步，假设 $\forall xP(x) \wedge \forall xQ(x)$ 为真。那么 $\forall xP(x)$ 为真，且 $\forall xQ(x)$ 为真。因此，如果 a 在论域中，那么 $P(a)$ 为真，且 $Q(a)$ 为真(因为 $P(x)$ 和 $Q(x)$ 对论域中所有元素都为真，所以这里用同样的值 a 不矛盾)。接着，对于所有的 a ， $P(a) \wedge Q(a)$ 为真。所以 $\forall x(P(x) \wedge Q(x))$ 为真。现在可以下结论

$$\forall x(P(x) \wedge Q(x)) \equiv \forall xP(x) \wedge \forall xQ(x)$$

1.3.9 否定量化表达式

我们常会考虑量化表达式的否定，例如，考虑语句

“班上每个学生都学过一门微积分课”

的否定。这个语句是全称量化语句，即

$$\forall xP(x)$$

其中 $P(x)$ 为语句“ x 学过一门微积分课”，论域是你们班的所有学生。这一语句的否定是“并非班上每个学生都学过一门微积分课”。这等价于“班上有个学生没有学过微积分课”。这也就是原命题函数之否定的存在量化，即

$$\exists x \neg P(x)$$

这个例子说明了下面的等价关系：

$$\neg \forall xP(x) \equiv \exists x \neg P(x)$$

为了说明 $\neg \forall xP(x)$ 和 $\exists x \neg P(x)$ 不论命题函数 $P(x)$ 和论域是什么都为逻辑等价，首先注意 $\neg \forall xP(x)$ 为真，当且仅当 $\forall xP(x)$ 为假。下一步，注意 $\forall xP(x)$ 为假当且仅当论域中有一个元素 x 使 $P(x)$ 为假。它成立当且仅当论域中有一个元素 x 使 $\neg P(x)$ 为真。最后，注意论域中有一个元素 x 使 $\neg P(x)$ 为真当且仅当 $\exists x \neg P(x)$ 为真。将这些步骤放在一起，可以得出结论 $\neg \forall xP(x)$ 为真当且仅当 $\exists x \neg P(x)$ 为真。接着便是 $\neg \forall xP(x)$ 和 $\exists x \neg P(x)$ 逻辑等价。

假定我们想否定一个存在量化语句。例如，考虑命题“班上有个学生学过一门微积分课。”就是存在量化语句

$$\exists xQ(x)$$

其中 $Q(x)$ 为语句“ x 学过一门微积分课”。这句话的否定是命题“并非班上有个学生学过微积分课”。这等价于“班上每个学生都没学过微积分课”，这也就是原命题函数之否定的全称量化，或用量词语言表示为

$$\forall x \neg Q(x)$$

这个例子说明了等价关系

$$\neg \exists x Q(x) \equiv \forall x \neg Q(x)$$

为了表示出 $\neg \exists x Q(x) \equiv \forall x \neg Q(x)$ ，即不论 $Q(x)$ 和论域是什么都为逻辑等价。首先注意 $\neg \exists x Q(x)$ 为真当且仅当 $\exists x Q(x)$ 为假。而这个为真当且仅当论域中没有 x 使 $Q(x)$ 为真。下一步，注意论域中没有 x 使 $Q(x)$ 为真当且仅当 $Q(x)$ 对论域中的每个元素 x 都为假。最后，注意 $Q(x)$ 对论域中每个 x 都为假当且仅当 $\neg Q(x)$ 对论域中所有 x 都为真。而它成立当且仅当 $\forall x \neg Q(x)$ 为真。将这些步骤放在一起，发现 $\neg \exists x Q(x)$ 为真当且仅当 $\forall x \neg Q(x)$ 为真。我们得出结论： $\neg \exists x Q(x)$ 和 $\forall x \neg Q(x)$ 逻辑等价。

量词否定的规则称为量词的德摩根定律。这些规则总结在了表 1-23 中。

表 1-23 量词的德摩根定律

否 定	等价语句	何时为真	何时为假
$\neg \exists x P(x)$	$\forall x \neg P(x)$	对每个 x ， $P(x)$ 为假	有 x ，使 $P(x)$ 为真
$\neg \forall x P(x)$	$\exists x \neg P(x)$	有 x 使 $P(x)$ 为假	对每个 x ， $P(x)$ 为真

注意 当谓词 $P(x)$ 的论域包含 n 个元素时，其中 n 是正整数，用于否定量化语句的规则恰与 1.2 节讨论的德摩根律相同。这是称这些规则为量词的德摩根定律的原因。当论域有 n 个元素 $x_1, x_2, \dots, x_n$ 时， $\neg \forall x P(x)$ 与 $\neg (P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n))$ 相等，而后者等价于 $\neg P(x_1) \vee \neg P(x_2) \vee \dots \vee \neg P(x_n)$ （由德摩根律），该式又等价于 $\exists x \neg P(x)$ 。类似地， $\neg \exists x P(x)$ 与 $\neg (P(x_1) \vee P(x_2) \vee \dots \vee P(x_n))$ 相等，后者由德摩根律等价于 $\neg P(x_1) \wedge \neg P(x_2) \wedge \dots \wedge \neg P(x_n)$ ，该式又等于 $\forall x \neg P(x)$ 。

例 20 和例 21 中解释了量化语句的否定。

例 20 语句“有的政治家诚实”和“所有美国人都吃干酪汉堡包”的否定是什么？

解 令 $H(x)$ 表示“ x 是诚实的”，则语句“有诚实的政治家”可以用 $\exists x H(x)$ 来表示，其中论域包括所有政治家。这个语句的否定是 $\neg \exists x H(x)$ ，它等价于 $\forall x \neg H(x)$ 。这个否定可以表达为“每个政治家都不诚实。”注意，在自然语言中，语句“所有政治家是不诚实的”是含糊的。在一般应用中，这个语句通常意味着“并不是所有的政治家都是诚实的”。因此我们不用这个语句表达它的否定。

令 $C(x)$ 为“ x 吃干酪汉堡包”，则语句“所有美国人都吃干酪汉堡包”可以用 $\forall x C(x)$ 来表示，其中论域包括所有美国人。这个语句的否定是 $\neg \forall x C(x)$ ，它等价于 $\exists x \neg C(x)$ 。这个否定可以用几种不同的方式来表达，包括“一些美国人不吃干酪汉堡包”和“有一个美国人不吃干酪汉堡包”。 ■

例 21 语句 $\forall x (x^2 > x)$ 和 $\exists x (x^2 = 2)$ 的否定是什么？

解 $\forall x (x^2 > x)$ 的否定是语句 $\neg \forall x (x^2 > x)$ ，它等价于 $\exists x \neg (x^2 > x)$ 。这个表达式可以重写为 $\exists x (x^2 \leq x)$ 。 $\exists x (x^2 = 2)$ 的否定是语句 $\neg \exists x (x^2 = 2)$ ，它等价于 $\forall x \neg (x^2 = 2)$ 。这个表达式可以重写为 $\forall x (x^2 \neq 2)$ 。这些语句的真值取决于论域。 ■

在例 22 中用量词的德摩根律。

例 22 证明 $\neg \forall x (P(x) \rightarrow Q(x))$ 和 $\exists x (P(x) \wedge \neg Q(x))$ 逻辑等价。

解 用全称量词德摩根律，我们知道 $\neg \forall x (P(x) \rightarrow Q(x))$ 和 $\exists x (\neg (P(x) \rightarrow Q(x)))$ 逻辑等价。用 1.2 节表 1-16 中第 5 个逻辑等价式，我们知道 $\neg (P(x) \rightarrow Q(x))$ 和 $P(x) \wedge \neg Q(x)$ 对每个 x 逻辑等价。因为在一个逻辑等价中可以将一个逻辑等价表达式转换为另一个，得到 $\neg \forall x (P(x) \rightarrow Q(x))$ 和 $\exists x (P(x) \wedge \neg Q(x))$ 逻辑等价。 ■

1.3.10 翻译语句为逻辑表达式

将汉语(或其他自然语言)语句翻译成逻辑表达式，这在数学、逻辑编程、人工智能、软件工程及许多其他学科中是一项重要的任务。在 1.1 节中，我们学习了把句子译成由命题和逻辑联接

词构成逻辑表达式的过程。那时，我们特意对要求谓词和量词的语句翻译避而不谈。当用到量词时，将语句翻译成逻辑表达式会变得更复杂。而且，有多种方式可用于翻译一个特定的语句。（因此，没有“菜谱”式的方法可供你按部就班地学习。）我们会给出一些例子说明如何将汉语语句翻译成逻辑表达式。翻译的目标是生成简单而有用的逻辑表达式。本节我们只讨论这样的语句，可将其翻译成使用单个量词的逻辑表达式。下一节会讨论一些更复杂的要求多个量词的语句。

例 23 使用谓词和量词表达语句“班上的每个学生都学过微积分课”。

解

首先，重写语句以确定要使用的量词。重写后可得：

“对班上的每一个学生，该学生学过微积分课。”

接着，引入变量 x ，语句就变成

“对班上的每一个学生 x ， x 学过微积分课。”

然后，引入谓词 $C(x)$ ，表示语句“ x 学过微积分课”。因此，如果 x 的论域是班上的学生，我们可以将语句翻译为 $\forall x C(x)$ 。

然而，还有其他正确的翻译方法，并可使用不同的论域和其他谓词。具体选择什么方法取决于后续要进行的推理。例如，我们可能对更广泛的人群感兴趣，而不仅仅是班上的学生。如果将论域改成所有人，则语句就要变成

“对每个人 x ，如果 x 是班上的学生，那么 x 学过微积分课。”

如果 $S(x)$ 表示语句 x 在这个班上，则我们的语句可表达为 $\forall x (S(x) \rightarrow C(x))$ 。（警告：语句不能被表达为 $\forall x (S(x) \wedge C(x))$ ，因为这表示的是所有人都是这个班上的学生并且学过微积分课。）

最后，如果我们对除微积分之外的其他主修课程感兴趣，可以使用双变量量词 $Q(x, y)$ 表示语句“学生 x 学过课程 y ”。这样在上述两种方法中我们就要把 $C(x)$ 替换成 $Q(x, \text{微积分})$ ，得到 $\forall x Q(x, \text{微积分})$ 或 $\forall x (S(x) \rightarrow Q(x, \text{微积分}))$ 。■

在例 23 中演示了采用谓词和量词表达相同语句的不同方法。不过要记住，应该总是采用适合后续推理使用的最简单的方法。

例 24 用谓词和量词表达语句“这个班上某个学生去过墨西哥”和“这个班上每个学生或去过加拿大，或去过墨西哥。”

解 语句“这个班上某个学生去过墨西哥”在意思是“在这个班上有个学生，他去过墨西哥”。引入变量 x ，因此语句变成“在这个班上有个学生 x ， x 去过墨西哥”。

引入谓词 $M(x)$ 表示语句“ x 去过墨西哥”。如果 x 的论域是这个班上的学生，我们就可以将第一个语句翻译为 $\exists x M(x)$ 。

然而，如果我们感兴趣的人不在这个班上，我们会看到这个语句会有些不同。语句可表达为“有这样一个人 x ： x 是这个班的学生，并且 x 去过墨西哥。”在这种情况下， x 的论域是所有人，我们引入谓词 $S(x)$ 表示语句“ x 是这个班上的一个学生”。答案就变成了 $\exists x (S(x) \wedge M(x))$ ，因为它表示有某个人 x 是这个班上的一个学生并且去过墨西哥。[警告：语句不能表示为 $\exists x (S(x) \rightarrow M(x))$ ，因为对于某个不在这个班的人它也是真的，由于对这样的 x ， $S(x) \rightarrow M(x)$ 就变成 $F \rightarrow T$ 或者 $F \rightarrow F$ ，两个都是真的。]

类似地，对第二个语句，能够表示成“对于在这个班上的每一个 x ， x 去过加拿大，或 x 去过墨西哥”。（注意：“或”在这里，假设是包容的，而不是排斥的。）我们令 $C(x)$ 表示语句“ x 去过加拿大”。由前面的推理，如果 x 的论域是这个班上的学生，则第二个语句可以表达为 $\forall x (C(x) \vee M(x))$ 。然而，如果 x 的论域是所有人，我们的语句就表示成：“对于每一个人 x ，如果 x 在这个班，则 x 去过加拿大，或 x 去过墨西哥。”因而语句被表示成 $\forall x (S(x) \rightarrow (C(x) \vee M(x)))$ 。

不同于使用谓词 $M(x)$ 和 $C(x)$ 分别表示 x 去过墨西哥和 x 去过加拿大，我们可以使用两个变量谓词 $V(x, y)$ 表示“ x 去过国家 y ”。在本例中， $V(x, \text{墨西哥})$ 和 $V(x, \text{加拿大})$ 与 $M(x)$ 和 $C(x)$ 意思相同并可以代替它们。如果碰到某人去过多个国家的语句，用这种双变量的方法更可取。否则为了简单将坚持用单变量谓词 $M(x)$ 和 $C(x)$ 。■

1.3.11 在系统说明中运用量词

在1.1节用命题来表示系统说明。然而，许多系统说明涉及谓词和量词。这将在例25中说明。

例25 用谓词和量词表达系统说明“每封大于1MB的邮件将被压缩”和“如果用户处于活动状态，那么至少有一个网络连接有效”。

Extra Examples

解 令 $S(m, y)$ 表示“邮件 m 大于 y MB”，其中变量 x 的论域是所有邮件，变量 y 是一个正实数。令 $C(m)$ 表示“邮件 m 将被压缩”，那么说明“每封大于1MB的邮件将被压缩”。可以表达为

$$\forall m(S(m, 1) \rightarrow C(m))$$

令 $A(u)$ 表示“用户 u 处于活动状态”，其中变量 u 的论域是所有用户。令 $S(n, x)$ 表示“网络连接 n 处于 x 状态”，其中 n 的论域是所有网络连接， x 的论域是所有可能的网络连接状态。那么说明“如果用户处于活动状态，那么至少有一个网络连接有效”可以表达为

$$\exists u A(u) \rightarrow \exists n S(n, \text{available})$$

1.3.12 选自 Lewis Carroll 的例子

Lewis Carroll(实际上是 C. L. Dodgson 的笔名)是《Alice in Wonderland》的作者，也是有关符号逻辑几篇论文的作者。他的书中有用量词推理的例子。下面的两个例子选自他的书《Symbolic logic》，选自这本书的其他例子在本节末的练习中给出。这些例子说明了怎样用量词表示各种类型的词句。

例26 考虑下面这些语句，其中头两句称为前提，第三句称为结论。作为一个整体它们被称为一个论证。

“所有狮子都是凶猛的。”

“有些狮子不喝咖啡。”

“有些凶猛的动物不喝咖啡。”

(1.5节将讨论判定结论部分是不是前提部分有效后果的问题，就这个例子来说，结论是有效的。)令 $P(x)$, $Q(x)$ 和 $R(x)$ 分别为语句“ x 是狮子”，“ x 是凶猛的”和“ x 喝咖啡”。假定所有动物的集合为论域，用量词及 $P(x)$ 、 $Q(x)$ 和 $R(x)$ 表示上面这些语句。

解 可以将这些句子表示为：

$$\forall x(P(x) \rightarrow Q(x))$$

$$\exists x(P(x) \wedge \neg R(x))$$

$$\exists x(Q(x) \wedge \neg R(x))$$

注意，第二句不能表示为 $\exists x(P(x) \rightarrow \neg R(x))$ 。原因是 $P(x) \rightarrow \neg R(x)$ 在 x 不是狮子时总是成真，所以只要有一只不是狮子的动物， $\exists x(P(x) \rightarrow \neg R(x))$ 就成真，即使所有狮子都喝咖啡它也成真。同样，第三句也不能写成

$$\exists x(Q(x) \rightarrow \neg R(x))$$

例27 考虑下面的语句，其中前3个语句为前提，第4个语句为有效结论。

“所有蜂鸟都五彩斑斓。”

“没有大鸟以蜜为生。”

“不以蜜为生的鸟都色彩单调。”

“蜂鸟都是小鸟。”

令 $P(x)$, $Q(x)$, $R(x)$ 和 $S(x)$ 分别为语句“ x 是只蜂鸟”，“ x 是大的”，“ x 以蜜为生”和“ x 五彩斑斓”。假定以所有鸟的集合为论域，用量词及 $P(x)$ 、 $Q(x)$ 、 $R(x)$ 和 $S(x)$ 表示上述语句。

解 可以把论证中的语句表示为

$$\forall x(P(x) \rightarrow S(x))$$

$$\neg \exists x(Q(x) \wedge R(x))$$

$$\forall x(\neg R(x) \rightarrow \neg S(x))$$

$$\forall x(P(x) \rightarrow \neg Q(x))$$

(注意，我们假定了“小”就是“不大”，“色彩单调”就是“不五彩斑斓”。为证明第4句是前3句的有效结论，需要用到将在1.5节讨论的推理规则。)

1.3.13 逻辑程序设计

有一类重要的程序设计语言，使用谓词逻辑的规则进行推理。其中之一就是 Prolog (Programming in Logic 的缩写)，这种语言开发于 20 世纪 70 年代，应用于人工智能领域。Prolog 程序包括一组声明，其中包括两类语句：**Prolog 事实**和**Prolog 规则**。Prolog 事实定义谓词，这是通过指定满足这些谓词的元素来实现的。Prolog 规则使用那些 Prolog 事实定义好的谓词来定义新的谓词。例 28 演示这些概念。

例 28 考虑一个 Prolog 程序，给出的事实是每门课程的教师和学生注册的课程。程序使用这些事实回答与教师授课情况相关的查询。这样的程序可使用谓词 $\text{instructor}(p, c)$ 和 $\text{enrolled}(s, c)$ 分别表示教授 p 是讲授课程 c 的老师及学生 s 注册课程 c 。例如，此程序中的 Prolog 事实可能包含：

```
instructor(chan, math273)
instructor(patel, ee222)
instructor(grossman, cs301)
enrolled(kevin, math273)
enrolled(juana, ee222)
enrolled(juana, cs301)
enrolled(kiko, math273)
enrolled(kiko, cs301)
```

(用小写字母表示项目是因为 Prolog 把以大写字母开始的名字当做变量。)

一个新的谓词 $\text{teaches}(p, s)$ 表示教授 p 教学生 s ，可以用 Prolog 规则来定义：

```
teaches(P, S) :- instructor(P, C), enrolled(S, C)
```

上述语句意味着如果教授 p 是课程 c 的老师，学生 s 注册课程 c ， $\text{teaches}(p, s)$ 为真。(注意，在 Prolog 中逗号用于表示谓词合取。类似地，分号表示谓词析取。)

Prolog 使用给定的事实和规则回答查询。例如，使用上述的事实和规则，查询

```
? enrolled(kevin, math273)
```

生成应答

```
yes
```

因为事实 $\text{enrolled}(\text{kevin}, \text{math273})$ 是一个输入。查询

```
? enrolled(X, math273)
```

生成应答

```
kevin
kiko
```

要生成上面的应答，Prolog 确定 $\text{enrolled}(X, \text{math273})$ 包含的作为 Prolog 事实的 X 所有可能的值。类似地，要找到 Juana 所选课程的教师，可以用查询

```
? teaches(X, juana)
```

这个查询返回

```
patel
grossman
```

练习

- 令 $P(x)$ 表示语句“ $x \leq 4$ ”。下列各项的真值是什么?
a) $P(0)$ b) $P(4)$ c) $P(6)$
- 令 $Q(x, y)$ 表示语句“ x 是 y 的首府。”下列各项的真值是什么?
a) $Q(\text{丹佛, 科罗拉多})$ b) $Q(\text{底特律, 密歇根})$ c) $Q(\text{马萨诸塞, 波士顿})$ d) $Q(\text{纽约, 纽约})$
- 令 $P(x)$ 为语句“ x 每个工作日都花 5 个多小时上课”, 其中 x 的论域是学生集合。用句子表达下列各量化语句。
a) $\exists x P(x)$ b) $\forall x P(x)$ c) $\exists x \neg P(x)$ d) $\forall x \neg P(x)$
- 将下列语句翻译成汉语语句, 其中 $C(x)$ 是“ x 是一个喜剧演员”, $F(x)$ 是“ x 很有趣”, 论域是所有人。
a) $\forall x (C(x) \rightarrow F(x))$ b) $\forall x (C(x) \wedge F(x))$ c) $\exists x (C(x) \rightarrow F(x))$ d) $\exists x (C(x) \wedge F(x))$
- 令 $P(x)$ 为语句“ x 说俄语”, $Q(x)$ 为语句“ x 了解计算机语言 C++”。用 $P(x)$ 、 $Q(x)$ 、量词和逻辑联接词表示下列各句子, 量词的论域为你校全体学生的集合。
a) 你校有个学生既会说俄语又了解 C++。 b) 你校有个学生会说俄语, 但不了解 C++。
c) 你校所有学生或会说俄语, 或了解 C++。 d) 你校没有学生会说俄语或了解 C++。
- 令 $P(x)$ 为语句“ $x = x^2$ ”。如果论域是整数集合, 下列各项的真值是什么?
a) $P(0)$ b) $P(1)$ c) $P(2)$
d) $P(-1)$ e) $\exists x P(x)$ f) $\forall x P(x)$
- 如果论域为整数集合, 下列各语句的真值是什么?
a) $\forall n (n+1 > n)$ b) $\exists n (2n = 3n)$ c) $\exists n (n = -n)$ d) $\forall n (n^2 \geq n)$
- 如果所有变量的论域为整数集合, 下列各语句的真值是什么?
a) $\forall n (n^2 \geq 0)$ b) $\exists n (n^2 = 2)$ c) $\forall n (n^2 \geq n)$ d) $\exists n (n^2 < 0)$
- 假设命题函数 $P(x)$ 的论域为整数 0, 1, 2, 3 和 4。使用析取、合取和否定写出下列命题。
a) $\exists x P(x)$ b) $\forall x P(x)$ c) $\exists x \neg P(x)$
d) $\forall x \neg P(x)$ e) $\neg \exists x P(x)$ f) $\neg \forall x P(x)$
- 假设命题函数 $P(x)$ 的论域为整数 1, 2, 3, 4 和 5。使用析取、合取和否定(而不使用量词)表达下列语句。
a) $\exists x P(x)$ b) $\forall x P(x)$ c) $\neg \exists x P(x)$ d) $\neg \forall x P(x)$
e) $\forall x ((x \neq 3) \rightarrow P(x)) \vee \exists x \neg P(x)$
- 给下列语句分别为真和假的相应论域。
a) 每一个人都学离散数学。 b) 每一个人都超过 21 岁。
c) 每两个人都有相同的妈妈。 d) 没有两个不同的人有相同的祖母。
- 使用谓词、量词和逻辑联接词, 以两种方式将下列语句翻译成逻辑表达式。首先, 令论域为班上的学生, 其次, 令论域为所有人。
a) 班上有人会说印地语。 b) 班上的每个人都很好。
c) 班上有个学生不是出生在加利福尼亚。 d) 班上有个学生曾演过电影。
e) 班上没有学生上过逻辑编程课程。
- 使用谓词、量词和逻辑联接词, 将下列语句翻译成逻辑表达式。
a) 人无完人。 b) 不是每个人都很完美。
c) 你的所有朋友都很完美。 d) 你的某个朋友很完美。
e) 每个人都是你的朋友并且很完美。 f) 不是每个人都是你的朋友或有人并不完美。
- 使用谓词(带有一个或两个变量), 通过改变论域, 以三种不同的方式将下列语句翻译成逻辑表达式。
a) 学校里的某个学生曾在越南居住过。 b) 学校里的有个学生不会说印地语。
c) 学校里的某个学生会用 Java、Prolog 和 C++。 d) 班上的每个学生都喜欢泰国食物。
e) 班上的某个学生不打曲棍球。
- 使用逻辑运算符、谓词和量词来表达下列语句。
a) 某些命题是永真式。 b) 矛盾的否定是永真式。
c) 两个可能式的析取可以是一个永真式。 d) 两个永真式的合取是一个永真式。

16. 假定 $Q(x, y, z)$ 的论域由 x, y 和 z 的三元组组成, 其中 $x=0, 1$ 或 $2, y=0$ 或 $1, z=0$ 或 1 。用析取和合取写出下列命题。
- a) $\forall yQ(0, y, 0)$ b) $\exists xQ(x, 1, 1)$ c) $\exists x\neg Q(0, 0, z)$ d) $\exists x\neg Q(x, 0, 1)$
17. 用量词表达下列语句, 然后取该语句的否定并使否定词不在量词的左边。再用简单语句表达这否定(不要简单地表达为“不是……”)。
- a) 一些年长的狗会学习新的技巧。 b) 没有会微积分的兔子。
c) 每只鸟都会飞。 d) 没有会说话的狗。
e) 班上没有人会法语和俄语。
18. 找出下列全称量化语句的反例(如果可能的话), 其中所有变量的论域是整数集合。
- a) $\forall x(x^2 \geq x)$ b) $\forall x(x > 0 \vee x < 0)$ c) $\forall x(x = 1)$
19. 用谓词和量词表达下列语句。
- a) 一名飞机乘客如果在一年中飞行的里程超过 25 000 英里, 或在那一年坐飞机的次数超过 25 次, 就表明他是高贵乘客。
b) 一名男选手, 若他的马拉松比赛过去最好成绩在 3 小时内, 他就有资格参赛。一名女选手, 若她的过去最好成绩在 3.5 小时内, 她就有资格参赛。
c) 一名学生要想取得硕士学位, 必须至少修满 60 个学分, 或至少修满 45 个学分并通过硕士论文答辩, 并且所有必修课程的成绩不低于 B。
d) 有某个学生在一个学期内修了 21 个学分并且全部课程的成绩都为 A。
- 练习 20~21 主要进行系统规范说明和逻辑表达式之间的翻译, 当中涉及量词。
20. 将下列规范说明翻译成语句, 其中 $F(p)$ 是“打印机 p 不能提供服务”, $B(p)$ 是“打印机 p 很忙”, $L(j)$ 是“打印作业 j 丢失了”, $Q(j)$ 是“打印作业 j 在队列中”。
- a) $\exists p(F(p) \wedge B(p)) \rightarrow \exists jL(j)$ b) $\forall pB(p) \rightarrow \exists jQ(j)$
c) $\exists j(Q(j) \wedge L(j)) \rightarrow \exists pF(p)$ d) $(\forall pB(p) \wedge \forall jQ(j)) \rightarrow \exists jL(j)$
21. 使用谓词、量词和逻辑联接词表达下列系统规范说明。
- a) 如果磁盘有 10MB 以上的空闲空间, 那么至少能保存一条邮件消息。
b) 每当有主动报警时, 所有排队的消息都被传送。
c) 诊断监控器跟踪所有系统的状态, 除了主控制台。
d) 对参与电话会议的每一方, 不在特殊列表上的主叫方应当付费。
22. 判断 $\forall x(P(x) \rightarrow Q(x))$ 和 $\forall xP(x) \rightarrow \forall xQ(x)$ 是否逻辑等价, 并证明。
23. 证明 $\exists x(P(x) \vee Q(x))$ 和 $\exists xP(x) \vee \exists xQ(x)$ 是逻辑等价的。

练习 24~25 给出了空量词规则, 当量化变元不出现在语句部分的时候就能用。

24. 证明下列逻辑等价关系, 其中 x 在 A 中不作为自由变元出现。假设论域非空。
- a) $(\forall xP(x)) \wedge A \equiv \forall x(P(x) \wedge A)$ b) $(\exists xP(x)) \wedge A \equiv \exists x(P(x) \wedge A)$
25. 证明下列逻辑等价关系, 其中 x 在 A 中不作为自由变元出现。假设论域非空。
- a) $\forall x(P(x) \rightarrow A) \equiv \exists xP(x) \rightarrow A$ b) $\exists x(P(x) \rightarrow A) \equiv \forall xP(x) \rightarrow A$
26. 证明 $\exists xP(x) \wedge \exists xQ(x)$ 和 $\exists x(P(x) \wedge Q(x))$ 不是逻辑等价的。
27. 下列语句的真值是什么?
- a) $\exists! xP(x) \rightarrow \exists xP(x)$ b) $\forall xP(x) \rightarrow \exists! xP(x)$ c) $\exists! x\neg P(x) \rightarrow \neg \forall xP(x)$
28. 给定例 28 的 Prolog 事实, 对下列查询 Prolog 返回的是什么?
- a)? instructor(chan, math273) b)? instructor(patel, cs301)
c)? enrolled(X, cs301) d)? enrolled(kiko, Y)
e)? teaches(grossman, Y)
29. 假定 Prolog 事实用于定义谓词 mother(M, Y) 和 father(F, X), 这表示 M 是 Y 的母亲, F 是 X 的父亲。给出一个 Prolog 规则定义谓词 sibling(X, Y), 表示 X 和 Y 是兄弟(也就是, 有相同的父亲和母亲)。

练习 30~31 是根据 Lewis Carroll 的《Symbolic Logic》一书中的问题编写的。

30. 令 $P(x)$ 、 $Q(x)$ 和 $R(x)$ 分别为语句“ x 是教授”、“ x 无知”和“ x 爱虚荣”。用量词、逻辑联接词和 $P(x)$ 、

$Q(x)$ 、 $R(x)$ 表达下列语句。假定论域是所有人的集合。

a) 没有无知的教授。

b) 所有无知者均受虚荣。

c) 没有受虚荣的教授。

d) 能从 a) 和 b) 推出 c) 吗？若不能，有没有一个正确的结论？

31. 令 $P(x)$ 、 $Q(x)$ 、 $R(x)$ 和 $S(x)$ 分别为语句“ x 是婴儿”、“ x 的行为符合逻辑”、“ x 能管理鳄鱼”和“ x 被人轻视”。假定 x 的论域是所有人的集合。用量词、逻辑联接词和 $P(x)$ 、 $Q(x)$ 、 $R(x)$ 和 $S(x)$ 表达下列语句。

a) 婴儿行为不合逻辑。

b) 能管理鳄鱼的人不被人轻视。

c) 行为不合逻辑的人被人轻视。

d) 婴儿不能管理鳄鱼。

*e) 能从 a)、b) 和 c) 推出 d) 吗？若不能，有没有一个正确的结论？

1.4 嵌套量词

1.4.1 引言

在 1.3 节定义了存在量词和全称量词，并展示了如何用它们来表示数学语句。我们也解释了如何用它们来将汉语语句翻译成逻辑表达式。本节我们将学习嵌套量词，它是出现在其他量词的作用域内的量词，如在语句 $\forall x \exists y (x+y=0)$ 中。注意量词范围内的所有事物都可以被认为是命题函数。比如， $\forall x \exists y (x+y=0)$ 和 $\forall x Q(x)$ 相同，其中 $Q(x)$ 指 $\exists y P(x, y)$ ，而 $P(x, y)$ 表示 $x+y=0$ 。嵌套量词通常在数学和计算机科学中出现。尽管嵌套量词有时比较难理解，但在 1.3 节介绍过的规则却有助于我们使用它们。

为理解这些涉及许多量词的语句，我们需要阐明出现的量词和谓词的含义。具体如例 1 和例 2 所示。

例 1 假定变量 x 和 y 的论域是所有实数的集合，语句

Extra
Example

$$\forall x \forall y (x+y = y+x)$$

说的是对所有实数 x 和 y ， $x+y=y+x$ 。这是实数加法的交换律。同样，语句

$$\forall x \exists y (x+y = 0)$$

说的是对所有实数 x ，有一个实数 y ，使得 $x+y=0$ 。也就是每个实数都有一个加法的逆。同样，语句

$$\forall x \forall y \forall z (x+(y+z) = (x+y)+z)$$

是实数加法的结合律。

例 2 将下列语句翻译成汉语句子

$$\forall x \forall y ((x > 0) \wedge (y < 0) \rightarrow (xy < 0))$$

其中变量 x 和 y 的论域都是全体实数。

解 这个语句表示对任意实数 x 和 y ，如果 $x > 0$ 且 $y < 0$ ，那么 $xy < 0$ 。也就是说，这个语句表示对实数 x 和 y ，如果 x 是正的且 y 是负的，那么 xy 就是负的。这可以更简洁地叙述为“一个正实数与一个负实数的积是一个负实数”。

循环量化考虑 对多个变量使用量词时，借助嵌套循环来思考是有益的。（当然，如果某个变量的论域有无穷多个元素，是无法真正对所有值做循环的。不过这种考虑方式对理解嵌套量词总是有益的。）例如，要决定 $\forall x \forall y P(x, y)$ 是否为真，我们先对 x 的所有值做循环，而对 x 的每个值再对 y 的所有值循环。如果我们发现对 x 和 y 的所有值 $P(x, y)$ 都为真，那么我们就判定了 $\forall x \forall y P(x, y)$ 为真。只要我们碰上一个 x 值，对这个值又有一个 y 值使 $P(x, y)$ 为假，那么就证明了 $\forall x \forall y P(x, y)$ 为假。

同样，要判定 $\forall x \exists y P(x, y)$ 是否为真，我们对 x 的所有值循环，对 x 的每个值，对 y 的值循环直到找到一个 y 使 $P(x, y)$ 为真。如果对 x 的所有值，我们都能碰上这样的 y 值，那么

$\forall x \exists y P(x, y)$ 为真。如果对某个 x 我们碰不上这样的 y , 那么 $\forall x \exists y P(x, y)$ 就为假。

要决定 $\exists x \forall y P(x, y)$ 是否为真, 需要对 x 的值循环直到找到某个 x , 就这个 x 对 y 的所有值循环时 $P(x, y)$ 总是成真。如果能找到这样的 x , $\exists x \forall y P(x, y)$ 就为真。如果总也碰不上这样的 x , 那么我们知道 $\exists x \forall y P(x, y)$ 为假。

最后要看 $\exists x \exists y P(x, y)$ 是否为真。我们对 x 的值循环, 循环时对 x 的每个值都对 y 的值循环, 直到找到 x 的一个值和 y 的一个值使 $P(x, y)$ 为真。只有当我们永远碰不上这样的 x 和 y 能使 $P(x, y)$ 成真时, 语句 $\exists x \exists y P(x, y)$ 才为假。

1.4.2 量词的顺序

许多数学语句需要对多变量命题函数作多重量化。除非所有量词均为全称量词或均为存在量词, 否则量词的顺序是重要的。例3~5说明了这一点。

Example 例3 令 $P(x, y)$ 为语句“ $x+y=y+x$ ”, 量化语句 $\forall x \forall y P(x, y)$ 和 $\forall y \forall x P(x, y)$ 的真值是什么? 这里所有变元的论域是全体实数。

解 量化语句

$$\forall x \forall y P(x, y)$$

表示的命题是

“对所有实数 x 和所有实数 $y, x+y=y+x$ 成立。”

因为 $P(x, y)$ 对所有实数 x 和 y 都为真, $\forall x \forall y P(x, y)$ 为真。注意语句 $\forall x \forall y P(x, y)$ 表示“对所有实数 y , 对所有实数 $x, x+y=y+x$ 。”这句话的意思和“对所有实数 x , 对所有实数 $y, x+y=y+x$ 意义相同。”也就是说, $\forall x \forall y P(x, y)$ 和 $\forall y \forall x P(x, y)$ 意义相同, 都为真。这说明在没有其他量词的语句中, 在不改变量化语句意义的前提下嵌套全称量词的顺序原则上可以改变。 ■

例4 令 $Q(x, y)$ 表示“ $x+y=0$ ”, 量化语句 $\exists y \forall x Q(x, y)$ 和 $\forall x \exists y Q(x, y)$ 的真值是什么? 这里所有变元的论域是全体实数。

解 量化语句

$$\exists y \forall x Q(x, y)$$

表示的命题是

“有个实数 y 能使 $Q(x, y)$ 对每一个实数 x 成立。”

不管 y 取什么值, 只有一个 x 的值能使 $x+y=0$ 成立。因为没有实数 y 能使 $x+y=0$ 对所有实数 x 成立, 语句 $\exists y \forall x Q(x, y)$ 为假。

量化语句

$$\forall x \exists y Q(x, y)$$

表示的命题是

“对每个实数 x 都有一个实数 y 使 $Q(x, y)$ 成立。”

给定一个实数 x , 总有一个实数 y 能使 $x+y=0$, 这个实数就是 $y=-x$ 。因此, 语句 $\forall x \exists y Q(x, y)$ 为真。 ■

例4说明量词出现的顺序不同含义也不同。语句 $\exists y \forall x P(x, y)$ 和 $\forall x \exists y P(x, y)$ 不是逻辑等价的, 语句 $\exists y \forall x P(x, y)$ 为真当且仅当存在一个 y , 使得 $P(x, y)$ 对每个 x 都成立。因此, 要使这一语句为真, 必须有一个特定的 y 值, 对这个值无论是什么样的 x , $P(x, y)$ 都成立; 另一方面, $\forall x \exists y P(x, y)$ 为真当且仅当对 x 的每一个值都有一个 y 的值使 $P(x, y)$ 成立。所以, 要使这个语句为真, 不管你选什么 x , 总有 y 的一个值(也许依赖于你选的 x)使 $P(x, y)$ 成立。换言之, 在第二种情况下, y 随着 x 而定, 但在第一种情况下, y 是与 x 无关的常数。

从这些观察可以看出, 如果 $\exists y \forall x P(x, y)$ 为真, 则 $\forall x \exists y P(x, y)$ 必定也为真; 可是如果 $\forall x \exists y P(x, y)$ 为真, $\exists y \forall x P(x, y)$ 不一定为真(参见本章补充练习24和25)。

表1-24是两个变量所有不同的可能量化方式的总结。

表 1-24 两个变量的量化

语 句	何时为真	何时为假
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	对每一对 x, y , $P(x, y)$ 均为真	有一对 x, y 使 $P(x, y)$ 为假
$\forall x \exists y P(x, y)$	对每个 x , 都有 y 使 $P(x, y)$ 为真	有 x , 使 $P(x, y)$ 对每个 y 总是假
$\exists x \forall y P(x, y)$	有一个 x , 使 $P(x, y)$ 对所有 y 均为真	对每个 x 都有 y 使 $P(x, y)$ 为假
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	有一对 x, y 使 $P(x, y)$ 为真	对每一对 x, y , $P(x, y)$ 均为假

超过两个变量的量词也很常见, 如例 5 所示。

例 5 令 $Q(x, y, z)$ 为语句“ $x+y=z$ ”, 语句 $\forall x \forall y \exists z Q(x, y, z)$ 和 $\exists z \forall x \forall y Q(x, y, z)$ 的真值是什么?

解 假定给 x 和 y 赋了值, 那么就有一个实数 z , 使得 $x+y=z$ 。于是量化语句

$$\forall x \forall y \exists z Q(x, y, z)$$

为真, 因为它其实就是语句

“对所有实数 x 和所有实数 y , 有实数 z , 使得 $x+y=z$ 。”

这里量词出现的顺序是重要的, 因为量化语句

$$\exists z \forall x \forall y Q(x, y, z)$$

也就是语句

“有实数 z 使得对所有实数 x 和所有实数 y , $x+y=z$ 。”

为假, 因为没有 z 的值能使 $x+y=z$ 对 x 和 y 的所有值都成立。 ■

1.4.3 将数学语句翻译成涉及嵌套量词的语句

用汉语表达的数学语句可以被翻译成逻辑表达式, 如例 6~8 所示。

例 6 将语句“两个正整数的和是正数”翻译成逻辑表达式。

解 要将这个语句翻译成逻辑表达式, 我们首先重写它以示出隐含的量词和论域: “对每两个整数, 如果它们都是正的, 那么它们的和是正数”。然后, 引入变量 x 和 y 就得到“对所有正整数 x 和 y , $x+y$ 是正数”。因此, 可以将这个语句表达为

$$\forall x \forall y ((x > 0) \wedge (y > 0) \rightarrow (x + y > 0))$$

其中这两个变量的论域是全体整数。注意, 也可以将正整数作为论域。那么语句“两个正整数的和是正整数”变为“对每两个正整数, 它们的和为正”。可以将它表达为

$$\forall x \forall y (x + y > 0)$$

其中两个变量的论域为全体正整数。 ■

例 7 将语句“每个实数(除了 0)都有乘法逆元”(实数 x 的乘法逆元是使 $xy=1$ 的实数 y) 翻译成逻辑表达式。

解 我们首先重写这个语句为“对每个实数 x (除了 0), x 有乘法逆元”, 然后可以再将之重写为“对每个实数 x , 如果 $x \neq 0$, 那么存在一个实数 y 使得 $xy=1$ ”。这可以重写为

$$\forall x ((x \neq 0) \rightarrow \exists y (xy = 1))$$

正如前面提到的, 量词常用于数学概念的定义, 你可能熟悉的极限概念即为一例。极限是微积分中的重要概念。

例 8 (需要微积分知识)用量词表示极限的定义。

解 回顾语句 $\lim_{x \rightarrow a} f(x) = L$ 的定义: 对每个实数 $\epsilon > 0$, 存在一个实数 $\delta > 0$, 使得对每一个 x , 只要 $0 < |x - a| < \delta$, 就有 $|f(x) - L| < \epsilon$ 。极限的这一定义用量词表示为

$$\forall \epsilon \exists \delta \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon)$$

其中 ϵ 和 δ 的论域是正实数集合, x 的论域是实数集合。

这一定义还可表示为

$$\forall \epsilon > 0 \exists \delta > 0 \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon)$$

其中 ϵ 和 δ 的论域为实数集合, 而不是正实数集合。(这里用到了约束量词。回忆 $\forall x > 0 P(x)$ 的意义是对所有 $x > 0$ 的数, $P(x)$ 为真。)

1.4.4 将嵌套量词翻译为汉语

表达汉语语句的带嵌套量词的表达式可以相当复杂。在翻译这样的表达式时, 第一步是写出表达式中的量词和谓词的含义, 第二步是用简单的句子来表达这个含义。例 9 和例 10 说明了这个过程。

例 9 把语句

$$\forall x(C(x) \vee \exists y(C(y) \wedge F(x, y)))$$

翻译成汉语。其中 $C(x)$ 是“ x 有台计算机”, $F(x, y)$ 是“ x 和 y 是朋友”, 而 x 和 y 的共同论域是学校全体学生的集合。

解 该语句说的是, 对学校的每个学生 x , 或者 x 有台计算机, 或者另有学生 y , 他有台计算机, 且 x 和 y 是朋友。换言之, 学校的每个学生或有计算机, 或有个有计算机的朋友。

例 10 把语句

$$\exists x \forall y \forall z ((F(x, y) \wedge F(x, z) \wedge (y \neq z)) \rightarrow \neg F(y, z))$$

翻译成汉语, 其中 $F(a, b)$ 的含义是 a 和 b 是朋友, 而 x 、 y 和 z 的论域是学校所有学生的集合。

解 这个语句说的是, 有一个学生 x , 对所有的学生 y 及不同于 y 的所有学生 z , 只要 x 和 y 是朋友, x 和 z 也是朋友, 那么 y 和 z 就不是朋友。换句话说, 有个学生, 他的朋友之间都不是朋友。

1.4.5 将汉语语句翻译成逻辑表达式

在 1.3 节展示了如何用量词来将句子翻译成逻辑表达式。然而, 当时没有谈及这样的句子, 即将其翻译成逻辑表达式要求使用嵌套量词。我们现在介绍如何翻译这样的句子。

例 11 将语句“如果某人是女性且为人父母, 这个人某人的母亲”翻译成逻辑表达式, 当中涉及谓词、量词(论域是所有人)以及逻辑联接词。

解 语句“如果某人是女性且为人父母, 这个人某人的母亲”可以表达为“对每个人 x , 如果 x 是女性且 x 是父母, 那么存在一个人 y 使 x 是 y 的母亲”。我们引入谓词 $F(x)$ 来表示“ x 是女性”, $P(x)$ 表示“ x 是父母”, $M(x, y)$ 表示“ x 是 y 的母亲”。则初始语句可以表示为

$$\forall x ((F(x) \wedge P(x)) \rightarrow \exists y M(x, y))$$

在 1.3 节练习 24 的 (b) 部分用空间量词规则, 可以把 $\exists y$ 移到左边, 以致它恰好出现在 $\forall x$ 之后, 因为 y 不在 $F(x) \wedge P(x)$ 中出现。这样得到一个等价的表达式

$$\forall x \exists y ((F(x) \wedge P(x)) \rightarrow M(x, y))$$

例 12 将语句“每个人都恰有一个最好的朋友”翻译成逻辑表达式, 当中涉及谓词、量词(论域是所有人)以及逻辑联接词。

解 语句“每个人都恰有一个最好的朋友”可以表达为“对每个人 x , x 恰有一个最好的朋友”。引入全称量词, 可以看到这个语句等价于“ $\forall x(x$ 恰有一个最好的朋友)”, 其中论域是所有人。

x 恰有一个最好的朋友意味着有一个人 y 是 x 的最好的朋友, 而且, 对每个人 z , 如果 z 不是 y , 那么 z 不是 x 的最好的朋友。当我们引入谓词 $B(x, y)$ 为语句“ y 是 x 的最好的朋友”, 则语句“ x 恰有一个最好的朋友”可以表示为

$$\exists y (B(x, y) \wedge \forall z ((z \neq y) \rightarrow \neg B(x, z)))$$

因此, 初始语句可以表示为

$$\forall x \exists y (B(x, y) \wedge \forall z ((z \neq y) \rightarrow \neg B(x, z)))$$

注意, 我们可以把这个语句写为 $\forall x \exists ! y B(x, y)$, 这里 $\exists !$ 是 1.3.4 节定义的存在唯一量词。

例 13 用量词表示语句“有位妇女已搭乘过世界上每一条航线上的航班”。

解 令 $P(w, f)$ 为“ w 搭乘过 f ”, 而 $Q(f, a)$ 为“ f 是 a 上的航班”。于是可将上述语句表示为

$$\exists w \forall a \exists f (P(w, f) \wedge Q(f, a))$$

其中, w 、 f 和 a 的论域分别为世界上所有妇女、所有空中航班和所有航线。

这个语句也可以表示为

$$\exists w \forall a \exists f R(w, f, a)$$

其中 $R(w, f, a)$ 为“ w 已搭乘过 a 上的 f ”。虽然这样表示更紧凑, 但它使变量之间的关系有点含糊不清, 因此, 第一个解要好些。

1.4.6 否定嵌套量词

Assignment: 带嵌套量词的语句可以通过连续地应用否定带单个量词的语句的规则成为否定的。具体如例 14~16 所示。

例 14 表达语句 $\forall x \exists y (xy=1)$ 的否定, 使得量词前面没有否定词。

Extra Examples: **解** 通过连续地应用否定量化语句的规则(见 1.3 节表 1-23), 可以将 $\neg \forall x \exists y (xy=1)$ 中的否定词移入所有量词里面。我们发现, $\neg \forall x \exists y (xy=1)$ 等价于 $\exists x \neg \exists y (xy=1)$, 而后者又等价于 $\exists x \forall y \neg (xy=1)$ 。由于 $\neg (xy=1)$ 可以简化为 $xy \neq 1$, 所以, 我们的否定语句可以表达为 $\exists x \forall y (xy \neq 1)$ 。

例 15 使用量词表达语句“没有一个妇女已搭乘过世界上每一条航线上的航班”。

解 这个语句是例 13 的语句“有位妇女已搭乘过世界上每一条航线上的航班”的否定。由例 13, 我们的语句可以表达为 $\neg \exists w \forall a \exists f (P(w, f) \wedge Q(f, a))$, 其中 $P(w, f)$ 是“ w 搭乘过 f ”, 而 $Q(f, a)$ 为“ f 是 a 上的航班”。通过连续地应用否定量化语句的规则(见 1.3 节表 1-23)并在最后一步应用德摩根律, 可发现我们的语句等价于如下语句:

$$\begin{aligned} \neg \exists w \neg \forall a \neg \exists f (P(w, f) \wedge Q(f, a)) &\equiv \forall w \exists a \neg \exists f (P(w, f) \wedge Q(f, a)) \\ &\equiv \forall w \exists a \forall f \neg (P(w, f) \wedge Q(f, a)) \\ &\equiv \forall w \exists a \forall f (\neg P(w, f) \vee \neg Q(f, a)) \end{aligned}$$

最后这条语句表示“对于每位妇女, 存在一条航线, 使得对所有的航班, 这位妇女要么没有搭乘过航班, 要么没有搭乘过不在这条航线的航班”。

例 16 (需要微积分知识)使用量词和谓词表达 $\lim_{x \rightarrow a} f(x)$ 不存在这一事实。

解 $\lim_{x \rightarrow a} f(x)$ 不存在意味着对全体实数 L , $\lim_{x \rightarrow a} f(x) \neq L$ 。根据例 8, $\lim_{x \rightarrow a} f(x) \neq L$ 可以表达为

$$\neg \forall \epsilon > 0 \exists \delta > 0 \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon)$$

连续地应用否定量化表达式的规则, 可得到如下等价语句序列:

$$\begin{aligned} \neg \forall \epsilon > 0 \exists \delta > 0 \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon) \\ &\equiv \exists \epsilon > 0 \neg \exists \delta > 0 \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon) \\ &\equiv \exists \epsilon > 0 \forall \delta > 0 \neg \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon) \\ &\equiv \exists \epsilon > 0 \forall \delta > 0 \exists x \neg (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon) \\ &\equiv \exists \epsilon > 0 \forall \delta > 0 \exists x (0 < |x - a| < \delta \wedge |f(x) - L| \geq \epsilon) \end{aligned}$$

在最后一步使用等价式 $\neg(p \rightarrow q) \equiv p \wedge \neg q$ 。这里依据 1.2 节表 1-16 的第 5 个等式。

由于“ $\lim_{x \rightarrow a} f(x)$ 不存在”意味着对全体实数 L , $\lim_{x \rightarrow a} f(x) \neq L$ 。这个语句可以表达为

$$\forall L \exists \epsilon > 0 \forall \delta > 0 \exists x (0 < |x - a| < \delta \wedge |f(x) - L| \geq \epsilon)$$

这最后一个语句表示, 对每个实数 L , 存在实数 $\epsilon > 0$ 使得对每个实数 $\delta > 0$, 存在实数 x 使得 $0 < |x - a| < \delta$ 和 $|f(x) - L| \geq \epsilon$ 。

练习

1. 将下列语句翻译成汉语句子, 其中每个变量的论域是全体实数。

- a) $\forall x \exists y (x < y)$ b) $\forall x \forall y ((x \geq 0) \wedge (y \geq 0)) \rightarrow (xy \geq 0)$
 c) $\forall x \forall y \exists z (xy = z)$
2. 令 $Q(x, y)$ 是语句“ x 已经发送电子邮件消息给 y ”, 其中 x 和 y 的论域都是班上的所有学生, 将下列量化式子表达成汉语句子。
 a) $\exists x \exists y Q(x, y)$ b) $\exists x \forall y Q(x, y)$ c) $\forall x \exists y Q(x, y)$
 d) $\exists y \forall x Q(x, y)$ e) $\forall y \exists x Q(x, y)$ f) $\forall x \forall y Q(x, y)$
3. 令 $W(x, y)$ 表示“ x 访问过 y ”, 其中 x 的论域是你校全体学生集合, y 的论域是所有网站的集合。用简单的句子表达下列语句。
 a) $W(\text{Sarah Smith}, \text{www.att.com})$
 b) $\exists x W(x, \text{www.imdb.org})$
 c) $\exists y W(\text{Jose Orez}, y)$
 d) $\exists y (W(\text{Ashok Puri}, y) \wedge W(\text{Cindy Yoon}, y))$
 e) $\exists y \forall z (y \neq (\text{David Belcher}) \wedge (W(\text{David Belcher}, z) \rightarrow W(y, z)))$
 f) $\exists x \exists y \forall z ((x \neq y) \wedge (W(x, z) \leftrightarrow W(y, z)))$
4. 令 $T(x, y)$ 表示学生 x 喜欢菜肴 y , 其中 x 的论域是学校的所有学生, y 的论域是所有菜肴。用简单的汉语句子表达下列语句。
 a) $\neg T(\text{Abdallah Hussein}, \text{Japanese})$ b) $\exists x T(x, \text{Korean}) \wedge \forall x T(x, \text{Mexican})$
 c) $\exists y (T(\text{Monique Arsenault}, y) \vee T(\text{Jay Johnson}, y))$ d) $\forall x \forall z \exists y ((x \neq z) \rightarrow \neg (T(x, y) \wedge T(z, y)))$
 e) $\exists x \exists z \forall y (T(x, y) \leftrightarrow T(z, y))$ f) $\forall x \forall z \exists y (T(x, y) \leftrightarrow T(z, y))$
5. 令 $L(x, y)$ 为语句“ x 爱 y ”, 其中 x 和 y 的论域都是全世界所有人的集合。用量词表达下列各语句。
 a) 每个人都爱 Jerry. b) 每个人都爱某个人. c) 有个人人都爱的人。
 d) 没有人爱所有的人. e) 有个 Lydia 不爱的人. f) 有个人人都不爱的人。
 g) 恰有一个人人都爱的人. h) Lynn 爱的人恰有两个. i) 每个人都爱自己。
 j) 有人除自己以外谁都不爱。
6. 令 $S(x)$ 为谓词“ x 是学生”, $F(x)$ 为谓词“ x 是教员”, 而 $A(x, y)$ 是谓词“ x 问过 y 问题”, 其中论域是你校所有人员的集合。用量词表达下列语句。
 a) Lois 问过 Michaels 教授问题. b) 每个学生都问过 Gross 教授问题。
 c) 每位教员都问过 Miller 教授问题或被 Miller 教授问过问题. d) 某个学生从未问过任何教员问题。
 e) 有位教员从未被学生问过问题. f) 有个学生问过所有教员问题。
 g) 有位教员问过所有其他教员问题. h) 有学生从未被教员问过问题。
7. 令 $M(x, y)$ 为“ x 已发给 y 电子邮件”, $T(x, y)$ 为“ x 给 y 打过电话”, 其中论域为你们班上所有学生。用量词表达下列语句。(假定所有被发出的电子邮件都能收到, 尽管事实并非如此。)
 a) Chou 从未给 Koko 发过电子邮件。
 b) Arlene 从未给 Sarah 发过电子邮件, 或未打过电话。
 c) Jose 从未收到过 Deborah 的电子邮件。
 d) 班上每个学生都给 Ken 发过电子邮件。
 e) 班上没有人给 Nina 打过电话。
 f) 班上每个人或给 Avi 打过电话或给他发过电子邮件。
 g) 班上有个学生给班上其他人都发过电子邮件。
 h) 班上有人给班上其他人或打过电话, 或发过电子邮件。
 i) 班上有两个学生互发过电子邮件。
 j) 班上有个学生给自己发过电子邮件。
 k) 班上有个学生既没从班上其他人那里收到过电子邮件, 也没有人给他打过电话。
 l) 班上每个学生都或从同班同学那里收到过电子邮件, 或接到过同班同学的电话。
 m) 班上至少有两个学生, 一个给另一个发过电子邮件, 第二个则给第一个打过电话。
 n) 班上有两个同学, 他们当中有一个给班上其余同学或发过电子邮件, 或打过电话。
8. 用量词和谓词表达下列语句。
 a) 每位学计算机科学的学生都需要上一门离散数学课。

- b) 班上有个学生有一台个人电脑。
 c) 班上每个学生至少选修了一门计算机科学课。
 d) 班上有个学生至少选修过一门计算机科学课。
 e) 班上每个学生都去过校园里每座建筑。
 f) 班上有个学生去过校园里至少一座楼的每个房间。
 g) 班上每个学生都去过校园里每座楼的至少一个房间。
9. 使用谓词、量词和逻辑联接词(如果有必要)表达下列系统规范说明。
 a) 每个用户恰能访问一个邮箱。
 b) 在所有错误条件下某个进程能继续运行, 仅当内核运行正确。
 c) 校园网的所有用户都能访问具有 .edu 后缀的 URL 的所有站点。
 *d) 恰有两个系统监控着每一个远程服务器。
10. 使用数学和逻辑运算符、谓词及量词表达下列语句, 论域是全体整数。
 a) 两个负整数的和是负数。
 b) 两个正整数的差不一定是正数。
 c) 两个整数的平方和大于等于它们的和的平方。
 d) 两个整数的积的绝对值等于它们的绝对值的积。
11. 使用谓词、量词、逻辑联接词和数学运算符表达语句“每个正整数是四个整数的平方的和”。
12. 使用谓词、量词、逻辑联接词和数学运算符表达下列数学语句:
 a) 两个负实数的积是正数。
 b) 一个实数与它自身的差是零。
 c) 每个正实数恰有两个平方根。
 d) 负实数没有实数平方根。
13. 将下列嵌套量化式子翻译成表达出数学事实的汉语语句。论域均为全体实数。
 a) $\exists x \forall y (xy = y)$
 b) $\forall x \forall y ((x < 0) \wedge (y < 0)) \rightarrow (xy > 0)$
 c) $\exists x \exists y ((x^2 > y) \wedge (x < y))$
 d) $\forall x \forall y \exists z (x + y = z)$
14. 假定所有变量的论域都是整数集合, 确定下列语句的真值。
 a) $\forall n \exists m (n^2 < m)$
 b) $\exists n \forall m (n < m^2)$
 c) $\forall n \exists m (n + m = 0)$
 d) $\exists n \forall m (nm = m)$
 e) $\exists n \exists m (n^2 + m^2 = 5)$
 f) $\exists n \exists m (n^2 + m^2 = 6)$
 g) $\exists n \exists m (n + m = 4 \wedge n - m = 1)$
 h) $\exists n \exists m (n + m = 4 \wedge n - m = 2)$
 i) $\forall n \forall m \exists p (p = (m + n)/2)$
15. 假定命题函数 $P(x, y)$ 的论域由 x 和 y 的序偶组成, 其中 x 是 1、2 或 3, y 是 1、2 或 3。用析取和合取写出下列命题。
 a) $\forall x \forall y P(x, y)$
 b) $\exists x \exists y P(x, y)$
 c) $\exists x \forall y P(x, y)$
 d) $\forall y \exists x P(x, y)$
16. 表达出下列语句的否定, 并且所有的否定词紧跟在谓词之前。
 a) $\forall x \exists y \forall z T(x, y, z)$
 b) $\forall x \exists y P(x, y) \vee \forall x \exists y Q(x, y)$
 c) $\forall x \exists y (P(x, y) \wedge \exists z R(x, y, z))$
 d) $\forall x \exists y (P(x, y) \rightarrow Q(x, y))$
17. 重写下列语句, 使否定只出现在谓词中(即否定词不在量词外边, 也不在含逻辑联接词的表达式外边)。
 a) $\neg \forall x \forall y P(x, y)$
 b) $\neg \exists y \exists x P(x, y)$
 c) $\neg \forall y \forall x (P(x, y) \vee Q(x, y))$
 d) $\neg (\exists x \exists y \neg P(x, y) \wedge \forall x \forall y Q(x, y))$
 e) $\neg \forall x (\exists y \forall z P(x, y, z) \wedge \exists z \forall y P(x, y, z))$
18. 给出变元 x, y, z 和 w 的一般论域, 使语句 $\forall x \forall y \forall z \exists w ((w \neq x) \wedge (w \neq y) \wedge (w \neq z))$ 为真, 给出另外的论域使其为假。
19. 用量词表达下列语句, 然后取该语句的否定并使否定词不在量词的左边。再用简单语句表达这否定(不要简单地表达为“不是……”)。
 a) 班上每个学生都恰好选修过本校两门数学课。
 b) 有人去过世界上除利比亚以外的每个国家。
 c) 没有人攀登过喜马拉雅山的每座山峰。
 d) 每位电影演员都或与 Kevin Bacon 同拍过电影, 或跟与 Kevin Bacon 同拍过电影的人同拍过电影。
20. 找出下列全称量化语句的反例(如果可能的话), 其中所有变量的论域是全体整数。
 a) $\forall x \forall y (x^2 = y^2 \rightarrow x = y)$
 b) $\forall x \exists y (y^2 = x)$
 c) $\forall x \forall y (xy \geq x)$

21. 用量词表达实数乘法的结合律。
22. 用量词和逻辑联接词表示事实：每一个线性多项式(也就是次数为1的多项式)，其中 x 带有非零实系数，其一定有一个实根。
23. 如果变量的论域分别如下，确定语句 $\forall x \exists y(xy=1)$ 的真值。
 a) 非零实数 b) 非零整数 c) 正实数
24. 证明语句 $\neg \exists x \forall y P(x, y)$ 和 $\forall x \exists y \neg P(x, y)$ 逻辑等价。这里在 $P(x, y)$ 中两个量词的第一和第二变元有相同的论域。
- *25. a) 证明 $\forall x P(x) \wedge \exists x Q(x)$ 和 $\forall x \exists y (P(x) \wedge Q(y))$ 逻辑等价。这里所有量词都有相同的非空论域。
 b) 证明 $\forall x P(x) \vee \exists x Q(x)$ 和 $\forall x \exists y (P(x) \vee Q(y))$ 逻辑等价。这里所有量词都有相同的非空论域。

一个语句称为前束范式(PNF)的充分必要条件是其表达形式为

$$Q_1 x_1 Q_2 x_2 \cdots Q_k x_k P(x_1, x_2, \dots, x_k)$$

其中每个 $Q_i (i=1, 2, \dots, k)$ 或是全称量词，或是存在量词；而 $P(x_1, x_2, \dots, x_k)$ 是不含量词的谓词。

例如 $\exists x \forall y (P(x, y) \wedge Q(y))$ 是前束范式，而 $\exists x P(x) \vee \forall x Q(x)$ 不是(因为并不是所有量词都先出现)。

将命题变量、谓词、T和F用逻辑联接词和量词组合在一起形成的每一个语句都等价于一个前束范式。

练习26要求的就是对这一事实的证明。

- **26. 给出并证明把任意语句变换为等价的前束范式的方法。

1.5 推理规则

1.5.1 引言

本章后一部分将学习证明。在数学中，证明是建立在数学命题真实性之上的有效论证。对于一个论证，是指表示一连串的命题最终得出结论。对于有效性，是计划得出结论或者论证的最终命题，其过程必须依据前面命题或者论证的前提的真实性。也就是说，一个论证是真实的，当且仅当它对所有的前提为真而结论为假是不可能的。为从已有的命题中推出新的命题，应用推理规则，它是构造有效论证的模板。推理规则是建立命题真实性的基本工具。

在学习数学证明之前，将看到涉及唯一复合命题的论证。我们将定义涉及复合命题论证的意义，然后介绍命题逻辑的一系列推理规则。这些规则在产生有效论证中是最重要的组成部分。在说明推理规则如何用于产生有效论证后，将描述不正确推理的常见形式，也就是谬误，它会导致无效论证。

在学习了命题逻辑中的推理规则后，将介绍量化语句的逻辑规则。我们将描述这些推理规则如何用于产生有效论证。这些涉及存在量词和全称量词的语句推理规则在计算机科学和数学中扮演了非常重要的角色，尽管用时常常没有明显提及它们。

最后，我们将显示命题的推理规则和量化语句的推理规则如何结合。这些推理规则地结合在复杂的论证中综合在一起使用。

1.5.2 命题逻辑的有效论证

考虑下面涉及命题的论证(从定义上讲，是一连串命题)：

“如果你有一个正确的密码，那么你可以登录到网络。”

“你有一个正确的密码。”

所以

“你可以登录到网络。”

我们想决定这是否是一个有效论证。也就是说，想要决定当前提“如果你有一个正确的密码，那么你可以登录到网络”和“你有一个正确的密码”都为真时，结论“你可以登录到网络”是否为真。

在讨论这个特定论证的有效性之前，将看它的形式化表示。用 p 代表“你有一个正确的密码”，用 q 代表“你可以登录到网络”，那么，这个论证形式化表示如下：

$$\begin{array}{c} p \rightarrow q \\ \therefore \frac{p}{q} \end{array}$$

其中 $\therefore$ 是表示“所以”的符号。

我们知道，当 p 和 q 是命题变量时，语句 $((p \rightarrow q) \wedge p) \rightarrow q$ 是个永真式。特别地，当 $p \rightarrow q$ 和 p 都为真时，我们知道 q 肯定为真。我们说语句的这种论证形式是有效的，因为无论什么时候，只要它的所有前提（论证中的所有语句，而不是最后一句结论）为真，那么结论也必须为真。现在假设“如果你有一个正确的密码，那么你可以登录到网络”和“你有一个正确的密码”都为真。当用 p 表示“你有一个正确的密码”，用 q 表示“你可以登录到网络”，那么接下来必然的结论是“你可以登录到网络”为真。这个论证有效是因为它的形式有效。注意，无论用什么命题替换 p 和 q ，且 $p \rightarrow q$ 和 p 都为真时，那么 q 也肯定为真。

当用命题将这个论证形式替换到 p 和 q ，其中 p 和 $p \rightarrow q$ 不都为真时会如何呢？比如，假设 p 代表“你可以入网”， q 代表“你可以改变你的年级”， p 为真，但是 $p \rightarrow q$ 为假。将 p 和 q 的值替换为论证形式，得到的论证为：

“如果你能入网，那么你可以改变你的年级”

“你可以入网”

$\therefore$ “你可以改变你的年级”

得到的这个论证是有效论证，但是因为其中一个前提，即第一个前提，为假，所以不能得出结论为真（有很大可能，这个结论为假）。

在讨论中，为分析一个论证，用命题变量代替命题。这将一个论证改变为一个论证形式。我们发现，一个论证的有效性取决于这个论证形式的有效性。用这些关键概念的定义来总结用于讨论论证有效性的术语。

定义 1 命题逻辑中的一个论证是一连串的命题。除了论证中最后一个命题外都叫前提，最后一个命题叫结论。当它的所有前提为真意味着结论为真时，一个论证是有效的。

命题逻辑中的论证形式是涉及命题变元的一连串复合命题。当在前提中无论哪个特定命题被转换为命题变元，当前提都为真时结论都为真，称为论证形式有效。

从有效论证形式的定义，当 $(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \rightarrow q$ 是永真式时，论证形式的前提 $p_1, p_2, \dots, p_n$ 以及结论 q 有效。

表示命题逻辑论证有效的关键是表示出它的论证形式有效。因此，我们想要表明论证形式有效的技术。现在我们将拓展完成这一任务的方法。

1.5.3 命题逻辑的推理规则

可以总是用一个真值表来表示一个论证形式有效。我们用无论何时只要前提为真，结论也就肯定为真来表明做到这一点。然而，这会是冗长乏味的方法。例如，当论证形式涉及 10 个不同变量，用真值表表示这个论证形式有效需要 $2^{10} = 1024$ 行。幸运的是，我们不必求助于真值表。取而代之，我们可以先建立一些比较简单的有效论证形式，称为推理规则。这些推理规则可以作为模块来构造更多复杂的有效论证形式。现在我们将介绍命题逻辑中最重要的推理规则。

永真式 $(p \wedge (p \rightarrow q)) \rightarrow q$ 是一个被称为假言推理或分离规则的推理规则的基础。（在拉丁文中假言推断是指模式确认。）这个永真式导致了以下有效论证形式，即在我们开头论证的讨论中已经看到的。（与前一样，这里符号 $\therefore$ 表示“所以”。）

$$\begin{array}{c} p \\ p \rightarrow q \\ \hline \therefore q \end{array}$$

用这个概念，将前提写在一列，随之是一横线，它是以因此符号开头以结论结尾的一行。特别地，假言推断告诉我们如果一个条件语句以及它的前提都为真，那么结论肯定为真。例 1 说明了假言推断的应用。

例 1 假设条件语句“若今天下雪，那么我们将滑雪”以及它的前提“今天正在下雪”为真。那么，根据假言推断，条件语句的结论“我们将滑雪”为真。 ■

就像前面提到的，当一个或更多前提为假时，一个有效论证可以导致错误的结论。在例2中将再次说明。

例2 决定如下给定的论证是否有效，并且决定由论证的有效性它的结论是否一定为真。

“如果 $\sqrt{2} > 3/2$ ，那么 $(\sqrt{2})^2 > (3/2)^2$ 。我们知道 $\sqrt{2} > 3/2$ ，因此 $(\sqrt{2})^2 = 2 > (3/2)^2 = 9/4$ 。”

解 令 p 为命题“ $\sqrt{2} > 3/2$ ”，令 q 为 $(\sqrt{2})^2 > (3/2)^2$ 。论证的前提 $p \rightarrow q$ 和 p ， q 是结论。这个论证是有效的，因为是用假言推断这个有效论证形式来构造的。然而，其中的前提 $\sqrt{2} > 3/2$ 为假。因此不能得出结论为真。此外，注意这个论证的结论为假，因为 $2 < 9/4$ 。 ■

表1-25列出了最重要的命题逻辑推理规则。1.2节练习5和8要求证明这些推理规则为有效论证形式。现在给出用这些推理规则论证的例子。在每一论证中，首先用命题变量表达论证中的命题。那么所得出的论证结果是表1-25中的推理规则。

表 1-25 推理规则

推理规则	永真式	名称
$\frac{p \quad p \rightarrow q}{\therefore q}$	$[p \wedge (p \rightarrow q)] \rightarrow q$	假言推理
$\frac{\neg q \quad p \rightarrow q}{\therefore \neg p}$	$[\neg q \wedge (p \rightarrow q)] \rightarrow \neg p$	取拒式
$\frac{p \rightarrow q \quad q \rightarrow r}{\therefore p \rightarrow r}$	$[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$	假言三段论
$\frac{p \vee q \quad \neg p}{\therefore q}$	$[(p \vee q) \wedge \neg p] \rightarrow q$	析取三段论
$\frac{p}{\therefore p \vee q}$	$p \rightarrow (p \vee q)$	附加
$\frac{p \wedge q}{\therefore p}$	$(p \wedge q) \rightarrow p$	化简
$\frac{p \quad q}{\therefore p \wedge q}$	$[(p) \wedge (q)] \rightarrow (p \wedge q)$	合取
$\frac{p \vee q \quad \neg p \vee r}{\therefore q \vee r}$	$[(p \vee q) \wedge (\neg p \vee r)] \rightarrow (q \vee r)$	消解

例3 说出哪个推理规则是下列论证的基础：“现在气温在冰点以下。因此，要么现在气温在冰点以下，要么现在下雨。”

解 设 p 是命题“现在气温在冰点以下”，而 q 是命题“现在正在下雨”。那么这个论证形如

$$\frac{p}{\therefore p \vee q}$$

这是使用附加规则的论证。 ■

例4 说出哪个推理规则是下列论证的基础：“现在气温在冰点以下并且现在下雨。因此，现在气温在冰点以下。”

解 设 p 是命题“现在气温在冰点以下”，而 q 是命题“现在正在下雨”。这个论证形如

$$\frac{p \wedge q}{\therefore p}$$

这个论证使用化简规则。 ■

例 5 说出在下列论证里使用哪个推理规则：

若今天下雨，则我们今天将不野餐。若我们今天不野餐，则我们明天将野餐。因此，若今天下雨，则我们明天将野餐。

解 设 p 是命题“今天下雨”，设 q 是命题“我们今天将不野餐”，而设 r 是命题“我们明天将野餐”。则这个论证形如

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$

因此，这个论证是假言三段论。

1.5.4 用推理规则建立论证

当有许多前提时，为了证明一个论证是有效的，常常需要多个推理规则。对此通过例 6 和例 7 来说明。在这些例子中每一步论证显示为不同的行，并明确地写出每步的理由。这些例子也说明如何使用推理规则来分析用自然语言表述的论证。

例 6 证明：前提“今天下午没有出太阳并且今天比昨天冷”，“只有今天下午出太阳，我们才去游泳”，“若我们不去游泳，则我们将乘独木舟游览”，以及“若我们乘独木舟游览，则我们将在黄昏时回家”，导致结论“我们将在黄昏时回家”。

解 设 p 是命题“今天下午出太阳”， q 是命题“今天比昨天冷”， r 是命题“我们将去游泳”， s 是命题“我们将乘独木舟游览”，而 t 是命题“我们将在黄昏时回家”。则这些前提成为 $\neg p \wedge q$ ， $r \rightarrow p$ ， $\neg r \rightarrow s$ ， $s \rightarrow t$ 。结论是 t 。我们需要用假设 $\neg p \wedge q$ ， $r \rightarrow p$ ， $\neg r \rightarrow s$ ，以及 $s \rightarrow t$ 和结论 t 给出有效论证。

如下构造一个论证来证明这些前提得出需要的结论。

步骤	理由
1. $\neg p \wedge q$	前提引入
2. $\neg p$	化简，用步骤 1
3. $r \rightarrow p$	前提引入
4. $\neg r$	取拒式，用步骤 2 和 3
5. $\neg r \rightarrow s$	前提引入
6. s	假言推理，用步骤 4 和 5
7. $s \rightarrow t$	前提引入
8. t	假言推理，用步骤 6 和 7

注意，可以用真值表表示无论何时，当 4 个前提为真，那么结论也为真。然而，因为用 5 个变元 p 、 q 、 r 、 s 和 t 来操作，这样的真值表需要 32 行。

例 7 证明：前提“若你发给我电子邮件消息，则我将完成编写程序”，“若你不发给我电子邮件消息，则我将早早地去睡觉”，以及“若我早早地去睡觉，则我将感觉精力充沛地醒来”，导致结论“若我不完成编写程序，则我将感觉精力充沛地醒来”。

解 设 p 是命题“你发给我电子邮件消息”， q 是命题“我将完成编写程序”， r 是命题“我将早早地去睡觉”，而 s 是命题“我将感觉精力充沛地醒来”。则这些前提是 $p \rightarrow q$ ， $\neg p \rightarrow r$ ， $r \rightarrow s$ 。需要的结论是 $\neg q \rightarrow s$ 。我们需要用假设 $p \rightarrow q$ ， $\neg p \rightarrow r$ ，以及 $r \rightarrow s$ 和结论 $\neg q \rightarrow s$ 给出有效论证。

下列论证证明这些前提得出需要的结论。

步骤	理由
1. $p \rightarrow q$	前提引入
2. $\neg q \rightarrow \neg p$	步骤 1 和逆否命题
3. $\neg p \rightarrow r$	前提引入
4. $\neg q \rightarrow r$	假言三段论，用步骤 2 和 3

5. $r \rightarrow s$ 前提引入
 6. $\neg q \rightarrow s$ 假言三段论, 用步骤4和5

1.5.5 消解

 已经开发出自动执行推理和证明定理任务的计算机程序。许多这类程序利用称为消解(resolution)的推理规则。这个推理规则基于永真式:

$$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$$

消解规则最后的析取 $q \vee r$ 称为消解式(resolvent)。当在此永真式中令 $q=r$ 时, 可得 $(p \vee q) \wedge (\neg p \vee q) \rightarrow q$ 。而且, 当令 $r=F$ 时, 可得 $(p \vee q) \wedge (\neg p) \rightarrow q$ (因为 $q \vee F \equiv q$)。这是永真式, 析取三段论规则便基于此式。

例8 使用消解证明假设“Jasmine 在滑雪或没有下雪”和“下雪了或 Bart 在打曲棍球”蕴含结论“Jasmine 在滑雪或 Bart 在打曲棍球。”

 **解** 令 p 为命题“下雪了”, q 为命题“Jasmine 在滑雪”, r 为命题“Bart 在打曲棍球”。我们可以将假设分别表示为 $\neg p \vee q$ 和 $p \vee r$ 。使用消解, 命题 $q \vee r$ 成立。

消解在基于逻辑规则的编程语言中扮演着重要的角色, 如在 Prolog 中(其中用到了量化语句的消解规则)。而且, 可以用消解来构建自动定理证明系统。要使用消解来构造命题逻辑中的证明, 假设和结论必须被表示为子句, 子句是变量的析取或这些变量的否定。我们可以将命题逻辑中非子句的语句用一个或多个等价的是子句的语句替换。例如, 假定有一个形如 $p \vee (q \wedge r)$ 的语句。因为 $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$, 便可以用两个子句 $p \vee q$ 和 $p \vee r$ 来代替 $p \vee (q \wedge r)$ 。我们可以用语句 $\neg p$ 和 $\neg q$ 来代替形如 $\neg(p \vee q)$ 的语句, 因为德摩根律表明 $\neg(p \vee q) \equiv \neg p \wedge \neg q$ 。也可以用等价的析取 $\neg p \vee q$ 来代替蕴含 $p \rightarrow q$ 。

例9 证明假设 $(p \wedge q) \vee r$ 和 $r \rightarrow s$ 蕴含结论 $p \vee s$ 。

解 可以将假设 $(p \wedge q) \vee r$ 重写为两个子句 $p \vee r$ 和 $q \vee r$ 。也可以将 $r \rightarrow s$ 替换为等价的子句 $\neg r \vee s$ 。使用子句 $p \vee r$ 和 $\neg r \vee s$, 通过消解便可得出 $p \vee s$ 。

1.5.6 谬误

几种常见的谬误都来源于不正确的论证。这些谬误看上去像是推理规则, 但是它们是基于偶然事件而不是永真式。在这里讨论这些谬误, 是为了说明在正确与不正确的推理之间的区别。

 **命题** $[(p \rightarrow q) \wedge q] \rightarrow p$ 不是永真式, 因为当 p 为假而 q 为真时, 它为假。不过, 存在许多把它当做永真式的不正确论证。换句话说, 对于前提 $p \rightarrow q$ 和 q 及结论 p 不是有效论证。这种类型的不正确推理称为肯定结论谬误。

例10 下列论证是否有效?

若你做本书的每一道练习, 则你将学习离散数学。你学习过离散数学。

因此, 你做过本书的每一道练习。

解 设 p 是命题“你做过本书的每一道练习”。设 q 是命题“你学习过离散数学”。这个论证形如: 若 $p \rightarrow q$ 并且 q , 则 p 。这是使用肯定结论谬误的不正确推理的例子。事实上, 你可能通过其他某种方式学习离散数学而没有做本书的每一道练习。(你可能通过阅读、听讲座、做本书的一些但不是全部练习等方式来学习离散数学。)

命题 $[(p \rightarrow q) \wedge \neg p] \rightarrow \neg q$ 不是永真式, 因为当 p 为假而 q 为真时, 它为假。许多不正确的论证都不正确地把它当作了推理规则。这种类型的不正确推理称为否定假设谬误。

例11 设 p 和 q 如同例10那样。若条件语句 $p \rightarrow q$ 为真, 并且 $\neg p$ 为真, 则得出 $\neg q$ 为真这是是否正确? 换句话说, 如果假定: 若你做本书里每一道练习, 则你将学习离散数学, 那么假定: 若你没有做过本书里每一道练习, 则你没有学习离散数学, 这是否正确?

解 即使你没有做过本书里每一道练习, 你也可能学过离散数学。这个不正确的论证具有形式: $p \rightarrow q$ 和 $\neg p$ 蕴含 $\neg q$, 这是否定假设谬误的例子。

1.5.7 带量词命题的推理规则

已经讨论了命题的推理规则。现在将要描述包含量词的命题的一些重要的推理规则。在数学论证里大量地使用这些推理规则，通常都没有明确地指出来。

全称例示 用它从前提 $\forall xP(x)$ 得出 $P(c)$ 为真，其中 c 是论域里的具体成员。当从命题“所有女人都聪明”得出“丽沙聪明”时，就使用了全称例示，其中丽沙是论域所有女人中的一员。

全称生成 它说明在对论域里所有元素 c 来说 $P(c)$ 都为真的前提下， $\forall xP(x)$ 为真。当通过从论域里拿出一个任意元素 c 并证明 $P(c)$ 为真来证明 $\forall xP(x)$ 为真时，就使用了全称生成。所选择的元素 c 必须是论域里一个任意的元素，而不是特定的元素。也就是说，当从 $\forall xP(x)$ 断言论域中的每一个元素 c ，而没有控制 c ，但不能得出任何其他关于 c 的假设，除非 c 属于论域中。在许多数学证明里都隐含的使用全称量词的引入，而很少明确地指出来。然而，当应用全称生成时，在错误的推理中增加关于任意元素 c 莫须有的假设是屡见不鲜的。

存在例示 在允许从已经 $\exists xP(x)$ 为真，得出在论域里存在一个使得 $P(c)$ 为真的元素 c 。在这里不能选择一个任意值的 c ，而必须是使得 $P(c)$ 为真的那个 c 。通常不知道 c 是什么，仅仅知道它存在。因为它存在，所以可以给它一个名称(c)而继续论证。

存在生成 用它在已知使 $P(c)$ 为真的一个具体的 c 时，得出 $\exists xP(x)$ 为真。即若知道论域里一个使 $P(c)$ 为真的元素 c ，则知道 $\exists xP(x)$ 为真。

这些推理规则总结在表 1-26 里。下面的例子将要说明如何使用带量词的命题的推理规则。

表 1-26 带量词的命题的推理规则

推 理 规 则	名 称
$\forall xP(x)$ $\therefore P(c)$	全称例示
$P(c), \text{任意 } c$ $\therefore \forall xP(x)$	全称生成
$\exists xP(x)$ $\therefore P(c), \text{对某个元素 } c$	存在例示
$P(c), \text{对某个元素 } c$ $\therefore \exists xP(x)$	存在生成

例 12 证明前提“在本离散数学课堂的每一个人学过一门计算机课程”和“玛拉是本课堂的学生”蕴含结论“玛拉学过一门计算机课程”。

解 设 $D(x)$ 表示“ x 在本离散数学课堂”，并且设 $C(x)$ 表示“ x 学过一门计算机课程”。
则前提是 $\forall x(D(x) \rightarrow C(x))$ 和 $D(\text{玛拉})$ ，结论是 $C(\text{玛拉})$ 。下列步骤可以用来从前提证明结论。

步 骤	理 由
1. $\forall x(D(x) \rightarrow C(x))$	前提引入
2. $D(\text{玛拉}) \rightarrow C(\text{玛拉})$	全称例示，用步骤 1
3. $D(\text{玛拉})$	前提引入
4. $C(\text{玛拉})$	假言推理，用步骤 2 和 3

例 13 证明前提“在这个班上的某个学生没有读过书”和“班上的每个人都通过了第一门考试”蕴含结论“通过考试的某个人没有读过书”。

解 令 $C(x)$ 表示“ x 在这个班中”， $B(x)$ 表示“ x 读过书了”， $P(x)$ 为“ x 通过了第一门考试”。
则前提是 $\exists x(C(x) \wedge \neg B(x))$ 和 $\forall x(C(x) \rightarrow P(x))$ ，结论是 $\exists x(P(x) \wedge \neg B(x))$ 。下列步骤可以用来从前提证明结论。

步骤	理由
1. $\exists x(C(x) \wedge \neg B(x))$	前提引入
2. $C(a) \wedge \neg B(a)$	存在例示, 用步骤 1
3. $C(a)$	化简, 用步骤 2
4. $\forall x(C(x) \rightarrow P(x))$	前提引入
5. $C(a) \rightarrow P(a)$	全称例示, 用步骤 4
6. $P(a)$	假言推理, 用步骤 3 和 5
7. $\neg B(a)$	化简, 用步骤 2
8. $P(a) \wedge \neg B(a)$	合取, 用步骤 6 和 7
9. $\exists x(P(x) \wedge \neg B(x))$	存在生成, 用步骤 8

1.5.8 命题推理和量化语句推理规则的结合

我们已经学习了命题推理规则和量化语句的推理规则。注意, 在例 12 和 13 中的论证既用了全称例示(量化语句推理规则)也用了假言推理(命题推理规则)。我们常常需要对推理规则结合使用。由于全称例示和假言推理结合使用如此广泛, 这个规则的结合有时称作全称假言推理。这个规则告诉我们: 如果 $\forall x(P(x) \rightarrow Q(x))$ 为真, 且如果 $P(a)$ 对在论域中的一个特定元素 a 为真, 那么 $Q(a)$ 也肯定为真。为便于理解, 注意用全称例示, $P(a) \rightarrow Q(a)$ 为真。那么, 用假言推理, $Q(a)$ 也肯定为真。可以将假言推理描述如下:

$$\begin{array}{l} \forall x(P(x) \rightarrow Q(x)) \\ P(a), \text{其中 } a \text{ 是论域中一个特定的元素} \\ \hline \therefore Q(a) \end{array}$$

全称假言推理常常用于数学论证中。这将在例 14 中说明。

例 14 假定“对所有正整数 n , 如果 n 比 4 大, 那么 n^2 小于 2^n ”为真。用全称假言推理证明 $100^2 < 2^{100}$ 。

解 令 $P(n)$ 表示“ $n > 4$ ”, $Q(n)$ 表示“ $n^2 < 2^n$ ”语句“对所有正整数 n , 如果 n 比 4 大, 那么 n^2 小于 2^n ”可以表示为 $\forall n(P(n) \rightarrow Q(n))$, 其中论域为所有正整数。假设 $\forall n(P(n) \rightarrow Q(n))$ 为真。注意 $P(100)$ 为真, 因为 $100 > 4$ 。接着用全称假言推理, $Q(n)$ 为真, 即 $100^2 < 2^{100}$ 。 ■

另一个重要的命题逻辑推理规则和量化语句的推理规则的结合是全称取拒式。全称取拒式结合了全称例示和取拒式, 可以用如下方式表达:

$$\begin{array}{l} \forall x(P(x) \rightarrow Q(x)) \\ \neg Q(a), \text{其中 } a \text{ 是论域中一个特定的元素} \\ \hline \therefore \neg P(a) \end{array}$$

将全称取拒式的证明留给读者(见练习 13)。

练习

1. 给出下列论证的论证形式化, 并决定是否有效。如果前提为真, 能断定结论为真吗?

如果苏格拉底是人, 那么苏格拉底是会死的。

苏格拉底是人

$\therefore$ 苏格拉底是会死的

2. 在下列每个论证里使用了什么推理规则?

a) 爱丽丝主修数学。因此, 爱丽丝主修数学或计算机科学。

b) 杰瑞主修数学和计算机科学。因此, 杰瑞主修数学。

c) 若天气下雨, 则游泳池将关闭。天气下雨。因此, 游泳池关闭。

d) 若今天下雪, 则大学将关闭。今天大学没有关闭。因此, 今天没有下雪。

e) 若我去游泳, 则我将在太阳下停留过久。若我在太阳下停留过久, 则我将有晒斑。因此, 若我去游泳, 则我将有晒斑。

3. 使用推理规则证明前提“兰迪努力地工作”，“若兰迪努力地工作，则他是一个笨孩子”，和“若兰迪是一个笨孩子，则他将得不到这个工作”蕴含着结论“兰迪将得不到这个工作”。
4. 在下面的著名论证里使用了什么推理规则？“所有的人都是要死的。苏格拉底是人。因此，苏格拉底是要死的”。
5. 对下列的每组前提，可以得出什么样的相关结论或一组结论？解释从前提获得每个结论所使用的推理规则。
 - a) “若我在某一天休息，则那天下雨或下雪”。“我在周二休息或在周四休息”。“周二出太阳”。“周四未下雪”。
 - b) “若我吃了辣的食物，则我做奇怪的梦”。“若当我睡觉时有雷声，则我做奇怪的梦”。“我没有做奇怪的梦”。
 - c) “我聪明或幸运”。“我不幸运”。“若我幸运，则我将赢得抽奖”。
 - d) “每个主修计算机科学的人都有个人电脑”。“拉尔夫没有个人电脑”。“安妮有个人电脑”。
 - e) “对公司有利的就对美国有利”。“对美国有利的就对你有利”。“对公司有利的就是你购买许多东西”。
 - f) “所有的啮齿类动物都啃它们的食物”。“老鼠是啮齿类动物”。“野兔不啃它们的食物”。“蝙蝠不是啮齿类动物”。
6. 证明如果论证形式由前提 $p_1, p_2, \dots, p_n, q$ 及结论 r 有效，则论证形式由前提 $p_1, p_2, \dots, p_n$ 及结论 $q \rightarrow r$ 有效。
7. 对下列每个论证，解释对每个步骤使用了哪条推理规则。
 - a) “本班学生道格知道如何用 Java 写程序。知道如何用 Java 写程序的每个人都可以得到高薪的工作。因此，本班的某个人可以得到高薪的工作。”
 - b) “本班的某个人喜欢观察鲸鱼。每个喜欢观察鲸鱼的人都关心海洋污染。因此，本班里某个人关心海洋污染。”
 - c) “本班的 93 个学生每人拥有一台个人电脑。拥有个人电脑的每个人都能使用字处理软件。因此，本班学生泽克能使用字处理软件。”
 - d) “新泽西的每个人都生活在距离海洋 50 英里之内。新泽西的某个人从来没有见过海洋。因此，生活在距离海洋 50 英里之内的某个人从来没有见过海洋。”
8. 判断下列论证是否正确并解释原因。
 - a) 班上的所有学生都理解逻辑。Xavier 是这个班上的学生，因此，Xavier 理解逻辑。
 - b) 每个计算机专业的学生都要学离散数学。Natasha 在学离散数学，因此，Natasha 是计算机专业的。
 - c) 所有鹦鹉都喜欢水果。我养的鸟不是鹦鹉，因此，我养的鸟不喜欢水果。
 - d) 每天吃麦片的人都很健康。Linda 不健康，因此，Linda 没有每天吃麦片。
9. 如下论证错在哪里？令 $H(x)$ 为“ x 很开心”。给定前提 $\exists x H(x)$ ，我们得出 $H(\text{Lola})$ 。因此，LoLa 很开心。
10. 判定下列每个论证是否有效。若论证是正确的，则使用了什么推理规则？若它不正确，则发生了什么谬误？
 - a) 若 n 是使 $n > 1$ 的实数，则 $n^2 > 1$ 。假定 $n^2 > 1$ 。于是 $n > 1$ 。
 - b) 若 n 是使 $n > 3$ 的实数，则 $n^2 > 9$ 。假定 $n^2 \leq 9$ ，于是 $n \leq 3$ 。
 - c) 若 n 是使 $n > 2$ 的实数，则 $n^2 > 4$ 。假定 $n \leq 2$ ，于是 $n^2 \leq 4$ 。
11. 哪些推理规则被用来证明在 1.3 节例 26 里所描述的卡洛尔(Lewis Carroll)论证的结论？
12. 指出如下由“ $\exists x P(x) \wedge \exists x Q(x)$ 为真，那么 $\exists x (P(x) \wedge Q(x))$ 为真”的论证有哪些错误。

1) $\exists x P(x) \wedge \exists x Q(x)$	前提引入
2) $\exists x P(x)$	化简，用步骤 1
3) $P(c)$	存在例示，用步骤 2
4) $\exists x Q(x)$	化简，用步骤 1
5) $Q(c)$	存在例示，用步骤 4
6) $P(c) \wedge Q(c)$	合取，用步骤 3 和 5
7) $\exists x (P(x) \wedge Q(x))$	存在生成
13. 通过证明前提 $\forall x (P(x) \rightarrow Q(x))$ 和 $\neg Q(a)$ ，其中是对论域中某个特定元素 a ，推出 $\neg P(a)$ ，来证明全称取拒式。

14. 用推理规则证明: 若 $\forall x(P(x) \rightarrow (Q(x) \wedge S(x)))$ 和 $\forall x(P(x) \wedge R(x))$ 为真, 则 $\forall x(R(x) \wedge S(x))$ 为真。
15. 用推理规则证明: 若 $\forall x(P(x) \vee Q(x))$ 和 $\forall x(\neg Q(x) \vee S(x))$, $\forall x(R(x) \rightarrow \neg S(x))$ 和 $\exists x \neg P(x)$ 为真, 则 $\exists x \neg R(x)$ 为真。
16. 使用消解法证明前提“天上没有下雨或 Yvette 带雨伞了”、“Yvette 没有带雨伞或她没有淋湿”和“天上下雨了或 Yvette 没有淋湿”蕴含“Yvette 没有淋湿”。
17. 用消解法证明复合命题 $(p \vee q) \wedge (\neg p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee \neg q)$ 不是可满足的。
- *18. 判定下列论证(来自 Kalish and Montague[KaMo64])是否有效:

假如超人能够并愿意防止邪恶, 则他将这样做。假如超人不能够防止邪恶, 则他将是无能的; 假如超人不愿意防止邪恶, 则他将是恶意的。超人没有防止邪恶。若超人存在, 则他是无能的或恶意的。因此, 超人不存在。

1.6 证明导论

1.6.1 引言

本节介绍证明的概念, 以及描述构造证明的方法。证明是建立在数学语句真实性基础上的有效论证。证明可以用定理(如果有的话), 假设为真的公理以及之前已经证明的定理的假设。用这些组成部分以及推理规则, 证明的最后一步是建立被证明命题的真实性。

在我们的讨论中, 从形式化定理证明转向非形式化证明。在 1.5 节介绍的涉及命题和量化语句为真的论证是形式化证明, 其中提供了所有步骤, 以及给出论证中每一步的所有规则。然而, 有用的定理形式化证明会很长且极难给出。实际上, 人们使用设计的定理证明几乎都是非形式化证明, 其中许多推理规则用于每一步, 其中的许多步骤被省略, 没有明确列出其中的假设公理和用到的推理规则。非形式化证明常常向人们解释定理为什么为真, 虽然计算机很愿意用自动推理系统产生形式化证明。

本节讨论的证明方法很重要, 不仅因为它们用于证明数学定理, 而且由于它们中的许多应用在计算机科学中。这些应用包括证明计算机程序是正确的, 建立安全的操作系统, 用人工智能做推理, 表明系统说明是一致的, 等等。因此, 在数学和计算机科学中理解用于证明的技术都非常关键。

1.6.2 一些专用术语

 正式地, 一个定理是一个能够表明是真的语句。在数学的描述中, 定理通常是用来表示被认为至少有些重要的语句。有时把不大重要的定理称为命题(定理也作为事实或结论被提到)。一个定理可以是带一个或多个前提的条件语句及一个结论全称量化。然而它可以是其他类型的逻辑语句。本节后面有一个例子, 示范一个带有证明的定理是真的。当建立了定理的真实性时, 一个证明就是个有效论证。用在证明中的语句包括公理(或假设), 假设这些语句是真的, 如果有的话, 定理的前提和以前已经证明的定理。公理用不要求定义的原始术语陈述, 而在定理和证明中所用的其他所有术语都必须定义。推理的规则和术语的定义一起用于从其他的断言推出结论, 并详列出证明的步骤。实际上, 证明的最后一步通常恰是定理的结论。然而, 为清晰起见, 在证明的最后一步, 通常重述定理的结论。

在其他结果证明中很有帮助的不大重要的定理称为引理(复数引理)。复杂证明用一系列引理来证明通常比较容易理解, 这里每一个引理都被独立证明。推论是从定理直接建立被证明的定理。猜想是被提出为真的命题, 通常是在一些依据的基础上, 启发式论证, 或是专家的一个直觉。当找到了一个猜想的证明时, 猜想就变成了定理。许多猜想表明是错误的, 因此它们不是定理。

1.6.3 定理陈述的理解

在介绍证明定理的方法之前, 我们需要理解有多少数学定理被陈述。许多定理断言一个性质对论域(比如整数或实数)中的所有元素都成立。虽然这些定理的准确陈述需要包含全称量词, 但是数学里的标准约定是省略全称量词。比如, 语句

“如果 $x > y$, 其中 x 和 y 是正实数, 那么 $x^2 > y^2$ 。”

其实意味着

“对所有正实数 x 和 y ，如果 $x > y$ ，那么 $x^2 > y^2$ 。”

此外，当证明这种类型的定理时，常常使用全称例示规则而不明确地指出来。证明的第一步通常涉及选择论域里的一个一般元素。随后的步骤是证明这个元素具有所考虑的性质。最后，全称生成蕴含着对论域里所有元素来说定理都成立。

1.6.4 证明定理的方法

Assessment: 现在将注意力转向数学定理的证明上。证明定理会很难。需要所有能帮助我们证明不同结果的“弹药”(方法)。现在介绍一系列不同证明方法。这些方法应该成为你证明定理的部分“拿手戏”。

为证明形如 $\forall x(P(x) \rightarrow Q(x))$ 的定理，我们的目标是表示出 $P(c) \rightarrow Q(c)$ 为真，其中 c 是论域中的任意一元素，然后应用全称生成。在这个证明中，需要表示出条件语句为真。回忆 $p \rightarrow q$ 为真，除非 p 为真且 q 为假。注意当证明语句 $p \rightarrow q$ 时，只需要证明如果 p 为真则 q 为真。下面的讨论将给出条件语句最常见的证明方法。之后，将讨论证明其他类型语句的方法。在 1.7 节，将介绍许多不同的可用于证明多种定理的证明技术。

当读到证明时，你将常常会发现词语“显然”或者“清楚的”。这些词意味着许多步骤已经省略并希望读者有能力补上。遗憾的是，这个假设往往无法保证，读者根本不确定怎么补上这些省略的步骤。我们将努力避免用这些词，并努力避免省略太多步骤。然而，如果包括了证明中所有的步骤，我们的证明将会非常冗长。

1.6.5 直接证明

构造一个条件语句 $p \rightarrow q$ 的直接证明，第一步假设 p 为真；接下来的步骤是用推理规则构造的，最后一步推出 q 也肯定为真。直接证明用于推出如果 p 为真，那么 q 也肯定为真，以致 p 为真、 q 为假的情况不会发生的方法表示出条件语句 $p \rightarrow q$ 为真。在直接证明中假定 p 为真，并且用公理、定义和前面证明过的定理以及推理规则，来证明出 q 也肯定为真。你将发现许多结果的直接证明相当直接，它是用一连串显然的步骤从假设隐含到结论。然而，直接证明有时要求特殊的洞察力并且可能会是需要技巧的。这里呈现的第一个直接证明相当直接；之后，你会看到一些不那么显而易见的证明。

我们将提供几个不同的直接证明。在给出第一个例子前，需要一个定义。

定义 1 整数 n 是偶数，如果存在一个整数 k 使得 $n = 2k$ ；整数 n 是奇数，如果存在一个整数 k 使得 $n = 2k + 1$ 。（注意，一个整数或为偶数或为奇数。）

例 1 给出定理“若 n 是奇数，则 n^2 是奇数”的直接证明。

Extra Example **解** 注意，这个定理表明 $\forall n(P(n) \rightarrow Q(n))$ ，这里 $P(n)$ 是“ n 是奇数”， $Q(n)$ 是“ n^2 是奇数”。正如已经说过的，我们将遵循数学证明中通常的惯例，证明 $P(n)$ 意味着 $Q(n)$ ，而不用全称例示。为开始这个定理的直接证明，假设这个条件语句的前提为真，即假设 n 是奇数。由奇整数的定义，有 $n = 2k + 1$ ，其中 k 是某个整数。想证明 n^2 也是奇数。把等式 $n = 2k + 1$ 两边平方，这样做得出 $n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$ 。由奇数定义，得到结论 n^2 是奇数（它是一个整数的 2 倍再加 1）。因此，证明了若 n 是奇数，则 n^2 是奇数。 ■

例 2 给出一个直接证明：如果 m 和 n 都是完全平方数，那么 nm 也是一个完全平方数。（若有一个整数 b ，使得 $a = b^2$ ，则整数 a 是一个完全平方数。）

解 为产生出这个定理的一个直接证明，假定这个条件语句的假设为真。即假定 m 和 n 都是完全平方数。用完全平方数的定义，则有整数 s 和 t 使得 $m = s^2$ ， $n = t^2$ 。证明的目的是表示出 mn 也是完全平方数。向前看到的，将两个等式 $m = s^2$ 和 $n = t^2$ 乘起来证明。这就得到 $mn = s^2 t^2$ ，也就是 $mn = (st)^2$ （用乘法交换律和结合律）。由完全平方数的定义， mn 也是一个完全平方数，因为它是 st 的平方，而 st 为整数。这就证明了如果 m 和 n 都是完全平方数，那么 nm 也是一个完全平方数。 ■

1.6.6 反证法

直接证明从定理的假设推出结论。它们从前提开始, 继续一连串的推演, 最终结束于结论。然而, 我们将发现直接证明方法常常行不通。我们需要形如 $\forall x(P(x) \rightarrow Q(x))$ 的定理的其他证明方法。这种类型的定理证明不是直接证明, 也就是说, 不从假设开始, 以结论结束, 这叫做间接证明。

一个非常有用的间接证明的类型叫做反证。反证用条件语句 $p \rightarrow q$ 等价于它的倒置 $\neg q \rightarrow \neg p$ 的事实。这意味着条件语句 $p \rightarrow q$ 可以用它的倒置 $\neg q \rightarrow \neg p$ 来证明。在用 $p \rightarrow q$ 的反证中, 将 $\neg q$ 作为假设, 用公理、定义和前面证明过的定理, 以及推理规则, 之后证明 $\neg p$ 。我们将用两个例子解释反证。这些例子表明当不容易找到直接证明时用反证会很有效。

例 3 给出定理“若 $3n+2$ 是奇数, 则 n 是奇数”的证明。

解 首先, 试图用直接证明。为构建直接证明, 首先假设 $3n+2$ 是奇整数。这意味着对某个整数 k , $3n+2=2k+1$ 。我们能由此证明 n 是奇数吗? 看到 $3n+1=2k$, 但不能用任何直接的方式得出 n 是奇数的结论。由于直接证明的想法失败, 下面尝试反证法。

反证法的第一步是假设条件语句“若 $3n+2$ 是奇数, 则 n 是奇数”的结论是假的; 也就是说, 假设 n 是偶数。于是由偶数定义, 对某个整数 k 有 $n=2k$ 。把 n 用 $2k$ 代入, 得到 $3n+2=3(2k)+2=6k+2=2(3k+1)$, 这告诉我们 $3n+2$ 是偶数(因为它是 2 的倍数), 因此不是奇数。这是定理前提的否定。因为对这个条件语句结论的否定蕴含着前提为假, 所以原来的条件语句为真。反证法成功, 证明了定理“若 $3n+2$ 是奇数, 则 n 是奇数”。

例 4 证明: 如果 $n=ab$, 其中 a 和 b 是正整数, 那么 $a \leq \sqrt{n}$ 或者 $b \leq \sqrt{n}$ 。

解 因为没有明显的方法从等式 $n=ab$ (其中 a 和 b 是正整数)直接证明 $a \leq \sqrt{n}$ 或者 $b \leq \sqrt{n}$, 所以尝试反证法。

反证法第一步是假定条件语句“如果 $n=ab$, 其中 a 和 b 是正整数, 那么 $a \leq \sqrt{n}$ 或者 $b \leq \sqrt{n}$ ”的结论为假。也就是说, 假定 $(a \leq \sqrt{n}) \vee (b \leq \sqrt{n})$ 为假。用析取和德摩根律, 发现这表明 $(a \leq \sqrt{n})$ 和 $(b \leq \sqrt{n})$ 都为假。可以将不等式乘起来(用的事实是如果 $0 < t$ 且 $0 < v$, 那么 $su < tv$)得到 $ab > \sqrt{n} * \sqrt{n} = n$ 。这表明了 $ab \neq n$, 这与命题 $n=ab$ 矛盾。

因为条件语句结论的否定意味着前提为假, 原来的条件语句为真。反证成功, 这就证明了如果 $n=ab$, 其中 a 和 b 是正整数, 那么 $a \leq \sqrt{n}$ 或者 $b \leq \sqrt{n}$ 。

空证明和平凡证明 当我们知道 p 为假时, 能够很快证明条件语句 $p \rightarrow q$ 为真, 因为当 p 为假时 $p \rightarrow q$ 一定为真。结果, 如果能证明出 p 为假, 那么就得到一个 $p \rightarrow q$ 的证明方法, 叫做空证明。空证明常用于证明一些定理的特殊情况, 这些定理说: 对所有正整数来说, 一个条件语句为真(即形如 $\forall n P(n)$ 的定理, 其中 $P(n)$ 是命题函数)。

例 5 证明命题 $P(0)$ 为真, 其中 $P(n)$ 是“若 $n > 1$, 则 $n^2 > n$ ”, 论域是所有整数。

解 注意命题 $P(0)$ 是条件语句“若 $0 > 1$, 则 $0^2 > 0$ ”。因为前提 $0 > 1$ 为假, 所以 $P(0)$ 自动地为真。

注意 条件语句的结论 $0^2 > 0$ 为假这个事实, 与条件语句的真值无关, 因为前提为假的条件语句保证为真。

如果我们知道结论 q 为真, 很快就能证明条件语句 $p \rightarrow q$ 。通过证明 q 为真, 紧接着得出 $p \rightarrow q$ 一定为真。用 q 为真的事实来证明 $p \rightarrow q$ 的方法叫做平凡证明。当证明定理的特殊情形(见 1.7 节分情形证明的讨论)时, 以及在数学归纳法中, 平凡证明常常是重要的。

例 6 设 $P(n)$ 是“若 a 和 b 是满足 $a \geq b$ 的正整数, 则 $a^n \geq b^n$ ”, 其中论域是所有整数。证明命题 $P(0)$ 为真。

解 命题 $P(0)$ 是“若 $a \geq b$, 则 $a^0 \geq b^0$ ”。因为 $a^0 = b^0 = 1$, 条件语句“若 $a \geq b$, 则 $a^0 \geq b^0$ ”的结论

为真。从而条件语句 $P(0)$ 为真。这是平凡证明法的一个例子。注意在这个证明里不需要前提，它是语句“ $a \geq b$ ”。

证明的小策略 我们已经介绍了证明形如 $\forall x(P(x) \rightarrow Q(x))$ 这类定理的两种重要方法：直接证明和反证法。我们还给出了示例说明如何使用每种方法。然而，当面临证明形如 $\forall x(P(x) \rightarrow Q(x))$ 定理时，要使用哪一种方法呢？这里将提供一些拇指法则，1.7 节将更详细讨论证明策略。当我们想要证明形如 $\forall x(P(x) \rightarrow Q(x))$ 的语句时，首先，快速评估一下直接证明是否可行。这通过扩展前提中的定义开始，然后使用它们（以及公理和可用的定理）进行推理。如果直接证明看来没什么希望，做同样的步骤试一下反证。回顾在反证中，要假设条件语句的结论为假，并使用直接证明法表明这意味着前提必为假。例 7 和例 8 演示了这种策略。在给出示例前，先介绍一个有用的定义。

定义 2 若存在整数 p 和 $q(q \neq 0)$ 使得 $r = p/q$ ，实数 r 是有理数。不是有理数的实数称为无理数。

例 7 证明两个有理数的和是有理数。（注意，如果这里包括隐含量词，这个定理我们想证明：“对于任意的实数 r 和 s ，如果 r 和 s 是有理数，则 $r+s$ 是有理数。”）

解 首先尝试直接证明。假设 r 和 s 是有理数。由有理数的定义，可知存在整数 p 和 $q(q \neq 0)$ 使得 $r = p/q$ ，存在整数 t 和 $u(u \neq 0)$ 使得 $s = t/u$ 。我们能利用这个信息证明 $r+s$ 是有理数吗？很显然，下一步是相加 $r = p/q$ 和 $s = t/u$ ，得到

$$r + s = \frac{p}{q} + \frac{t}{u} = \frac{pu + qt}{qu}$$

因为 $q \neq 0$ 且 $u \neq 0$ ，所以 $qu \neq 0$ 。因此， $r+s$ 已经被表示为两个整数 $pu+qt$ 和 qu 的比，其中 $qu \neq 0$ 。这表示 $r+s$ 是有理数。我们已经完成了两个有理数的和是有理数的证明，直接证明成功了。

例 8 证明：如果 n 是整数且 n^2 是奇数，则 n 是奇数。

解 首先尝试直接证明。假设 n 是整数且 n^2 是奇数。那么，存在整数 k 使得 $n^2 = 2k+1$ 。我们能利用这个信息证明 n 是奇数吗？看来没有明显可行的方法，因为求解 n 会得出等式 $n = \pm \sqrt{2k+1}$ ，这没有什么用。

因为直接证明没有达到目的，下面尝试反证。我们将语句“ n 不是奇数”作为前提。因为每个整数不是奇数便是偶数，这意味着 n 为偶数。这表示存在整数 k 使得 $n = 2k$ 。要证明这个定理，我们需证明这个前提蕴含着“ n^2 不是奇数”的结论，即 n^2 是偶数。我们能利用 $n = 2k$ 实现这个目标吗？对这个等式的两边取平方，可得 $n^2 = 4k^2 = 2(2k^2)$ ，这表示 n^2 也是偶数，因为 $n^2 = 2t$ （其中 $t = 2k^2$ ）。我们已经证明了如果 n 是整数且 n^2 是奇数，则 n 是奇数。反证成功了。

1.6.7 归谬证明

假设我们想证明 p 是真的，假定可以找到矛盾式 q 使得 $\neg p \rightarrow q$ 为真，因为 q 是假的， $\neg p \rightarrow q$ 为真，我们能够得出 $\neg p$ 必然为假，这意味着 p 为真。怎样才能找到矛盾式 q ，使其帮助我们证明 p 是真的？

因为无论 r 是什么命题， $r \wedge \neg r$ 都是矛盾式。如果我们能够证明对一些命题 r ， $\neg p \rightarrow (r \wedge \neg r)$ 为真时，就能证明 p 是真的。这种类型的证明称为归谬证明。由于归谬证明不能直接证明一个结果，因此它是另一种间接证明的类型。

下面给出几个归谬证明的例子。第一个例子是鸽巢原理（将在 3.2 节深入介绍）的应用。

例 9 证明：在任意 22 天中至少有 4 天属于一个星期的同一天。

解 令 p 为命题“在任意 22 天中至少有 4 天属于一个星期的同一天”。假设 $\neg p$ 为真。这意味着 22 天中至多有 3 天属于一个星期的同一天。因为一个星期有 7 天，所以至多可以选择 21 天。这样得到一个与 22 天的选择矛盾的假说。也就是说，如果 r 是命题“22 天被选择”，当知道 p 是真的，那么我们能够推出 $\neg p \rightarrow (r \wedge \neg r)$ 。因此，已经证明了任意 22 天中至少有 4 天属于一个星期的同一天。

例 10 通过给出归谬证明来证明： $\sqrt{2}$ 是无理数。

解 设 p 是命题“ $\sqrt{2}$ 是无理数”。假定 $\neg p$ 为真。从归谬证明开始，假定 $\neg p$ 为真。注意 $\neg p$ 表示“并非 $\sqrt{2}$ 是无理数”，这就是说 $\sqrt{2}$ 是有理数。我们将要证明 $\neg p$ 为真的假设导致矛盾。

如果 $\sqrt{2}$ 是有理数，则存在整数 a 和 b 满足 $\sqrt{2} = a/b$ ，其中 a 和 b 没有公因子（所以分数 a/b 是既约的）（这里用到了事实：每个有理数都能写成既约分数）。因为 $\sqrt{2} = a/b$ ，所以当这个等式的两端都平方时，就得出 $2 = a^2/b^2$ 。因此， $2b^2 = a^2$ 。

根据偶数的定义，则 a^2 是偶数。下面应用事实：若 a^2 是偶数，则 a 也一定是偶数。另外，因为 a 是偶数，由偶数的定义，所以对某个整数 c ，有 $a = 2c$ 。因此 $2b^2 = 4c^2$ 。

等式两边除以 2 得： $b^2 = 2c^2$ 。

由偶数定义，这意味着 b^2 是偶数。再应用事实：如果一个整数的平方是偶数，那么这个数自身也一定是偶数。因此，我们得出 b 也必然是偶数。

现在，已经证明假设 $\neg p$ 导致等式 $\sqrt{2} = a/b$ ，其中 a 和 b 没有公因子，但 a 和 b 都是偶数，也就是说 2 整除 a 和 b 。注意到命题 $\sqrt{2} = a/b$ ，其中 a 和 b 没有公因子，实际上意味着 2 不能整除 a 和 b 。由于对 $\neg p$ 的假设导致 2 整除 a 和 b 与 2 不能整除 a 和 b 矛盾， $\neg p$ 一定是假的。也就是说，命题 p ：“ $\sqrt{2}$ 是无理数”是真的。我们完成了证明 $\sqrt{2}$ 是无理数。 ■

归谬证明能够用于证明条件语句。在这样的证明中，首先假设结论的否定。然后应用定理前提和结论否定得到一个矛盾式。（这样证明是有效的推理原因是依据 $p \rightarrow q$ 和 $(p \wedge \neg q) \rightarrow F$ 的逻辑等价。为发现这些语句等价，简单注意每个在特定的一种情况下为假，即当 p 为真且 q 为假时。）

注意，对一个条件语句的反证可以改写成归谬证明。在 $p \rightarrow q$ 的反证里，假定 $\neg q$ 为真而证明 $\neg p$ 也必然为真。为了把 $p \rightarrow q$ 的间接证明改写成归谬证明，假定 p 和 $\neg q$ 都为真。然后利用 $\neg q \rightarrow \neg p$ 的证明步骤，来证明 $\neg p$ 也必然为真。这样就得出矛盾式 $p \wedge \neg p$ ，由此完成归谬证明。例 11 说明如何把对条件语句的反证改写成归谬证明。

例 11 给出定理“若 $3n+2$ 是奇数，则 n 是奇数”的归谬证明。

解 假定 p 表示“ $3n+2$ 是奇数”， q 表示“ n 是奇数”。为构造归谬证明，假设 p 和 $\neg q$ 都为真。也就是假设 $3n+2$ 是奇数而 n 不是奇数。因为 n 不是奇数，所以 n 是偶数。按照在例 3 解答里步骤（归谬证明），可以证明若 n 为偶数则 $3n+2$ 是偶数。首先因为 n 偶数，存在整数 k 使得 $n = 2k$ 。这意味着 $3n+2 = 3(2k)+2 = 6k+2 = 2(3k+1)$ 。由于 $3n+2$ 是 $2t$ ，这里 $t = 3k+1$ ， $3n+2$ 是偶数。注意到 $\neg p$ 表示“ $3n+2$ 是偶数”，因为一个整数是偶数当且仅当它不是奇数。由于设 p 和 $\neg p$ 都为真，得出一个矛盾式。这完成了一个归谬证明，证明了如果 $3n+2$ 是奇数，则 n 是奇数。证毕。 ■

注意，也能用矛盾式证明 $p \rightarrow q$ 是真的，通过假设 p 和 $\neg q$ 都为真，来证明 q 也一定为真。这意味着 q 和 $\neg q$ 都为真，矛盾。这一点告诉我们，能够转变一个直接证明为归谬证明。

等价性证明 为了证明一个定理是双条件的，即形如 $p \leftrightarrow q$ 的语句，我们来证明 $p \rightarrow q$ 和 $q \rightarrow p$ 都是真的。这个方法的有效性建立在

$$(p \leftrightarrow q) \leftrightarrow [(p \rightarrow q) \wedge (q \rightarrow p)]$$

是重言式。

例 12 证明定理“整数 n 是奇数当且仅当 n^2 是奇数”。

Extra Examples **解** 这个定理是形如“ p 当且仅当 q ”，其中 p 是“ n 是奇数”而 q 是“ n^2 是奇数”。（通常不明确地处理为全称量化。）为了证明这个定理，需要证明 $p \rightarrow q$ 和 $q \rightarrow p$ 都为真。

已经证明了（在例 1 里） $p \rightarrow q$ 为真且 $q \rightarrow p$ 为真（在例 8 里）。

因为已经证明了 $p \rightarrow q$ 和 $q \rightarrow p$ 都为真，所以就已经证明了这个定理为真。 ■

有时候定理说几个命题都是等价的。这样的定理说命题 $p_1, p_2, p_3, \dots, p_n$ 都是等价的。这个定理可以写成

$$p_1 \leftrightarrow p_2 \leftrightarrow \dots \leftrightarrow p_n$$

它说所有的 n 个命题都具有相同的真值，因此对所有的 i 和 j ，其中 $1 \leq i \leq n$ ， $1 \leq j \leq n$ ， p_i 和 p_j 是等价的。证明这些命题互相等价的一种方式是使用永真式

$$[p_1 \leftrightarrow p_2 \leftrightarrow \cdots \leftrightarrow p_n] \leftrightarrow [(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_3) \wedge \cdots \wedge (p_n \rightarrow p_1)]$$

这个永真式说明，若可以证明条件语句 $p_1 \rightarrow p_2$ ， $p_2 \rightarrow p_3$ ， $\cdots$ ， $p_n \rightarrow p_1$ 都为真，则命题 p_1 ， p_2 ， p_3 ， $\cdots$ ， p_n 都是等价的。

这更加充分地证明对所有的 $i \neq j$ ， $1 \leq i \leq n$ ， $1 \leq j \leq n$ ，都有 $p_i \rightarrow p_j$ 。

当要证一组命题等价时，我们可以建立一个条件语句链，只要能够证明链中任一个推出另一个。例如，通过证明 $p_1 \rightarrow p_3$ 、 $p_3 \rightarrow p_2$ 、 $p_2 \rightarrow p_1$ ，就能够证明 p_1 、 p_2 、 p_3 是等价的。

例 13 对于整数 n ，证明下列三个语句等价：

p_1 ： n 是偶数

p_2 ： $n-1$ 是奇数

p_3 ： n^2 是偶数

解 为了证明这些语句等价，证明条件语句 $p_1 \rightarrow p_2$ ， $p_2 \rightarrow p_3$ 和 $p_3 \rightarrow p_1$ 都为真。

将用直接证明来证明 $p_1 \rightarrow p_2$ 为真。假定 n 为偶数，则对某个整数 k 来说，有 $n=2k$ 。因此， $n-1=2k-1=2(k-1)+1$ 。这意味着 $n-1$ 是奇数，因为它形如 $2m+1$ ，其中 $m=k-1$ 。

仍用直接证明来证明 $p_2 \rightarrow p_3$ 。现在假定 $n-1$ 是奇数，则对某个整数 k 来说，有 $n-1=2k+1$ 。因此， $n=2k+2$ ， $n^2=(2k+2)^2=4k^2+8k+4=2(2k^2+4k+2)$ 。这意味着 n^2 是整数 $2k^2+4k+2$ 的 2 倍，所以 n^2 是偶数。

要证明 $p_3 \rightarrow p_1$ ，用反证法。也就是说，证明如果 n 不是偶数，则 n^2 也不是偶数。这等同于证明如果 n 是奇数，那么 n^2 是奇数，这在例 1 中已证明过了。证毕。 ■

反例 在 1.3 节曾提到，证明形如 $\forall x P(x)$ 的语句为假，只要能找到一个反例，即存在一个例子使 $P(x)$ 为假。当提交一个形如 $\forall x P(x)$ ，我们相信它是假的，或者排斥所有的企图证明，就寻找反例。例 14 说明了反例的应用。

例 14 证明语句“每个正整数都是两个整数的平方和”为假。

反例

解 为了证明此语句为假，我们寻找一个反例，即寻找一个特殊的整数，它不是两个数的平方和。很快发现反例，3 不能写成两个数的平方和。为表达出这是事实，请注意可用的不超过 3 的平方数只有 $0^2=0$ 、 $1^2=1$ ，而 0、1 的每两项相加都得出 3，因此证明了“每个正整数都是两个整数的平方和”为假。 ■

1.6.8 证明中的错误

在构造数学证明时，容易犯一些错误。这里简述其中的一些。这当中最常见的错误是算术和基本代数方面的。甚至专门的数学家也会犯这种错误，尤其是在处理复杂的公式时。任何时候进行这样的计算都应当尽可能仔细地检查。（你应当复习基本代数中让你头疼的方面。）

数学证明的每一步都应当是正确的，并且结论要从之前的步骤中逻辑地导出。许多错误是源于引入不是前面步骤的逻辑推导。下面的例 15~17 说明了这一点。

例 15 下面这个著名的假定“证明”（即 $1=2$ ）错在哪里？

“证明”步骤如下，其中 a 和 b 是两个相等的正整数。

步骤

理由

1. $a=b$

给定的前提

2. $a^2=ab$

步骤 1 两边乘以 a

3. $a^2-b^2=ab-b^2$

步骤 2 两边减去 b^2

4. $(a-b)(a+b)=b(a-b)$

步骤 3 两边分解因式

5. $a+b=b$

步骤 4 两边除以 $a-b$

6. $2b=b$

步骤 5 把 a 替换成 b ，因为 $a=b$ 并化简

7. $2=1$

步骤 6 两边除以 b

解 除了步骤5两边除以 $(a-b)$ 之外，每个步骤都有效，错误在于 $a-b$ 等于零；一个等式两边用同一个数相除只有在除数不是零时才是有效的。

例 16 下面这个“证明”错在哪里？

“定理”：若 n^2 是正数，则 n 是正数。

“证明”：假定 n^2 是正数。因为条件命题“若 n 是正数，则 n^2 是正数”为真，所以可以得出 n 是正数。

解 令 $P(n)$ 为“ n 是正数”， $Q(n)$ 为“ n^2 是正数”。则前提是 $Q(n)$ 。命题“如果 n 是正数，则 n^2 是正数”也就是 $\forall n(P(n) \rightarrow Q(n))$ 。从前提 $Q(n)$ 和 $\forall n(P(n) \rightarrow Q(n))$ 不能得出 $P(n)$ ，因为没有使用有效的推理规则。相反，这是一个断言结论的谬误的示例。一个反例是当 $n=-1$ 时， $n^2=1$ 为正数，但 n 却是负的。

例 17 下面的“证明”错在哪里？

“定理”：如果 n 不是正数，则 n^2 不是正数。

“证明”：假定 n 不是正数。因为条件语句“如果 n 是正数，则 n^2 是正数”为真，所以可得 n^2 不是正数。

解 如例16那样设定 $P(n)$ 和 $Q(n)$ 。则前提是 $\neg P(n)$ ，语句“如果 n 是正数，则 n^2 是正数”即是 $\forall n(P(n) \rightarrow Q(n))$ 。从前提 $\neg P(n)$ 和 $\forall n(P(n) \rightarrow Q(n))$ 不能得出 $\neg Q(n)$ ，因为没有使用有效的推理规则。相反，这是一个否定前提的谬误示例。如例16那样， $n=-1$ 即为反例。

最后，简要讨论一种比较难应付的错误。许多不正确的论证都基于一种称为偷用论题的谬误。当证明的一个或多个步骤基于待证明命题的正确性时，会发生这样的谬误。换句话说，当命题使用自身或等价于自身的命题来进行证明时会产生这种谬误。所以这种谬误也称为循环论证。

例 18 下面的论证是否正确？

假定 n^2 是偶数，则存在某个整数 k 使 $n^2=2k$ 。令存在某个整数 l 使得 $n=2l$ 。这证明了 n 是偶数。

解 这个论证不正确。证明中没有给出论证说明 n 可以写为 $2l$ ， l 为某整数。这是一个循环论证，因为这个命题等价于待证的命题（即 n 是偶数）。当然，结果本身是正确的，只是证明方法不对。

在证明中犯错是学习过程的一部分。当你犯了某个错误并被别人发现时，应该仔细分析哪里出了错误并确保不再犯同样的错误。

1.6.9 仅仅是开始

现在已经开发了一个基本的证明方法库。在下一节将介绍其他重要的证明方法。在第3章将介绍组合证明的概念。

本节介绍了几个形如 $\forall x(P(x) \rightarrow Q(x))$ 的定理证明方法，包括直接证明和反证法。存在着许多定理，通过直接利用前提和定理里的名词的定义，就容易找出其证明。不过，通常要是不借助于灵活地利用反证—归谬证明，或一些其他的证明技术，就难以证明一个定理。在1.7节中，将讲述证明策略。我们将描述当直接方法行不通时的各种可以用于证明的方法。构造证明是一种只能通过经验学习的艺术，包括写证明，让他人评论你的证明，以及读和分析其他证明。

练习

1. 用直接证法证明：两个奇数之和是偶数。
2. 用直接证法证明：偶数的平方是偶数。
3. 证明：如果 $m+n$ 和 $n+p$ 都是偶数，其中 m 、 n 和 p 都是整数，那么 $m+p$ 也是偶数。你用的是什么证明方法？
4. 用直接证法证明：每个奇数都不同于2的平方。
5. 证明：一个无理数与一个有理数之和是无理数，使用归谬证明。
6. 证明或反驳：两个无理数之积是无理数。
7. 证明：若 x 是无理数，则 $1/x$ 是无理数。
8. 使用归谬证明：如果 $x+y \geq 2$ ，这里 x 、 y 是实数，那么 $x \geq 1$ ，或者 $y \geq 1$ 。
9. 证明：若 n 是整数而且 n^3+5 是奇数，则 n 是偶数，使用

a)反证法证明

b)归谬证明

10. 证明命题 $P(0)$, 其中 $P(n)$ 是命题“若 n 是个大于 1 的正整数, 则 $n^2 > n$ ”。你使用什么类型的证明?
11. 设 $P(n)$ 是命题“若 a 和 b 是正实数, 则 $(a+b)^n \geq a^n + b^n$ ”。证明 $P(1)$ 为真。你使用什么类型的证明?
12. 证明: 在任意 64 天中至少有 10 天指向一个星期的同一天。
13. 用归谬法证明: 没有有理数 r 使得 $r^3 + r + 1 = 0$ 。[提示: 假设 $r = a/b$ 是一个根, 这里 a, b 是整数, a/b 是既约的。通过乘以 b^3 , 得到一个整数的等式。那么看看是否 a 和 b 是某一个奇数或偶数。]
14. 证明: 若 n 是正整数, 则 n 是奇数当且仅当 $5n+6$ 是奇数。
15. 证明或反驳: 若 m 和 n 是整数使得 $mn=1$, 则 $m=1$ 并且 $n=1$, 或者 $m=-1$ 并且 $n=-1$ 。
16. 证明下面三个语句等价: (i) $3x+2$ 是偶数, (ii) $x+5$ 是奇数, (iii) x^2 是偶数。
17. 证明下面三个语句等价: (i) x 是无理数, (ii) $3x+2$ 是无理数, (iii) $x/2$ 是无理数。
18. 下列求解方程 $\sqrt{x+3}=3-x$ 的步骤是否正确? (1) $\sqrt{x+3}=3-x$ 是给定的; (2) $x+3=x^2-6x+9$, 对 (1) 的两边平方; (3) $0=x^2-7x+6$, 从 (2) 的两边都减去 $x+3$; (4) $0=(x-1)(x-6)$, 对 x^2-7x+6 进行因式分解; (5) $x=1$ 或 $x=6$, 因为 $ab=0$ 蕴含 $a=b$ 或 $b=0$, 故从 (4) 可得到解。
19. 证明: 可以通过证明条件语句 $p_1 \rightarrow p_4, p_3 \rightarrow p_1, p_4 \rightarrow p_2, p_2 \rightarrow p_5$ 和 $p_5 \rightarrow p_3$ 来证明命题 p_1, p_2, p_3, p_4 和 p_5 等价。
20. 证明: 在实数 $a_1, a_2, \dots, a_n$ 中至少有一个数大于或等于这些数的平均值。你使用什么类型的证明?
21. 证明: 若 n 是整数, 则下面 4 个语句等价: (i) n 是偶数; (ii) $n+1$ 是奇数; (iii) $3n+1$ 是奇数; (iv) $3n$ 是偶数。

1.7 证明的方法和策略

1.7.1 引言

在 1.6 节介绍了各种不同的证明方法, 并说明了每一种方法如何使用。在这一节, 继续这方面的讨论。我们将介绍几种其他重要的证明方法, 包括考虑个别不同案例的证明和具有预期性质物体的存在性证明。

在 1.6 节只简要讨论了构造证明的策略。这些策略包括选择证明方法, 然后以这个方法为基础, 一步步地成功建立论据。本节在我们已经成功建立一个广泛的证明方法库之后, 将研究一些艺术和科学证明之外的方面。对于如何寻找一个定理的证明方法将提供一些忠告。我们也将描述一些窍门, 包括如何从工作背景中发现证明方法, 并适应存在性证明。

对于数学家的工作, 他们制定猜想, 企图证明或驳斥。这里将简要描述瓦片与骨牌和其他类型的棋盘棋子证明结果的过程。考察这种类型的瓦片, 我们将能够迅速制定猜想和证明定理而不先发展理论。

我们将通过开放问题的角色讨论得到这部分内容。特别地, 要讨论一些有趣的问题, 它们或者是开发数百年后已经被解决的, 或者仍然保持开放的问题。

1.7.2 穷举证明和分情形证明

有时用单一论据通过保持所有的可能情况并不能证明定理。现在介绍一种通过分别考虑不同的情况, 能够用于证明定理的方法。这个方法建立在现在要介绍的一个推理规则上。为了证明形如

$$(p_1 \vee p_2 \vee \dots \vee p_n) \rightarrow q$$

的条件语句, 可以用永真式 $[(p_1 \vee p_2 \vee \dots \vee p_n) \rightarrow q] \leftrightarrow [(p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge \dots \wedge (p_n \rightarrow q)]$ 作为推理规则。这个推理规则说明, 可以通过分别证明每个条件语句 $p_i \rightarrow q (i=1, 2, \dots, n)$ 来证明由命题 $p_1, p_2, \dots, p_n$ 的析取式组成前提的原条件语句。这种论证称为分情形证明。有时为了证明条件语句 $p \rightarrow q$ 为真, 方便的做法是用析取式 $p_1 \vee p_2 \vee \dots \vee p_n$ 代替 p 作为条件语句的前提, 其中 p 与 $p_1 \vee p_2 \vee \dots \vee p_n$ 等价。

穷举证明 有些定理能够通过有关的小数量例子测试来证明。这样的证明叫做穷举证明。因为这些证明通过穷尽所有可能进行。一个穷举证明是分情形证明的特殊类型, 这里每一种情形都作为独立例子检查。现在对穷举证明给些说明。

例1 证明：当 n 是一个正整数，且 $n \leq 4$ 时， $(n+1)^3 \geq 3^n$ 。

Extra Examples

解 用穷举证明。我们只需检验当 $n=1, 2, 3, 4$ 时， $(n+1)^3 \geq 3^n$ 不等式。对于 $n=1$ ， $(n+1)^3 = 2^3 = 8$ ，且 $3^n = 3^1 = 3$ ；对于 $n=2$ ， $(n+1)^3 = 3^3 = 27$ ，且 $3^n = 3^2 = 9$ ；对于 $n=3$ ， $(n+1)^3 = 4^3 = 64$ ，且 $3^n = 3^3 = 27$ ；对于 $n=4$ ， $(n+1)^3 = 5^3 = 125$ ，且 $3^n = 3^4 = 81$ 。在这四种情况的每一种情形，都得到 $(n+1)^3 \geq 3^n$ 。用穷举证明法方法证明了当 n 是一个正整数，且 $n \leq 4$ 时， $(n+1)^3 \geq 3^n$ 。

例2 证明：在 100 以内，连续的正整数是全幂数的只有 8 和 9（全幂数是指它能写成 n^a ，其中 a 是大于 1 的整数）。

解 可以通过证明在 $n < 100$ 时，只有一对连续的正整数 $n, n+1$ 它们都是全幂数来证明这个事实，即 $n=8$ 时。我们能够证明此事实，通过检验在 100 以内的正整数 n ，首先检查 n 是否是全幂数。如果是，检查 $n+1$ 是否也是全幂数。做这件事最快的方法是简单地看 100 以内所有的全幂数，检查是否它下一个大的整数也是全幂数。100 以内正整数的平方有 1、4、9、16、25、36、49、64、81 和 100。100 以内正整数的立方有 1、8、27 和 64。100 以内正整数的 4 次幂有 1、16 和 81。100 以内正整数的 5 次幂有 1 和 32，100 以内正整数的 6 次幂有 1 和 64。除了 1 以外，100 以内没有正整数的 6 次幂大于 6 次。观察不超过 100 的一系列全幂数，发现只有 $n=8$ 时， n 是全幂数，而 $n+1$ 也是全幂数。即 $2^3=8$ ， $3^2=9$ ，是 100 以内唯一两个连续全幂数。证毕。

当只需要检查有关的小数量命题例子的时候，人们可以进行穷举证明。当要求计算机检查非常庞大的命题例子时，它不会抱怨，但它们仍然有局限性。注意，当不能列出所有要检查的例子时，计算机甚至不能检查所有的例子。

分情形证明 分情形证明一定要覆盖定理中出现的所有可能情况。我们用一组例子分情形说明，在每一个例子中，我们应该检查所有覆盖的可能性。

例3 证明：当 n 为整数时，有 $n^2 \geq n$ 。

解 对每一个正整数，通过考虑 $n=0$ ， $n \geq 1$ 和 $n \leq -1$ 三种情况来证明 $n^2 \geq n$ 。分解证明为三种情况：因为通过考虑 0，正整数和负整数可以直接证明这个结果。

Extra Examples

情况(i) 当 $n=0$ 时，因为 $0^2=0$ ，从而 $0^2 \geq 0$ 。这表明在这种情况下， $n^2 \geq n$ 是真的。

情况(ii) 当 $n \geq 1$ 时，把不等式 $n \geq 1$ 两边同时乘以正整数 n ，得到 $n \cdot n \geq n \cdot 1$ 。这表明对 $n \geq 1$ ，有 $n^2 \geq n$ 。

情况(iii) 当 $n \leq -1$ 时，然而 $n^2 \geq 0$ ，因而有 $n^2 \geq n$ 。

因为在所有的三种情况中，有不等式 $n^2 \geq n$ ，能够得出结论：当 n 为整数时，有 $n^2 \geq n$ 。

例4 用分情形证明法证明 $|xy| = |x||y|$ ，其中 x 和 y 是实数。（ $|a|$ 是 a 的绝对值，当 $a \geq 0$ 时， $|a| = a$ ；当 $a \leq 0$ 时， $|a| = -a$ 。）

解 在这个定理的证明中，用事实：当 $a \geq 0$ 时， $|a| = a$ ；当 $a \leq 0$ 时， $|a| = -a$ ，消除绝对值。由于 $|x|$ 和 $|y|$ 两个在公式中，将需要分成四种情况：(i) x 和 y 都非负；(ii) x 非负， y 是负的；(iii) x 是负的， y 非负；(iv) x 是负的， y 是负的。

（注意：我们通过每一种情况的恰当的符号选择去掉绝对值符号。）

情况(i) 我们看到 $p_1 \rightarrow q$ ，因为当 $x \geq 0$ 且 $y \geq 0$ 时 $xy \geq 0$ ，因此 $|xy| = xy = |x||y|$ 。

情况(ii) 要证明 $p_2 \rightarrow q$ ，注意若 $x \geq 0$ 且 $y < 0$ ，则 $xy \leq 0$ ，因此 $|xy| = -xy = x(-y) = |x||y|$ 。（因为 $y < 0$ ，我们有 $|y| = -y$ 。）

情况(iii) 要证明 $p_3 \rightarrow q$ ，可遵循前一种情形的推理过程，只需将 x 和 y 的角色互换。

情况(iv) 要证明 $p_4 \rightarrow q$ ，注意当 $x < 0$ 且 $y < 0$ 时， $xy > 0$ 。因此 $|xy| = xy = (-x)(-y) = |x||y|$ 。

因为已经完成了所有的四项，这些情况包含了所有的可能情况。能够得出结论当 x 和 y 是实数时， $|xy| = |x||y|$ 。

利用分情形证明 已经给出的说明分情形证明的例子提供了一些何时应用这种证明方法的启发。特别地，当同时考虑所有的证明情况不可能时，应该考虑分情形证明。何时应该应用这种证

明方法呢？一般地，当没有明显的思路开始证明，而每一种情况的额外信息又能推进证明，这时寻求分情形证明。例5说明了如何有效地应用分情形证明。

例5 对于十进制数字，一个整数的平方的最末位数字，给出一个猜想，并证明你的结论。

解 最小的完全平方数分别是1、4、9、16、25、36、49、64、81、100、121、144、169、196、225，等等。注意到，完全平方数的最末位数字是：0、1、4、5、6和9，而2、3、7、8从不出现在完全平方数的最末位数字。猜想这样的结论：一个完全平方数的最末位数字是：0、1、4、5、6或9。如何证明这个结论呢？

首先注意到，把整数 n 表示为 $10a+b$ ，这里 a 和 b 是正整数， b 是0、1、2、3、4、5、6、7、8或9。这里 a 是从 n 中分解出10的倍数，从此抽取出 n 的十位数获得的整数。其次， $(10a+b)^2 = 100a^2 + 20ab + b^2 = 10(10a^2 + 2ab) + b^2$ ，因而， n^2 的十进制最末位数字与 b^2 的十进制最末位数字相同。进一步， b^2 的最末位十进制数字与 $(10-b)^2 = 100 - 20b + b^2$ 相同。因此，把证明缩减为以下6种情况：

情况(i) n 的最末尾数字是1或9，这样 n^2 的最后十进制数字是 $1^2=1$ ，或 $9^2=81$ ，即为1。

情况(ii) n 的最末尾数字是2或8，这样 n^2 的最后十进制数字是 $2^2=4$ ，或 $8^2=64$ ，即为4。

情况(iii) n 的最末尾数字是3或7，这样 n^2 的最后十进制数字是 $3^2=9$ ，或 $7^2=49$ ，即为9。

情况(iv) n 的最末尾数字是4或6，这样 n^2 的最后十进制数字是 $4^2=16$ ，或 $6^2=36$ ，即为6。

情况(v) n 的十进制最末尾数字是5，这样 n^2 的十进制最后数字是 $5^2=25$ ，即为5。

情况(vi) n 的最末尾数字是0，这样 n^2 的十进制最后数字是 $0^2=0$ ，即为0。

因为考虑了所有的6种情况，能够得出：当 n 是整数， n^2 的十进制最后数字是0、1、4、5、6或9。 ■

在分情形证明中，有时能估计所有的，但有很少的几个例子。例6就是如此。

例6 证明：对于整数 x 、 y ， $x^2+3y^2=8$ 没有解。

解 由于当 $|x| \geq 3$ 时， $x^2 > 8$ ，同时当 $|y| \geq 2$ 时， $3y^2 > 8$ 。因此能够很快缩减证明，只检查几种简单的情况。这样只剩下 x 取值-2、-1、0、1、2，而 y 取值-1、0、1的情况，我们就能给出一个全面的证明。对于保留的情况，注意到， x^2 的可能取值是0、1、4， $3y^2$ 的可能取值是0和3， x^2 与 $3y^2$ 可能取值的最大和是7。因此，当 x 、 y 是整数时， $x^2+3y^2=8$ 是不可能的。 ■

不失一般性 在例4的证明中，除掉情形(iii) $x < 0$ ， $y \geq 0$ ，因为它与情形(ii) $x \geq 0$ ， $y < 0$ 的 x 和 y 角色互换后相同。为了简短证明，不失一般性，我们可以通过假设，把情形(ii)和(iii)的证明合在一起为 $x \geq 0$ ， $y < 0$ 。这意味着，我们通过情形 $x \geq 0$ ， $y < 0$ 来完成情形 $x < 0$ ， $y \geq 0$ ，道理如上，但要有明显的变化。一般地，术语“不失一般性”用在证明中(缩写为WLOG)时，我们得到如果没有另外的证据要求证明其他指定的情形，可通过证明定理的其中一种情况。也就是说，其他的一系列情况通过简单的变化来论证，或者添加到一些简单的初始步骤中。当然，这个原理的不正确应用会导致不幸的错误发生。有时假设会导致失去一般性。一些假设使得忽略一个情况可以是非常不同于其他情形。这样会导致不完整及不可挽救的证明。事实上，许多著名定理的不正确证明原来依赖于应用“不失一般性”的想法，来论证不能很快从简单情形证明而建立的情形。

现在来说明有效应用不失一般性的证明。

例7 证明： $(x+y)^r < x^r + y^r$ 。这里 x 、 y 是正实数， r 是 $0 < r < 1$ 的实数。

解 不失一般性，假设 $x+y=1$ 。[注意：假设对于 $x+y=1$ 已经证明了定理。设 $x+y=t$ ，那么 $x/t+y/t=1$ ，这意味着 $(x/t+y/t)^r < (x/t)^r + (y/t)^r$ 。最后式子两边乘以 t^r ，证得 $(x+y)^r < x^r + y^r$ 。]

假设 $x+y=1$ ，因为 x 、 y 都为正，所以 $0 < x < 1$ ， $0 < y < 1$ 。因为 $0 < r < 1$ ，因此， $0 < 1-r < 1$ ，故 $x^{1-r} < 1$ ， $y^{1-r} < 1$ 。这意味着 $x < x^r$ ， $y < y^r$ 。因此， $x^r + y^r > x+y=1$ 。从而 $(x+y)^r = 1^r < x^r + y^r$ 。这样对于 $x+y=1$ 证明了定理。

因为不失一般性，我们假设 $x+y=1$ ，得到 $(x+y)^r < x^r + y^r$ 这里 x 、 y 是正实数， r 是 $0 < r < 1$ 的实数。 ■

穷举证明和分情形证明的常见错误 推理的一种常见错误是从例子中得出不正确结论。不管考虑多少不同的例子，除非每一种可能情况都覆盖了，否则不能从例子证明定理。证明定理的问题类似于要证明计算机程序总能输出所期望的。除非所有的输入值都测试了，否则无论测试多少输入值，也不能得到程序总会有正确的输出。

例 8 语句“每个正整数都是 18 个四次方整数之和”是否为真？

解 要判断 n 是否可写为 18 个四次方整数的和，我们先从最小的正整数开始考察。因为整数的四次方分别是 0, 1, 16, 81, …，如果能从这些数中选择 18 个项后相加得 n ，则命题得证。可以证明，从 1 到 78 的所有正整数都可以写成 18 个四次方整数的和（细节留给读者证明）。然而，如果认为这就检查够了，那就会得出错误的结论，因为 79 并不是 18 个四次方整数的和（读者请自行验证）。所以，题设语句不为真。 ■

另一个常见错误是由于未考虑到所有情形，导致不正确的证明，因而做出无根据的假设。如例 9 所示。

例 9 下面的“证明”错在哪里？

“定理”：如果 x 是实数，则 x^2 是正实数。

“证明”：令 p_1 为“ x 是正数”， p_2 为“ x 是负数”， q 为“ x^2 是正数”。要证明 $p_1 \rightarrow q$ ，注意，当 x 是正数时， x^2 为正数，因为两个正数 x 和 x 的积肯定为正。要证明 $p_2 \rightarrow q$ ，注意当 x 是负数时， x^2 是正数，因为 x^2 是两个负数的积。证毕。

解 上面的证明存在的问题是忘了考虑 $x=0$ 的情形。当 $x=0$ 时， $x^2=0$ 非正，因此假设的定理为假。如果 p 是“ x 是实数”，那我们可以分三种情形 p_1 、 p_2 和 p_3 来证明结果，其中 p_1 是“ x 是正数”， p_2 是“ x 是负数”， p_3 是“ $x=0$ ”，因为 $p \leftrightarrow p_1 \vee p_2 \vee p_3$ 。 ■

1.7.3 存在性证明

许多定理都断言存在特定类型的对象。这种类型的定理是形如 $\exists xP(x)$ 的命题，其中 P 是谓词。对形如 $\exists xP(x)$ 的命题的证明称为存在性证明。有多种方式来证明这种类型的定理。有时通过找出一个使得 $P(a)$ 为真的元素 a 来给出 $\exists xP(x)$ 的存在性证明。这样的存在性证明称为构造性的。给出非构造性的证明也是可能的；即不是找出使 $P(a)$ 为真的元素 a ，而是以某种其他方式来证明 $\exists xP(x)$ 为真。给出非构造性证明的一种普通方法是使用归谬证明，证明该存在量词化的否定式蕴含着矛盾。下列例 10 说明构造性的存在性证明的概念。例 11 说明非构造性的存在性证明的概念。

例 10 构造性的存在性证明 证明存在某个正整数，可以用两种不同的方式将其表示为正整数的立方和。

Extra Example

解 经过大量的计算（如使用计算机搜索）可找到

$$1729 = 10^3 + 9^3 = 12^3 + 1^3$$

因为 1729 满足题设要求，所以得证。 ■

例 11 非构造性的存在性证明 证明存在无理数 x 和 y 使得 x^y 是有理数。

解 由 1.6 节例 10 可知 $\sqrt{2}$ 是无理数，考虑数 $\sqrt{2}^{\sqrt{2}}$ 。如果它是有理数，那就存在两个无理数 x 和 y 且 x^y 是有理数，即 $x=\sqrt{2}$ ， $y=\sqrt{2}$ 。另一方面，如果 $\sqrt{2}^{\sqrt{2}}$ 是无理数，那么可以令 $x=\sqrt{2}^{\sqrt{2}}$ 且 $y=\sqrt{2}$ ，因此 $x^y = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{(\sqrt{2} \cdot \sqrt{2})} = \sqrt{2}^2 = 2$ 。

这是一个非构造性存在性证明的例子，即我们并没有找出无理数 x 和 y 使得 x^y 是有理数。而我们证明了或者 $x=\sqrt{2}$ ， $y=\sqrt{2}$ ，或者 $x=\sqrt{2}^{\sqrt{2}}$ ， $y=\sqrt{2}$ ，可能满足性质，但并不知道这两对哪一个满足。 ■

非构造性存在性证明非常微妙，如例 12 所示。

例 12 Chomp 是两个人玩的游戏。在这个游戏中，曲奇饼放在矩形格中。左上角的曲奇饼被毒化了。如图 1-1a 所示。两个玩家轮流移动。每一个玩家都要吃一块留下的曲奇饼，两人都要使右边和/或它下面所有的曲奇饼都被取走（例如如图 1-1b 所示）。如果不要吃毒化

的曲奇饼就没有选择的玩家为失败。请问：两个玩家之一是否有获胜的策略，即其中一个玩家能够一直移动保证其获胜。

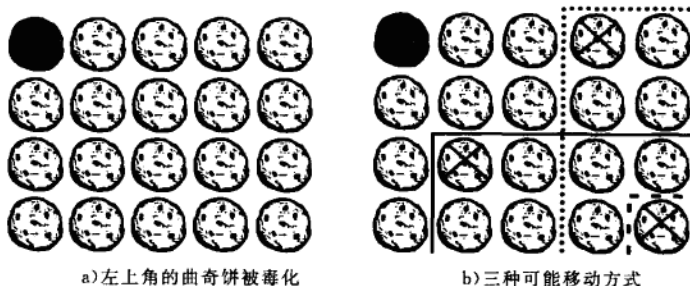


图 1-1 Chomp

解 给第一个玩家获胜策略的非构造性存在证明。也就是说，将证明第一个玩家总有获胜的策略，但对必须怎么移动没有明确的描述。

首先，游戏结束后，不能是一个平局，因为每一步移动至少要吃一块曲奇饼，因此游戏结束不会超过移动 $m \times n$ 步，这里初始网格是 $m \times n$ 。现假设在游戏开始，第一个玩家吃掉了右下角的曲奇饼，这有两种可能，一这是第一个玩家获胜策略的第一步，二或者这是第二个玩家获胜策略的第一步。在第二种情况下，第一个玩家替代恰好要吃右下角的曲奇饼，可以采用第二个玩家获胜策略的第一步相同的步骤(然后继续循环这个获胜策略)。这将保证第一个玩家获胜。

注意，这里证明了获胜策略的存在性，但是没有指定具体的获胜策略。因此，这个证明是非构造性存在性证明。事实上，没有人能够描述 Chomp 的获胜策略，适用于所有的长方形网格通过描述第一个玩家应该遵循的移动方式。然而，获胜策略能够被描述成某种特殊的情况，比如当网格是正方形和当网格只有两行曲奇饼时。

1.7.4 唯一性证明

一些定理断言具有特定性质的元素唯一存在。换句话说，这些定理断言恰有一个元素具有这个性质。要证明这类语句，需要证明有某个元素具有这个性质，且没有其他元素有此性质。唯一性证明的两个部分如下：

存在性：证明存在某个元素 x 具有期望的性质。

唯一性：证明若 $y \neq x$ ，则 y 不具有期望的性质。

等价地，我们可以证明如果 x 和 y 都具有期望的性质，则 $x = y$ 。

注意 证明存在某个唯一元素 x 等同于证明语句 $\exists x(P(x) \wedge \forall y(y \neq x \rightarrow \neg P(y)))$ 。

我们在例 13 中要说明唯一性证明的组成。

例 13 证明：如果 a, b 是实数，并且 $a \neq 0$ ，那么有一个唯一的 r ，使得 $ar + b = 0$ 。

解 首先，注意到实数 $r = -b/a$ 是 $ar + b = 0$ 的一个解，因为 $a(-b/a) + b = -b + b = 0$ 。因此，对于 $ar + b = 0$ 实数 r 是存在的。这是证明的存在性部分。

Example 其次，假设 s 是 $as + b = 0$ 成立的实数。因此 $ar + b = as + b$ ，这里 $r = -b/a$ 。从两边减去 b ，得到 $ar = as$ 。最后式子两边同除以 a ，这里 a 是非零的，得到 $r = s$ 。这意味着如果 $r \neq s$ ，则 $as + b \neq 0$ 。这建立了证明的唯一性部分。

1.7.5 证明策略

发现证明是一项挑战性的工作。当面对待证命题时，应该先用定义替换术语，再仔细分析假设和结论的含义。这样做之后，用一种可用的证明方法去试图证明结果。一般情况下，如果语句是条件语句，就应该首先尝试直接证明；如果这样不行，就尝试间接证明。如果这些方法都不行，就尝试归谬证明。

前推与后推 无论选择什么证明方法，都需要一个起点。条件语句的直接证明就从假设开始。利用这些假设以及公理和已知定理，用导向结论的一系列步骤来构造证明。这类推理称为前推，是用来证明相对简单结果的一类最常见推理。同样，要开始间接证明，就从假设的否定开始，用一系列步骤来得出前提的否定。

但是，前推常常难以用来证明更复杂的结果，因为得出想要的结论所需要的推理可能相当复杂。在这种情况下，使用后推可能会有帮助。为了后推证明命题 q ，就寻找能证明具有性质 $p \rightarrow q$ 的语句 p 。（注意，寻找能证明使得 $q \rightarrow r$ 的语句 r 不会有帮助，因为从 $q \rightarrow r$ 和 r 得出 q 为真，这种推论是错误的。）例 14 和例 15 解释后推。

例 14 给定两个正实数 x 和 y ，其算术均值是 $(x+y)/2$ ，其几何均值是 $\sqrt{xy}$ 。当比较成对不同正实数的算术和几何均值时，发现算术均值总是大于几何均值。（例如，当 $x=4$ 和 $y=6$ 时，有 $5=(4+6)/2 > \sqrt{4 \cdot 6} = \sqrt{24}$ 。）能否证明这个不等式恒为真？

Extra Example 解 要证明当 x 和 y 是不同正实数时 $(x+y)/2 > \sqrt{xy}$ ，可进行回溯，构造一系列等价的不等式（请读者来证明每对相邻的不等式都是等价的）。这些等价的不等式是：

$$\begin{aligned}(x+y)/2 &> \sqrt{xy}, & (x+y)^2/4 &> xy \\ (x+y)^2 &> 4xy, & x^2 + 2xy + y^2 &> 4xy \\ x^2 - 2xy + y^2 &> 0, & (x-y)^2 &> 0\end{aligned}$$

由于当 $x \neq y$ 时， $(x-y)^2 > 0$ ，所以最后一个不等式为真。由于所有这些不等式都等价，所以当 $x \neq y$ 时， $(x+y)/2 > \sqrt{xy}$ 。一旦构造了这个证明，就能轻而易举地构造使用前推的证明。现在给出这个证明。

假设 x 和 y 是两个不同实数，那么 $(x-y)^2 > 0$ ，因为非零实数的平方是正的。由于 $(x-y)^2 = x^2 - 2xy + y^2$ ，这意味着 $x^2 - 2xy + y^2 > 0$ 。两边同时加 $4xy$ ，得 $x^2 + 2xy + y^2 > 4xy$ 。因为 $x^2 + 2xy + y^2 = (x+y)^2$ ，因此 $(x+y)^2 > 4xy$ ，两边同时除以 4 得， $(x+y)^2/4 > xy$ 。最后，两边同时开平方（因为不等式两边都是正的，开方有保序性）得 $(x+y)/2 > \sqrt{xy}$ 。得到，如果 x 和 y 是两个不同实数，那么它们的算术均值 $(x+y)/2$ 大于它们的几何均值 $\sqrt{xy}$ 。 ■

例 15 假定甲、乙两人玩游戏，轮流从最初有 15 块石头的堆中每次取 1、2 或 3 块石头。取最后一块石头的人赢得游戏。证明：无论乙如何取，先取的甲都能赢得游戏。

解 为了证明甲总能赢得游戏，进行回溯。在最后一步，如果石头堆中剩下 1、2 或 3 块石头，甲就能获胜。如果乙不得不从有 4 块石头的堆中取石头，就迫使乙留下 1、2 或 3 块石头。因此，甲要获胜的一种方法是在倒数第二步，应该给乙留下 4 块石头。当乙不得不从 8 块石头的堆中取石头时，此后就会留下 5、6 或 7 块石头，甲再取石头后就能留下 4 块石头。因此，为使乙留下 5、6 或 7 块石头，甲就要在倒数第三步给乙留下 8 块石头。这意味着当轮到甲取时还有 9、10 或 11 块石头。同样，当甲走第一步时，应该留下 12 块石头。把这个论证倒过来就证明了无论乙如何取，甲总是有石头取，所以甲赢得游戏。这些步骤依次给乙留下 12、8 和 4 块石头。 ■

改造现有证明 用于证明语句方法的很好思路是利用现有的证明。一个现有的证明时常适合于证明一个新结果。即使不是这样，这个现有证明想法的一部分也是有帮助的。因为现有证明为新证明提供了线索，你应该去阅读和理解在学习过程中遇到的这些证明。例 16 对这个过程进行了说明。

例 16 在 1.6 节例 10 中证明了 $\sqrt{2}$ 是无理数。现在推测 $\sqrt{3}$ 是无理数。我们能够改造在 1.6 节例 10 的证明来证明 $\sqrt{3}$ 是无理数的吗？

Extra Example 解 为改造在 1.6 节例 10 的证明，开始先模仿这个证明的步骤。但要用 $\sqrt{3}$ 代替 $\sqrt{2}$ 。首

先，假设 $\sqrt{3} = c/d$ ，这里分数 c/d 是既约的。等式的两端平方得到 $3 = c^2/d^2$ 。因此， $3d^2 = c^2$ 。类似在 1.6 节例 10 中由等式 $2b^2 = a^2$ 证明 2 是 a 和 b 的公因子的方法，用这个等式能证明 3 一定是 c 和 d 的公因子吗？（回忆如果 t/s 是整数，则整数 s 是整数 t 的因子。一个整数 n 是偶

数, 当且仅当 2 是 n 的因子。) 这里我们仅勾画一些剩下的证明。因为 3 是 c^2 的因子, 它也必然是 c 的因子。并且, 因为 3 是 c 的因子, 9 就是 c^2 的因子, 这意味着 9 是 $3d^2$ 的因子。因此 3 是 d^2 的因子, 这意味着 3 是 d 的因子。因而 3 是 c 和 d 的因子, 这是矛盾的。之后添加这些步骤的理由, 通过改造 $\sqrt{2}$ 是无理数的证明, 将能证明 $\sqrt{3}$ 是无理数。注意: 这个证明可以推广到 $\sqrt{n}$ 是无理数, 每当 n 是一个非完全平方的正整数。证毕。 ■

当你遇到要证明一个新理论时, 一个好的途径是寻找现有的可以改造的证明, 尤其是当新理论与你已经证明过的理论相似时。

1.7.6 寻找反例

在 1.5 节, 介绍了应用反例证明某些语句是错的。当面对一个推测, 你一定首先试图去证明这个猜测, 如果你的尝试没有成功, 你就要试图找到一个反例。如果你不能找到反例, 你就一定要再试图证明这个语句。无论如何, 寻找反例都是相当重要的方法, 它时常能提供对问题的领悟。下面用一些例子说明反例的作用。

例 17 在 1.6 节例 14 中, 通过寻找反例证明了语句“每个正整数都是两个整数的平方和”为假。也就是说, 有正整数不能写成两个整数的平方和。尽管不能写出每一个正整数作为两个整数平方和, 也许我们能写出每一个正整数作为三个整数平方和。也就是语句“每个正整数都是三个整数的平方和”为真为假呢?

Extra Example **解** 因为我们知道并不是每个正整数都是两个整数的平方和, 可能最初怀疑每一个正整数作为三个整数平方和。因此, 首先寻找反例。也就是说, 证明语句“每个正整数都是三个整数的平方和”为假, 如果能够找到一个特殊的整数不是三个整数的平方和。为寻找反例, 试着写出可满足语句条件的连续的正整数。可以发现 $1=0^2+0^2+1^2$, $2=0^2+1^2+1^2$, $3=1^2+1^2+1^2$, $4=0^2+0^2+2^2$, $5=0^2+1^2+2^2$, $6=1^2+1^2+2^2$, 但无法找到方法将 7 写为三个整数的平方和。要证明 7 是一个反例, 请注意可用的不超过 7 的平方数只有 0、1 或 4, 而三个项(每项为 0、1 或 4)相加却得不出 7, 所以 7 是一个反例。得到结论: 语句“每个正整数都是三个整数的平方和”为假。

我们已经证明了并不是每个正整数都是三个整数的平方和, 下一个问题要问是不是每个正整数都是四个整数的平方和。一些实验提供了证据, 答案是对的。例如, $7=1^2+1^2+1^2+2^2$, $25=4^2+2^2+2^2+1^2$ 和 $87=9^2+2^2+1^2+1^2$ 。这个产生猜测: “每个正整数都是四个整数的平方和”是真的。对于证明参见[R05]。 ■

1.7.7 行动证明策略

仿佛数学事实是刻在石头上的, 一般认为数学是这样。数学教科书(包括这本书的绝大部分)规范地提出定理和它的证明。而在数学中的介绍不揭示发现的过程。这个过程以探讨概念和例子、提问题、创造猜测和企图通过证明或者通过反例解决一些猜测。这些是数学中日常的工作。相信或者不信, 在教科书中所讲授的材料都是以这个方式原始地展开。

人们基于许多类型的可能证据创造猜想。为能够识别可能的模式, 特殊情况下的考试能够导致猜想。改变已知定理的假设和结论也能导致似是而非的猜测。其次, 猜测也建立在直觉和结果所持有的信念的基础上。无论怎样, 猜测产生了, 一旦它被明确描述, 目标就是证明或者驳斥它。当数学家确信猜测是真的, 他们就会努力寻找证明; 如果他们不能找到证明, 他们就会寻找反例。当他们不能找到反例时, 他们又会转回来再次试图证明猜测。尽管许多猜测很快被解决, 而一些猜测要抵制数百年的攻击, 这导致数学新内容的发展。在以后的章节中将提到几个著名的猜想。

1.7.8 填充

Extra Example 通过对棋盘填充的简要研究, 能够说明证明策略的一些方面。研究棋盘的填充是快速发现和证明许多不同结果的很有效方法, 而这些证明用到了各种各样的证明方法。在这个方面也几乎创造了无穷多的猜想和研究。首先, 需要定义一些术语。一个棋盘是一个由水平

和垂直线分成同样大小方块组成的矩形。跳棋游戏是在8行和8列的木板上进行；这个板称为标准棋盘，如图1-2所示。在这一节，用术语**镶板**指任意矩形大小的棋盘，而且棋盘的一部分包含删除一个或更多个方格组成。一个**骨牌**是一块长方形，由两个方格组成，如图1-3所示。当它的所有方格由不重叠的骨牌覆盖，而没有骨牌悬垂在镶板上，我们说一个镶板由骨牌所填充。现在研究一些有关用骨牌填充镶板的结果。

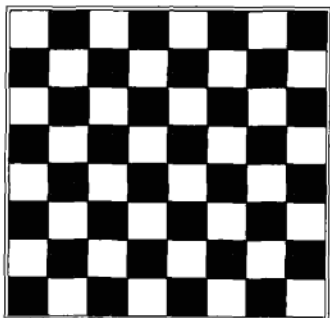


图 1-2 标准棋盘

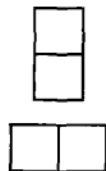


图 1-3 两种骨牌

例 18 我们能用骨牌填充标准棋盘吗？

解 我们找到许多用骨牌填充标准棋盘的方法。例如，可以水平放32块骨牌填充它，如图1-4所示。这样填充的存在性完成了一个构造性的存在证明。当然，做这个填充还有大量其他的方法。可以在板上垂直放32块骨牌，或者水平地和垂直地填充它。但对于一个构造性存在证明需要找到仅仅一个这样的填充就可以。

例 19 我们能填充从标准棋盘中去掉四个角的方格之一得到的镶板吗？

解 为了回答这个问题，注意一个标准棋盘有64个方格，去掉一个方块这样由63个方格产生一个镶板。现在假设能够填充一个从一个标准棋盘中去掉一个角的方格的镶板。因为每一个骨牌盖住两个方格，又没有两个骨牌重叠，没有骨牌悬垂在板上，所以板上一定有偶数个方格。因此，可以用归谬证明法证明去掉一个方格的标准棋盘不能用骨牌填充，因为这样一个板有奇数个方格。

现在考虑一个棘手的情况。

例 20 去掉标准棋盘中左上角和右下角方格得到的镶板，如图1-5所示，能填充这个镶板吗？

解 去掉标准棋盘中两个方格得到的镶板包含 $64 - 2 = 62$ 个方格。因为62是偶数，不能很快排除去掉左上角和右下角方格的标准棋盘填充的存在性，不像例19，排除了去掉一个方格的标准棋盘用骨牌填充的存在性。试图构造这个镶板的填充，读者应该企图的第一个方法一定是成功地放骨牌。然而，无论怎么试验，都不能找到这样的填充。因为我们的努力不能产生一个填充，所以得出一个猜测：填充不存在。

我们一定要试图证明填充不存在，然而证明显示：成功地在镶板上放骨牌已经到了穷途末路。为构造这样的证明，不得不考虑出现的所有可能情况，就像我们做过的成功地放骨牌的所有可能选择一样。例如，覆盖第一行第二栏中的方格有两种选择，而旁边是去掉的左角上方格。我们能用水平放置的填充或者垂直

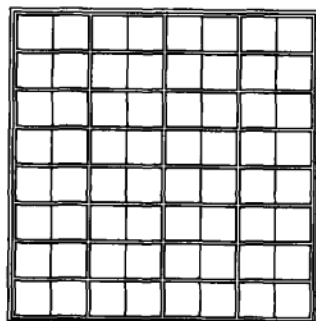


图 1-4 填充标准棋盘

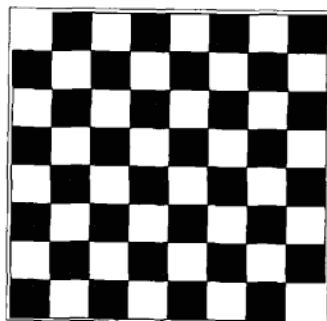


图 1-5 去掉左上角和右下角方格的标准棋盘

放置的填充。这两种选择的每一种都会导致进一步的选择，等等。对于一个人来说，这个并不要很久，就能看到这不是卓有成效的解决计划，尽管计算机通过穷举能够完成这样的证明(练习 11 要求你完成这样的证明，不能填充一个去掉对角的 4×4 棋盘)。

现在需要另一个方法。或许有一个比较容易的方法证明不存在去掉两个对角的填充。正如许多证明，一个关键的观察能启发我们。我们给棋盘的方格涂上颜色，用来改变白和黑方格，如图 1-2 所示。观察得到，对这样镶板的填充，一个骨牌覆盖一个白方格和一个黑方格。其次注意到，这样的镶板白色方格和黑色方格数量不等。我们能用这样的观察通过归谬证明来证明：一个去掉两个对角的标标准棋盘不能用骨牌填充。现在给出这个证明。

证 假如能用骨牌对去掉两个对角的标标准棋盘填充。注意到去掉两个对角的标标准棋盘包含 $64 - 2 = 62$ 个方块。它可以用 $62/2 = 31$ 个骨牌填充。在这个填充中，每个骨牌盖住一个白的和一个黑的方格。因此，这个填充盖住 31 个白的和 31 个黑的方格。然而，当去掉两个对角方格时，或者保留 32 块白的、30 块黑的，或者保留 30 块白的、32 块黑的。这驳斥了能用骨牌覆盖去掉两个对角的标标准棋盘的假设，完成证明。 ■

我们还能用除骨牌之外的其他类型的块来填充。代替骨牌，研究沿着它们的边来连接构建的全等正方形这样相同形状的块来填充。这样的块被称为是多方块。这个术语是由数学家所罗门·戈洛姆在 1953 年杜撰的，他是一本娱乐书籍的作者 [Go94]。两个多方块用同样数量的方形，如果通过旋转和/或翻动四角之一而得到另一个，则将考虑它们相同。例如，有两种类型的三格板(见图 1-6)，它是由三个边连接的正方形组成的多方块。三格板的一种类型是直三格板，它有三个水平连接的正方形；另一个是右三格板，像字母 L 的形状，如果有必要，翻动和/或旋转它。这里将研究用直三格板填充棋盘。

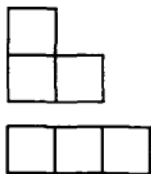


图 1-6 一个右三格板和一个直三格板

例 21 你能用直三格板填充标准棋盘吗？

解 标准棋盘含有 64 个方格，每一个三格板覆盖三个方格。因此，如果三格板填充了一个镶板，镶板的方格数量一定是 3 的倍数。因为 64 不是 3 的倍数，所以三格板不能覆盖 8×8 棋盘。 ■

下面的例 22，考虑了用直三格板填充去掉一个角的标准棋盘问题。

例 22 我们能用直三格板填充去掉四个角的任一个角的标准棋盘吗？一个 8×8 棋盘去掉一个角后包含 $64 - 1 = 63$ 个方格。用直三格板填充任一个去掉一个角的棋盘其四种情况的任一种情况都要用 $63/3 = 21$ 个直三格板。然而当我们试验时，不能找到一个直三格板填充任一个去掉一个角的四种情况的任一种情况。由穷举证明显示这个没有希望。我们能改造例 20 的证明来证明这样的填充不存在吗？

解 我们将给棋盘的方格涂上颜色，企图改造在例 20 给出的归谬证明法证明：应用骨牌填充去掉对角的标标准棋盘是不可能的。因为用直三格板而不是骨牌，我们用三种颜色区分方格而不是两种颜色，如图 1-7 所示。注意到在这个着色中有 21 个灰色方格，21 个黑色方格，22 个白色方格。其次，做一个重要的观察，一个直三格板覆盖棋盘的三个方格，它覆盖一个灰色的、一个黑色的和一个白色的方格。然后注意到三个颜色的每一个都出现在一个角的方格中。于是，在一般没有去掉的时候，可以假设轮换颜色，使去掉的方格是灰色的。因此假设留下的板包含 20 个灰色方格，21 个黑色方格，22 个白色方格。

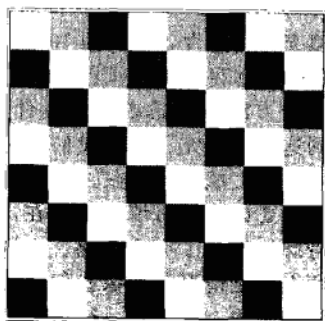


图 1-7 用三种颜色对删除对角的标标准棋盘方格着色

如果能用直三格板填充这块镶板，那么将用 $63/3=21$ 个直三格板。这些直三格板覆盖 21 个灰色方格，21 个黑色方格，21 个白色方格。这与这个镶板包含 20 个灰色方格，21 个黑色方格，22 个白色方格矛盾。因此不能用直三格板填充这个镶板。 ■

1.7.9 未解决问题的作用

在数学中，许多进展是人们在试图解决著名的但尚未解决的问题中而获得的。在过去的 20 年中，许多未解决的问题最后被解决，比如数论中 300 多年前产生的一个猜想的证明。这个猜想断言被称为费尔马大定理的语句为真。

定理 1 费尔马大定理 只要 n 是满足 $n>2$ 的整数，方程

$$x^n + y^n = z^n$$

就没有满足 $xyz \neq 0$ 的整数解 x 、 y 和 z 。

 **注意** 方程 $x^2 + y^2 = z^2$ 有无穷多个整数解 x 、 y 和 z 。这些解称为毕达哥拉斯三元组[⊖]，对应于具有整数边长的直角三角形的边长。

这个问题有一段很有意思的典故。在 17 世纪，费尔马在一本丢番图的著作的空白处匆匆写道，他有了“巧妙的证明”：当 n 是大于 2 的整数时， $x^n + y^n = z^n$ 没有非平凡的整数解。但他从来没有发表过这个证明（费尔马几乎没有发表过任何东西），在他死后留下的文章中也找不到任何证明。数学家们花了 300 年寻找这个证明却没有成功，尽管许多人相信能找到相对简单的证明。（特殊情形的证明，比如欧拉的当 $n=3$ 时情形的证明和费尔马本人的 $n=4$ 情形的证明，已经找到。）年复一年，少数有声誉的数学家认为他们证明了这个定理。在 19 世纪，这些失败的尝试之一导致了所谓代数数论的数论分支的发展。直到 20 世纪 90 年代，当安德鲁·怀尔斯（Adrew Wiles）采用从所谓椭圆曲线论的深奥数论领域中最近发展的思想来证明费尔马大定理时，才找到了几百页长的高等数学的正确证明。从公共电视台 Nova 系列的节目中可以看出，利用这个强有力的理论来寻找费尔马大定理的证明，怀尔斯找了将近 10 年！（感兴趣的读者可以查阅 [Ro05] 来了解关于费尔马大定理的更多信息和关于这个问题及其解决的其他参考资料。）

下面我们给出一个未解决问题，这个问题描述很简单，但却很难求解。

 **例 23 $3x+1$ 猜想** 设 $f(x)$ 是把偶数 x 转换成 $x/2$ 、把奇数 x 转换成 $3x+1$ 的变换。有个著名的猜想（有时称为 $3x+1$ 猜想）说：对所有正整数 x 来说，当反复地应用变换 T 时，最终会得到整数 1。例如，从 $x=13$ 开始，发现 $T(13)=3 \cdot 13+1=40$ ， $T(40)=40/2=20$ ， $T(20)=20/2=10$ ， $T(10)=10/2=5$ ， $T(5)=3 \cdot 5+1=16$ ， $T(16)=8$ ， $T(8)=4$ ， $T(4)=2$ ， $T(2)=1$ 。对于直到 $5.6 \cdot 10^{13}$ 的所有整数都验证了 $3x+1$ 猜想。

$3x+1$ 猜想具有有趣的历史，从 20 世纪 50 年代以来就吸引了数学家们的注意力。这个猜想被多次提出，具有许多其他名称，包括：Collatz 问题、Hasse 算法、Ulam 问题、Syracuse 问题以及 Kakutani 问题等。许多数学家抛开原有工作来花时间解决这个猜想。这引起一则笑话，说这个问题是旨在减缓美国数学研究的阴谋的一部分。参见 Jeffrey Lagaris 的文章 [La85] 来了解对这个问题有意思的讨论以及试图解决这个问题的数学家们所发现的结果。 ■

1.7.10 其他证明方法

本章介绍了证明中使用的基本方法，也描述了如何利用这些方法来证明各种结果。在第 2 章，将用这些证明方法证明有关集合、函数、算法和数论的结果。在这些定理中，将证明著名的停机定理，即存在不能用任何过程来解决的问题。但除了这些讨论过的方法之外还有许多重要的证明方法。2.4 节会介绍康托尔对角化方法，用来证明关于无穷集大小的结果。第 3 章介绍组合证明的概念，用来证明采用计数论证的结果。读者应当注意已经有相关书籍专门描述本节讨论的内容，包括乔治·波利亚（George Polya）的许多优秀著作（[Po61]，[Po71]，[Po90]）。

⊖ 也叫做勾股数组。——译者注

最后, 请注意: 本书没有给出在数学中能够用于证明定理的步骤。这是一个不存在这样步骤的深奥的数理逻辑定理。

练习

1. 证明: 当 n 是 $1 \leq n \leq 4$ 的正整数时, 有 $n^2 + 1 \geq 2^n$ 。
2. 证明: 若 x 和 y 都是实数, 则 $\max(x, y) + \min(x, y) = x + y$ 。[提示: 使用分情形证明, 两种情形分别对应于 $x \geq y$ 和 $x < y$ 。]
3. 证明三角不等式: 若 x 和 y 都是实数, 则 $|x| + |y| \geq |x + y|$ (其中 $|x|$ 表示 x 的绝对值, 若 $x \geq 0$ 则它等于 x , 若 $x < 0$ 则它等于 $-x$)。
4. 证明: 存在 100 个连续的不是完全平方的正整数。你的证明是构造性的还是非构造性的?
5. 证明: 存在一对连续的整数, 其中一个整数是完全平方的, 另一个是完全立方的。
6. 证明或驳斥: 存在有理数 x 和无理数 y , 使得 x^y 是无理数。
7. 证明下列每一个命题均可用于表达这样的事实: 存在一个唯一的元素 x 使得 $P(x)$ 为真。(注意, 这等同于命题 $\exists! x P(x)$ 。)
 - a) $\exists x \forall y (P(y) \leftrightarrow x = y)$
 - b) $\exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \rightarrow x = y)$
 - c) $\exists x (P(x) \wedge \forall y (P(y) \rightarrow x = y))$
8. 假定 a 和 b 是奇数且 $a \neq b$ 。证明: 存在唯一的整数 c 满足 $|a - c| = |b - c|$ 。
9. 证明: 若 n 是奇数, 则存在唯一的整数 k , 使得 n 是 $k - 2$ 和 $k + 3$ 之和。
10. 证明: 给定实数 x , 存在唯一的数 n 和 ϵ 使得 $x = n - \epsilon$, n 是整数且 $0 \leq \epsilon < 1$ 。
11. 两个整数 x 和 y 的调和均值是 $\frac{2xy}{x+y}$ 。通过计算不同成对正整数的调和均值和几何均值, 形成关于这两种均值相对大小的猜想并证明之。
- *12. 在黑板上写数字 $1, 2, \dots, 2n$, 其中 n 是奇数, 从中挑出两个数 j 和 k , 在黑板上写 $|j - k|$ 并擦掉 j 和 k 。继续这个过程, 直到黑板上只写一个整数为止。证明: 这个整数必为奇数。
13. 已知如下有关十进制数问题的一个猜想: 一个整数 4 次方的十进制最后位数字。用分情形方法证明你的猜想。
14. 证明: 不存在正整数 n 使得 $n^2 + n^3 = 100$ 。
15. 证明: 方程 $x^4 + y^4 = 625$ 没有整数解 x 和 y 。
16. 改造 1.6 节例 4 的证明, 证明: 如果 a, b, c 是正整数, $n = abc$, 则 $a \leq \sqrt[3]{n}$, $b \leq \sqrt[3]{n}$ 或者 $c \leq \sqrt[3]{n}$ 。
17. 证明: 任两个有理数之间都有一个无理数。
- *18. 设 $S = x_1 y_1 + x_2 y_2 + \dots + x_n y_n$, 其中 $x_1, x_2, \dots, x_n$ 和 $y_1, y_2, \dots, y_n$ 是排序的两个不同正实数序列, 各自有 n 个元素。
 - a) 证明: 在这两个序列的所有排序当中, 当两个序列都排序 (使得每个序列中的元素都以非降序排列) 时, S 取最大值。
 - b) 证明: 在这两个序列的所有排序当中, 当一个序列排成非降顺序, 另一个序列排成非升顺序时, S 取最小值。
19. 对下列这些整数验证 $3x + 1$ 猜想:
 - a) 6
 - b) 7
 - c) 17
 - d) 21
20. 证明或驳斥: 你能用骨牌填充去掉两个相邻角 (也就是说不是对角) 的标准棋盘。
21. 证明: 你能用骨牌填充带有偶数个方格的长方形棋盘。
22. 通过穷举法证明: 用骨牌填充去掉两个对角的 4×4 棋盘是不存在的。[提示: 首先证明假设去掉左上角和右下角的方格。棋盘原始方格是 1 到 16, 从第一行开始, 在这一行向右移动, 然后在第 2 行最左边的方格开始向右移动, 等等。去掉方块 1 和 16。开始证明时, 注意到 2 或者被一个水平放置的骨牌覆盖, 这个骨牌覆盖 2 和 3, 或者这个骨牌垂直地覆盖 2 和 6。分别考虑每一种情况, 证明产生的所有子情况。]
23. 证明: 从一个 8×8 的棋盘去掉两块白的和两块黑的方格 (颜色如教材所示), 用骨牌填充棋盘留下的方格是不可能的。
- *24. a) 画五种不同的格板, 每一个格板都是由四个角的四个格组成的。
b) 证明或驳斥, 用五种格板中的每种不同格板, 可填充标准棋盘。

关键术语和结果

术语

命题：一个或为真或为假的语句。

命题变元：代表一个命题的变量。

真值：真或假。

$\neg p$ (p 的否定)：与 p 的真值相反的命题。

逻辑运算符：用于组合命题的运算符。

复合命题：用逻辑运算符组合命题构造出的命题。

真值表：显示命题真值的表。

$p \vee q$ (p 和 q 的析取)：除非 p 和 q 均为假，否则为真的命题。

$p \wedge q$ (p 和 q 的合取)：只有在 p 和 q 均为真时才为真的命题。

$p \oplus q$ (p 和 q 的异或)：当 p 和 q 中恰有一个为真时才为真的命题。

$p \rightarrow q$ (p 蕴含 q)：只有在 p 为真而 q 为假时才为假的命题。

$p \rightarrow q$ 的逆蕴含：条件语句 $q \rightarrow p$ 。

$p \rightarrow q$ 的倒置蕴含：条件语句 $\neg q \rightarrow \neg p$ 。

$p \rightarrow q$ 的反蕴含：条件语句 $\neg p \rightarrow \neg q$ 。

$p \leftrightarrow q$ (双蕴含)：只有 p 和 q 真值相同时才为真的命题。

字位：0 或 1。

布尔变量：以 0 或 1 为值的变量。

字位运算：一个或多个字位上的运算。

位串：一串字位。

按位运算：位串上的运算，对一个位串的字位和另一位串的对应该字位作运算。

永真式：永远为真的复合命题。

矛盾：永远为假的复合命题。

可能式：有时成真有时为假的复合命题。

一致复合命题：为变量赋真值，使所有这些命题为真的复合命题。

复合命题的逻辑等价：总有同样真值的复合命题。

谓词：句子中代表主语属性的那部分。

命题函数：包含一个或多个变量的语句，当每一个变量被指派或被量词约束时，就变成命题。

论域：命题函数中变量可能取到的所有值。

$\exists x P(x)$ ($P(x)$ 的存在量化)：当且仅当在论域中存在一个 x 使 $P(x)$ 为真时才为真的命题。

$\forall x P(x)$ ($P(x)$ 的全称量化)：当且仅当论域中的所有 x 值 $P(x)$ 均为真时才为真的命题。

逻辑等价表达式：无论应用什么样的命题函数和论域，真值都相同的表达式。

自由变量：命题函数中未绑定的变量。

绑定变量：被量化的变量。

量词的作用域：语句中量词所绑定变量的那部分。

论证：一连串的命题。

论证形式：包含命题变量的一连串的复合命题。

前提：论证或论证形式中最后命题以外的命题。

结论：论证或论证形式中最后的命题。

有效论证形式：包含命题变量的一连串的复合命题，它的所有前提为真蕴含结论为真。

有效论证：具有有效论证形式的论证。

推理规则：用于示范论证有效的有效论证形式。

谬误：作为一个推理规则，常常(有时或者更一般)用于不正确推理的无效论证形式。

循环论证或偷用论题：一步或多个步骤基于待证明命题的正确性的推理。

定理：可以证明为真的数学断言。

猜想：真值未知的数学断言。

证明：对定理为真的说明。

公理：常作为证明定理的基础，并假设为真的命题。

引理：用来证明其他定理的简单定理。

推论：可以作为刚刚证明的定理的后果而被证明的命题。

空证明：基于 p 为假的事实而对蕴含式 $p \rightarrow q$ 的证明。

平凡证明：基于 q 为真的事实而对蕴含式 $p \rightarrow q$ 的证明。

直接证明：通过证明当 p 为真时 q 必然为真而进行的对蕴含式 $p \rightarrow q$ 的证明。

反证法：通过证明当 q 是假时 p 一定是假，来证明 $p \rightarrow q$ 为真。

归谬证明：基于蕴含式 $\neg p \rightarrow q$ 的真值(其中 q 是矛盾式)而对命题 p 为真的证明。

穷举证明：通过检查一系列的所有情况所建立的结果得到的证明。

分情形证明：把情况分解为覆盖所有可能的单独情况的证明。

不失一般性：假定一个证明可以通过减少需要证明的情况来证明的一个法则。

反例：使得 $P(x)$ 为假的元素 x 。

构造性的存在性证明：证明具有特定性质的元素存在，通过显式的方式来寻找这样的元素。

非构造性的存在性证明：证明具有特定性质的元素存在，但不是显式地寻找这样的元素。

有理数：可以表示为两个整数 p 和 q (其中 $q \neq 0$) 之比的数。

唯一性证明：证明具有特定性质的元素唯一地存在。

结果

1.2 节表 1-15、表 1-16、表 1-17 给出的逻辑等价关系。

德摩根定律

命题演算的推理规则

带量词命题的推理规则

复习题

1. a) 定义命题的否定。
b) “这是一门烦人的课程”的否定是什么？
2. a) (用真值表) 定义命题 p 和 q 的析取、合取、异或、蕴含和双蕴含。
b) “今晚我去看电影”和“我将完成离散数学作业”的析取、合取、异或、蕴含和双蕴含是什么？
3. a) 给出至少五种用汉语表达条件语句 $p \rightarrow q$ 的方式。
b) 定义条件语句的逆蕴含和倒置。
c) “如果明天阳光明媚，我将到林中散步。”给出此条件语句的逆蕴含和倒置。
4. a) 两个命题逻辑等价的含义是什么？
b) 描述证明两个复合命题逻辑等价的不同方法。
c) 至少用两种方法证明 $\neg p \vee (r \rightarrow \neg q)$ 和 $\neg p \vee \neg q \vee \neg r$ 等价。
5. (依赖于 1.2 节的练习)
a) 给出一个真值表，解释怎样用析取范式构造一个复合命题使其真值表就是你给出的真值表。
b) 解释为什么 a) 说明运算符 $\wedge$, $\vee$ 和 $\neg$ 是功能完全的。
c) 是否有一个运算符使得只含这个运算符的集合是功能完全的？
6. 什么是谓词 $P(x)$ 的全称和存在量化？什么是它们的否定？
7. a) 量化语句 $\exists x \forall y P(x, y)$ 和 $\forall y \exists x P(x, y)$ 的区别何在？其中 $P(x, y)$ 为谓词。
b) 给出谓词 $P(x, y)$ 的一个实例，使 $\exists x \forall y P(x, y)$ 和 $\forall y \exists x P(x, y)$ 有不同的真值。
8. 描述在命题逻辑中，有效论证是什么意思，并且证明：“如果地球是平的，那你就能行驶到地球边缘”；“你不能行驶到地球边缘”，因此，“地球不是平的”是有效论证。
9. 用推理规则证明：如果前提“所有的斑马都有条纹”，“马克是一匹斑马”是真的，那么结论“马克有条纹”是真的。
10. a) 描述一下条件语句 $p \rightarrow q$ 的直接证明、反证法和归谬证明分别是什么意思。
b) 分别给出语句：“若 n 是偶数，则 $n+4$ 是偶数”的直接证明、反证法和归谬证明。

11. a) 描述证明双条件语句 $p \leftrightarrow q$ 的一种方式。
b) 证明命题：“整数 $3n+2$ 是奇数当且仅当整数 $9n+5$ 是偶数，其中 n 是整数”。
12. 为了证明语句 p_1 , p_2 , p_3 和 p_4 都是等价的，证明条件语句 $p_4 \rightarrow p_2$, $p_3 \rightarrow p_1$ 和 $p_1 \rightarrow p_2$ 都是有效的，这样做是否充分？若不充分，则给出可用来证明这四个语句都是等价的另外一组条件语句。
13. a) 假定形如 $\forall x P(x)$ 的条件语句为假。如何可以证明它为假？
b) 证明语句“对每个正整数 n 来说， $n^2 \geq 2n$ ”为假。
14. 构造性与非构造性存在性证明之间的差异是什么？对这两种证明分别举一个例子。
15. 证明：存在唯一的元素 x 使得 $P(x)$ 为真（其中 $P(x)$ 是命题函数）。这种证明的基础是什么？
16. 阐释如何用分情形证明以下绝对值的结果：对所有实数 x 和 y ，有 $|xy| = |x||y|$ 。

补充练习

1. 令 p 为命题“我将完成本书中的每一道练习”， q 为命题“这门课程我将得‘A’”。将下列各项表示为 p 和 q 的组合。
 - a) 只有完成本书中的每道练习，这门课程我才能得‘A’。
 - b) 这门课程我将得‘A’，而且我将完成本书中每一道练习。
 - c) 或者这门课程我将得‘A’，或者我不完成本书中的每一道练习。
 - d) 这门课程我得‘A’的充分必要条件是完成本书中每一道练习。
2. 求复合命题 $(p \vee q) \rightarrow (p \wedge \neg r)$ 的真值表。
3. 证明下列命题为永真式。
 - a) $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$
 - b) $((p \vee q) \wedge \neg p) \rightarrow q$
4. 给出下列条件语句的逆蕴含、反蕴含和倒置。
 - a) 如果今天下雨，我就开车上班。
 - b) 如果 $|x| = x$ ，那么 $x \geq 0$ 。
 - c) 若 n 大于 3，那么 n^2 大于 9。
5. 给出条件语句 $p \rightarrow q$ 的逆的反蕴含、反的反蕴含、倒置。
6. 给出条件语句 $p \rightarrow q$ 的逆的逆蕴含、反的逆蕴含、倒置的逆蕴含。
7. 用命题变量 p , q , r 和 s 构造一个复合命题，使它在这些命题变量中恰有三个为真时取真值，其他情况下为假。
8. 证明下列语句是不一致的：“如果 Sergei 拿到工作签证，那他将获得签约奖金。”“如果 Sergei 拿到工作签证，那他将拿到高薪。”“如果 Sergei 拿到签约奖金，那他将拿不到高薪。”“Sergei 拿到了工作签证。”
9. 证明下列语句是不一致的：“如果 Miranda 没有修过离散数学课程，那她将不能毕业。”“如果 Miranda 不能毕业，那她将不能胜任那份工作。”“如果 Miranda 读了这本书，那她将能胜任那份工作。”“Miranda 没有修过离散数学课程，但她读过这本书。”
10. 假定你在 1.1 节例 18 中描述的“武士和流氓”共居的岛上遇见三个人 A、B 和 C。如果 A 说“我是流氓，B 是骑士”且 B 说“我们三人中恰有一人是骑士”，那么 A、B 和 C 分别是哪种人？
11. (摘自 [Sm78]) 假定在一个岛上住着三类人：武士、流氓和普通人。武士总是说真话，流氓总是说谎话，普通人有时说谎话有时说真话。侦探为了找出罪犯，询问了岛上的三个人 Amy、Brenda 和 Claire。侦探知道三人中有一人犯罪了，但不知是哪个人。他们也知道罪犯是一个骑士，另两个人不是骑士。此外，侦探还记录了如下供述。Amy 说：“我是清白的。”Brenda 说：“Amy 说的是真的。”Claire 说：“Brenda 不是普通人。”经过分析，侦探找到了罪犯。他是谁？
12. 证明：如果 S 是一个命题，这里 S 是条件命题：“若 S 是真的，则独角兽是存在的”，那么“独角兽是存在的”是真的。证明 S 不能是一个命题（这是著名的罗素悖论）。
13. 证明：假设“牙齿仙女是真人”，“真理仙女就不是真人”，结论“你能在彩虹尽头找到金子”是有效论证。能证明结论是真的吗？
14. 令 $P(x)$ 为语句“学生 x 会微积分”， $Q(y)$ 为“ y 班上有个学生会微积分”。用 $P(x)$ 和 $Q(y)$ 的量化表示下列各项。
 - a) 某个学生会微积分。
 - b) 不是每个学生都会微积分。
 - c) 每个班上都有一个学生会微积分。
 - d) 每个班上的每个学生都会微积分。

e) 至少有一个班没有学生会微积分。

15. 令 $P(m, n)$ 为语句“ m 除尽 n ”, 其中变量 m 和 n 的论域均为正整数集合。给出下列命题的真值。

- a) $P(4, 5)$ b) $P(2, 4)$ c) $\forall m \forall n P(m, n)$
 d) $\exists m \forall n P(m, n)$ e) $\exists n \forall m P(m, n)$ f) $\forall n P(1, n)$

16. 确定量词的论域, 使语句 $\exists x \exists y (x \neq y) \wedge \forall z ((z \neq x) \wedge (z \neq y))$ 为真。

17. 确定量词的论域, 使语句 $\exists x \exists y (x \neq y) \wedge \forall z ((z = x) \wedge (z = y))$ 为假。

18. 用量词表达语句“没有人有多于三个的祖母”, 使用命题函数 $G(x, y)$, 它表示“ x 是 y 的祖母。”

19. 用量词表达语句“每个人恰有两个亲生父母”, 使用命题函数 $P(x, y)$, 它表示“ x 是 y 的亲生父母。”

20. 量词 $\exists_n$ 表示“有准确的 n ”, 因此 $\exists_n x P(x)$ 意思是在论域中存在准确的 n 值, 使 $P(x)$ 为真。由所有实数组成论域, 决定以下命题的真值:

- a) $\exists_0 x (x^2 = -1)$ b) $\exists_1 x (|x| = 0)$ c) $\exists_2 x (x^2 = 2)$ d) $\exists_3 x (x = |x|)$

21. 用存在量词、全称量词和命题逻辑表示以下每一个命题, 这里 $\exists_n$ 如练习 20 定义:

- a) $\exists_0 x P(x)$ b) $\exists_1 x P(x)$ c) $\exists_2 x P(x)$ d) $\exists_3 x P(x)$

22. 令 $P(x, y)$ 为命题函数。证明蕴含关系 $\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y)$ 为永真式。

23. 令 $P(x)$ 和 $Q(x)$ 为命题函数。求证 $\exists x (P(x) \rightarrow Q(x))$ 和 $\forall x P(x) \rightarrow \exists x Q(x)$ 有同样的真值。

24. 如果 $\forall y \exists x P(x, y)$ 为真, $\exists x \forall y P(x, y)$ 是否必定也为真?

25. 如果 $\forall x \exists y P(x, y)$ 为真, $\exists x \forall y P(x, y)$ 是否必定也为真?

26. 找出下列语句的否定。

- a) 如果今天下雪, 那么我明天去滑雪。 b) 班上每个人都懂数学归纳法。
 c) 班上有些学生不喜欢离散数学。 d) 每堂数学课都有某个学生上着课就睡着了。

27. 用量词表示“班上每个学生都选修过数学学院每个系的某门课”。

28. 用量词表示“在美国某学院的校园里有座楼的每间屋子都漆成了白色”。

29. 用唯一量词表示以下每一个语句: “在这个学校里, 恰有一个数学班, 而本班只有一个学生”, 然后再用量词而不用唯一量词表示这个语句。

30. 描述一个推理规则, 可用它来证明论域中恰有两个元素 x 和 y 使得 $P(x)$ 和 $P(y)$ 为真。用汉语句子表达这个推理规则。

31. 使用推理规则证明如果前提 $\forall x (P(x) \rightarrow Q(x))$ 、 $\forall x (Q(x) \rightarrow R(x))$ 和 $\neg R(a)$ 为真, 那么结论 $\neg P(a)$ 为真, 其中 a 在论域中。

32. 证明: 若 x^3 是无理数, 则 x 是无理数。

33. 证明: 若 x 是无理数且 $x \geq 0$, 则 $\sqrt{x}$ 是无理数。

34. 证明: 给定一个非负整数 n , 存在一个唯一的非负整数 m 使得 $m^2 \leq n < (m+1)^2$ 。

35. 证明: 存在一个整数 m 使得 $m^2 > 10^{1000}$ 。你的证明是构造性的还是非构造性的?

36. 证明存在这样的正整数: 它可以被以两种不同的方式写成正整数的立方和。使用计算机或计算器来完成。

37. 反驳如下命题: 每个正整数均可表示为 8 个非负整数的立方和。

38. 反驳如下命题: 每个正整数均可表示为至多两个非负整数的平方与一个非负整数的立方的和。

39. 反驳如下命题: 每个正整数均可表示为 36 个非负整数的 5 次方的和。

40. 假设以下定理的真实性: 对任何不是完全平方数的正整数 n , $\sqrt{n}$ 是个无理数。由此证明: $\sqrt{2} + \sqrt{3}$ 是无理数。

计算机题目

按给定的输入和输出写程序。

- 已知命题 p 和 q 的真值, 求它们的合取、析取、异或、条件语句和双条件语句的真值。
- 已知长度为 n 的两个位串, 求它们的按位 AND、按位 OR 及按位 XOR。
- 已知模糊逻辑中命题 p 和 q 的真值, 求 p 和 q 的析取和合取的真值(参看 1.1 节练习 21)。

*4. 给定正整数 m, n , 交互玩 Chomp 游戏。

*5. 给出部分棋盘, 用各种不同类型的四角填充, 包括骨牌、两种类型的三格板和大四角。

计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 找出这样的正整数：它不是 9 个不同的正整数的立方和。
2. 找出大于 79 的这样的正整数：它不是 18 个正整数的四次方的和。
3. 找出尽可能多这样的正整数：它可以用正整数的立方和来表示。用两种不同的方式，对 1729 使用这个性质。
- *4. 试图找出不同初始设置曲奇饼的 Chomp 游戏的获胜策略。
- *5. 寻找棋盘和带有四角的部分棋盘的填充。

写作题目

用课本以外的资料，按下列要求写成短文。

1. 讨论如下逻辑悖论：克里特人 Epimenides 的悖论(又叫说谎者悖论)、Jourdain 的纸牌悖论以及理发师悖论。说明如何解决它们。
2. 描述模糊逻辑怎样用于实际应用。参考为一般读者写的一本或几本最近出版的模糊逻辑书。
3. 描述由 Layman Allen 提出的“WFF'N PROOF, The Game of Modern Logic”的基本规则。给出 WFF'N PROOF 中包含的一些博弈示例。
4. 读一些卡罗尔(Lewis Carroll)关于符号逻辑的作品。详细描述他用于表示逻辑论证和推理规则的一些模型。
5. 扩展 1.3 节对 Prolog 的讨论，进一步解释 Prolog 如何使用消解。
6. 讨论在计算逻辑中使用的一些技术，包括 Skolem 的规则。
7. “自动定理证明”是使用计算机来机械地证明定理的任务。讨论自动定理证明的目标和应用，以及在开发自动定理证明器上取得的进步。
8. 讨论如何使用 DNA 计算来求解可满足性问题的示例。
9. 讨论有关 Chomp 游戏已知的获胜策略。
10. 描述在乔治波利亚有关推理的著作中讨论的各种证明策略，包括[Po62]、[Po71]和[Po90]。
11. 举例说明在[Go94]和[Ma91]中描述的几个有关四角棋盘的填充问题和结果。

基本结构：集合、函数、数列与求和

离散数学的许多内容用以表示离散对象，研究离散结构。许多重要的离散结构是由所谓的对象集构建的集合。用联接词构建集合之间的离散结构，这些无序的对象集被广泛地用于计数；关系，代表着对象之间联系的有序对集合。图，是节点集和连接节点的边的集合；有限状态机用于模型计算机。这些是后续章节研究的一些主题。

函数的概念在离散数学中是非常重要的。函数把集合中的每一个元素准确地指派给集合的一个元素。函数对整个离散数学起着重要的作用，用以表示算法的计算复杂性，研究的集合的大小，计算对象的数量，以及无数其他的应用方式。排序和字符串是特殊类型的函数。

2.1 集合

2.1.1 引言

我们将在这一部分学习基本的离散结构，以及由此而得到的各种各样的离散结构，换言之，集合。集合用于把对象组合在一起。通常在一个集合中的对象都有相似的性质。例如，你们学校中目前在册的所有学生可以组成一个集合。同样，任何学校正选修一门离散数学课的学生可以组成一个集合。此外，你们学校在册学生中正选修一门离散数学课的所有学生组成一个集合，这个集合可以从上述两个集合中取共同的元素得到。像集合那样，集合语言是以构造的方式来研究的工具。现在提供集合的一种定义。这是一种直观的定义，不属于正规的集合定义。

定义 1 集合是一组无序的对象。

注意，我们并没有说明对象是什么就已经使用这个术语了。以对象的直观概念为基础把集合描述为一组对象，这是德国数学家康托尔 1895 年首先给出的。从集合的这一直观定义得到的理论导出悖论，即逻辑上的不一致性。这是英国哲学家罗素 1902 年提出的。以称为公理的基本假设起点建立集合理论可以避免逻辑上的不一致。我们将不展开集合理论的公理描述，而是使用康托尔的称为朴素集合理论的原始描述，这是因为本书中考虑的所有集合均可用康托尔的原始理论处理而无不一致性。

定义 2 集合中的对象也称为该集合的元素，或成员。也说集合包含它的元素。

现在介绍用于描述集合成员的方法。我们记 $a \in A$ ，也就是表示 a 是集合 A 中一个元素。而 $a \notin A$ 就表示 a 不是集合 A 中的一个元素。注意，小写字母通常用于表示集合中的元素。

有几种方式描述集合，一种方式是在可能的情况下一一列出集合中的元素。我们采用在花括号之间列出所有元素的方法。例如， $\{a, b, c, d\}$ 表示含 4 个元素 a, b, c 和 d 的集合。

例 1 英语字母中所有元音字母的集合 V 可以表示为 $V = \{a, e, i, o, u\}$ 。 ■

例 2 小于 10 的正奇数集合 O 可以表示为 $O = \{1, 3, 5, 7, 9\}$ 。 ■

例 3 尽管集合常用来表示一组具有共同性质的元素，但有的集合中的元素表面上看起来毫不相干。例如，集合 $\{a, 2, \text{Fred}, \text{New Jerseg}\}$ 包含了 4 个元素： $a, 2, \text{Fred}$ 和 New Jerseg 。 ■

有时用花括号表示集合但并不列出它的所有元素。先列出集合中的某些元素，然后当元素的一般形式很明显时就用省略号($\dots$)表示。

例 4 小于 100 的正整数集合可以表示为 $\{1, 2, 3, \dots, 99\}$ 。 ■

描述集合的另一种方式是使用集合的构造符号。我们给出作为集合的成员必须具有的性质，以此来刻画集合的所有元素。例如，小于 10 的所有奇数的集合 O 可以写成

$$O = \{x \mid x \text{ 是小于 } 10 \text{ 的正奇数}\}$$

或者，指定正整数集合的一个范围，如

$$O = \{x \in \mathbf{Z}^+ \mid x \text{ 为奇数}, x < 10\}$$

当无法给出集合所有元素时常用这一类方法来描述集合。例如，所有正有理数集合，可以被写成

$$\mathbf{Q}^+ = \{x \in \mathbf{R} \mid x = p/q, p, q \text{ 为正整数}\}$$

这些集合用黑体表示，它们在离散数学中发挥着重要的作用：

$$\mathbf{N} = \{0, 1, 2, 3, \dots\}, \text{ 自然数集}$$

$$\mathbf{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}, \text{ 整数集}$$

$$\mathbf{Z}^+ = \{1, 2, 3, \dots\}, \text{ 正整数集}$$

$$\mathbf{Q} = \{p/q \mid p \in \mathbf{Z}, q \in \mathbf{Z}, \text{ 和 } q \neq 0\}, \text{ 有理数集}$$

$$\mathbf{R}, \text{ 实数集}$$

(注意，一些人认为 0 不是自然数成员，所以当你阅读其他书籍的时候要检查自然数成员的术语是怎样用的。)

集合可以包含其他的集合，以例 5 为例说明。

例 5 集合 $\{\mathbf{N}, \mathbf{Z}, \mathbf{Q}, \mathbf{R}\}$ 包含了 4 个元素，其中每一个都是集合。这 4 个元素是： $\mathbf{N}$ ，自然数集； $\mathbf{Z}$ ，整数集； $\mathbf{Q}$ ，有理数集，以及 $\mathbf{R}$ ，实数集。 ■

注意 计算机科学中的数据类型或类型的概念建立在集合这一概念之上。数据类型或类型是集合的一个称号，它是连同对集合中的对象实施操作的集合。例如，布尔是集合 $\{0, 1\}$ 的一个称号，连同其上一个或多个元素实施运算，像并且、或者和非。

由于许多数学语句需要断定两组以不同方式描述的对象实际上是同一个集合，我们需要理解两个集合相等的含义。

定义 3 两个集合相等当且仅当它们有同样的元素。也就是说，若集合 A 和集合 B 是相等的，当且仅当 $\forall x(x \in A \leftrightarrow x \in B)$ 。如果 A 和 B 是相等的集合，记为 $A=B$ 。

例 6 集合 $\{1, 3, 5\}$ 和 $\{3, 5, 1\}$ 相等，因为它们有同样的元素。注意集合中元素的顺序不起作用，还要注意同一个元素被列出来不止一次也没关系。所以 $\{1, 3, 3, 3, 5, 5, 5, 5\}$ 和 $\{1, 3, 5\}$ 是同一个集合，因为它们有同样的元素。 ■

还可以用文氏图形象地表示集合。文氏图(或称维恩图)是以英国数学家 John Venn 的名字命名的，他在 1881 年介绍了这种图的使用。我们所考虑的所有对象的集合 U ，称为全集。在文氏图中全集用长方形表示。在长方形内部，圆或其他几何图形用于表示集合，有时用点来表示集合中特定的元素。文氏图常用于表示集合之间的关系。我们用下面的例子解释怎样使用文氏图。

例 7 画一个表示英语字母中元音字母集合 V 的文氏图。

解 画一个长方形表示全集 U ，也就是 26 个英文字母的集合。在长方形中画一个圆表示集合 V ，在圆中用点表示集合 V 的元素(见图 2-1)。 ■

有一个不含任何元素的特殊集合，称为空集，用 $\emptyset$ 表示。空集也可以用 $\{\}$ 表示(这是用花括号内列出所有元素的方式表示集合)。经常具有一定性质的元素组成的集合其实就是空集。例如，大于自身的平方的所有正整数的集合是空集。只有一个元素的集合叫做单元元素集合。

通常的错误是把空集 $\emptyset$ 与单元元素集合 $\{\emptyset\}$ 搞混。集合 $\{\emptyset\}$ 的唯一元素是空集本身！记住这个区别的一个形象比喻是考虑计算机文件系统中的文件夹。空集可以比做一个空的文件夹，包含一个空集合的集合可以比作一个文件夹里只有一个空的文件夹文件。

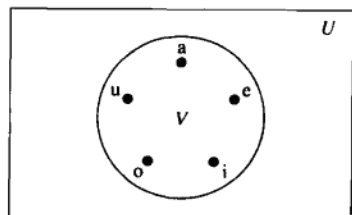


图 2-1 元音字母集合的文氏图

定义 4 集合 A 是集合 B 的子集当且仅当 A 的每个元素也是 B 的元素。我们用记号 $A \subseteq B$ 表

示 A 是 B 的子集。

我们看到, $A \subseteq B$ 当且仅当量化语句

$$\forall x(x \in A \rightarrow x \in B)$$

为真。

例 8 小于 10 的所有正奇数的集合, 是小于 10 的所有正整数的集合的子集。有理数集是实数集的一个子集。你们学校主修计算机科学的学生的集合是你们学校全体学生集合的子集。集合{全体中国人民}是{全体中国人民}的子集(也就是说, 它本身)。

定理 1 表明, 每个非空集合 S 都至少有两个子集, 一个是空集, 一个是集合 S 本身, 即 $\emptyset \subseteq S$ 和 $S \subseteq S$ 。

定理 1 对于任意集合 S , (i) $\emptyset \subseteq S$ 和 (ii) $S \subseteq S$ 。

证 这里证明 (i), (ii) 留作练习。

令 S 为一个集合。为了证明 $\emptyset \subseteq S$, 必须证明 $\forall x(x \in \emptyset \rightarrow x \in S)$ 为真。因为空集没有元素, 所以 $x \in \emptyset$ 总是假。因此 $x \in \emptyset \rightarrow x \in S$ 总是真, 因为其假设为假, 并且具有假的假设的条件语句为真。即 $\forall x(x \in \emptyset \rightarrow x \in S)$ 为真。这完成了 (i) 的证明。这是空证明的示例。

如果我们要强调集合 A 是集合 B 的子集, 但 $A \neq B$, 就写成 $A \subset B$, 并说 A 是 B 的真子集。若 $A \subset B$ 是真的, 则必有 $A \subseteq B$, 且必有 B 的某个元素 x 不属于 A , 即如果

$$\forall x(x \in A \rightarrow x \in B) \wedge \exists x(x \in B \wedge x \notin A)$$

为真, 那么 A 是 B 的真子集。文氏图可以用来表示集合 A 是集合 B 的子集。我们把全集 U 画成长方形。在这长方形中画一表示 B 的圆。由于 A 是 B 的子集, 我们在代表 B 的圆内画表示 A 的圆。图 2-2 所示就是这一关系。

证明两个集合具有相同元素的一个方法是证明它们互为另一个的子集。换言之, 可以证明, 如果 A 和 B 均为集合, 且 $A \subseteq B$ 和 $B \subseteq A$, 则 $A = B$ 。实践表明, 这是证明集合相等的一种有用的方式。也就是说, $A = B$, 其中 A 和 B 是集合, 当且仅当 $\forall x(x \in A \rightarrow x \in B)$ 和 $\forall x(x \in B \rightarrow x \in A)$; 或者 A 和 B 等价, 当且仅当 $\forall x(x \in A \leftrightarrow x \in B)$ 。

集合可以以其他集合作为它的成员。例如, 下面列出的就是这样的集合:

$$A = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}, B = \{x \mid x \text{ 是集合 } \{a, b\} \text{ 的子集}\}$$

注意, 这两个集合相等, 即 $A = B$ 。同时注意 $\{a\} \in A$, 但是 $a \notin A$ 。

集合广泛用于计数问题。这类问题需要讨论集合的大小。

定义 5 令 S 为集合。若 S 中恰有 n 个不同的元素, n 是非负整数, 就说 S 是有限集合, 而 n 是 S 的基数。 S 的基数用 $|S|$ 表示。

例 9 令 A 为小于 10 的正奇数集合, 则 $|A| = 5$ 。

例 10 令 S 为英语字母集, 那么 $|S| = 26$ 。

例 11 由于空集没有元素, 所以 $|\emptyset| = 0$ 。

我们对不是有限的集合也有兴趣。

定义 6 如果一个集合不是有限的, 就说它是无限的。

例 12 正整数集合是无限的。

无限集合的基数将在 2.4 节讨论。

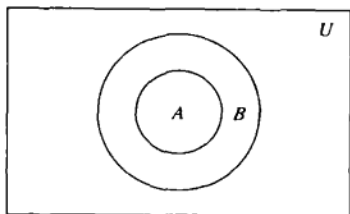


图 2-2 表示 A 是 B 的子集的文氏图

2.1.2 幂集合

许多问题都需要检查一个集合的元素所有可能的组合, 看它们是否具有某种性质。为了考虑集合 S 中元素所有可能的组合, 我们构造一个新集合, 它以 S 的所有子集作为它的元素。

定义 7 已知集合 S , S 的幂集合是集合 S 所有子集的集合。 S 的幂集合用 $P(S)$ 表示。

例 13 集合 $\{0, 1, 2\}$ 的幂集合是什么?

解 幂集合 $P(\{0, 1, 2\})$ 是 $\{0, 1, 2\}$ 所有子集的集合。因此

$$P(\{0, 1, 2\}) = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}$$

注意, 空集和 $\{0, 1, 2\}$ 自身都是这个子集集合的成员。

例 14 空集的幂集合是什么? 集合 $\{\emptyset\}$ 的幂集合是什么?

解 空集只有一个子集, 这就是它自己。因此

$$P(\emptyset) = \{\emptyset\}$$

集合 $\{\emptyset\}$ 有两个子集, 即 $\emptyset$ 和集合 $\{\emptyset\}$ 自身。于是

$$P(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$$

如果一个集合有 n 个元素, 那么它的幂集合有 2^n 个元素。我们将在本书以后的几节中以不同的方式证明这一事实。

2.1.3 笛卡儿积

一组元素中元素的次序往往是重要的。由于集合是无序的, 必须用不同的结构来表示有序的一组元素, 这个结构由有序 n 元组来提供。

定义 8 有序 n 元组 $(a_1, a_2, \dots, a_n)$ 是以 a_1 为第 1 个元素, a_2 为第 2 个元素, $\dots$, a_n 为第 n 个元素的有序组。

只有在两个有序 n 元组每一对对应的元素都相等时, 它们才相等。换言之, $(a_1, a_2, \dots, a_n) = (b_1, b_2, \dots, b_n)$ 当且仅当对于 $i=1, 2, \dots, n$, $a_i = b_i$ 。二元组特称为有序偶。有序偶 (a, b) 和 (c, d) 相等当且仅当 $a=c$ 和 $b=d$ 。注意, 除非 $a=b$, 否则 (a, b) 和 (b, a) 不相等。

在随后几章中将要学习的许多离散结构均以 René Descartes 命名的集合的笛卡儿积的概念为基础。我们先定义两个集合的笛卡儿积。

定义 9 令 A 和 B 为集合。 A 和 B 的笛卡儿积用 $A \times B$ 表示, 是所有有序偶 (a, b) 的集合, 其中 $a \in A$ 而 $b \in B$ 。于是

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$$

例 15 令 A 为某大学所有学生的集合, B 表示该大学开设的所有课程的集合。 A 和 B 的笛卡儿积 $A \times B$ 是什么?

解 笛卡儿积 $A \times B$ 由形为 (a, b) 的所有有序偶组成, 其中 a 是学生, 而 b 是该校开的一门课。集合 $A \times B$ 可以用来表示该校学生选课的所有可能情况。

例 16 $A = \{1, 2\}$ 和 $B = \{a, b, c\}$ 的笛卡儿积是什么?

解 笛卡儿积 $A \times B$ 是

$$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}$$

笛卡儿积 $A \times B$ 的子集 R 称为从集合 A 到集合 B 的关系。 R 的元素是有序偶, 其中第一个元素属于 A , 第二个元素属于 B 。例如, $R = \{(a, 0), (a, 1), (a, 3), (b, 1), (b, 2), (c, 0), (c, 3)\}$ 是从集合 $\{a, b, c\}$ 到集合 $\{0, 1, 2, 3\}$ 的关系。我们将在第 5 章详细研究关系。

除非 $A = \emptyset$ 或 $B = \emptyset$ (这样 $A \times B = \emptyset$) 或除非 $A = B$, 否则 $A \times B$ 和 $B \times A$ 不相等。这在下例中说明。

例 17 证明笛卡儿积 $B \times A$ 和 $A \times B$ 不相等, 其中 A 和 B 为例 16 中的集合。

解 笛卡儿积 $B \times A$ 是

$$B \times A = \{(a, 1), (a, 2), (b, 1), (b, 2), (c, 1), (c, 2)\}$$

这不等于例 16 中得到的 $A \times B$ 。

对于两个以上的集合也可以定义笛卡儿积。

定义 10 集合 $A_1, A_2, \dots, A_n$ 的笛卡儿积用 $A_1 \times A_2 \times \dots \times A_n$ 表示, 这是有序 n 元组

$(a_1, a_2, \dots, a_n)$ 的集合, 其中对于 $i=1, 2, \dots, n, a_i \in A_i$. 换言之,

$$A_1 \times A_2 \times \dots \times A_n = \{(a_1, a_2, \dots, a_n) \mid a_i \in A_i, i=1, 2, \dots, n\}$$

例 18 笛卡儿积 $A \times B \times C$ 是什么, 其中 $A=\{0, 1\}, B=\{1, 2\}, C=\{0, 1, 2\}$?

解 这一笛卡儿积由所有有序三元组 (a, b, c) 组成, 其中 $a \in A, b \in B, c \in C$. 因此,

$$A \times B \times C = \{(0, 1, 0), (0, 1, 1), (0, 1, 2), (0, 2, 0), (0, 2, 1), (0, 2, 2), \\ (1, 1, 0), (1, 1, 1), (1, 1, 2), (1, 2, 0), (1, 2, 1), (1, 2, 2)\}$$

2.1.4 使用带量词的集合符号

有时我们在符号中显式说明语句的论域。特别地, $\forall x \in S(P(x))$ 表示 $P(x)$ 的全称量化, 其中论域是集合 S . 换句话说, $\forall x \in S(P(x))$ 是 $\forall x(x \in S \rightarrow P(x))$ 的简写。类似地, $\exists x \in S(P(x))$ 表示 $P(x)$ 的存在量化, 论域是 S . 也就是说, $\exists x \in S(P(x))$ 是 $\exists x(x \in S \wedge P(x))$ 的简写。

例 19 语句 $\forall x \in \mathbf{R}(x^2 \geq 0)$ 和 $\exists x \in \mathbf{Z}(x^2 = 1)$ 的含义是什么?

解 语句 $\forall x \in \mathbf{R}(x^2 \geq 0)$ 声称对任意实数 $x, x^2 \geq 0$. 这个语句可以表达为“任意实数的平方是非负的”。这是真语句。

语句 $\exists x \in \mathbf{Z}(x^2 = 1)$ 声称存在一个整数 x 使得 $x^2 = 1$. 这个语句可以表达为“有某个整数, 其平方是 1”。这个语句也是真语句, 因为 $x=1$ 就是这样一个整数 (-1 也是)。

2.1.5 量词的真值集合

我们把集合理论和谓词逻辑的概念结合在一起。给定谓词 P 和论域 D , 定义 P 的真值集合: D 中元素 x 使 $P(x)$ 为真的元素组成的集合。 $P(x)$ 的真值集合记为 $\{x \in D \mid P(x)\}$ 。

例 20 谓词 $P(x), Q(x), R(x)$ 的真值集合都是什么? 其中论域是整数集合, $P(x)$ 是“ $|x|=1$ ”, $Q(x)$ 是“ $x^2=2$ ”, $R(x)$ 是“ $|x|=x$ ”。

解 P 的真值集合 $\{x \in \mathbf{Z} \mid |x|=1\}$ 是满足 $|x|=1$ 的整数集合。因为当 $x=1$ 或 $x=-1$ 时, $|x|=1$, 而没有其他的整数。 P 的真值集合是 $\{-1, 1\}$ 。

Q 的真值集合 $\{x \in \mathbf{Z} \mid x^2=2\}$ 是满足 $x^2=2$ 的整数集合。因为没有整数 x 满足 $x^2=2$, 所以这是个空集。

R 的真值集合 $\{x \in \mathbf{Z} \mid |x|=x\}$ 是满足 $|x|=x$ 的整数集合。因为当且仅当 $x \geq 0$ 时, $|x|=x$, 所以 R 的真值集合是 $\mathbf{N}$, 非负整数集合。

注意, 在域 U 上 $\forall x P(x)$ 为真当且仅当 P 的真值集合是集合 U 。同样, 域 U 上 $\exists x P(x)$ 为真, 当且仅当 P 的真值集合非空。

练习

1. 列出下述集合的成员。

a) $\{x \mid x \text{ 是使得 } x^2=1 \text{ 的实数}\}$

b) $\{x \mid x \text{ 是小于 } 12 \text{ 的正整数}\}$

c) $\{x \mid x \text{ 是某个整数的平方且 } x < 100\}$

d) $\{x \mid x \text{ 是整数且 } x^2=2\}$

2. 判断下面每对集合是否相等。

a) $\{1, 3, 3, 3, 5, 5, 5, 5, 5\}, \{5, 3, 1\}$

b) $\{\{1\}\}, \{1, \{1\}\}$

c) $\emptyset, \{\emptyset\}$

3. 对下面的每个集合, 判断 2 是否为它的元素。

a) $\{x \in \mathbf{R} \mid x \text{ 是大于 } 1 \text{ 的整数}\}$

b) $\{x \in \mathbf{R} \mid x \text{ 是某整数的平方}\}$

c) $\{2, \{2\}\}$

d) $\{\{2\}, \{\{2\}\}\}$

e) $\{\{2\}, \{2, \{2\}\}\}$

f) $\{\{\{2\}\}\}$

4. 判断下列语句是真还是假。

a) $0 \in \emptyset$

b) $\emptyset \in \{0\}$

c) $\{0\} \subset \emptyset$

d) $\emptyset \subset \{0\}$

e) $\{0\} \in \{0\}$

f) $\{0\} \subset \{0\}$

g) $\{\emptyset\} \subseteq \{\emptyset\}$

5. 判断下列语句是真还是假。

a) $x \in \{x\}$

b) $\{x\} \subseteq \{x\}$

c) $\{x\} \in \{x\}$

d) $\{x\} \in \{\{x\}\}$

e) $\emptyset \subseteq \{x\}$

f) $\emptyset \in \{x\}$

6. 用文氏图说明在一年所有的月份集合中不包含字母 R 的所有月份的集合。
7. 用文氏图说明集合关系 $A \subset B$ 和 $B \subset C$ 。
8. 假定 A 、 B 和 C 为集合，且 $A \subseteq B$ ， $B \subseteq C$ 。证明 $A \subseteq C$ 。
9. 下列各集合的基数是什么？
- a) $\{a\}$ b) $\{\{a\}\}$ c) $\{a, \{a\}\}$ d) $\{a, \{a\}, \{a, \{a\}\}\}$
10. 找出下列各集合的幂集合。
- a) $\{a\}$ b) $\{a, b\}$ c) $\{\emptyset, \{\emptyset\}\}$
11. 下列各集合各有多少个元素？
- a) $P(\{a, b, \{a, b\}\})$ b) $P(\{\emptyset, a, \{a\}, \{\{a\}\}\})$ c) $P(P(\emptyset))$
12. 令 $A = \{a, b, c, d\}$ ， $B = \{y, z\}$ 。求
- a) $A \times B$ b) $B \times A$
13. 什么是笛卡儿积 $A \times B \times C$ ，其中 A 是所有航线的集合， B 和 C 都是所有美国城市的集合。
14. 令 A 为集合，求证 $\emptyset \times A = A \times \emptyset = \emptyset$ 。
15. 如果 A 有 m 个元素， B 有 n 个元素， $A \times B$ 有多少个元素？
16. 解释为什么 $A \times B \times C$ 和 $(A \times B) \times C$ 不同。
17. 将下列量化表达式翻译成汉语句子并确定其真值。
- a) $\forall x \in \mathbb{R}(x^2 \neq -1)$ b) $\exists x \in \mathbb{Z}(x^2 = 2)$ c) $\forall x \in \mathbb{Z}(x^2 > 0)$ d) $\exists x \in \mathbb{R}(x^2 = x)$
18. 给出以下各个谓词的真值集合，这里域是整数集合。
- a) $P(x)$ ：“ $x^2 < 3$ ” b) $Q(x)$ ：“ $x^2 > x$ ” c) $R(x)$ ：“ $2x + 1 = 0$ ”
- *19. 证明：如果有序偶 (a, b) 可以用集合术语定义为 $\{\{a\}, \{a, b\}\}$ ，那么 $(a, b) = (c, d)$ 当且仅当 $a = c$ 且 $b = d$ 。[提示：首先证明 $\{\{a\}, \{a, b\}\} = \{\{c\}, \{c, d\}\}$ 当且仅当 $a = c$ 且 $b = d$ 。]
- *20. 给出可以列出有限集合所有子集的步骤。

2.2 集合运算

2.2.1 引言

两个集合可以以许多不同的方式结合在一起。例如，从学校主修数学课的集合和主修计算机科学课的集合入手，可以构成主修数学或计算机科学的学生集合，既主修数学又主修计算机科学的学生集合，不主修数学的学生集合，等等。

定义 1 令 A 和 B 为集合。 A 和 B 的并集用 $A \cup B$ 表示，这是 A 或 B 中或同时在 A 和 B 中的元素组成的集合。

元素 x 属于 A 和 B 的并集当且仅当 x 属于 A 或 x 属于 B 。这说明

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

图 2-3 中的文氏图表示集合 A 和 B 的并集。代表集合 A 的圆圈上和代表 B 的圆圈内的阴影区域表示 A 和 B 的并集。

我们将给出集合并集的例子。

例 1 集合 $\{1, 3, 5\}$ 和集合 $\{1, 2, 3\}$ 的并集是集合 $\{1, 2, 3, 5\}$ ，即

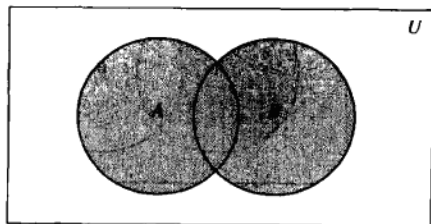
$$\{1, 3, 5\} \cup \{1, 2, 3\} = \{1, 2, 3, 5\}$$

例 2 学校主修计算机科学的学生集合与主修数学的学生集合的并集，是或主修数学或主修计算机科学或同时主修这两门课的学生的集合。

定义 2 令 A 和 B 为集合。 A 和 B 的交集用 $A \cap B$ 表示，这是既在 A 中又在 B 中的那些元素的集合。

元素 x 属于集合 A 和 B 的交集当且仅当 x 属于 A 而且 x 属于 B 。这说明

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



$A \cup B$ 为阴影区

图 2-3 表示 A 和 B 并集的文氏图

图 2-4 中的文氏图表示集合 A 和 B 的交集。同时在代表 A 和 B 的两个圆之内的阴影区域表示 A 和 B 的交集。

我们给出交集的几个例子。

例 3 集合 $\{1, 3, 5\}$ 和 $\{1, 2, 3\}$ 的交集是 $\{1, 3\}$ ，即 $\{1, 3, 5\} \cap \{1, 2, 3\} = \{1, 3\}$ 。

例 4 学校所有主修计算机科学的学生集合与所有主修数学的学生集合，是所有既主修计算机科学又主修数学的学生的集合。

定义 3 如果两个集合的交集为空集，就说它们不相交。

例 5 令 $A = \{1, 3, 5, 7, 9\}$ 而 $B = \{2, 4, 6, 8, 10\}$ 。因为 $A \cap B = \emptyset$ ，所以 A 和 B 不相交。

还有其他组合集合的重要方式。

定义 4 令 A 和 B 为集合。 A 和 B 的差集用 $A - B$ 表示，这是包含只属于 A 而不属于 B 的所有元素的集合。 A 和 B 的差集也称为 B 对于 A 的补集。

元素 x 属于 A 和 B 的差集当且仅当 $x \in A$ 且 $x \notin B$ ，这说明

$$A - B = \{x \mid x \in A \wedge x \notin B\}$$

图 2-5 中的文氏图表示集合 A 和 B 的差集。既在表示集合 A 的圆圈内部也在表示集合 B 的圆圈外部的阴影区域表示 $A - B$ 。

让我们举几个差集的例子。

例 6 集合 $\{1, 3, 5\}$ 和 $\{1, 2, 3\}$ 的差集是 $\{5\}$ ；即 $\{1, 3, 5\} - \{1, 2, 3\} = \{5\}$ 。这不同于 $\{1, 2, 3\}$ 和 $\{1, 3, 5\}$ 的差集 $\{2\}$ 。

例 7 学校主修计算机科学的学生集合和主修数学的学生集合的差集是学校主修计算机科学但不主修数学的学生集合。

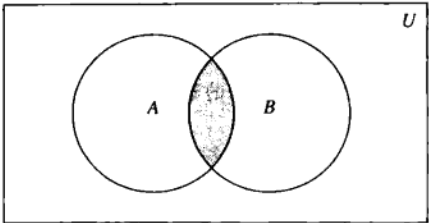
一旦指定了全集 U ，就可以定义集合的补集。

定义 5 令 U 为全集。集合 A 的补集用 $\bar{A}$ 表示，这是 A 对于 U 的补集。集合 A 的补集是 $U - A$ 。

元素 x 属于 $\bar{A}$ 当且仅当 $x \notin A$ 。这说明

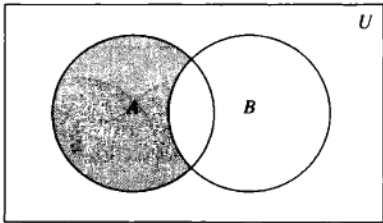
$$\bar{A} = \{x \mid x \notin A\}$$

图 2-6 中代表集合 A 的圆圈外面的阴影区域表示 $\bar{A}$ 。



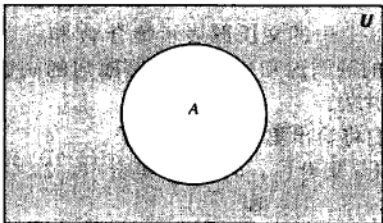
$A \cap B$ 为阴影区

图 2-4 表示 A 和 B 交集的文氏图



$A - B$ 为阴影区

图 2-5 A 和 B 的差集的文氏图



$\bar{A}$ 为阴影区

图 2-6 集合 A 的补集的文氏图

我们举几个补集的例子。

例 8 令 $A = \{a, e, i, o, u\}$ (全集为英语字母集)。那么

$$\bar{A} = \{b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z\}$$

例 9 令 A 为大于 10 的正整数的集合 (全集为正整数集合)。那么

$$\bar{A} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

2.2.2 集合恒等式

表 2-1 列出了最重要的集合恒等式。本节将用三种不同的方法证明其中的几个。介绍这些方法是想说明，对一个问题的解往往有不同的途径。表中未证明的恒等式留给读者练习。读者应该注意这些集合恒等式和 1.2 节讨论的逻辑等价的相似之处。事实上，这里给出的集合恒等式可以直接由对应的逻辑等价证明。不仅如此，而且这两者都是布尔代数中成立的恒等式的特殊情况。

表 2-1 集合恒等式

等 式	名 称
$A \cup \emptyset = A$ $A \cap U = A$	恒等律
$A \cup U = U$ $A \cap \emptyset = \emptyset$	支配律
$A \cup A = A$ $A \cap A = A$	幂等律
$\overline{(\overline{A})} = A$	补集律
$A \cup B = B \cup A$ $A \cap B = B \cap A$	交换律
$A \cup (B \cap C) = (A \cup B) \cap C$ $A \cap (B \cup C) = (A \cap B) \cup C$	结合律
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	分配律
$\overline{A \cup B} = \overline{A} \cap \overline{B}$ $\overline{A \cap B} = \overline{A} \cup \overline{B}$	德摩根定律
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	吸收律
$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$	补律

证明集合相等的一种方法是证明两者中任何一个都是另一个的子集。为了证明一个集合是另一个集合的子集，可以通过证明一个元素如果属于第一个集合，必定属于第二个集合。也就是用直接的方法证明。我们将以确立德摩根第二定律为例说明这一方法。

例 10 证明 $\overline{A \cap B} = \overline{A} \cup \overline{B}$ 。

解 为了证明 $\overline{A \cap B} = \overline{A} \cup \overline{B}$ ，我们需要证明 $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$ 和 $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$ 。

首先，证明 $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$ 。假定 $x \in \overline{A \cap B}$ ，根据补的定义， $x \notin A \cap B$ ，由交的定义， $\neg((x \in A) \wedge (x \in B))$ 为真，(由逻辑)用德摩根定律可得 $\neg(x \in A) \vee \neg(x \in B)$ 。因此，根据否定的定义，有 $x \notin A$ 或 $x \notin B$ 。再由补的定义， $x \in \overline{A}$ 或 $x \in \overline{B}$ ；再由并， $x \in \overline{A} \cup \overline{B}$ ，因此，得证 $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$ 。

然后，证明 $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$ 。现假设 $x \in \overline{A} \cup \overline{B}$ ；由并的定义， $x \in \overline{A}$ 或 $x \in \overline{B}$ ；用补的定义， $x \in A$ 或 $x \notin B$ ；所以， $\neg(x \in A) \vee (x \in B)$ 为真。(由逻辑)用德摩根定律可得， $\neg(x \in A) \wedge (x \in B)$ 为真。由交的定义， $\neg(x \in A \cap B)$ 成立；由补的定义，由 $x \in \overline{A \cap B}$ ，这表明 $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$ 。由于已证明这两个集合中，一个是另一个的子集，所以，这两个集合相等，即得证恒等。 ■

另一个更简洁的方法是使用集合构造符，如例 11 所示。

例 11 用集合构造符和逻辑等价证明 $\overline{A \cap B} = \overline{A} \cup \overline{B}$ 。

解 下列一串等式提供了这一恒等关系的证明：

$$\begin{aligned}
 \overline{A \cap B} &= \{x \mid x \notin A \cap B\} && \text{根据补的定义} \\
 &= \{x \mid \neg(x \in (A \cap B))\} && \text{根据不属于符号的含义} \\
 &= \{x \mid \neg(x \in A \wedge x \in B)\} && \text{根据交的定义} \\
 &= \{x \mid \neg(x \in A) \vee \neg(x \in B)\} && \text{根据逻辑等价式的德摩根第一定律} \\
 &= \{x \mid x \notin A \vee x \notin B\} && \text{根据不属于符号的含义} \\
 &= \{x \mid x \in \overline{A} \vee x \in \overline{B}\} && \text{根据补的定义} \\
 &= \{x \mid x \in \overline{A} \cup \overline{B}\} && \text{根据并的定义} \\
 &= \overline{A} \cup \overline{B} && \text{根据集合表示的含义}
 \end{aligned}$$

注意，除了补、并、集合成员、集合表示的定义，这个证明还用到了逻辑等价式的德摩根第一定律。 ■

要证明涉及两个以上集合的集合恒等，可以证明恒等式的每一边是另一边的子集，这我们可用例 12 来说明。

例 12 证明表 2-1 中的第一个分配律：对任意集合 A , B 和 C ，证明

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

解 我们将通过说明等式的每一边是另一边的子集来证明这个恒等式。

假定 $x \in A \cap (B \cup C)$ ，那么 $x \in A$ 且 $x \in B \cup C$ 。根据集合并的定义，可得 $x \in A$ 且 $x \in B$ 或 $x \in C$ ；因此， $x \in A$ 且 $x \in B$ 或 $x \in A$ 且 $x \in C$ 。根据交的定义，可知 $x \in A \cap B$ 或 $x \in A \cap C$ ；使用并的定义，可得出 $x \in (A \cap B) \cup (A \cap C)$ ；由此 $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$ 。

现在假定 $x \in (A \cap B) \cup (A \cap C)$ ，那么，由并的定义， $x \in A \cap B$ 或 $x \in A \cap C$ ；由交的定义，可得 $x \in A$ 且 $x \in B$ 或 $x \in A$ 且 $x \in C$ ；由此可知， $x \in A$ ， $x \in B$ 或 $x \in C$ ；因此，由并的定义可知， $x \in A$ 且 $x \in B \cup C$ 。而且，由交的定义，可得 $x \in A \cap (B \cup C)$ ；最终得出 $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$ 。这便完成了恒等式的证明。 ■

还可以用成员表来证明集合恒等式。我们考虑一个元素可能属于的集合的每一种组合，并证明在同样的集合组合中的元素属于恒等式两边的集合。用 1 表示元素属于一个集合，用 0 表示元素不属于一个集合（读者应注意成员表和真值表的相似之处）。

例 13 用成员表证明 $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ 。

解 表 2-2 给出了这些集合组合的成员表。这表格有 8 行。由于对应于 $A \cap (B \cup C)$ 和 $(A \cap B) \cup (A \cap C)$ 的两列相同，恒等式有效。 ■

表 2-2 分配性质的成员表

A	B	C	$B \cup C$	$A \cap (B \cup C)$	$A \cap B$	$A \cap C$	$(A \cap B) \cup (A \cap C)$
1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1
1	0	1	1	1	0	1	1
1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0

用我们已经证明的这些集合恒等式可以证明其他的集合恒等式。考虑下面的例子。

例 14 令 A, B, C 为集合。证明

$$\overline{A \cup (B \cap C)} = (\bar{C} \cup \bar{B}) \cap \bar{A}$$

解 我们有

$$\begin{aligned} \overline{A \cup (B \cap C)} &= \bar{A} \cap \overline{(B \cap C)} && \text{由德摩根第一律} \\ &= \bar{A} \cap (\bar{B} \cup \bar{C}) && \text{由德摩根第二律} \\ &= (\bar{B} \cup \bar{C}) \cap \bar{A} && \text{由交集的交换律} \\ &= (\bar{C} \cup \bar{B}) \cap \bar{A} && \text{由并集的交换律} \end{aligned}$$

2.2.3 扩展的并集和交集

由于集合的并集和交集满足结合律，所以只要 A, B, C 为集合， $A \cup B \cup C$ 和 $A \cap B \cap C$ 均有定义。也就是说，这样表示的含义都是很明确的。我们不需要用括号来指明哪个运算在前，因为 $A \cup (B \cup C) = (A \cup B) \cup C$ 及 $A \cap (B \cap C) = (A \cap B) \cap C$ 。注意 $A \cup B \cup C$ 包含那些至少属于 A, B, C 中一个集合的元素，而 $A \cap B \cap C$ 包含那些属于 A, B, C 全部三个集合的元素。三个集合 A, B, C 的这两种组合如图 2-7 所示。

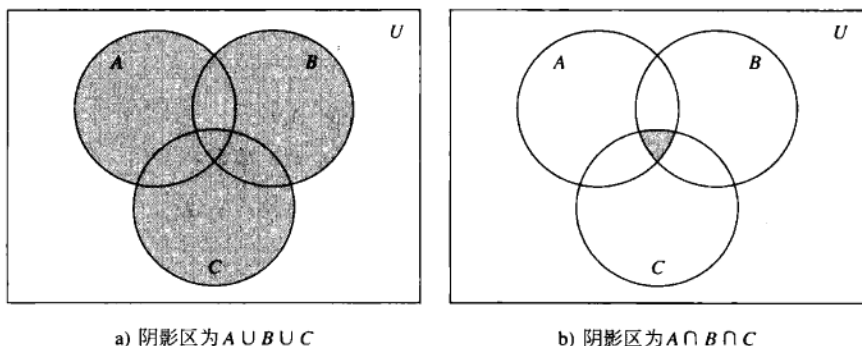


图 2-7 集合 A, B, C 的并集和交集

例 15 令 $A = \{0, 2, 4, 6, 8\}$, $B = \{0, 1, 2, 3, 4\}$, $C = \{0, 3, 6, 9\}$ 。 $A \cup B \cup C$ 和 $A \cap B \cap C$ 是什么集合？

解 $A \cup B \cup C$ 包括那些至少属于 A, B, C 之一的元素。所以

$$A \cup B \cup C = \{0, 1, 2, 3, 4, 6, 8, 9\}$$

集合 $A \cap B \cap C$ 包括那些属于全部三个集合的元素。因此

$$A \cap B \cap C = \{0\}$$

可以考虑任意多个集合的并集和交集。我们采用下面的定义。

定义 6 一组集合的并集是包含那些至少是这组集合中一个集合成员的元素的集合。
用记号

$$A_1 \cup A_2 \cup \cdots \cup A_n = \bigcup_{i=1}^n A_i$$

表示集合 $A_1, A_2, \dots, A_n$ 的并集。

定义 7 一组集合的交集是包含那些属于这组集合中所有成员集合的元素的集合。
用记号

$$A_1 \cap A_2 \cap \cdots \cap A_n = \bigcap_{i=1}^n A_i$$

表示集合 $A_1, A_2, \dots, A_n$ 的交集。

我们用例 16 说明扩展的并集和交集。

例 16 令 $A_i = \{i, i+1, i+2, \dots\}$ 。那么

$$\bigcup_{i=1}^n A_i = \bigcup_{i=1}^n \{i, i+1, i+2, \dots\} = \{1, 2, 3, \dots\}$$

而

$$\bigcap_{i=1}^n A_i = \bigcap_{i=1}^n \{i, i+1, i+2, \dots\} = \{n, n+1, n+2, \dots\}$$

我们可以扩展已经介绍的并集和交集到类似的集合。尤其可以使用公式

$$A_1 \cup A_2 \cup \dots \cup A_n \cup \dots = \bigcup_{i=1}^{\infty} A_i$$

去表示集合 $A_1, A_2, \dots, A_n, \dots$ 的并集。类似地，这些集合的交集可以表示为

$$A_1 \cap A_2 \cap \dots \cap A_n \cap \dots = \bigcap_{i=1}^{\infty} A_i$$

大体上，公式 $\bigcap_{i \in I} A_i$ 和 $\bigcup_{i \in I} A_i$ 分别用于表示对于 $i \in I$ 集合 A_i 的并集和交集。注意，我们这样使用：

$$\bigcap_{i \in I} A_i = \{x \mid \forall i \in I (x \in A_i)\}$$

和

$$\bigcup_{i \in I} A_i = \{x \mid \exists i \in I (x \in A_i)\}$$

例 17 对于 $i=1, 2, 3, \dots$ ，集合 $A_i = \{1, 2, 3, \dots, i\}$ 。那么，

$$\bigcup_{i=1}^{\infty} A_i = \bigcup_{i=1}^{\infty} \{1, 2, 3, \dots, i\} = \{1, 2, 3, \dots\} = \mathbf{Z}^+$$

和

$$\bigcap_{i=1}^{\infty} A_i = \bigcap_{i=1}^{\infty} \{1, 2, 3, \dots, i\} = \{1\}$$

可以看到，这些集合的并集是正整数集，每一个正整数至少属于一个集合，因为整数 n 属于 $A_n = \{1, 2, \dots, n\}$ ，并集中的每一个元素都是正整数。注意到属于所有的集合的元素只有 1，即这些集合的交集。也就是说， $A_1 = \{1\}$ ，而且 $1 \in A_i (i=1, 2, \dots)$ 。

2.2.4 计算机表示集合的方式

计算机表示集合的方式多种多样。一种办法是把集合的元素无序地存储起来。可是如果这样做，在做集合的并集、交集或差集等运算时会浪费时间，因为这些运算将需要大量的元素检索。我们将要介绍一种利用全集元素的一个任意排序存放元素以表示集合的方法。集合的这种表示法使我们很容易计算集合的组合。

假定全集 U 是有限的（而且大小合适，使 U 的元素个数不超过计算机能使用的内存量）。首先为 U 的元素任意规定一个顺序，例如 $a_1, a_2, \dots, a_n$ 。于是可以用长度为 n 的位串表示 U 的子集 A ：如果 a_i 属于 A ，则位串中第 i 位是 1；如果 a_i 不属于 A ，则位串中第 i 位是 0。下面的例子阐明了这一方法。

例 18 令 $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ ，而且 U 的元素从小到大排序，即 $a_i = i$ 。表示 U 中所有奇数的子集、所有偶数的子集和不超过 5 的整数的子集的位串是什么？

解 表示 U 中所有奇数的子集即 $\{1, 3, 5, 7, 9\}$ 的位串，其第 1、3、5、7、9 位为 1，其他位为 0，即

$$10\ 1010\ 1010$$

（我们已把长度为 10 的位串分成长度为 4 的片段组合以便阅读，因为长位串不易读）。类似地， U 中所有偶数的子集即 $\{2, 4, 6, 8, 10\}$ ，由位串

$$01\ 0101\ 0101$$

表示。 U 中不超过5的所有整数的集合即 $\{1, 2, 3, 4, 5\}$ ，由位串

$$11\ 1110\ 0000$$

表示。

用位串表示集合便于计算集合的补集、并集、交集和差集。要从表示集合的位串计算它的补集的位串，只需简单地把每个1改为0，每个0改为1，因为 $x \in A$ 当且仅当 $x \notin \bar{A}$ 。注意，当把每个字位看成是真值时，上述运算对应于取每个字位的非，因为1表示真，0表示假。

例 19 我们已经知道集合 $\{1, 3, 5, 7, 9\}$ 的位串(全集为 $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$)是

$$10\ 1010\ 1010$$

它的补集的位串是什么？

解 用0取代1，用1取代0，即可得到此集合的补集的位串

$$01\ 0101\ 0101$$

这对应着集合 $\{2, 4, 6, 8, 10\}$ 。

要得到两个集合的并集和交集的位串，我们可以对表示这两个集合的位串按位做布尔运算。只要两个位串的第 i 字位有一个是1，则并集的位串的第 i 位是1，当两个字位都是0时为0。因此，并集的位串是两个集合位串的按位或(bitwise OR)。当两个位串的第 i 字位均为1时，交集的位串第 i 位为1，否则为0。因此交集的位串是两个集合位串的按位与(bitwise AND)。

例 20 集合 $\{1, 2, 3, 4, 5\}$ 和 $\{1, 3, 5, 7, 9\}$ 的位串分别是11 1110 0000 和 10 1010 1010。用位串找出它们的并集和交集。

解 这两个集合的并集的位串是

$$11\ 1110\ 0000 \vee 10\ 1010\ 1010 = 11\ 1110\ 1010$$

它表示的集合是 $\{1, 2, 3, 4, 5, 7, 9\}$ 。这两个集合的交集的位串是

$$11\ 1110\ 0000 \wedge 10\ 1010\ 1010 = 10\ 1010\ 0000$$

它表示的集合是 $\{1, 3, 5\}$ 。

练习

1. 令 A 为住在离学校一英里以内的所有学生的集合， B 是走路上学的所有学生的集合。描述一下下列各集合中的学生：

$$\text{a) } A \cap B \qquad \text{b) } A \cup B \qquad \text{c) } A - B \qquad \text{d) } B - A$$

2. 令 $A = \{1, 2, 3, 4, 5\}$ ， $B = \{0, 3, 6\}$ 。求

$$\text{a) } A \cup B \qquad \text{b) } A \cap B \qquad \text{c) } A - B \qquad \text{d) } B - A$$

在练习3~5中，假定 A 是某个未知全集 U 的子集。

3. 证明表2-1中的补集律： $\overline{\bar{A}} = A$ 。

4. 证明表2-1中的支配律：

$$\text{a) } A \cup U = U \qquad \text{b) } A \cap \emptyset = \emptyset$$

5. 证明表2-1中的交换律：

$$\text{a) } A \cup \bar{A} = U \qquad \text{b) } A \cap \bar{A} = \emptyset$$

6. 已知 A 和 B 为两个集合，试证明表2-1中的交换律：

$$\text{a) } A \cup B = B \cup A \qquad \text{b) } A \cap B = B \cap A$$

7. 证明表2-1中的第二个吸收律：如果 A 和 B 为两个集合，那么 $A \cap (A \cup B) = A$ 。

8. 通过以下两种方式证明表2-1中的第一个德摩根律：如果 A 和 B 为两个集合，那么 $\overline{A \cup B} = \bar{A} \cap \bar{B}$ 。

a) 证明两边互为子集 b) 使用成员表

9. 设 A, B, C 为集合。证明 $\overline{A \cap B \cap C} = \bar{A} \cup \bar{B} \cup \bar{C}$ 。

a) 用证明等式两边互为子集的办法 b) 用成员表

10. 如果 A 和 B 为集合，证明 $A - B = A \cap \bar{B}$ 。

11. 证明表2-1中的第一个结合律：如果 A, B, C 均为集合，那么 $A \cup (B \cap C) = (A \cup B) \cap C$ 。

12. 证明表2-1中的第二个分配律：如果 A, B, C 均为集合，那么 $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ 。

13. 令 $A = \{0, 2, 4, 6, 8, 10\}$, $B = \{0, 1, 2, 3, 4, 5, 6\}$, $C = \{4, 5, 6, 7, 8, 9, 10\}$ 。求
 a) $A \cap B \cap C$ b) $A \cup B \cup C$ c) $(A \cup B) \cap C$ d) $(A \cap B) \cup C$
14. 画出以下集合 A, B, C, D 的各个组合的文氏图:
 a) $A \cap (B - C)$ b) $(A \cap B) \cup (A \cap C)$ c) $(A \cap \bar{B}) \cup (A \cap \bar{C})$
15. 如果有关集合 A 与 B 的下列性质为真, 你能就 A 和 B 说些什么?
 a) $A \cup B = A$ b) $A \cap B = A$ c) $A - B = A$
 d) $A \cap B = B \cap A$ e) $A - B = B - A$
16. 令 A 和 B 为全集 U 的子集。证明 $A \subseteq B$ 当且仅当 $\bar{B} \subseteq \bar{A}$ 。

集合 A 和 B 的对称差, 用 $A \oplus B$ 表示, 是属于 A 或属于 B 但不同时属于 A 与 B 的元素组成的集合。

17. 求某校主修计算机科学的学生集合与主修数学的学生集合的对称差。
 18. 证明 $A \oplus B = (A \cup B) - (A \cap B)$ 。
 19. 证明: 若 A 是全集 U 的子集, 则
 a) $A \oplus A = \emptyset$ b) $A \oplus \emptyset = A$ c) $A \oplus U = \bar{A}$ d) $A \oplus \bar{A} = U$
20. 如果 $A \oplus B = A$, 你能就集合 A 和 B 说些什么?
 *21. 假定 A, B, C 为集合, 使得 $A \oplus C = B \oplus C$ 。是否必定有 $A = B$?
 22. 若 A, B, C, D 为集合, $(A \oplus B) \oplus (C \oplus D) = (A \oplus D) \oplus (B \oplus C)$ 是否成立?
 *23. 令 $A_i = \{1, 2, 3, \dots, i\}$, $i = 1, 2, 3, \dots$ 。求

a) $\bigcup_{i=1}^n A_i$ b) $\bigcap_{i=1}^n A_i$

24. 令 A_i 为所有长度不超过 i 的非空位串(即长度至少为 1)的集合。求

a) $\bigcup_{i=1}^n A_i$ b) $\bigcap_{i=1}^n A_i$

25. 试求 $\bigcup_{i=1}^{\infty} A_i$ 和 $\bigcap_{i=1}^{\infty} A_i$, 如果对于任一正整数 i ,
 a) $A_i = \{-i, -i+1, \dots, -1, 0, 1, \dots, i-1, i\}$
 b) $A_i = \{-i, i\}$
 c) $A_i = [-i, i]$, 即满足 $-i \leq x \leq i$ 的实数 x 的集合
 d) $A_i = [i, \infty]$, 即满足 $x \geq i$ 的实数 x 的集合

26. 使用上题中的同一个全集, 求下列位串各自代表的集合。
 a) 11 1100 1111 b) 01 0111 1000 c) 10 0000 0001

27. 对应于两个集合之差的位串是什么?

28. 令 $A = \{a, b, c, d, e\}$, $B = \{b, c, d, g, p, t, v\}$, $C = \{c, e, i, o, u, x, y, z\}$, $D = \{d, e, h, i, n, o, t, u, x, y\}$ 。说明怎样用位串的按位运算求下列集合:

a) $A \cup B$ b) $A \cap B$ c) $(A \cup D) \cap (B \cup C)$ d) $A \cup B \cup C \cup D$

29. 集合 A 的后继是 $A \cup \{A\}$ 。求下列集合的后继。

a) $\{1, 2, 3\}$ b) $\emptyset$ c) $\{\emptyset\}$ d) $\{\emptyset, \{\emptyset\}\}$

有时一个元素在一个无序集中出现的次数也有意义。当同一个元素作为成员可以出现不止一次时, 这个无序元素集就是多重集。符号 $\{m_1 \cdot a_1, m_2 \cdot a_2, \dots, m_r \cdot a_r\}$ 表示的是 a_1 出现 m_1 次, a_2 出现 m_2 次等的多重集。整数 m_i 称为 a_i 的重数, $i = 1, 2, \dots, r$ 。

令 P 和 Q 为多重集, 多重集 P 和 Q 的并集是个多重集, 其中每个元素的重数是该元素在 P 和 Q 中的重数的最大值。 P 和 Q 的交集是个多重集, 其中每个元素的重数是该元素在 P 和 Q 中重数的最小值。 P 和 Q 的差集是个多重集, 其中每个元素的重数是该元素在 P 中的重数减去它在 Q 中的重数, 当然这不能是个负数, 若是负数就以 0 作重数。 P 和 Q 的和集是个多重集, 其中每个元素的重数是该元素在 P 和 Q 中的重数的和, P 和 Q 的并集、交集和差集分别用 $P \cup Q$ 、 $P \cap Q$ 和 $P - Q$ 表示(不要混淆这些运算和集合上的类似运算)。 P 和 Q 的和集用 $P + Q$ 表示。

30. 令 A 和 B 分别为多重集 $\{3 \cdot a, 2 \cdot b, 1 \cdot c\}$ 和 $\{2 \cdot a, 3 \cdot b, 4 \cdot d\}$ 。求

a) $A \cup B$ b) $A \cap B$ c) $A - B$ d) $B - A$ e) $A + B$

人工智能中使用模糊集合。全集 U 中每个元素在模糊集合 S 中都有个成员度，即 0 和 1 之间(包括 0 和 1)的实数。模糊集合 S 的表示法是列出其元素及成员度(成员度为 0 的元素不列)。例如，用 $\{0.6 \text{ Alice}, 0.9 \text{ Brian}, 0.4 \text{ Fred}, 0.1 \text{ Oscar}, 0.5 \text{ Rita}\}$ 表示名人集合 F ，说明 F 中 Alice 的成员度为 0.6，Brian 为 0.9，Fred 为 0.4，Oscar 为 0.1，而 Rita 为 0.5(因此这些人里面 Brian 最出名而 Oscar 最不出名)。再假定 R 是富人集合， $R = \{0.4 \text{ Alice}, 0.8 \text{ Brian}, 0.2 \text{ Fred}, 0.9 \text{ Oscar}, 0.7 \text{ Rita}\}$ 。

31. 模糊集合 S 的补是集合 $\bar{S}$ ，元素在 $\bar{S}$ 中的成员度等于 1 减去该元素在 S 中的成员度。求 $\bar{F}$ (不出名者的模糊集合)和 $\bar{S}$ (不富裕者的模糊集合)。
32. 模糊集合 S 和 T 的交集是模糊集合 $S \cap T$ ，其中每个元素的成员度是该元素在 S 和 T 中的成员度的最小值。求既出名又富裕者的模糊集合 $F \cap R$ 。

2.3 函数

2.3.1 引言

在许多情况下我们都会为一个集合的每个元素指派另一个集合(可以就是第一个集合)的一个特定元素。例如，假定对离散数学课的每个学生从 $\{A, B, C, D, F\}$ 中选一个字母作为他的得分。再假定 Adams 的得分是 A ，Chou 的得分是 C ，Goodfriend 的得分是 B ，Rodriguez 的得分是 A ，而 Stevens 的得分是 F 。这一打分如图 2-8 所示。

这种打分就是一个函数。离散数学中函数的概念分外重要。在定义像序列和字符串这样的离散结构时用到函数。函数还用于表示解决一定规模的问题需要的计算机机时。许多计算机程序和子程序被设计用来计算函数值。借助自身定义的函数称为递归函数。

定义 1 令 A 和 B 为非空集合。从 A 到 B 的函数 f 是对元素的一种指派，对 A 的每个元素恰好指派 B 的一个元素。如果 f 指派给 A 中元素 a 的唯一的 B 中元素是 b ，就写成 $f(a) = b$ 。如果 f 是从 A 到 B 的函数，就写成 $f: A \rightarrow B$ 。

注意 函数有时也被称作映射或者变换。

有许多描述函数的方式。有时明确说明指派关系(如图 2-8 所示)，但常用的方式是给出指派公式来定义函数，如 $f(x) = x + 1$ 。有时也用计算机程序来描述函数。

函数 $f: A \rightarrow B$ 也能由从 A 到 B 的关系来定义。见 2.1 节， A 到 B 的关系就是集合 $A \times B$ 的子集。对于 A 到 B 的关系，对每一个 $a \in A$ 的元素都有且仅有一个有序对 (a, b) ，则定义了 A 到 B 的函数 f 。这个函数通过 $f(a) = b$ 来定义。在这个以 a 为第一个元素的关系中， (a, b) 是唯一的有序对。

定义 2 如果 f 是从 A 到 B 的函数，就说 A 是 f 的定义域，而 B 是 f 的伴域。如果 $f(a) = b$ ，就说 b 是 a 的像而 a 是 b 的原像。 A 中元素的所有像元素的集合称为 f 的值域。若 f 是从 A 到 B 的函数，有时也说成 f 把 A 映射到 B 。

图 2-9 表示 A 到 B 的函数。

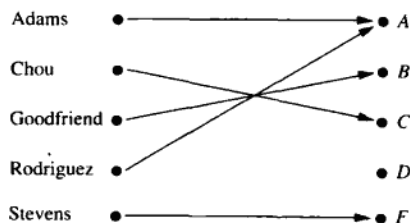


图 2-8 离散数学课成绩

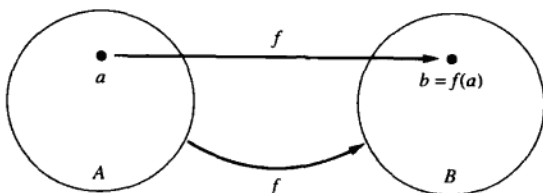


图 2-9 函数 f 把 A 映射到 B

当定义一个函数的时候，我们需要指定它的定义域、伴域、元素定义域到伴域的映射。当有相同的定义域、伴域，相同的映射关系时，我们说这两个函数是相等的。注意，如果改变函数的

定义域或伴域, 那么我们将得到一个不同的函数。如果改变元素的映射关系, 也会得到一个不同的函数。

例1~5 提供了函数的例子。在每个例子中, 我们都描述了定义域、伴域、值域和法则。

例1 引用本节开头的例子, 给学生打分来描述什么是函数的定义域、伴域、值域。

解 令 G 为函数, 表示在离散数学课上一个学生的得分, 例如 $G(\text{Adams}) = A$ 。则 G 的定义域是集合 $\{\text{Adams}, \text{Chou}, \text{Goodfriend}, \text{Rodriguez}, \text{Stevens}\}$, 伴域是集合 $\{A, B, C, D, F\}$ 。 G 的值域是 $\{A, B, C, F\}$, 因为没有学生得 D 。

例2 令 R 为包含有序对 $(\text{Abdul}, 22)$, $(\text{Brenda}, 24)$, $(\text{Carla}, 21)$, $(\text{Desire}, 22)$, $(\text{Eddie}, 24)$ 和 $(\text{Felicia}, 22)$ 的一个关系, 每一对表示一个学生的得分及其年龄。那么, 该关系 R 确定的函数是什么?

解 这个关系定义了函数 f , 有 $f(\text{Abdul}) = 22$, $f(\text{Brenda}) = 24$, $f(\text{Carla}) = 21$, $f(\text{Desire}) = 22$, $f(\text{Eddie}) = 24$, $f(\text{Felicia}) = 22$ 。在这里, 定义域是 $\{\text{Abdul}, \text{Brenda}, \text{Carla}, \text{Desire}, \text{Eddie}, \text{Felicia}\}$ 。为了定义函数 f , 需要指定一个伴域, 我们可以把正整数作为伴域以确保包含每一个学生可能的年龄(注意, 可以选择小一点的伴域, 但是这会改变函数), 最终值域是集合 $\{21, 22, 24\}$ 。

Extra Example

例3 令 f 为函数, 指派两个长度或者更大的位串的最后两位。例如, $f(11010) = 10$ 。

那么, f 的定义域就是所有两个长度或者更大的位串的集合。而伴域和值域都是集合 $\{00, 01, 10, 11\}$ 。

例4 令 f 为从 $\mathbb{Z}$ 到 $\mathbb{Z}$ 的函数, 它指派给每个整数的是该整数的平方。于是 $f(x) = x^2$, 而 f 的定义域是所有整数的集合, f 的伴域可以从所有整数的集合中选择, f 的值域是所有非负整数中那些完全平方的集合, 即 $\{0, 1, 4, 9, \dots\}$ 。

例5 函数的定义域和伴域往往以程序语言描述。例如 Java 语句

```
int floor(float real){...}
```

和 Pascal 语句

```
function floor(x:real):integer
```

都说的是, floor 函数的定义域是实数集合, 而它的伴域是整数集合。

具有相同定义域的两个实数值函数可以相加和相乘。

定义3 令 f_1 和 f_2 是从 A 到 $\mathbb{R}$ 的函数, 那么 $f_1 + f_2$ 和 $f_1 f_2$ 也是从 A 到 $\mathbb{R}$ 的函数, 其定义为

$$(f_1 + f_2)(x) = f_1(x) + f_2(x)$$

$$(f_1 f_2)(x) = f_1(x) f_2(x)$$

注意 $f_1 + f_2$ 和 $f_1 f_2$ 的定义是利用 f_1 和 f_2 在 x 的值计算它们在 x 的值。

例6 令 f_1 和 f_2 是从 $\mathbb{R}$ 到 $\mathbb{R}$ 的函数, 且 $f_1(x) = x^2$ 而 $f_2(x) = x - x^2$ 。函数 $f_1 + f_2$ 和 $f_1 f_2$ 是什么?

解 从函数的和与积的定义知

$$(f_1 + f_2)(x) = f_1(x) + f_2(x) = x^2 + (x - x^2) = x$$

$$(f_1 f_2)(x) = x^2(x - x^2) = x^3 - x^4$$

若 f 是从集合 A 到集合 B 的函数, 则可以定义 A 的子集的像。

定义4 令 f 为从集合 A 到集合 B 的函数, S 为 A 的一个子集。 S 的像是由 S 中元素的像组成的 B 的子集。我们用 $f(S)$ 表示 S 的像, 于是

$$f(S) = \{t \mid \exists s \in S (t = f(s))\}$$

我们也用简写 $\{f(s) \mid s \in S\}$ 来表示这个集合。

注意, S 的像是具有潜在不明确性的。这里, $f(S)$ 表示一个集合, 不是集合 S 的值域。

例7 令 $A = \{a, b, c, d, e\}$ 而 $B = \{1, 2, 3, 4\}$, 且 $f(a) = 2$, $f(b) = 1$, $f(c) = 4$, $f(d) = 1$ 及 $f(e) = 1$ 。子集 $S = \{b, c, d\}$ 的像是集合 $f(S) = \{1, 4\}$ 。

2.3.2 一对一函数和映上函数

有些函数在它们的定义域的不同成员上有不同的像。这种函数称为一对一的。

定义 5 函数 f 称为一对一的或单射的，当且仅当对于 f 的定义域中的所有 a 和 b ， $f(a) = f(b)$ 蕴含着 $a = b$ 。一对一的函数称为单射。

注意 函数 f 是一对一的，当且仅当只要 $a \neq b$ 就有 $f(a) \neq f(b)$ 。这种表达 f 为一对一函数的方式是对定义中的蕴含倒置而来。我们可以用量词，如 $\forall a \forall b (f(a) = f(b) \rightarrow a = b)$ 或等价地 $\forall a \forall b (a \neq b \rightarrow f(a) \neq f(b))$ ，来表达 f 是一对一的，其中论域是函数的定义域。

我们通过一对一的函数和不是一对一的函数示例来说明这个概念。

Assignment **例 8** 判断从 $\{a, b, c, d\}$ 到 $\{1, 2, 3, 4, 5\}$ 的函数 f 是否为一对一的， f 的定义是 $f(a) = 4$ ， $f(b) = 5$ ， $f(c) = 1$ 而 $f(d) = 3$ 。

Extra Examples **解** f 是一对一的，因为 f 在它定义域的四个元素上取不同的值。图 2-10 说明了这一点。

例 9 判断从整数集到整数集的函数 $f(x) = x^2$ 是否为一对一的。

解 函数 $f(x) = x^2$ 不是一对一的，因为，例如 $f(1) = f(-1) = 1$ ，但 $1 \neq -1$ 。

注意，若定义域限制为 $\mathbb{Z}^+$ ，函数 $f(x) = x^2$ 就是一对一的。(技术上说，当限定一个函数的定义域的时候，我们得到了一个新的函数，被限制的元素的值域与原来是相同的，而被限制的定義域以外的原来定义域的元素就不被限制的函数定义了。)

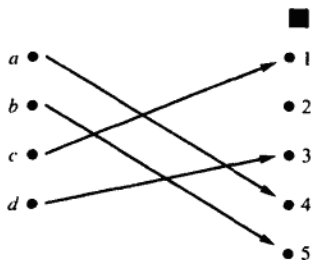


图 2-10 一个一对一函数

例 10 判断函数 $f(x) = x + 1$ 是否为实数集到它自身的一对一函数。

解 函数 $f(x) = x + 1$ 是一对一的。要证明这一点，只需注意在 $x \neq y$ 时 $x + 1 \neq y + 1$ 。

现在我们给出保证函数为一对一的某些条件。

定义 6 定义域和伴域都是实数集子集的函数 f 称为递增的，如果对 f 的定义域中的 x 和 y ，只要 $x < y$ 就有 $f(x) \leq f(y)$ (若对于 $x < y$ ，恒有 $f(x) < f(y)$ ，则称函数 f 为严格递增的)。类似地， f 是递减的，如果对 f 的定义域中的 x 和 y ，只要 $x < y$ 就有 $f(x) \geq f(y)$ (若对于 $x < y$ ，恒有 $f(x) > f(y)$ ，称函数 f 为严格递减的) (定义中严格一词意味着严格不等式)。

注意 如果 $\forall x \forall y (x < y \rightarrow f(x) \leq f(y))$ ，则函数 f 是递增的；如果 $\forall x \forall y (x < y \rightarrow f(x) < f(y))$ ，则函数 f 是严格递增的。如果 $\forall x \forall y (x < y \rightarrow f(x) \geq f(y))$ ，则函数 f 就是递减的；如果 $\forall x \forall y (x < y \rightarrow f(x) > f(y))$ ，则函数 f 就是严格递减的。这里论域均为函数 f 的定义域。

从上述定义可知，只要函数是严格递增的或者严格递减的，它必定是一对一的。但是，如果一个函数不是严格意义上的递增或递减，就不必然一对一了。

有些函数的值域和伴域相等，也就是说，伴域中的每个成员都是定义域中某个元素的像。具有这一性质的函数称为映上函数。

定义 7 从 A 到 B 的函数 f 称为映上的或满射的，当且仅当对每个 $b \in B$ ，有元素 $a \in A$ 使得 $f(a) = b$ 。如果函数 f 是映上的，就说它是满射函数。

注意 如果 $\forall y \exists x (f(x) = y)$ ，函数 f 就是映上的，其中 x 的论域是函数的定义域， y 的论域是函数的伴域。

我们现在举几个映上函数和非映上函数的例子。

例 11 令 f 为从 $\{a, b, c, d\}$ 到 $\{1, 2, 3\}$ 的函数，其定义为 $f(a) = 3$ ， $f(b) = 2$ ， $f(c) = 1$ 及

$f(d)=3$. f 是映上函数吗?

解 由于伴域中所有 3 个元素均为定义域中元素的像, f 是映上的. 图 2-11 说明了这一点. 注意, 若伴域是 $\{1, 2, 3, 4\}$, f 就不是映上的. ■

例 12 从整数集到整数集的函数 $f(x)=x^2$ 是映上的吗?

解 f 不是映上的, 因此, 比如说没有 x 使 $x^2=-1$. ■

例 13 从整数集到整数集的函数 $f(x)=x+1$ 是映上的吗?

解 这个函数是映上的, 因为对每个整数 y , 都有一个整数 x 使 $f(x)=y$. 为看出这一点, 只要注意 $f(x)=y$ 的充分必要条件是 $x+1=y$, 而这只要令 $x=y-1$ 就成立.

定义 8 若函数 f 既是一对一的, 又是映上的, 就说它是一一对应或双射的.

下面的例 14 和例 15 阐述双射的概念.

例 14 令 f 为从 $\{a, b, c, d\}$ 到 $\{1, 2, 3, 4\}$ 的函数, 其定义为 $f(a)=4, f(b)=2, f(c)=1$ 及 $f(d)=3$. f 是双射吗?

解 函数 f 是一对一的和映上的. 它是一对一的, 是因为函数值都不同; 它是映上的, 是因为伴域中的所有 4 个元素, 均为定义域的元素像. 于是 f 是双射. ■

图 2-12 给出了 4 个函数: 其中第 1 个是一对一的, 但不是映上的; 第 2 个是映上的, 但不是对一的; 第 3 个, 既是一对一的, 又是映上的; 第 4 个既不是一对一的, 也不是映上的. 图 2-12 中的第 5 个对应关系不是函数, 因为它把一个元素传递给两个不同的元素.

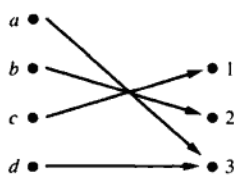


图 2-11 一个映上函数

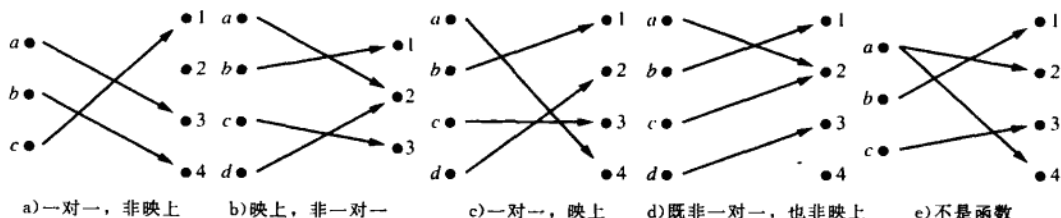


图 2-12 不同类型的对应关系的例子

假定 f 是从集合 A 到它自己的函数。如果 A 是有限的, 那么 f 是一对一的当且仅当它是映上的。当 A 为无限集时, 这一结论不一定成立(参看 2.4 节)。

例 15 令 A 为集合。 A 上的恒等函数是函数 $\iota_A: A \rightarrow A$, 其中

$$\iota_A(x) = x$$

对所有 $x \in A$ 。换言之, 恒等函数 ι_A 是这样的函数, 它赋给每个元素的是这个元素自身。函数 ι_A 是一对一的和映上的, 所以是双射. ■

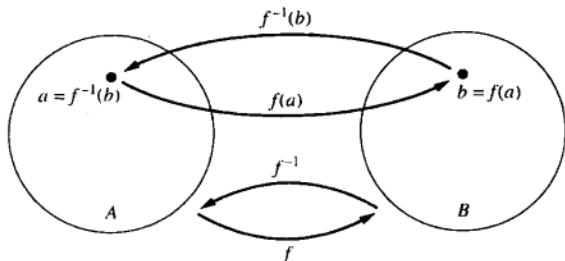
2.3.3 反函数和函数组合

现在考虑从集合 A 到集合 B 的一一对应 f 。由于 f 是映上的函数, B 的每个元素都是 A 中某元素的像。又由于 f 还是一对一的函数, B 的每个元素都是 A 中唯一的一个元素的像。于是我们可以定义一个从 B 到 A 的新函数, 把 f 给出的对应关系颠倒过来。

定义 9 令 f 为从集合 A 到集合 B 的一一对应, f 的反函数是这样的函数, 它指派给 B 中元素 b 的是 A 中使得 $f(a)=b$ 唯一元素 a 。 f 的反函数用 f^{-1} 表示, 于是在 $f(a)=b$ 时 $f^{-1}(b)=a$ 。

注意 切勿将函数 f^{-1} 与 $1/f$ 混淆, 后者表示定义域中每个元素 x 对应函数值为 $1/f(x)$ 的一个函数, 而且仅当 $f(x)$ 为非 0 实数时有意义, 二者是不同的。

图 2-13 给出了反函数概念的说明。

图 2-13 函数 f^{-1} 是函数 f 的反函数

如果函数 f 不是一一对应, 就无法定义反函数。若 f 不是一一对应, 那么它或者不是一对一的, 或者不是映上的; 如果 f 不是一对一的, 则伴域中的某元素 b 是定义域中多个元素的像。如果 f 不是映上的, 那么伴域中某个元素 b 不是定义域中任何元素的像, 即定义域中不存在元素 a 使 $f(a)=b$ 。总之, 如果 f 不是一一对应, 就不能为伴域中每个元素 b 都指派定义域中唯一的元素 a , 使 $f(a)=b$ (因为对某个 b 或者有多个这样的 a , 或者没有这样的 a)。

一一对应关系称为可逆的, 因为作为函数可以定义它的反函数。如果函数不是一一对应关系, 就说它是不可逆的, 因为这样的函数没有反函数。

例 16 令 f 为从 $\{a, b, c\}$ 到 $\{1, 2, 3\}$ 的函数, 使 $f(a)=2$, $f(b)=3$ 及 $f(c)=1$ 。 f 可逆吗? 如果可逆, 其反函数是什么?

解 f 是可逆的, 因为它是一个一对一的对应关系。其反函数 f^{-1} 颠倒 f 给出的对应关系, 所以 $f^{-1}(1)=c$, $f^{-1}(2)=a$ 而 $f^{-1}(3)=b$ 。

例 17 令 $f: \mathbf{Z} \rightarrow \mathbf{Z}$, 使得 $f(x)=x+1$ 。 f 可逆吗? 如果可逆, 其反函数是什么?

解 f 可逆, 因为前面已证明它是一一对应关系。要颠倒对应关系, 设 y 是 x 的像, 即 $y=x+1$ 。从而 $x=y-1$ 。即 $y-1$ 是 $\mathbf{Z}$ 的唯一元素, 在 f 之下与 y 对应, 于是 $f^{-1}(y)=y-1$ 。

例 18 令 f 是从 $\mathbf{R}$ 到 $\mathbf{R}$ 的函数, 使 $f(x)=x^2$ 。 f 可逆吗?

解 由于 $f(-2)=f(2)=4$, f 不是一一对应的, 要想定义反函数, 就得为 4 指派两个元素。因此 f 是不可逆的。

有时, 可以限制函数的定义域或者伴域, 或者两者都设限, 得到一个可逆的函数, 正如图 19。

例 19 如果我们限制例 18 中的函数 $f(x)=x^2$ 的定义域是非负数, 那么 f 就是可逆的。

解 因为被限制在非负数范围内, 所以 f 是一一对应的。比如说, 如果 $f(x)=f(y)$, 那么 $x^2=y^2$ 。所以 $x^2-y^2=(x+y)(x-y)=0$ 。也就是说, $x+y=0$ 或者 $x-y=0$, 则 $x=y$ 或者 $x=-y$ 。因为 x 和 y 都是非负的, 那就只可能 $x=y$ 。因此, 这个函数是一一对应的。同时, 当伴域是非负数时, 对于 $f(x)=x^2$ 是映上的。因为每一个非负数只有一个正平方根。当 y 是非负数时, 刚好 $x=\sqrt{y}$, 也就是 $x^2=y$ 。因此, 函数 $f(x)=x^2$ 在非负数上是一一对应和映上的, 它的反函数为 $f^{-1}(y)=\sqrt{y}$ 。

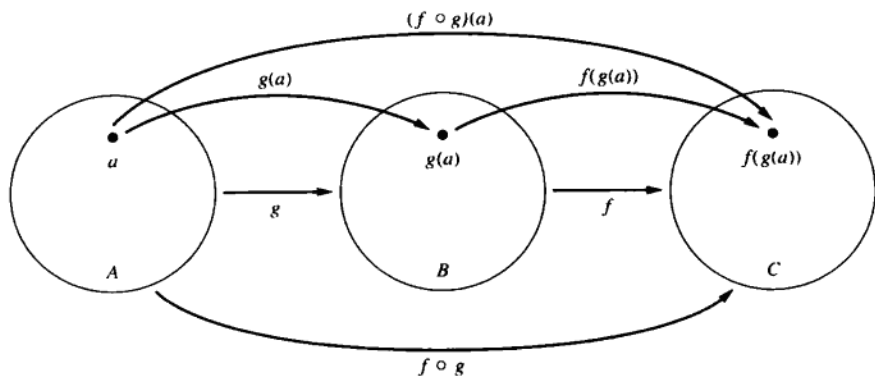
定义 10 令 g 为从集合 A 到集合 B 的函数, f 是从集合 B 到集合 C 的函数, 函数 f 和 g 的组合用 $f \circ g$ 表示, 定义为

$$(f \circ g)(a) = f(g(a))$$

换句话说, 函数 $f \circ g$ 指派给 A 的元素 a 的就是 f 指派给 $g(a)$ 的。注意, 如果 g 的值域不是 f 的定义域的子集, 就无法定义 $f \circ g$, 图 2-14 解释了函数组合的概念。

例 20 令 g 为从 $\{a, b, c\}$ 到它自己的函数, 使 $g(a)=b$, $g(b)=c$, 而 $g(c)=a$ 。令 f 为从 $\{a, b, c\}$ 到 $\{1, 2, 3\}$ 的函数, 使 $f(a)=3$, $f(b)=2$, 而 $f(c)=1$ 。 f 和 g 的组合是什么? g 和 f 的组合是什么?

解 $f \circ g$ 的定义是 $(f \circ g)(a) = f(g(a)) = f(b) = 2$, $(f \circ g)(b) = f(g(b)) = f(c) = 1$, 而 $(f \circ g)(c) = f(g(c)) = f(a) = 3$ 。

图 2-14 函数 f 和 g 的组合

注意 $g \circ f$ 是没有定义的，因为 f 的值域不是 g 的定义域的一部分。

例 21 令 f 和 g 为从整数集到整数集的函数，其定义为 $f(x) = 2x + 3$ 和 $g(x) = 3x + 2$ 。 f 和 g 的组合是什么？ g 和 f 的组合是什么？

解 $f \circ g$ 和 $g \circ f$ 均有定义，即

$$(f \circ g)(x) = f(g(x)) = f(3x + 2) = 2(3x + 2) + 3 = 6x + 7$$

及

$$(g \circ f)(x) = g(f(x)) = g(2x + 3) = 3(2x + 3) + 2 = 6x + 11$$

注意 尽管例 21 中对函数 f 和 g 而言 $f \circ g$ 和 $g \circ f$ 均有定义， $f \circ g$ 和 $g \circ f$ 并不相等。换言之，对函数组合而言交换律不成立。

在构造函数和它的反函数的组合时，不论以什么次序组合，得到的都是恒等函数。要看清这一点，假定 f 是从集合 A 到集合 B 的一一对应关系，那么存在反函数 f^{-1} ，而且 f^{-1} 是从 B 到 A 的一一对应关系。反函数把原函数的对应关系颠倒过来，所以当 $f(a) = b$ 时 $f^{-1}(b) = a$ ，当 $f^{-1}(b) = a$ 时， $f(a) = b$ 。因此，

$$(f^{-1} \circ f)(a) = f^{-1}(f(a)) = f^{-1}(b) = a$$

及

$$(f \circ f^{-1})(b) = f(f^{-1}(b)) = f(a) = b$$

从而 $f^{-1} \circ f = \iota_A$ ，而 $f \circ f^{-1} = \iota_B$ ，其中 ι_A 和 ι_B 分别是集合 A 和 B 上的恒等函数。这就是说 $(f^{-1})^{-1} = f$ 。

2.3.4 函数的图像

从 A 到 B 的每个函数都对应着 $A \times B$ 中的一个有序偶集合，这个有序偶集合称为该函数的图像，并且往往用图来表示以帮助理解函数的行为。

定义 11 令 f 为从集合 A 到集合 B 的函数，函数 f 的图像是序偶集合 $\{(a, b) \mid a \in A \text{ 且 } f(a) = b\}$ 。

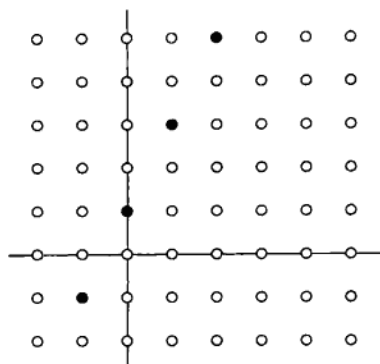
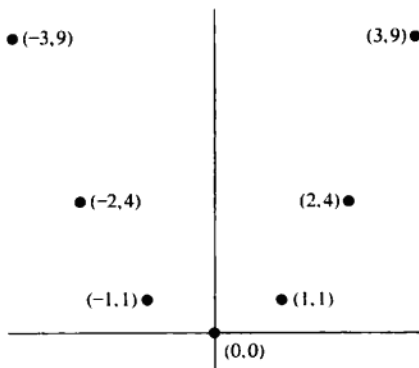
根据定义，从 A 到 B 的函数的图像是 $A \times B$ 的子集，这个子集包含的序偶中第二项等于 B 中由 f 指派给第一项的那个元素。

例 22 图示从整数集到整数集的函数 $f(n) = 2n + 1$ 的图像。

解 f 的图像是形为 $(n, 2n + 1)$ 的序偶的集合，其中 n 为整数。图 2-15 是其图像显示。

例 23 图示整数集到整数集的函数 $f(x) = x^2$ 。

解 f 的图像是形为 $(x, f(x)) = (x, x^2)$ 的序偶的集合，其中 x 为整数。图 2-16 是其图像显示。

图 2-15 从 $\mathbb{Z}$ 到 $\mathbb{Z}$ 的函数 $f(n)=2n+1$ 的图像图 2-16 从 $\mathbb{Z}$ 到 $\mathbb{Z}$ 的 $f(x)=x^2$ 的图像

2.3.5 几个重要的函数

下面介绍离散数学中两个重要的函数，即下取整函数和上取整函数。令 x 为实数。下取整函数把 x 下舍入到小于或等于 x 又最接近 x 的整数，而上取整函数则把 x 上舍入到大于或等于 x 又最接近 x 的整数。在统计对象个数时常使用这两个函数。在分析解一定规模的问题的计算机过程使用的步数时，这两个函数起着重要的作用。

定义 12 下取整函数指派给实数 x 的是小于或等于 x 的最大整数。下取整函数在 x 的值用 $\lfloor x \rfloor$ 表示。上取整函数指派给实数 x 的是大于或等于 x 的最小整数。上取整函数在 x 的值用 $\lceil x \rceil$ 表示。

注意 下取整函数也常称为最大整数函数，这时往往用 $[x]$ 表示。

例 24 下面是下取整函数和上取整函数的若干值：

$$\begin{aligned} \lfloor 1/2 \rfloor &= 0, & \lceil 1/2 \rceil &= 1, & \lfloor -1/2 \rfloor &= -1, & \lceil -1/2 \rceil &= 0, \\ \lfloor 3.1 \rfloor &= 3, & \lceil 3.1 \rceil &= 4, & \lfloor 7 \rfloor &= 7, & \lceil 7 \rceil &= 7 \end{aligned}$$

图 2-17 显示的是下取整函数和上取整函数的图像。在图 2-17a 中，显示下取整函数 $\lfloor x \rfloor$ 。这个函数在 $[n, n+1)$ 区间取同样的值 n ，然后，当 $x=n+1$ 时，取值跳到 $n+1$ 。在图 2-17b 中，显示上取整函数 $\lceil x \rceil$ 。这个函数在 $(n, n+1]$ 区间取同样的值 $n+1$ ，然后，当 x 取略大于 $n+1$ 时，取值跳到 $n+2$ 。

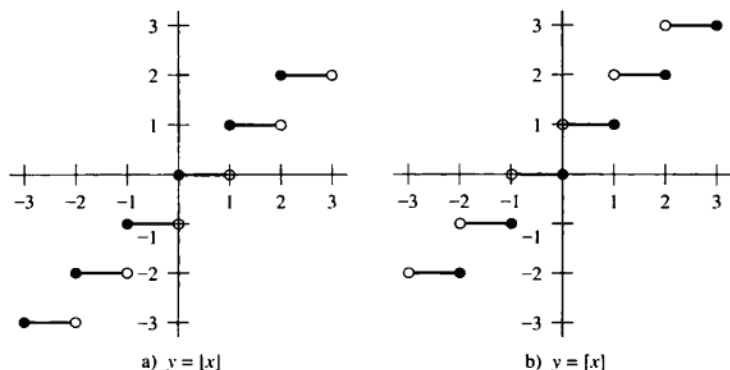


图 2-17 a) 下取整函数图像；b) 上取整函数图像

下取整函数和上取整函数有广泛的应用，包括用于数据存储和数据传输。考虑例 25 和例 26，这是研究数据库和数据通信问题时要完成的典型的基本计算。

例 25 存在计算机磁盘上的数据或数据网络上传输的数据通常表示为字节串。每个字节由 8 个字位组成, 要表示 100 字位的数据需要多少字节?

解 要决定需要的字节数, 就要找出最小的整数, 它至少要与 100 除以 8 的商一样大, 8 是每个字节的字位数。于是, 需要的字节数是 $\lceil 100/8 \rceil = \lceil 12.5 \rceil = 13$ 。 ■

例 26 异步传输模式(ATM)(用于骨干网络上的通信协议)下, 数据按 53 个字节分组, 每组称为一个信元。以速率每秒 500 千字位传输数据的连接上一分钟能传输多少个 ATM 信元?

解 一分钟内这一连接能传输 $500\,000 \times 60 = 30\,000\,000$ 字位。每个 ATM 信元长度是 53 字节, 也就是 $53 \times 8 = 424$ 字位。要计算一分钟能传输多少信元, 需计算不超过 $30\,000\,000$ 除以 424 的最大整数。于是在每秒 500 千字位的连接上一分钟能传输的 ATM 信元数是

$$\lfloor 30\,000\,000/424 \rfloor = 70\,754$$

以 x 代表实数的表 2-3 给出了下取整函数和上取整函数的若干简单但重要的性质。由于这两个函数在离散数学中出现得十分频繁, 看一看表中的恒等式是有益的。表中的每条性质都可以用下取整函数和上取整函数的定义来建立。性质 (1a), (1b), (1c) 和 (1d) 可以直接由定义得出。例如, (1a) 说的是 $\lfloor x \rfloor = n$ 当且仅当 n 小于或等于 x 而 $n+1$ 大于 x 。这恰恰就是 n 为不超过 x 的最大整数的含义, 也就是 $\lfloor x \rfloor = n$ 的定义。性质 (1b)、(1c) 和 (1d) 可以类似地建立起来。我们使用直接证明法来证明 (4a) 是成立的。

表 2-3 上取整函数和下取整函数的有用性质(n 为整数)

(1a) $\lfloor x \rfloor = n$ 当且仅当 $n \leq x < n+1$
(1b) $\lceil x \rceil = n$ 当且仅当 $n-1 < x \leq n$
(1c) $\lfloor x \rfloor = n$ 当且仅当 $x-1 < n \leq x$
(1d) $\lceil x \rceil = n$ 当且仅当 $x \leq n < x+1$
(2) $x-1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x+1$
(3a) $\lfloor -x \rfloor = -\lceil x \rceil$
(3b) $\lceil -x \rceil = -\lfloor x \rfloor$
(4a) $\lfloor x+n \rfloor = \lfloor x \rfloor + n$
(4b) $\lceil x+n \rceil = \lceil x \rceil + n$

证 假定 $\lfloor x \rfloor = m$, 其中 m 为整数。由 (1a) 知, $m \leq x < m+1$ 。在这一不等式中加上 n 得 $m+n \leq x+n < m+n+1$ 。再次利用 (1a), 可知 $\lfloor x+n \rfloor = m+n = \lfloor x \rfloor + n$, 这正是我们要证明的。其他性质的证明推迟到练习中去做。 □

除了表 2-3 列出的性质, 上取整函数和下取整函数还有许多其他有用的性质。也有许多关于这些函数的语句看似正确, 实则不然。我们将在例 27 和例 28 中考虑与上取整函数和下取整函数有关的语句。

在考虑下取整函数相关的语句时, 一个有用的方法是令 $x = n + \epsilon$, 其中 $n = \lfloor x \rfloor$ 是一个整数, ϵ 是 x 的分数部分, 满足不等式 $0 \leq \epsilon < 1$ 。类似地, 考虑上取整函数时, 通常写 $x = n - \epsilon$, 其中 $n = \lceil x \rceil$ 且 $0 \leq \epsilon < 1$ 。

例 27 证明: 若 x 是一个实数, 则 $\lfloor 2x \rfloor = \lfloor x \rfloor + \left\lfloor x + \frac{1}{2} \right\rfloor$ 。

解 要证明这个语句, 令 $x = n + \epsilon$, 其中 n 是正整数且 $0 \leq \epsilon < 1$ 。依据 ϵ 是否小于 $\frac{1}{2}$, 有两种情况要考虑。(选择这两种情况的原因见下面的证明。)

首先考虑 $0 \leq \epsilon < \frac{1}{2}$ 的情况。此时, $2x = 2n + 2\epsilon$ 且 $\lfloor 2x \rfloor = 2n$, 因为 $0 \leq 2\epsilon < 1$ 。类似地, $x + \frac{1}{2} = n + \left(\frac{1}{2} + \epsilon\right)$, 使得 $\left\lfloor x + \frac{1}{2} \right\rfloor = n$, 因为 $0 < \frac{1}{2} + \epsilon < 1$ 。因此

$$\lfloor 2x \rfloor = 2n \text{ 且 } \lfloor x \rfloor + \left\lfloor x + \frac{1}{2} \right\rfloor = n + n = 2n$$

接下来, 考虑 $\frac{1}{2} \leq \epsilon < 1$ 的情况。此时, $2x = 2n + 2\epsilon = (2n+1) + (2\epsilon-1)$ 。由于 $0 \leq 2\epsilon-1 < 1$,

可得 $\lfloor 2x \rfloor = 2n+1$ 。因为 $\left\lfloor x + \frac{1}{2} \right\rfloor = \left\lfloor n + \left(\frac{1}{2} + \epsilon \right) \right\rfloor = \left\lfloor n+1 + \left(\epsilon - \frac{1}{2} \right) \right\rfloor$ 且 $0 \leq \epsilon - \frac{1}{2} < 1$, 可得 $\left\lfloor x + \frac{1}{2} \right\rfloor = n+1$ 。因此, $\lfloor 2x \rfloor = 2n+1$ 且 $\lfloor x \rfloor + \left\lfloor x + \frac{1}{2} \right\rfloor = n + (n+1) = 2n+1$ 。证毕。 ■

例 28 对全体实数 x 和 y , 证明或否定 $\lceil x+y \rceil = \lceil x \rceil + \lceil y \rceil$ 。

解 尽管这个语句看似合理, 但它其实是假的。一个反例就是令 $x = \frac{1}{2}$ 且 $y = \frac{1}{2}$, 此时

$$\lceil x+y \rceil = \left\lceil \frac{1}{2} + \frac{1}{2} \right\rceil = \lceil 1 \rceil = 1, \text{ 但 } \lceil x \rceil + \lceil y \rceil = \left\lceil \frac{1}{2} \right\rceil + \left\lceil \frac{1}{2} \right\rceil = 1+1=2. \quad \blacksquare$$

有几类函数是课本一直要用到的, 其中包括多项式、对数和指数函数。本书中用记号 $\log x$ 表示 x 以 2 为基的对数, 因为 2 是我们将经常使用的对数基数。我们用 $\log_b x$ 表示以 b 为基的对数, 其中 b 是大于 1 的任意实数。用 $\ln x$ 表示自然对数。

另一个我们将在本书中使用的函数是阶乘函数 $f: \mathbf{N} \rightarrow \mathbf{Z}^+$, 表示为 $f(n) = n!$ 。 $f(n) = n!$ 的值是前几个正整数的积, 因此 $f(n) = 1 \cdot 2 \cdots (n-1) \cdot n$ [$f(0) = 0! = 1$]。

例 29 我们有 $f(1) = 1! = 1$, $f(2) = 2! = 1 \cdot 2 = 2$, $f(6) = 6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$, $f(20) = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10 \cdot 11 \cdot 12 \cdot 13 \cdot 14 \cdot 15 \cdot 16 \cdot 17 \cdot 18 \cdot 19 \cdot 20 = 2\,432\,902\,008\,176\,640\,000$ 。 ■

例 29 表明阶乘函数随着 n 的增加而迅速递增。通过斯特林公式, 阶乘函数的递增更加清楚, 由高等数学可知, $n! \sim \sqrt{2\pi n} (n/e)^n$ 。我们用 $f(n) \sim g(n)$ 这样的表示法, 意思是随着 n 的无限递增, $f(n)/g(n)$ 趋近于 1 (即 $\lim_{n \rightarrow \infty} f(n)/g(n) = 1$)。符号 $\sim$ 读作“近似于”。斯特林公式以 18 世纪的苏格兰数学家詹姆斯·斯特林命名。

练习

1. 为什么下列几问中的 f 不是从 $\mathbf{R}$ 到 $\mathbf{R}$ 的函数?

a) $f(x) = 1/x$

b) $f(x) = \sqrt{x}$

c) $f(x) = \pm \sqrt{(x^2+1)}$

2. 判断 f 是否为从所有位串的集合到整数集合的函数:

a) $f(S)$ 是 S 中某个 0 字位的位置。

b) $f(S)$ 是 S 中 1 字位的个数。

c) $f(S)$ 是最小整数 i 使 S 中的第 i 字位为 1, 当 S 是不含字位的空串时 $f(S) = 0$ 。

3. 求下列函数的定义域和值域。(注意: 为求得函数定义域, 只需确定被该函数指派了值的元素集合。)

a) 函数在每个位串的值是串中 1 字位个数与 0 字位个数之差。

b) 函数在每个位串的值是串中 0 字位个数的 2 倍。

c) 函数在每个位串的值是把串分成字节 (8 个字位为一字节) 时不够一个字节的字位数。

d) 函数在每个正整数的值是不超过此整数的最大完全平方。

4. 求下列函数的定义域和值域。

a) 函数在每对正整数的值是这两个整数中的最大的整数。

b) 函数在每个正整数的值是 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 中未在整数数位中出现的数。

c) 函数在一个位串的值是串中块 11 出现的次数。

d) 函数在一个位串的值是串中第一个 1 的数值位置, 若位串为全 0, 就赋值 0。

5. 求下列各值:

a) $\lceil 3/4 \rceil$

b) $\lfloor 7/8 \rfloor$

c) $\lceil -3/4 \rceil$

d) $\lfloor -7/8 \rfloor$

e) $\lceil 3 \rceil$

f) $\lfloor -1 \rfloor$

g) $\lfloor 1/2 + \lceil 3/2 \rceil \rfloor$

h) $\lfloor 1/2 \cdot \lfloor 5/2 \rfloor \rfloor$

6. 判断在下列情况下函数 $f: \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}$ 是否是映上的?

a) $f(m, n) = m+n$

b) $f(m, n) = m^2 + n^2$

c) $f(m, n) = m$

d) $f(m, n) = |n|$

e) $f(m, n) = m-n$

7. 给出从整数集到正整数集的函数的明确公式, 使得:

- a) 一对一但非映上。 b) 映上但非一对一。
c) 既映上又一对一。 d) 既不映上又非一对一。

8. 判断下列各函数是否是从 $\mathbf{R}$ 到 $\mathbf{R}$ 的双射。

- a) $f(x) = 2x + 1$ b) $f(x) = x^2 + 1$
c) $f(x) = x^3$ d) $f(x) = (x^2 + 1)/(x^2 + 2)$

9. 令 $f: \mathbf{R} \rightarrow \mathbf{R}$ 且 $f(x) > 0$ 。证明: $f(x)$ 严格递减当且仅当函数 $g(x) = 1/f(x)$ 严格递增。

10. 试给出一个定义域与伴域均为实数集的递减而非一对一的函数实例。

11. 证明: 从实数集到非负实数集的函数 $f(x) = |x|$ 不是可逆的, 而若将其定义域限制到非负实数集, 则所得函数为可逆的。

12. 令 $f(x) = \lfloor x^2/3 \rfloor$ 。求 $f(S)$, 如果

- a) $S = \{-2, -1, 0, 1, 2, 3\}$ b) $S = \{0, 1, 2, 3, 4, 5\}$
c) $S = \{1, 5, 7, 11\}$ d) $S = \{2, 6, 10, 14\}$

13. 假定 g 是从 A 到 B 的函数, f 是从 B 到 C 的函数。

- a) 证明: 如果 f 和 g 均为一对一函数, 那么 $f \circ g$ 也是一对一函数。
b) 证明: 如果 f 和 g 均为到映上函数, 那么 $f \circ g$ 也是映上函数。

*14. 如果 f 和 $f \circ g$ 都是映上的, 能否得出结论 g 也是映上的? 说明理由。

15. 求证: 从 $\mathbf{R}$ 到 $\mathbf{R}$ 的函数 $f(x) = ax + b$ 是可逆的, 其中 a 和 b 为常数且 $a \neq 0$ 。找出 f 的反函数。

令 f 为从集合 A 到集合 B 的函数, S 为 B 的子集。 S 的逆像定义为 A 的子集, 该子集包含 S 的所有元素的原像。我们用 $f^{-1}(S)$ 表示 S 的逆像, 于是 $f^{-1}(S) = \{a \in A \mid f(a) \in S\}$ 。(警告: 符号 f^{-1} 有两种不同的使用方式。不要将这里引入的符号与可逆函数 f 在 y 处的逆 $f^{-1}(y)$ 混淆。还要注意 $f^{-1}(S)$ 表示集合 S 的逆像, 对所有函数 f 都有意义, 而不仅仅是可逆函数 f 。)

16. 令 $g(x) = \lfloor x \rfloor$ 。求

- a) $g^{-1}(\{0\})$ b) $g^{-1}(\{-1, 0, 1\})$ c) $g^{-1}(\{x \mid 0 < x < 1\})$

17. 令 f 为从 A 到 B 的函数, S 为 B 的子集。求证 $f^{-1}(\bar{S}) = \overline{f^{-1}(S)}$ 。

18. 证明: $\lceil x - 1/2 \rceil$ 是最接近 x 的整数, 除非 x 恰为两个(相邻)整数的中间数, 此时, 它为这两个整数中较小的一个。

19. 证明: 对实数 x , $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$ 。

20. 若 x 为实数, n 为整数, 求证

- a) $x < n$ 当且仅当 $\lfloor x \rfloor < n$ 。 b) $n < x$ 当且仅当 $n < \lceil x \rceil$ 。

21. 证明: 若 n 为整数, 则当 n 为偶数时 $\lfloor n/2 \rfloor = n/2$, n 为奇数时 $\lfloor n/2 \rfloor = (n-1)/2$ 。

22. 有些计算器上有 INT 函数, 当 x 为非负实数时 $\text{INT}(x) = \lfloor x \rfloor$, 当 x 为负实数时 $\text{INT}(x) = \lceil x \rceil$ 。求证这一函数 INT 满足等式 $\text{INT}(-x) = -\text{INT}(x)$ 。

23. 令 a 和 b 的实数, 且 $a < b$, 用下取整函数及/或上取整函数表示使 $a < n < b$ 的整数 n 的数目。

24. 含 n 个字节的数据需要用几个字节编码? 其中 n 为

- a) 7 b) 17 c) 1001 d) 28800

25. 数据在某以太网上以 1500 个 8 位字节为信息块传输。下面的数据量在这个以太网上各需要多少个信息块传输(注意字节就是 8 位字节)?

- a) 150 千字节的数据。 b) 384 千字节的数据。
c) 1.544 兆字节的数据。 d) 45.3 兆字节的数据。

26. 画出从 $\mathbf{R}$ 到 $\mathbf{R}$ 的函数 $f(x) = \lfloor 2x \rfloor$ 的图像。

27. 画出从 $\mathbf{R}$ 到 $\mathbf{R}$ 的函数 $f(x) = \lfloor x \rfloor + \lfloor x/2 \rfloor$ 的图像。

28. 画出下列各函数的图像:

- a) $f(x) = \lfloor x + 1/2 \rfloor$ b) $f(x) = \lfloor 2x + 1 \rfloor$ c) $f(x) = \lceil x/3 \rceil$
d) $f(x) = \lceil 1/x \rceil$ e) $f(x) = \lceil x - 2 \rceil + \lfloor x + 2 \rfloor$ f) $f(x) = \lfloor 2x \rfloor \lfloor x/2 \rfloor$
g) $f(x) = \lceil \lfloor x - 1/2 \rfloor + 1/2 \rceil$

29. 求 $f(x) = x^3 + 1$ 的反函数。

30. 令 S 为全集 U 的子集。 S 的特征函数 f_S 是从 U 到集合 $\{0, 1\}$ 的函数, 使得若 x 属于 S 则 $f_S(x)=1$, 若 x 不属于 S , 则 $f_S(x)=0$ 。令 A, B 为集合。求证

- a) $f_{A \cap B}(x) = f_A(x) \cdot f_B(x)$ b) $f_{A \cup B}(x) = f_A(x) + f_B(x) - f_A(x) \cdot f_B(x)$
 c) $f_{\bar{A}}(x) = 1 - f_A(x)$ d) $f_{A \oplus B}(x) = f_A(x) + f_B(x) - 2f_A(x)f_B(x)$

31. 证明或否定下列关于上取整函数和下取整函数的语句。

- a) 对任意实数 x , $\lceil \lfloor x \rfloor \rceil = \lfloor x \rfloor$ 。
 b) 只要 x 是实数, $\lfloor 2x \rfloor = 2 \lfloor x \rfloor$ 。
 c) 只要 x 和 y 是实数, $\lceil x \rceil + \lceil y \rceil - \lceil x+y \rceil = 0$ 或 1 。
 d) 对任意实数 x 和 y , $\lceil xy \rceil = \lceil x \rceil \lceil y \rceil$ 。
 e) 对任意实数 x , $\left\lceil \frac{x}{2} \right\rceil = \left\lfloor \frac{x+1}{2} \right\rfloor$ 。

32. 证明: 若 x 是正实数, 则

- a) $\lfloor \sqrt{\lfloor x \rfloor} \rfloor = \lfloor \sqrt{x} \rfloor$ b) $\lceil \sqrt{\lceil x \rceil} \rceil = \lceil \sqrt{x} \rceil$

用来计算函数值的程序也许不能给出该函数在其所有元素上的正确值。例如程序可能因无限循环或内存溢出而不能给出某个正确值。类似地, 在抽象数学中, 我们常常要讨论仅在实数子集上定义的函数, 如 $1/x$, $\sqrt{x}$ 和 $\arcsin(x)$ 。另外, 我们可能还会用到“最小孩子”函数(此函数对没有孩子的夫妇没有定义)和“日出时间”函数(此函数对北极圈没有定义)。

为研究这种情况, 我们用到部分函数的概念。从集合 A 到集合 B 的部分函数 f 是为 A 的一个子集的每个元素 a 指定 B 中唯一值 b 的指派。 A 的这个子集称为 f 的定义区域。集合 A 和 B 分别称为 f 的定义域和伴域。我们说 f 在 A 中不属于定义区域的元素上未定义。我们用 $f: A \rightarrow B$ 表示 f 是从 A 到 B 的部分函数(这个符号与表示函数的符号一样。符号的上下文将决定 f 究竟是部分函数还是全函数)。当 f 的定义区域等于 A 时, 就说 f 是全函数。

33. 对下列各个部分函数求它的定义域、伴域、定义区域及其未定义的值集合。另外判断它是否为全函数。

- a) $f: \mathbf{Z} \rightarrow \mathbf{R}, f(n) = 1/n$ b) $f: \mathbf{Z} \rightarrow \mathbf{Z}, f(n) = \lceil n/2 \rceil$
 c) $f: \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Q}, f(m, n) = m/n$ d) $f: \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}, f(m, n) = mn$
 e) $f: \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}, f(m, n) = m - n$, 如果 $m > n$

34. a) 证明: 若 S 为基数为 m 的集合, m 为正整数, 则在集合 S 与集合 $\{1, 2, \dots, m\}$ 之间存在一个一一对应。

b) 证明: 若 S, T 均为基数为 m 的集合, m 为正整数, 则在集合 S 与集合 T 之间存在一个一一对应。

*35. 证明函数 $f: \mathbf{Z}^+ \times \mathbf{Z}^+ \rightarrow \mathbf{Z}^+$ 是一对一且是映上的, 其中 $f(m, n) = (m+n-2)(m+n-1)/2 + m$ 。

2.4 序列与求和

2.4.1 引言

序列是一系列有序的元素。在离散数学中用许多方式使用序列。第4章将介绍用序列表示某些计数问题的解。在计算机科学中序列也是重要的数据结构。本节包括用来表示序列及序列项之和的记号的阐述。

当可以列举无穷集合的元素时, 这个集合称为可数的。本节末尾讨论可数和不可数集合, 证明有理数集合是可数的而实数集合是不可数的。

2.4.2 序列

序列是用来表示有序表的离散结构。例如 $1, 2, 3, 5, 8$ 是一个含有五项的序列, 而 $1, 3, 9, 27, 81, \dots, 30, \dots$ 是一个无穷序列。

定义1 序列是从整数集合的子集(通常是 $\{0, 1, 2, \dots\}$ 或 $\{1, 2, 3, \dots\}$)到集合 S 的函数。用记号 a_n 表示整数 n 的像。称 a_n 为序列的一个项。

用记号 $\{a_n\}$ 描述序列。(注意 a_n 表示序列 $\{a_n\}$ 的个别项。还要注意序列记号 $\{a_n\}$ 与集合记号有冲突, 但使用这个记号的上下文总是分得清楚何时在处理序列而何时在处理集合。还要注意字母 a

的选择是任意的。)

描述序列是通过按照下标升序来列举序列项。

例 1 考虑序列 $\{a_n\}$, 其中

$$a_n = 1/n$$

从 a_1 开始的这个序列项的表, 即

$$a_1, a_2, a_3, a_4, \dots$$

开头是:

$$1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots$$

定义 2 几何序列是如下形式的序列

$$a, ar, ar^2, \dots, ar^n, \dots$$

其中初项 a 和公比 r 都是实数。

注意 几何序列是指数函数 $f(x) = ar^x$ 的离散对应物。

例 2 满足 $b_n = (-1)^n$ 、 $c_n = 2 \cdot 5^n$ 和 $d_n = 6 \cdot (1/3)^n$ 的序列 $\{b_n\}$ 、 $\{c_n\}$ 和 $\{d_n\}$ 都是几何序列, 初项和公比分别等于 1 和 -1 , 2 和 5 以及 6 和 $1/3$ 。表 $b_0, b_1, b_2, b_3, b_4, \dots$ 的开头是:

$$1, -1, 1, -1, 1, \dots$$

表 $c_0, c_1, c_2, c_3, c_4, \dots$ 的开头是:

$$2, 10, 50, 250, 1250, \dots$$

表 $d_0, d_1, d_2, d_3, d_4, \dots$ 的开头是:

$$6, 2, 2/3, 2/9, 2/27, \dots$$

定义 3 等差序列是如下形式的序列

$$a, a+d, a+2d, \dots, a+nd, \dots$$

其中初项 a 和公差 d 都是实数。

注意 等差序列是线性函数 $f(x) = dx + a$ 的离散对应物。

例 3 满足 $s_n = -1 + 4n$ 和 $t_n = 7 - 3n$ 的序列 $\{s_n\}$ 和 $\{t_n\}$ 都是等差序列, 初项和公差分别等于 -1 和 4 以及 7 和 -3 。从 $n=0$ 这项开始的表 $s_0, s_1, s_2, s_3, \dots$ 的开头是:

$$-1, 3, 7, 11, \dots$$

表 $t_0, t_1, t_2, t_3, \dots$ 的开头是:

$$7, 4, 1, -2, \dots$$

在计算机科学中经常使用形如 $a_1, a_2, \dots, a_n$ 的序列。这些有穷序列也称为串。这个串也记作 $a_1 a_2 \dots a_n$ 。(回忆一下, 位串是比特位的有限序列, 在 1.1 节介绍)。串 S 的长度是这个串中的项数。空串是没有任何项的串, 记作 λ 。空串长度为 0。

例 4 串 $abcd$ 是长度为 4 的串。

2.4.3 特殊的整数序列

离散数学中的常见问题是找出构造序列项的公式或一般规则。有时候只知道解决问题的序列的一部分项, 目标是确定序列。尽管序列的初始项不能确定整个序列(毕竟存在无穷多个从任何有穷初始项集合开始的序列), 但知道前几项仍有助于形成关于序列本身的合理猜想。一旦形成猜想, 就可以尝试验证找到了正确序列。

当试图从初始项推导序列项的可能公式或规则时, 尝试一下找出这些项中的模式, 再观察一下能否确定如何从前项产生后项。有许多问题可问, 但某些较有用的问题是:

- 是否有相同值连续出现?
- 是否给前项加上常量或与序列中位置有关的量就得出后项?
- 是否给前项乘以特定量就得出后项?

- 是否按照某种方式组合前项就得出后项?
- 是否在各项之间存在循环?

例 5 求具有下列前 5 项的序列公式:

(a) $1, 1/2, 1/4, 1/8, 1/16$ (b) $1, 3, 5, 7, 9$ (c) $1, -1, 1, -1, 1$

解 (a) 可以看出分母都是 2 的幂。满足 $a_n = 1/2^n$ 的序列是候选解序列。这个候选序列是几何序列, 满足 $a=1$ 和 $r=1/2$ 。

(b) 注意每一项都是对前一项加 2 而得到的。满足 $a_n = 2n+1$ 的序列是候选解序列。这个候选序列是等差序列, 满足 $a=1$ 和 $d=2$ 。

(c) 各项轮流取值 1 和 -1。满足 $a_n = (-1)^n$ 的序列是候选解序列。这个候选序列是几何序列, 满足 $a=1$ 和 $r=-1$ 。

例 6 和例 7 说明如何通过分析序列来发现构造项。

例 6 如果前 10 项是 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 则如何产生序列项?

解 注意整数 1 出现 1 次, 整数 2 出现 2 次, 整数 3 出现 3 次, 整数 4 出现 4 次。产生这个序列的合理规则是整数 n 恰好出现 n 次, 所以序列的下 5 项可能都是 5, 随后 6 项可能都是 6, 等等。这种方式产生的序列是候选解序列。

例 7 如果前 10 项是 5, 11, 17, 23, 29, 35, 41, 47, 53, 59, 则如何产生序列项?

解 注意这个序列第一项之后的前 10 项每项都是给前项加 6 得到的(从相邻项之差为 6 就看出这一点)。因此从 5 开始总共加 $(n-1)$ 次 6 就产生第 n 项, 合理猜测第 n 项为 $5+6(n-1)=6n-1$ 。(这是个等差序列, 满足 $a=5$ 和 $d=6$ 。)

另一种求序列项生成规则的有用技术是对比所求的序列项与熟知的整数序列项, 比如等差序列项、几何序列项、完全平方、完全立方, 等等。表 2-4 显示若干应当记住的序列的前 10 项。

表 2-4 若干有序序列

第 n 项	前 10 项
n^2	1, 4, 9, 16, 25, 36, 49, 64, 81, 100, ...
n^3	1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, ...
n^4	1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10 000, ...
2^n	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, ...
3^n	3, 9, 27, 81, 243, 729, 2187, 6561, 19 683, 59 049, ...
$n!$	1, 2, 6, 24, 120, 720, 5040, 40 320, 362 880, 3 628 800, ...

例 8 如果序列 $\{a_n\}$ 的前 10 项为 1, 7, 25, 79, 241, 727, 2185, 6559, 19 681, 59 047, 试猜想 a_n 的简单公式。

解 要解决这个问题, 先查看相邻项的差, 但没有看出模式。当计算相邻项的比来查看每项是否为前项的倍数时, 发现这个比虽然不是常数却接近于 3。所以有理由怀疑与 3^n 有关的公式产生了这个序列的各项。比较这些项与序列 $\{3^n\}$ 的对应项, 注意到第 n 项要比对应的 3 的幂少 2。看到对于 $1 \leq n \leq 10$ 来说 $a_n = 3^n - 2$, 因而猜想对所有 n 来说, 这个公式成立。

贯穿本书, 可以看到整数序列广泛出现在离散数学中。已经遇到或即将遇到的序列包括: 素数序列、排序 n 个离散对象的方式数(第 3 章)、解决著名的 n 个盘子汉诺塔智力题需要的步数(第 4 章)以及在 n 个月后岛上的兔子数(第 4 章)等。

整数序列还出现在离散数学以外的十分广泛的专题领域中, 包括生物学、工程、化学、物理学以及智力题等。数学家内尔·斯朗(Neil Sloane)在 20 世纪 60 年代构造了超过 100 000 个五花八门妙趣横生的整数序列, 这些序列可以在 1995 年出版的《整数序列百科全书》(Online Encyclopedia of Integer Sequences) ([SIPI95]) 中找到。经过扩充的序列表出现在网络上, 定期地加入新序列。还有通过网络访问的程序, 用来从百科全书中查找给定初始项的解序列。

2.4.4 求和

下一步介绍求和记号。首先描述用来表达序列 $\{a_n\}$ 中

$$a_m, a_{m+1}, \dots, a_n$$

这些项之和的记号。用记号

$$\sum_{j=m}^n a_j, \quad \sum_{j=m}^n a_j \quad \text{或} \quad \sum_{1 \leq j \leq n} a_j$$

表示

$$a_m + a_{m+1} + \dots + a_n$$

此处变量 j 称为求和下标，字母 j 是任意选择作为变量的；换句话说，可以使用任何其他字母，比如 i 或 k 。或者用记号写成

$$\sum_{j=m}^n a_j = \sum_{i=m}^n a_i = \sum_{k=m}^n a_k$$

此处求和下标依次等于从下限 m 开始到上限 n 为止的所有整数。用大写希腊字母 Σ 表示求和。通常的算术定律也适用于求和。例如，当 a 和 b 均为实数时，有

$$\sum_{j=1}^n (ax_j + by_j) = a \sum_{j=1}^n x_j + b \sum_{j=1}^n y_j$$

这里， $x_1, x_2, \dots, x_n$ 及 $y_1, y_2, \dots, y_n$ 均为实数（此处我们不给出该恒等式的正式证明）。

下面给出求和记号的若干例子。

例 9 表示序列 $\{a_n\}$ 前 100 项之和，对 $n=1, 2, 3, \dots, a_n=1/n$ 。

解 求和下标下限为 1，上限为 100。把这个和写成

$$\sum_{j=1}^{100} \frac{1}{j}$$

例 10 $\sum_{j=1}^5 j^2$ 的值是什么？

解 我们有

$$\begin{aligned} \sum_{j=1}^5 j^2 &= 1^2 + 2^2 + 3^2 + 4^2 + 5^2 \\ &= 1 + 4 + 9 + 16 + 25 \\ &= 55 \end{aligned}$$

例 11 $\sum_{k=4}^8 (-1)^k$ 的值是什么？

解 我们有

$$\begin{aligned} \sum_{k=4}^8 (-1)^k &= (-1)^4 + (-1)^5 + (-1)^6 + (-1)^7 + (-1)^8 \\ &= 1 + (-1) + 1 + (-1) + 1 \\ &= 1 \end{aligned}$$

有时候平移一下和中的求和下标会有用处。通常这样做是当两个和需要相加而求和下标却不一致时。当平移求和下标时，重要的是适当地修改对应求和项。下面的例子解释这一点。

例 12 假定有和

$$\sum_{j=1}^5 j^2$$

但是希望求和下标在 0 和 4 之间（而不是在 1 和 5 之间）依次取值。为此令 $k=j-1$ 。于是新的求和下标从 0 到 4 依次取值，项 j^2 变成 $(k+1)^2$ 。因此

$$\sum_{j=1}^5 j^2 = \sum_{k=0}^4 (k+1)^2$$

容易验证两个和都是 $1+4+9+16+25=55$ 。

几何序列各项之和经常出现(这种和称为几何级数)。定理1给出几何序列各项之和的公式。

定理1 若 a 和 r 都是实数且 $r \neq 0$, 则

$$\sum_{j=0}^n ar^j = \begin{cases} \frac{ar^{n+1} - a}{r - 1} & \text{若 } r \neq 1 \\ (n+1)a & \text{若 } r = 1 \end{cases}$$

证 设

$$S = \sum_{j=0}^n ar^j$$

要计算 S , 先给等式两边同乘 r , 然后如下变换得出的和:

$$\begin{aligned} rS &= r \sum_{j=0}^n ar^j && \text{用求和公式代替 } S \\ &= \sum_{j=0}^n ar^{j+1} && \text{分配律} \\ &= \sum_{k=1}^{n+1} ar^k && \text{转换求和下标, 令 } k = j+1 \\ &= \left(\sum_{k=0}^n ar^k \right) + (ar^{n+1} - a) && \text{去除 } k = n+1 \text{ 项, 添加 } k = 0 \text{ 项} \\ &= S + (ar^{n+1} - a) && \text{用 } S \text{ 代替求和公式} \end{aligned}$$

从这些等式看出

$$rS = S + (ar^{n+1} - a)$$

解出 S 就证明了若 $r \neq 1$, 则

$$S = \frac{ar^{n+1} - a}{r - 1}$$

若 $r=1$, 则显然这个和等于 $(n+1)a$. □

例13 很多情况下要求双重求和(比如在计算机程序中嵌套循环的分析中)。双重求和的例子是

$$\sum_{i=1}^4 \sum_{j=1}^3 ij$$

要计算双重求和, 先展开内层求和, 再继续计算外层求和:

$$\begin{aligned} \sum_{i=1}^4 \sum_{j=1}^3 ij &= \sum_{i=1}^4 (i + 2i + 3i) = \sum_{i=1}^4 6i \\ &= 6 + 12 + 18 + 24 = 60 \end{aligned}$$

还用求和记号把函数的所有值相加, 或把指标集的各项加起来, 其中求和下标依次等于集合中的所有值。换句话说, 用

$$\sum_{s \in S} f(s)$$

表示对 S 中所有元素 s 的值 $f(s)$ 求和。

例14 $\sum_{s \in \{0, 2, 4\}} s$ 的值是什么?

解 由于 $\sum_{s \in \{0, 2, 4\}} s$ 表示对集合 $\{0, 2, 4\}$ 中所有元素 s 的值求和, 所以

$$\sum_{s \in \{0, 2, 4\}} s = 0 + 2 + 4 = 6$$

某些求和在离散数学中反复出现。掌握这种求和的一组公式会有用处, 所以表2-5为常见求和提供袖珍公式表。

定理1推导了表中的第一个公式。接下来三个公式给出了前 n 个正整数的求和、平方求和以及立方求和。可用许多不同方式推导这三个公式(例如, 参见本节末练习11)。表中最后两个公式与无穷级数有关, 很快就会讨论到。

表 2-5 若干有用的求和公式

和	闭 形 式	和	闭 形 式
$\sum_{k=0}^n ar^k (r \neq 0)$	$\frac{ar^{n+1} - a}{r - 1}, r \neq 1$	$\sum_{k=1}^n k^3$	$\frac{n^2(n+1)^2}{4}$
$\sum_{k=1}^n k$	$\frac{n(n+1)}{2}$	$\sum_{k=0}^{\infty} x^k, x < 1$	$\frac{1}{1-x}$
$\sum_{k=1}^n k^2$	$\frac{n(n+1)(2n+1)}{6}$	$\sum_{k=1}^{\infty} kx^{k-1}, x < 1$	$\frac{1}{(1-x)^2}$

例 15 解释了表 2-5 中的公式是如何使用的。

例 15 求 $\sum_{k=50}^{100} k^2$ 。

解 首先注意由于 $\sum_{k=1}^{100} k^2 = \sum_{k=1}^{49} k^2 + \sum_{k=50}^{100} k^2$, 所以有

$$\sum_{k=50}^{100} k^2 = \sum_{k=1}^{100} k^2 - \sum_{k=1}^{49} k^2$$

利用表 2-5 的公式 $\sum_{k=1}^n k^2 = n(n+1)(2n+1)/6$, 看出

$$\sum_{k=50}^{100} k^2 = \frac{100 \cdot 101 \cdot 201}{6} - \frac{49 \cdot 50 \cdot 99}{6} = 338\,350 - 40\,425 = 297\,925$$

若干无穷级数 尽管本书中大多数求和都是有穷求和, 但在离散数学某些部分中无穷级数却是重要的。通常在微积分课程中研究无穷级数, 甚至这些级数的定义也需要使用微积分, 但有时它们也会出现在离散数学中, 因为离散数学也可以处理无穷离散对象集。尤其是将来在离散数学的研究中, 我们将会发现例 16 和例 17 中无穷级数的闭形式非常有用。

例 16 (本例要求微积分知识) 设 x 是实数满足 $|x| < 1$ 。求 $\sum_{n=0}^{\infty} x^n$ 。

Example 解 根据定理 1, 令 $a=1$ 和 $r=x$ 就看出 $\sum_{n=0}^k x^n = \frac{x^{k+1} - 1}{x - 1}$ 。由于 $|x| < 1$, 所以当 k 趋于无穷时, x^{k+1} 趋于 0。所以

$$\sum_{n=0}^{\infty} x^n = \lim_{k \rightarrow \infty} \frac{x^{k+1} - 1}{x - 1} = \frac{-1}{x - 1} = \frac{1}{1 - x}$$

对已有公式进行微分或积分就产生新的求和公式。

例 17 (本例要求微积分知识) 对下列方程两边微分

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

根据例 16 就发现

$$\sum_{k=1}^{\infty} kx^{k-1} = \frac{1}{(1-x)^2}$$

(根据关于无穷级数的定理, 对 $|x| < 1$ 这个微分有效。)

2.4.5 基数

回忆一下, 2.1 节把有限集的基数定义成集合中的元素个数。换句话说, 有限集合的基数告诉我们何时两个有限集大小相同, 何时一个比另一个大。那么我们是否可以将这种思想推广到无穷集合呢? 回忆一下 2.3 节的练习 34: 任何两个相同大小的有限集之间必然存在一个一一对应。这个发现使得我们可将基数概念推广到所有集合, 包括有限集和无限集(详见定义 4)。

定义 4 集合 A 和集合 B 有相同的基数, 当且仅当存在从 A 到 B 的一一对应。

现在把无限集分为两组, 一组与自然数集合基数相同, 另一组基数不同。

定义 5 有限集或与自然数集基数相同的集合都称为可数的。不是可数的集合称为不可数的。

如果一个无穷集合 S 是可数的，我们用符号 $\aleph_0$ 来表示集合 S 的基数 ($\aleph$ 是希伯来语，希伯来字母表的第一个字母)。记 $|S| = \aleph_0$ ，且说 S 有基数“希伯来零”。

现在给出可数集合与不可数集合的例子。

例 18 证明：正奇数集合是可数集合。

解 要证明正奇数集合是可数的，就说明这个集合与正整数集合之间的一一对应。考虑从 $\mathbb{Z}^+$ 到正奇数集合的函数

$$f(n) = 2n - 1$$

证明 f 既是单的又是满的，即证明 f 是一一对应的。为了看出 f 是单的，假定 $f(n) = f(m)$ 。于是 $2n - 1 = 2m - 1$ ，所以 $n = m$ 。为了看出 f 是满的，假定 t 是正奇数。于是 t 比偶数 $2k$ 少 1，其中 k 是自然数。因此 $t = 2k - 1 = f(k)$ 。图 2-18 显示这个一一对应。

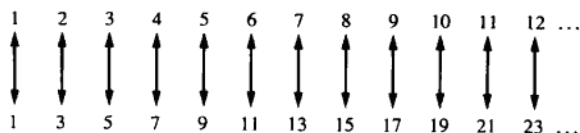


图 2-18 在 $\mathbb{Z}^+$ 和正奇数集合之间的一一对应

无限集是可数的，当且仅当可以把集合中的元素列成序列 (下标是正整数)。这是因为从正整数集到集合 S 的一一对应 f 可表示成序列 $a_1, a_2, \dots, a_n, \dots$ ，其中 $a_1 = f(1)$, $a_2 = f(2)$, $\dots$, $a_n = f(n)$, $\dots$ 例如，奇数集列成序列 $a_1, a_2, \dots, a_n, \dots$ ，其中 $a_n = 2n - 1$ 。

我们可以通过列举其所包含元素来证明所有整数的集合是可数的。

例 19 证明所有整数的集合是可数的。

证 我们可用以 0 开头，正负整数交替出现的序列来列出所有整数：0, 1, -1, 2, -2, $\dots$ 这样，可以确定正整数集与整数集之间的一一对应 (证明函数 $f(n) = n/2$ (n 为偶数)，及 $f(n) = -(n-1)/2$ (n 为奇数) 即为满足条件的一个函数，具体过程留给读者完成)。据此，所有整数的集合是可数的。

奇数集与整数集均为可数集合并不令人感到意外，但许多人对于有理数集也是可数集合的结果颇为惊讶，详见例 20 的证明。

例 20 证明：正有理数集合是可数的。

解 正有理数集合是可数的，这似乎令人惊讶，但下面说明如何把正有理数列成序列 $r_1, r_2, \dots, r_n, \dots$ 首先，注意每个正有理数都是两个正整数之商 p/q 。这样排列正有理数：在第 1 行列出分母 $q=1$ 的有理数，在第 2 行列出分母 $q=2$ 的有理数，等等，如图 2-19 所示。

把有理数列成序列的关键是：沿着图 2-19 所示路线，先列出满足 $p+q=2$ 的正有理数 p/q ，再列出满足 $p+q=3$ 的正有理数，然后列出满足 $p+q=4$ 的正有理数，等等。每当遇到已经列出过的数 p/q ，就不再次列出了。例如，当遇到 $2/2=1$ 时就不列出了，因为已经列出过 $1/1=1$ 。所构造的正有理数表的初始项是 1, $1/2$, 2, 3, $1/3$, $1/4$, $2/3$, $3/2$, 4, 5, 等等。由于所有有理数都只列出一次 (读者可以验证它)，所以已经证明了有理数集合是可数的。

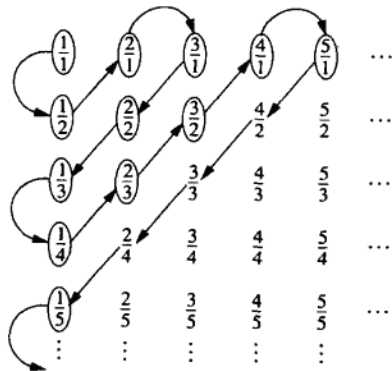


图 2-19 正有理数是可数的

我们已经看到有理数集也是可数集合，那么是否有候选的不可数集合呢？对此，第一个需关注的集合就是实数集。在例 21 中，我们使用乔治·康托尔于 1879 年引入的所谓康托尔对角化论证的重要证明方法来证明实数集合是不可数的。在数理逻辑和计算理论中大量

地使用这个证明方法。

例 21 证明: 实数集合是不可数集合。

解 要证明实数集合是不可数的, 就假定实数集合是可数的并得出矛盾。于是, 所有落在 0 和 1 之间的实数所成的子集合也是可数的(因为可数集合的所有子集合都是可数的)。在此假设下, 在 0 和 1 之间的实数按照某种顺序列出, 比如说 $r_1, r_2, r_3, \dots$, 设这些实数的十进制表示为

$$\begin{aligned} r_1 &= 0. d_{11} d_{12} d_{13} d_{14} \dots \\ r_2 &= 0. d_{21} d_{22} d_{23} d_{24} \dots \\ r_3 &= 0. d_{31} d_{32} d_{33} d_{34} \dots \\ r_4 &= 0. d_{41} d_{42} d_{43} d_{44} \dots \\ &\vdots \end{aligned}$$

其中 $d_{ij} \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ 。(例如, 如果 $r_1 = 0.237\,941\,02\dots$, 就有 $d_{11} = 2, d_{12} = 3, d_{13} = 7$, 等等。)于是, 构造新的实数具有十进制展开式 $r = 0. d_1 d_2 d_3 d_4 \dots$, 其中用下列规则确定十进制数字:

$$d_i = \begin{cases} 4 & \text{若 } d_{ii} \neq 4 \\ 5 & \text{若 } d_{ii} = 4 \end{cases}$$

(例如, 假定 $r_1 = 0.237\,941\,02\dots, r_2 = 0.445\,901\,38\dots, r_3 = 0.091\,187\,64\dots, r_4 = 0.805\,539\,00\dots$, 等等, 于是 $r = 0. d_1 d_2 d_3 d_4 \dots = 0.4544\dots$, 其中因为 $d_{11} \neq 4, d_1 = 4$; 因为 $d_{22} = 4, d_2 = 5$; 因为 $d_{33} \neq 4, d_3 = 4$; 因为 $d_{44} \neq 4, d_4 = 4$; 等等。)

每个实数都有唯一的十进制展开式(展开式结尾全部由数字 9 组成的可能性除外)。于是, 实数 r 不等于 $r_1, r_2, \dots$ 中的任何一个, 因为对每个 i 来说, r 的十进制展开式与 r_i 的十进制展开式在小数点右边第 i 位是不同的。

由于存在着不在列表中而在 0 和 1 之间的实数 r , 所以可以列出在 0 和 1 之间所有实数的假设就必定为假。因此, 不能列出在 0 和 1 之间的所有实数, 在 0 和 1 之间的实数集合是不可数的。任何含有不可数子集合的集合都是不可数的(参见本节末练习 19)。因此实数集合是不可数的。 ■

练习

- 求序列 $\{a_n\}$ 的下列各项, 其中 $a_n = 2 \cdot (-3)^n + 5^n$.
 - a_0
 - a_1
 - a_4
 - a_5
- 序列 $\{a_n\}$ 的项 a_0, a_1, a_2 和 a_3 是什么? 其中 a_n 等于
 - $2^n + 1$
 - $(n+1)^{n+1}$
 - $\lfloor n/2 \rfloor$
 - $\lfloor n/2 \rfloor + \lceil n/2 \rceil$
- 列出下列各序列的前 10 项。
 - 从 2 开始, 每个后续项都比前一项多 3 的序列。
 - 按升序把每个正整数列出 3 次的序列。
 - 按升序把每个正奇数列出 2 次的序列。
 - 第 n 项是 $n! - 2^n$ 的序列。
 - 从 3 开始, 每个后续项都是前一项的 2 倍的序列。
 - 前两项都是 1, 每个后续项都是前两项之和的序列。(这是著名的斐波那契序列, 后文有述。)
 - 第 n 项是数 n 的二进制展开式的位数的序列。
 - 第 n 项是第 n 个英文单词中包含的字母数的序列。
- 求至少 3 个不同的序列, 初始项都是 1, 2, 4, 用简单的公式或规则产生各项。
- 对于下列每个整数表, 给出简单的公式或规则, 以产生从给定的表开始的整数序列项。假定你给出的公式或规则是正确的, 据此写出相应序列的后续三项。
 - 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, ...
 - 1, 2, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, ...
 - 1, 0, 2, 0, 4, 0, 8, 0, 16, 0, ...
 - 3, 6, 12, 24, 48, 96, 192, ...
 - 15, 8, 1, -6, -13, -20, -27, ...
 - 3, 5, 8, 12, 17, 23, 30, 38, 47, ...
 - 2, 16, 54, 128, 250, 432, 686, ...
 - 2, 3, 7, 25, 121, 721, 5041, 40321, ...

**6. 证明: 如果 a_n 表示不是完全平方的第 n 个正整数, 则 $a_n = n + \{\sqrt{n}\}$, 其中 $\{x\}$ 表示最接近于实数 x 的整数。

7. 下列各和的值是什么？

$$a) \sum_{k=1}^5 (k+1)$$

$$b) \sum_{j=0}^4 (-2)^j$$

$$c) \sum_{i=1}^{10} 3$$

$$d) \sum_{j=0}^8 (2^{j+1} - 2^j)$$

8. 下列每个几何序列这些项之和的值是什么？

$$a) \sum_{j=0}^5 3 \cdot 2^j$$

$$b) \sum_{j=1}^8 2^j$$

$$c) \sum_{j=2}^8 (-3)^j$$

$$d) \sum_{j=0}^8 2 \cdot (-3)^j$$

9. 计算下列各双重和。

$$a) \sum_{j=1}^2 \sum_{i=1}^3 (i+j)$$

$$b) \sum_{i=0}^2 \sum_{j=0}^3 (2i+3j)$$

$$c) \sum_{i=1}^3 \sum_{j=0}^2 i$$

$$d) \sum_{i=0}^2 \sum_{j=1}^3 ij$$

10. 证明： $\sum_{j=1}^n (a_j - a_{j-1}) = a_n - a_0$ ，其中 $a_0, a_1, \dots, a_n$ 是实数序列。这种类型的和称为递进。

11. 从 $k=1$ 到 $k=n$ 对恒等式 $k^2 - (k-1)^2 = 2k-1$ 两边求和，并且利用练习 10，求：

$$a) \sum_{k=1}^n (2k-1) \text{ 的公式 (前 } n \text{ 个奇自然数之和)}$$

$$b) \sum_{k=1}^n k \text{ 的公式}$$

12. 用表 2-5 求 $\sum_{k=100}^{200} k$ 。

*13. 当 m 是正整数时，求 $\sum_{k=0}^m \lfloor \sqrt{k} \rfloor$ 的公式。

对于乘积也有一个特殊记号。 $a_m, a_{m+1}, \dots, a_n$ 的乘积表示为 $\prod_{j=m}^n a_j$ 。

14. 下列乘积的值是什么？

$$a) \prod_{i=0}^{10} i$$

$$b) \prod_{i=5}^8 i$$

$$c) \prod_{i=1}^{100} (-1)^i$$

$$d) \prod_{i=1}^{10} 2$$

回忆一下，阶乘函数在正整数 n 上的值(表示成 $n!$)是从 1 到 n 的正整数之积。另外规定 $0! = 1$ 。

15. 求 $\sum_{j=0}^i j!$ 。

16. 确定下列各集合是否可数或不可数。对可数的集合，说明在自然数集合和该集合之间的一一对应。

a) 负整数

b) 偶数

c) 0 和 $1/2$ 之间的实数

d) 7 的整倍数

17. 确定下列各集合是否可数或不可数。对可数的集合，说明在自然数集合和该集合之间的一一对应。

a) 不包含位 0 的全部位串

b) 表示成两整数之商时分母不小于 4 的全部正有理数

c) 十进制表示形式不包含 0 的实数

d) 十进制表示形式包含有限个 1 的实数

18. 如果 A 是不可数集合而 B 是可数集合，那么 $A-B$ 一定不可数吗？

19. 假设 A, B 是集合，证明：如果 A 是不可数集合并且 $A \subseteq B$ ，则 B 是不可数的。

20. 证明：若 A 和 B 是具有相同基数的两个集合， C 和 D 是具有相同基数的两个集合，则 $A \times C$ 与 $B \times D$ 具有相同基数。

**21. 证明：可数的多个可数集合的并集是可数的。

*22. 证明：所有有限位串的集合是可数的。

*23. 证明：用具体的程序设计语言编写的所有计算机程序的集合是可数的。[提示：可以认为用程序设计语言编写的计算机程序是有穷字母表上的符号串。]

*24. 如果存在着求函数值的计算机程序，就说这个函数是可计算的。用练习 23 证明：存在着不可计算的函数。

关键术语和结果

术语

集合：一组互不相同的对象。

公理：理论的一个基本假设。

悖论：逻辑上的不一致性。

集合的元素、成员：集合中的一个对象。

$\emptyset$ (空集)：没有成员的集合。

全集：包含考虑中的所有对象的集合。

文氏图：一个或多个集合的一种图形表示。

$S=T$ (集合相等)： S 和 T 有相同的元素。

$S \subseteq T$ (S 是 T 的子集)： S 的每个元素也是 T 的元素。

$S \subset T$ (S 是 T 的真子集)： S 是 T 的子集，且 $S \neq T$ 。

有限集：含 n 个元素的集合，其中 n 是非负整数。

无限集：不是有限集的集合。

$|S|$ (S 的基数)： S 中元素的个数。

$P(S)$ (S 的幂集合)： S 的所有子集的集合。

$A \cup B$ (A 和 B 的并集)：包含那些至少属于 A 和 B 之一的元素的集合。

$A \cap B$ (A 和 B 的交集)：包含那些既属于 A 又属于 B 的元素的集合。

$A - B$ (A 和 B 的差集)：包含那些只属于 A 而不属于 B 的元素的集合。

$\bar{A}$ (A 的补集)：全集中不属于 A 的元素的集合。

$A \oplus B$ (A 和 B 的对称差)：包含恰属于 A 和 B 之一的那些元素的集合。

成员表：显示集合中元素的成员关系的表格。

从 A 到 B 的函数：一种指派，为 A 中每个元素指派 B 中恰好一个元素。

f 的定义域：指集合 A ，若 f 是从 A 到 B 的函数。

f 的伴域：指集合 B ，若 f 是从 A 到 B 的函数。

b 是 f 之下 a 的像： $b=f(a)$ 。

a 是 f 之下 b 的原像： $f(a)=b$ 。

f 的值域： f 的像的集合。

映上函数，满射：从 A 到 B 的函数，使 B 的每个元素都是 A 中某元素的像。

一对一函数，内射：定义域中每个元素的像都不相同的函数。

一一对应，双射：既是一对一又是映上的函数。

f 的逆：(当 f 是双射时)颠倒由 f 给出的对应关系的函数。

$f \circ g$ (f 和 g 的组合)：为 x 指派 $f(g(x))$ 的函数。

$\lfloor x \rfloor$ (下取整函数)：不超过 x 的最大整数。

$\lceil x \rceil$ (上取整函数)：大于或等于 x 的最小整数。

序列：以整数集合的子集作为定义域的函数。

几何序列：形如 $a, ar, ar^2, \dots$ 的序列，其中 a 和 r 都是实数。

等差序列：形如 $a, a+d, a+2d, \dots$ 的序列，其中 a 和 d 都是实数。

串：有穷序列。

空串：长度为 0 的串。

$\sum_{i=1}^n a_i$ ：和 $a_1 + a_2 + \dots + a_n$ 。

$\prod_{i=1}^n a_i$ ：积 $a_1 a_2 \dots a_n$ 。

可数集：有穷集或可与正整数集一一对应的集合。

不可数集：不是可数的集合。

康托尔对角化论证：可用来证明实数集合是不可数的证明技术。

结果

2.2 节表 2-1 给出诸多集合恒等式。

2.4 节表 2-5 给出求和公式。

有理数集是可数的。
实数集是不可数的。

复习题

1. 解释一个集合是另一个集合子集的含义，如何证明一个集合是另一个集合的子集？
2. 空集表示什么？证明空集是任何集合的子集。
3. a) 定义 $|S|$ (集合 S 的基数)。
b) 给出 $|A \cup B|$ 的公式，其中 A, B 均为集合。
4. a) 定义集合 S 的幂集。
b) 何时空集包含在集合 S 的幂集中？
c) 包含 n 个元素的集合 S ，其幂集含有多少个元素？
5. a) 定义两个集合的并集、交集、差集以及对称差。
b) 正整数集与奇数集的并集、交集、差集及对称差分别是什么？
6. a) 解释两集合相等的含义。
b) 描述尽可能多的方法以证明两集合相等。
c) 用至少两种不同的方法证明 $A - (B \cap C)$ 与 $(A - B) \cup (A - C)$ 是相等的。
7. 解释逻辑等价与集合恒等之间的关系。
8. a) 定义一个函数的定义域、伴域及值域。
b) 令 $f(n) = n^2 + 1$ 为从整数集到整数集的一个函数，则其定义域、伴域、值域分别是什么？
9. a) 解释从正整数集到正整数集的一对一函数的含义。
b) 解释从正整数集到正整数集的映上函数的含义。
c) 列举一个从正整数集到正整数集的既一对一又映上的函数。
d) 列举一个从正整数集到正整数集的一对一而非映上的函数。
e) 列举一个从正整数集到正整数集的非一对一但映上的函数。
f) 列举一个从正整数集到正整数集的既非一对一又非映上的函数。
10. a) 定义一个函数的逆函数。
b) 何时一个函数存在逆函数？
c) 从整数集到整数集的函数 $f(n) = 10 - n$ 是否有逆函数？若有，其逆函数是什么？
11. a) 定义从实数集到整数集的下取整函数和上取整函数。
b) 对于哪些 x 而言， $\lfloor x \rfloor = \lceil x \rceil$ ？
12. 以 8, 14, 32, 86, 248 开头的序列之项推测一个表达式，并据此求出该序列后续三项。
13. 当 $r \neq 1$ 时，几何序列各项之和： $a + ar + \cdots + ar^n$ 是什么？
14. 证明：奇数集合是不可数的。
15. 给一个不可数集合的例子。

补充练习

1. 令 A 为包含字母 x 的英文单词集合， B 为包含字母 q 的英文单词集合。试表示下列 A, B 组合而得的集合：
 - a) 不包含字母 x 的英文单词集合
 - b) 不包含字母 x 和 q 的英文单词集合
 - c) 包含字母 x 而不包含字母 q 的英文单词集合
 - d) 不包含字母 x 或 q 的英文单词集合
 - e) 包含 x 或 q 但不同时包含二者的英文单词集合
2. 证明：若 A 是 B 的子集，则 A 的幂集是 B 的幂集的子集。
3. 假定 A 和 B 均为集合且满足 A 的幂集是 B 的幂集的子集，则 A 一定是 B 的子集吗？
4. 令 E 表示偶数集合， O 表示奇数集合， Z 一如往常表示整数集合。判断下列每个集合所代表的含义：
 - a) $E \cup O$
 - b) $E \cap O$
 - c) $Z - E$
 - d) $Z - O$
5. 证明：若 A 和 B 均为集合，则 $A - (A - B) = A \cap B$ 。
6. 令 A 和 B 为两个集合，证明： $A \subseteq B$ 当且仅当 $A \cap B = A$ 。
7. 令 A, B, C 为三个集合，证明： $(A - B) - C$ 不一定等于 $A - (B - C)$ 。

8. 假定 A 和 B 均为集合, 证明或否定: $(A-B)-C=(A-C)-B$.
9. 假定 A, B, C 和 D 均为集合, 证明或否定: $(A-B)-(C-D)=(A-C)-(B-D)$.
10. 证明: 若 A 和 B 均为有限集合, 则 $|A \cap B| \leq |A \cup B|$ 并判断何时等号成立.
11. 令 A 和 B 为有限全集 U 的两个子集, 按照递增顺序列举出下列各项:
- a) $A, |A \cup B|, |A \cap B|, |U|, |\emptyset|$
- b) $|A-B|, |A \oplus B|, |A|+|B|, |A \cup B|, |\emptyset|$
12. 令 A 和 B 为有限全集 U 的两个子集, 证明: $|\bar{A} \cap \bar{B}| = |U| - |A| - |B| + |A \cap B|$.
13. 令 f 和 g 分别为从 $\{1, 2, 3, 4\}$ 到 $\{a, b, c, d\}$ 和从 $\{a, b, c, d\}$ 到 $\{1, 2, 3, 4\}$ 的两个函数, 且满足 $f(1)=d, f(2)=c, f(3)=a, f(4)=b$ 和 $g(a)=2, g(b)=1, g(c)=3, g(d)=2$. 则
- a) f 是一一对应函数吗? g 呢?
- b) f 是映上函数吗? g 呢?
- c) f 或 g 是否有逆函数? 若有, 求出其逆函数.
14. 令 f 是从集合 A 到集合 B 的一一对应函数, S, T 均为 A 的子集. 证明: $f(S \cap T) = f(S) \cap f(T)$.
15. 举例证明: 对于非一一对应函数 f , 练习 14 中的等式不一定成立.

假定 f 是从 A 到 B 的一个函数. 我们按照如下规则定义从 $P(A)$ 到 $P(B)$ 的函数 S_f : 对于 A 的每个子集 X , 有 $S_f(X) = f(X)$. 类似地, 我们定义从 $P(B)$ 到 $P(A)$ 的函数 $S_{f^{-1}}$ (按照如下规则: 对于 B 的每个子集 Y , 有 $S_{f^{-1}}(Y) = f^{-1}(Y)$). 并且, 将使用 2.3 节的定义 4 以及集合逆像的定义.

- *16. a) 证明: 若 f 是从 A 到 B 的一一对应函数, 则 S_f 是从 $P(A)$ 到 $P(B)$ 的一一对应函数.
- b) 证明: 若 f 是从 A 到 B 的映上函数, 则 S_f 是从 $P(A)$ 到 $P(B)$ 的映上函数.
- c) 证明: 若 f 是从 A 到 B 的映上函数, 则 $S_{f^{-1}}$ 是从 $P(B)$ 到 $P(A)$ 的一一对应函数.
- d) 证明: 若 f 是从 A 到 B 的一一对应函数, 则 $S_{f^{-1}}$ 是从 $P(B)$ 到 $P(A)$ 的映上函数.
- e) 利用 a) 并通过 d) 得出结论: 若 f 是从 A 到 B 的一一对应, 则 S_f 是从 $P(A)$ 到 $P(B)$ 的一一对应且 $S_{f^{-1}}$ 是从 $P(B)$ 到 $P(A)$ 的一一对应.
17. 证明: 若 f 和 g 均为从 A 到 B 的函数且满足 $S_f = S_g$ (使用练习 16 前言中的定义), 则对于所有 $x \in A$ 必有 $f(x) = g(x)$.
18. 证明: 若 n 是整数, 则 $n = \lceil n/2 \rceil + \lfloor n/2 \rfloor$.
19. 对于哪些实数 x, y 而言, $\lfloor x+y \rfloor = \lfloor x \rfloor + \lfloor y \rfloor$ 成立?
20. 对于哪些实数 x, y 而言, $\lceil x+y \rceil = \lceil x \rceil + \lceil y \rceil$ 成立?
21. 对于哪些实数 x, y 而言, $\lceil x+y \rceil = \lceil x \rceil + \lfloor y \rfloor$ 成立?
22. 证明: 对于任一整数 n , $\lfloor n/2 \rfloor \lceil n/2 \rceil = \lfloor n^2/4 \rfloor$ 成立.
23. 证明: 若 m 是整数而 x 非整数, 则 $\lfloor x \rfloor + \lfloor m-x \rfloor = m-1$, 若 x 也是整数, 则等于 m .
24. 证明: 若 x 是实数, 则 $\lfloor \lfloor x/2 \rfloor / 2 \rfloor = \lfloor x/4 \rfloor$.
25. 证明: 若 n 是奇数, 则 $\lceil n^2/4 \rceil = (n^2+3)/4$.
26. 证明: 若 m, n 均为正整数, x 为实数, 则 $\left\lfloor \frac{\lfloor x \rfloor + n}{m} \right\rfloor = \left\lfloor \frac{x+n}{m} \right\rfloor$.
- *27. 证明: 若 m 是正整数, x 是实数, 则 $\lfloor mx \rfloor = \lfloor x \rfloor + \left\lfloor x + \frac{1}{m} \right\rfloor + \left\lfloor x + \frac{2}{m} \right\rfloor + \dots + \left\lfloor x + \frac{m-1}{m} \right\rfloor$.
- *28. 下面我们来定义乌拉姆数并规定: $u_1=1, u_2=2$. 深入一步, 在判断了小于 n 的整数是否是乌拉姆数之后, 若 n 可以被唯一地记作两个不同乌拉姆数之和, 则令 n 等于下一个乌拉姆数. 注意: $u_3=3, u_4=4, u_5=6, u_6=8$.
- a) 求前 20 个乌拉姆数.
- b) 证明存在无穷多个乌拉姆数.
29. 求 $\prod_{k=1}^{100} \frac{k+1}{k}$ 的值 (此处所使用的乘积符号在 2.4 节练习 14 前言有过定义).
- *30. 试给出一个规则以产生以 1, 3, 4, 8, 15, 27, 50, 92... 开头的序列项, 并据此求出该序列的后续四项.
- *31. 试给出一个规则以产生以 2, 3, 3, 5, 10, 13, 39, 43, 172, 177, 885, 891... 开头的序列项, 并据此求出该序列的后续四项.
- *32. 证明: 若 A, B 均为可数集合, 则 $A \times B$ 也为可数集合.

计算机题目

按给定的输入和输出写程序。

1. 给出含有 n 个元素的集合的两个子集 A, B , 利用位串求出 $\bar{A}, A \cup B, A \cap B, A - B, A \oplus B$ 。
2. 给出来自同一全集的两个多重集合 A, B , 试求 $A \cup B, A \cap B, A - B$ 以及 $A + B$ (参见 2.2 节练习 30 前言)。
3. 给出模糊集合 A, B , 试求 $\bar{A}, A \cup B$ 以及 $A \cap B$ (参见 2.2 节练习 31 前言)。
4. 给出一个从 $\{1, 2, \dots, n\}$ 到整数集合的函数 f , 判断 f 是否是一对一函数。
5. 给出一个从 $\{1, 2, \dots, n\}$ 到其自身的函数 f , 判断 f 是否是映上函数。
6. 给出一个从 $\{1, 2, \dots, n\}$ 到其自身的双射 f , 求 f^{-1} 。

计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 给出两个有限集, 试列出这两个集合笛卡儿积中的所有元素。
2. 给出一个有限集, 试列出其幂集中的所有元素。
3. 计算从集合 S 到集合 T 的一对一函数的个数, 其中 S, T 均为任意大小的有限集。你能否给出这样的函数总个数的一个公式? (我们将在第 3 章给出该公式。)
4. 计算从集合 S 到集合 T 的映上函数的个数, 其中 S, T 均为任意大小的有限集。你能否给出这样的函数总个数的一个公式? (我们将在第 4 章给出该公式。)
- *5. 对用以产生序列项的不同规则, 随机选取这些规则中的一个, 并使用该规则产生特定序列的程序进行汇总。利用提示序列后续项的交互式程序部分判断解答是否是预期的后续项。

写作题目

用课本以外的资料, 按下列要求写成短文。

1. 讨论一套公理化集合理论是如何避免罗素悖论而形成的?
2. 调查函数概念最早出现在哪, 试描述当时这种思想是如何被运用的。
3. 解释发现《整数序列百科全书》巨大作用的不同方式, 并描述几个非比寻常的序列以及它们是如何出现的。
4. 定义最新提出的 EKG 序列并描述它的某些性质以及有关它的一些未解问题。
5. 查阅超越数的定义。解释如何证明这样的数存在以及这些数是如何被构造出来的。考虑哪些典型数可以作为超越数, 而哪些典型数是否是超越数至今仍未可知。
6. 讨论无穷基数及连续统假设。

计 数

组合数学这一研究个体安排的学科，是离散数学的重要部分。早在17世纪就开始了这类课题的研究，当时在赌博游戏的研究中出现了组合问题。枚举，具有确定性质的个体的计数，是组合数学的一个重要的部分。我们必须对个体计数来求解许多不同类型的问题。例如，用计数确定算法的复杂性。计数也用于确定是否存在能够充分满足需求的电话号码或因特网地址。此外，计数技术也广泛用于计算事件的概率。

3.1节将要研究的基本计数规则可以求解各种各样的问题。例如，可以用这些规则来计数美国各种不同可能的电话号码，计算机系统中允许使用的密码，以及在比赛结束时赛跑运动员的不同的名次。另一个重要的组合工具是鸽巢原理，将放在3.2节研究。这个原理指出，当把物体放在盒子里时，若物体比盒子多，那么存在一个盒子包含着至少两个物体。例如，我们可以用这个原理证明在15个或者更多的学生中至少有3人出生在相同的星期几。

我们可以用集合中个体的有序或无序安排来描述许多计数问题。这些安排称作排列和组合，在许多的计数问题中都会用到它们。例如，在2000个学生参加的考试竞赛中最终将有100个获胜者被邀请赴宴。我们可以枚举将被邀请的100个学生的可能的组合，以及最终10名获奖者的产生方式。

组合数学的另一个问题涉及生成某个特定类型的所有排列。这在计算机模拟中通常是很重要的。我们将设计算法来生成各种类型的排列。

3.1 计数基础

3.1.1 引言

一个计算机系统的密码由6、7或8个字符组成。每个字符必须是数字或字母表中的字母。每个密码必须至少包含一位数字。问有多少个这样的密码？本节将介绍回答这个问题及各种各样的其他计数问题所需要的技术。

数学和计算机科学中存在着计数问题。例如，我们必须为成功的实验结果和所有可能的实验结果计数，以确定离散事件的概率。我们需要对某个算法用到的操作数计数，以便研究它的时间复杂性。

本节将介绍基本的计数方法。这些方法是几乎所有计数技术的基础。

3.1.2 基本的计数原则

 我们将提出两个基本的计数原则，乘积法则和求和法则。然后将说明怎样用它们来求解许多不同的计数问题。

当一个过程由独立的任务组成时使用乘积法则。

乘积法则 假定一个过程可以被分解成两个任务。如果完成第一个任务有 n_1 种方式，在第一个任务完成之后有 n_2 种方式完成第二个任务，那么完成这个过程有 $n_1 n_2$ 种方式。

 **例1至例10**显示怎样使用乘积法则。

例1 一个新建公司中只有两个雇员 Sanchez 和 Patel，公司租用了一个大楼的底层，共12个办公室。有多少种方法为这两个雇员分配办公室？

解 对这两个雇员分配办公室的过程是这样的：为 Sanchez 分配办公室，有12种方法，为 Patel 分配一个不同的办公室，有11种方法。根据乘法规则，为这两个雇员分配办公室共有 $12 \cdot 11 = 132$ 种方法。 ■

例2 用一个字母和一个不超过100的正整数给礼堂的座位编号。那么不同编号的座位最多有多少?

解 给一个座位编号的过程由两个任务组成,即从26个字母中先选择一个字母分配给这个座位,然后再从100个正整数中选择一个整数分配给它。乘积法则表明一个座位可以有 $26 \cdot 100 = 2600$ 种不同的编号方式。因此,不同编号的座位数至多是2600。 ■

例3 某个计算机中心有32台微机,每台微机有24个端口。问在这个中心里有多少个不同的单机端口?

解 选择一个端口的过程由两个任务组成。首先挑一台微机,然后在这台微机上挑一个端口。因为有32种方式选择微机,而不管选择了哪台微机,又有24种方式选择端口,由乘积法则存在 $32 \cdot 24 = 768$ 个端口。 ■

经常会用到推广的乘积法则。假定一个过程由执行任务 $T_1, T_2, \dots, T_m$ 来完成。如果在完成任务之后用 n_i 种方式来完成 T_i ,那么完成这个过程有 $n_1 \cdot n_2 \cdot \dots \cdot n_m$ 种方式。可以由两个任务的乘积法则通过数学归纳法证明推广的乘积法则。

例4 有多少个不同的7位二进制串?

解 每位有两种选择方式,可以是0或1。因此,乘积法则表明总共有 $2^7 = 128$ 个不同的7位二进制串。 ■

例5 如果每个车牌由3个字母后跟3个数字的序列构成(任何字母的序列都允许),那么有多少个不同的有效的车牌?

解 对3个字母中的每个字母有26种选择,对3个数字中的每个数字有10种选择。因此,由乘积法则总共有 $26 \cdot 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 = 17\,576\,000$ 个可能的车牌。 ■



例6 计数函数 从一个 n 元集到一个 m 元集存在多少个函数?

解 一个函数对于定义域中 m 个元素中的每个元素都要选择陪域中 n 个元素中的一个元素来对应。因此,由乘积法则存在 $n \cdot n \cdot \dots \cdot n = n^m$ 个从 m 元集到 n 元集的函数。例如,从一个3元集到一个5元集存在 5^3 个不同的函数。 ■

例7 计数一对一函数 从一个 m 元集到一个 n 元集存在多少个一对一函数?

解 首先注意到当 $m > n$ 时没有从 m 元集到 n 元集的一对一函数。现在令 $m \leq n$ 。假设定义域中的元素是 $a_1, a_2, \dots, a_m$ 。有 n 种方式选择函数在 a_1 的值。因为函数是一对一的,可以有 $n-1$ 种方式选择函数在 a_2 的值(因为 a_1 用过的值不能再用)。一般地,有 $n-k+1$ 种方式选择函数在 a_k 的值。由乘积法则,从一个 m 元集到一个 n 元集存在着 $n(n-1)(n-2)\dots(n-m+1)$ 个一对一函数。例如,从一个3元集到一个5元集存在 $5 \cdot 4 \cdot 3 = 60$ 个一对一函数。 ■

例8 电话编号计划 在北美,电话号码的格式是由一个编号计划规定的。一个电话号码由10个数字组成,这些数字由一个3位的地区代码、一个3位的局代码以及一个4位的话机代码组成。出于信号的考虑,在一些数字上有某种限制。为了规定允许的格式,令 X 表示可以在0到9之间任意选取的数字, N 表示可以在2到9之间选取的数字,而 Y 表示必须取0或1的数字。下面讨论两个编号计划,分别称为老计划和新计划(老计划是20世纪60年代使用的,已经被新计划代替了,但目前对新号码需求的迅速增长甚至使得这个新计划也将显得落伍了)。正如将要证明的,新计划允许使用更多的号码。

在老计划中,地区代码、局代码和话机代码的格式分别为 NYX 、 NNX 和 $XXXX$,因而电话号码形式为 $NYX-NNX-XXXX$ 。在新计划中,这些代码的格式分别为 NXX 、 NXX 和 $XXXX$,因而电话号码形式为 $NXX-NXX-XXXX$ 。在老计划和新计划下分别可能有多少个不同的北美电话号码?

解 由乘积法则,格式为 NYX 的地区代码有 $8 \cdot 2 \cdot 10 = 160$ 个,格式为 NXX 的地区代码有 $8 \cdot 10 \cdot 10 = 800$ 个。类似地,由乘积法则,存在 $8 \cdot 8 \cdot 10 = 640$ 个格式为 NNX 的局代码。乘积

法则也表明存在着 $10 \cdot 10 \cdot 10 \cdot 10 = 10\,000$ 个格式为 XXXX 的话机代码。

因此, 再次使用乘积法则, 结果在老计划下存在

$$160 \cdot 640 \cdot 10\,000 = 1\,024\,000\,000$$

个不同的北美有效的电话号码。在新计划下存在

$$800 \cdot 800 \cdot 10\,000 = 6\,400\,000\,000$$

个不同的电话号码。

例 9 执行下面的代码以后 k 的值是什么?

```

k := 0
for  $i_1 := 1$  to  $n_1$ 
  for  $i_2 := 1$  to  $n_2$ 
    :
    for  $i_m := 1$  to  $n_m$ 
      k := k + 1

```

解 k 的初值是 0。这个嵌套的循环每执行一次, k 就加 1。令 T_i 表示执行第 i 个循环的任务, 那么循环执行的次数就是完成任务 $T_1, T_2, \dots, T_m$ 的方法数。因为对每个整数 $i_j, 1 \leq i_j \leq n_j$, 第 j 个循环都执行一次, 执行任务 $T_j (j=1, 2, \dots, m)$ 的方法数就是 n_j 。由乘积法则, 这个嵌套的循环执行了 $n_1 n_2 \cdots n_m$ 次。因此 k 最后的值是 $n_1 n_2 \cdots n_m$ 。

例 10 计数有穷集的子集 用乘积法则证明一个有穷集 S 的不同的子集数是 $2^{|S|}$ 。

解 设 S 是有穷集。按任意的顺序将 S 的元素列成一个表。考虑到在 S 的子集和长为 $|S|$ 的二进制串之间存在着一对一的对应, 即如果表的第 i 个元素在这个子集里, 则该子集对应的二进制的第 i 位为 1, 否则这位为 0。由乘积法则, 存在着 $2^{|S|}$ 个长为 $|S|$ 的二进制串。因此 $|P(S)| = 2^{|S|}$ 。

乘积法则也常用集合的语言表述如下: 如果 $A_1, A_2, \dots, A_m$ 是有穷集, 那么在这些集合的笛卡儿积中的元素数是每个集合的元素数之积。为把这种表述与乘积法则联系起来, 注意到在笛卡儿积 $A_1 \times A_2 \times \cdots \times A_m$ 中选一个元素的任务是通过在 A_1 中选一个元素, A_2 中选一个元素, $\dots, A_m$ 中选一个元素来完成的。由乘积法则得到

$$|A_1 \times A_2 \times \cdots \times A_m| = |A_1| \cdot |A_2| \cdot \cdots \cdot |A_m|$$

现在引入求和法则。

求和法则 如果完成第一项任务有 n_1 种方式, 完成第二项任务有 n_2 种方式, 并且这些任务不能同时完成, 那么完成第一或第二项任务有 $n_1 + n_2$ 种方式。

例 11 说明怎样使用求和法则。

例 11 假定要选一位数学学院的教师或数学专业的学生作为校委会的代表。如果有 37 位数学学院的教师和 83 位数学专业的学生, 那么这个代表有多少种不同的选择?

解 完成第一项任务, 选一位数学学院的教师, 可以有 37 种方式。完成第二项任务, 选一位数学专业的学生, 有 83 种方式。根据求和法则, 结果有 $37 + 83 = 120$ 种可能的方式来挑选这个代表。

可以把求和法则推广到多于两项任务的情况。假定任务 $T_1, T_2, \dots, T_m$ 分别有 $n_1, n_2, \dots, n_m$ 种完成的方式, 并且任何两项任务都不能同时做, 那么完成其中一项任务的方式数是 $n_1 + n_2 + \cdots + n_m$ 。正如例 12 和例 13 所示, 这个推广的求和法则在计数问题中常常用到。这个求和法则可以从两个集合的求和法则使用数学归纳法加以证明(见本节后的练习 30)。

例 12 一个学生可以从三个表中的一个表选择一个计算机课题。这三个表分别包含 23、15 和 19 个可能的课题。那么课题的选择可能有多少种?

解 这个学生有 23 种方式从第一个表中选择课题, 有 15 种方式从第二个表中选择课题, 有

19 种方式从第三个表中选择课题。因此, 共有 $23+15+19=57$ 种选择课题的方式。 ■

例 13 在下面的代码执行后 k 的值是什么?

```

k := 0
for i1 := 1 to n1
    k := k + 1
for i2 := 1 to n2
    k := k + 1
    ⋮
for im := 1 to nm
    k := k + 1

```

解 k 的初值是 0。这个代码块由 m 个不同的循环构成。循环中的每次执行 k 都要加 1。令 T_i 是执行第 i 个循环的任务。因为第 i 个循环被执行 n_i 次, 所以任务 T_i 可以用 n_i 种方式完成。由于任何两个任务不能同时执行, 求和法则证明 k 的最后值, 即完成任务 $T_i (i=1, 2, \dots, m)$ 的方式数是 $n_1+n_2+\dots+n_m$ 。 ■

求和法则可以用集合的语言表述: 如果 $A_1, A_2, \dots, A_m$ 是不交的集合, 那么在其并集中的元素数是每个集合的元素数之和。为把这种表述与求和法则联系起来, 令 T_i 是从 $A_i (i=1, 2, \dots, m)$ 中选取一个元素的任务。有 $|A_i|$ 种方式做 T_i 。由于任何两个任务不可能同时做, 根据求和法则, 从其中某个集合选择一个元素的方式数, 即在并集中的元素数是

$$|A_1 \cup A_2 \cup \dots \cup A_m| = |A_1| + |A_2| + \dots + |A_m|$$

这个等式仅适用于问题中的集合是不相交的情况。当这些集合含有公共元素时, 情况要复杂得多。本节的后面将对这种情况进行简要的讨论, 更深入的讨论放在第 4 章。

3.1.3 比较复杂的计数问题

许多计数问题不能仅仅使用求和法则或者乘法法则来求解。但是, 许多复杂的计数问题可以使用这两个法则来求解。

Extra Example

例 14 在计算机语言 BASIC 的某个版本中, 变量的名字是含有一个或两个字符的符号串, 其中的大写和小写字母是不加区分的 (一个字符或者取自 26 个英文字母, 或者取自 10 个数字)。此外, 变量名必须以字母作为开始, 并且必须和由两个字符构成的用于程序设计的 5 个保留字相区别。在 BASIC 的这个版本中有多少个不同的变量名?

解 令 V 等于在这个 BASIC 版本中的不同的变量名个数, V_1 是单字符的变量名个数, V_2 是两个字符的变量名个数。那么由求和法则, $V=V_1+V_2$ 。由于单字符变量名必须是字母, 故 $V_1=26$ 。又根据乘法法则存在 $26 \cdot 36$ 个以字母打头以字母数字结尾的 2 位字符串。但是其中 5 个不包含在内, 因此 $V_2=26 \cdot 36-5=931$ 。从而在这个 BASIC 版本中存在 $V=V_1+V_2=26+931=957$ 个不同的变量名。 ■

例 15 计算机系统的每个用户有一个 6~8 个字符构成的密码, 其中每个字符是一个大写字母或者数字, 且每个密码必须至少包含一个数字。有多少可能的密码?

解 设 P 是可能的密码总数, 且 P_6, P_7, P_8 分别表示 6、7 或 8 位的可能的密码数。由求和法则, $P=P_6+P_7+P_8$ 。我们现在找 P_6, P_7 和 P_8 。直接找 P_6 是困难的。而找 6 个大写字母和数字构成的字符串数是容易的, 其中包含那些没有数字的串在内, 然后从中减去没有数字的串数就得到 P_6 。由乘法法则, 6 个字符的串数是 36^6 , 而没有数字的字符串数是 26^6 。因此,

$$P_6 = 36^6 - 26^6 = 2\,176\,782\,336 - 308\,915\,776 = 1\,867\,866\,560$$

类似地得到

$$P_7 = 36^7 - 26^7 = 78\,364\,164\,096 - 8\,031\,810\,176 = 70\,332\,353\,920$$

和

$$P_8 = 36^8 - 26^8 = 2\,821\,109\,907\,456 - 208\,827\,064\,576 = 2\,612\,282\,842\,880$$

从而

$$P = P_6 + P_7 + P_8 = 2\,684\,483\,063\,360$$



例 16 计数因特网网址 在由计算机的物理网络互连而构成的因特网中, 每台计算机(或者更精确地说是计算机的每个网络连接)被分配一个因特网地址。目前正在使用的因特网协议版本 4(IPv4)中, 一个地址是一个 32 位的二进制串。它以网络标识(netid)开始, 后面跟随着主机标识(hostid), 该标识把一个计算机认定为某个指定网络的成员。

根据网络标识和主机标识位数的不同, 使用 3 种地址形式。用于最大网络的 A 类地址, 由 0 后跟 7 位的网络标识和 24 位的主机标识构成。用于中等规模网络的 B 类地址, 由 10 后跟 14 位的网络标识和 16 位的主机标识构成。用于最小网络的 C 类地址, 由 110 后跟 21 位的网络标识和 8 位的主机标识构成。由于特定用途, 对地址有着某些限制: 1111111 在 A 类网络的网络标识中是无效的, 全 0 和全 1 组成的主机标识对任何网络都是无效的。因特网上的一台计算机有一个 A 类、B 类或 C 类地址。(除了 A 类、B 类和 C 类地址外, 还有 D 类地址和 E 类地址。D 类地址在多台计算机同时编址时用于多路广播, 它由 1110 后跟 28 位组成。E 类地址保留为将来应用, 由 11110 后跟 27 位组成。D 类和 E 类地址不会分配给因特网中的一台计算机作为 IP 地址。)图 3-1 显示了 IPv4 的编址。(A 类和 B 类网络标识的数量限制已经使得 IPv4 编址不够用了; 将代替 IPv4 的 IPv6 使用 128 位地址来解决这个问题。)

位数	0	1	2	3	4	8	16	24	31	
A类	0	网络标识				主机标识				
B类	1	0	网络标识				主机标识			
C类	1	1	0	网络标识				主机标识		
D类	1	1	1	0	多路广播地址					
E类	1	1	1	1	0	地址				

图 3-1 因特网地址(IPv4)

对因特网上的计算机有多少不同的有效 IPv4 地址?

解 令 x 是因特网上计算机的有效地址数, x_A 、 x_B 和 x_C 分别表示 A 类、B 类和 C 类的有效地址数。由求和法则, $x = x_A + x_B + x_C$ 。

为了找到 x_A , 由于 1111111 是无效的, 故存在 $2^7 - 1 = 127$ 个 A 类的网络标识。对于每个网络标识, 存在 $2^{24} - 2 = 16\,777\,214$ 个主机标识, 这是由于全 0 和全 1 组成的主机标识是无效的。因此,

$$x_A = 127 \cdot 16\,777\,214 = 2\,130\,706\,178$$

为了找到 x_B 和 x_C , 首先注意到存在 $2^{14} = 16\,384$ 个 B 类网络标识和 $2^{21} = 2\,097\,152$ 个 C 类网络标识。对每个 B 类网络标识存在着 $2^{16} - 2 = 65\,534$ 个主机标识, 而对每个 C 类网络标识存在着 $2^8 - 2 = 254$ 个主机标识, 这也是考虑到全 0 和全 1 组成的主机标识是无效的。因而,

$$x_B = 1\,073\,709\,056, \quad x_C = 532\,676\,608$$

我们可以断言 IPv4 有效地址的总数是

$$x = x_A + x_B + x_C = 2\,130\,706\,178 + 1\,073\,709\,056 + 532\,676\,608 = 3\,737\,091\,842$$

3.1.4 容斥原理

当同时做两个任务时, 我们不能使用求和法则来计数完成其中一个任务的方式。把对每个任务的方式数加起来将导致计数结果的增大, 因为同时完成两个任务的那些方式被计数了两次。为了正确地计数完成其中一个任务的方式, 我们先把完成每个任务的方式数加起来, 然后再减去同时完成

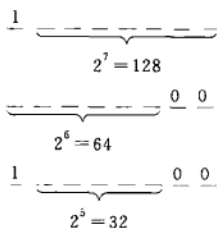
两个任务的方式数。这个技术叫做容斥原理。例17显示了怎样用这个原理来求解计数问题。

例17 以1开始或者以00结束的8位二进制符号串有多少个?

解 第一个任务,构造以1开始的8位二进制字符串,完成它有 $2^7=128$ 种方式,这是由乘积法则得到的。因为第一位只有一种选择方式,而其他7位中的每位有两种选择方式。

第二个任务,构造以00结束的8位二进制字符串,完成它有 $2^6=64$ 种方式,这也是由乘积法则得到的。因为前6位的每位有两种选择方式,而最后两位只有一种选择方式。

同时完成两个任务,构造以1开始以00结束的8位二进制符号串,完成它有 $2^5=32$ 种方式。这里也使用了乘积法则,因为第一位只有一种选法,从第二位到第六位每位可以有二种选法,最后两位也只有一种选法。因而,以1开始或者以00结束的8位二进制符号串个数,即完成第一或第二个任务的方式数,等于 $128+64-32=160$ 。



我们可以用集合的语言表述这一计数原理。令 A_1 和 A_2 是集合, T_1 是从 A_1 选择一个元素的任务, T_2 是从 A_2 选择一个元素的任务。完成 T_1 有 $|A_1|$ 种方式,完成 T_2 有 $|A_2|$ 种方式。完成 T_1 或 T_2 的方式数是完成 T_1 的方式数与完成 T_2 的方式数之和减去同时完成 T_1 、 T_2 两个任务的方式数。因为存在 $|A_1 \cup A_2|$ 种方式完成 T_1 或 T_2 , $|A_1 \cap A_2|$ 种方式完成 T_1 和 T_2 , 我们有

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

这就是在2.2节给出的计数两个集合并集中元素的公式。

我们给出一个例子说明如何使用容斥原理的公式求解计数问题。

例18 某计算机公司收到了350份计算机毕业生设计一组新网络服务器工作的工作申请书。假如这些申请人中有220人主修的是计算机专业,有147人主修的是商务专业,有51人既主修了计算机专业又主修了商务专业。那么,有多少个申请人既没有主修计算机专业又没有主修商务专业?

解 为了求出既没有主修计算机专业又没有主修商务专业的申请人的个数,可以从总的申请人数中减去主修计算机专业的人数,或减去主修商务专业的人数(或减去二者人数之和)。设 A_1 是主修计算机专业学生的集合, A_2 是主修商务专业学生的集合,那么 $A_1 \cup A_2$ 是主修计算机专业或主修商务专业学生的集合, $A_1 \cap A_2$ 是既主修计算机专业又主修商务专业学生的集合。根据容斥原理,主修计算机专业或主修商务专业(或二者都主修)学生的人数为

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2| = 220 + 147 - 51 = 316$$

因此得到结论:有 $350-316=34$ 个申请人既没有主修计算机专业又没有主修商务专业。

容斥原理可以推广来求完成 n 个不同任务中的一个任务的方式数,换句话说,就是找 n 个集合的并集中的元素数,其中 n 是正整数。我们将在第4章研究容斥原理和它的某些广泛应用。

3.1.5 树图

可以使用树图求解计数问题。一棵树由根、从根出发的许多分支以及可能从其他分支端点出发的新的分支构成。(我们将在第7章详细地研究树。)为了在计数中使用树,我们用一个分支表示每个可能的选择,用树叶表示可能的结果。这些树叶是某些分支的端点,从这些端点不再进一步分支。

注意,当用树图求解计数问题时,为到达一片树叶所作的选择个数可能是不同的(作为例子,见例20)。

例19 有多少不含连续两个1的4位二进制串?

解 图3-2的树图给出了所有不含连续两个1的4位二进制串。我们看出存在8个不含连续两个1的4位二进制串。

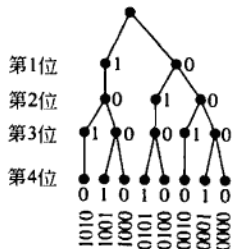


图3-2 不含连续两个1的4位二进制串

例20 在两个队(队1和队2)之间的决赛至多由5场比赛构成。先胜3次的队赢得决赛。决赛可能出现多少种不同的方式?

解 在图 3-3 的树图中, 以每次比赛的得胜者给出了决赛可能进行的所有方式。我们看到了 20 种不同的决赛的方式。

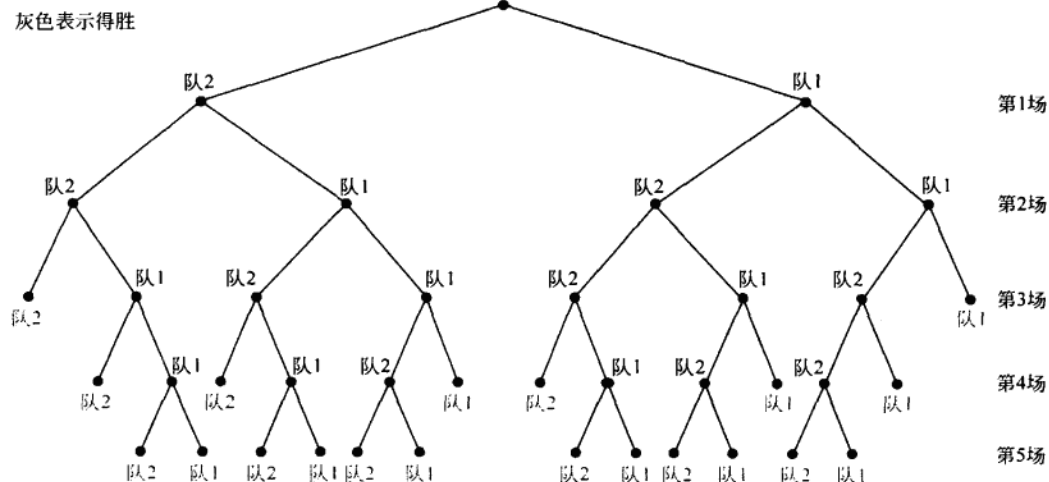


图 3-3 5 次决赛胜 3 次

例 21 假设“我爱新泽西”T 恤衫有 5 种不同的规格: S、M、L、XL 和 XXL。又知道 XL 规格只有三种颜色, 红色、绿色和黑色, XXL 规格只有绿色和黑色。除此之外, 其他规格有四种颜色, 白色、红色、绿色和黑色。如果每种规格和颜色的 T 恤衫至少一件, 一个纪念品商店必须库存多少件不同的 T 恤衫?

解 图 3-4 的树图给出了所有规格和颜色的配对。从中可知这个纪念品商店老板必须库存 17 件不同的 T 恤衫。

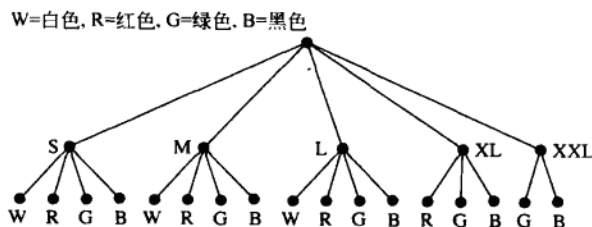


图 3-4 计数不同的 T 恤衫

练习

- 一个学院有 18 个数学专业和 325 个计算机科学专业的学生。
 - 选两个代表, 使得一个是数学专业的且另一个是计算机科学专业的, 有多少种方式?
 - 选一个数学专业或计算机科学专业的代表又有多少种方式?
- 一次多项选择考试包含 10 个问题。每个问题有 4 个可能的答案。
 - 在这次考试中如果每个问题都要回答, 一个学生回答这些问题可能有多少种方式?
 - 在这次考试中如果允许某些答案空缺, 一个学生回答这些问题可能有多少种方式?
- 从纽约到丹佛有 6 条不同的航线, 而从丹佛到旧金山有 7 条。如果选一个到丹佛的航班, 接着选一个到旧金山的航班, 那么从纽约经丹佛到旧金山的旅行有多少种不同的可能性?
- 如果用 3 个字母作为姓名的缩写, 人们可以有多少种不同的选择?
- 如果这 3 个字母以 A 开始, 人们又可以有多少种不同的选择?

6. 首尾都是 1 的 10 位二进制串有多少个?
7. 位数不超过 n 且全由 1 组成的二进制串有多少个? 这里的 n 是正整数。
8. 位数不超过 4 且由小写字母构成的串有多少个?
9. 由 5 个 ASCII 码构成且至少(在符号位)包含一个 @ 字符的串有多少个? (注: 有 128 个不同的 ASCII 码。)
10. 在 50 到 100 之间有多少个正整数
 - a) 能被 7 整除? 这些整数是什么?
 - b) 能被 11 整除? 这些整数是什么?
 - c) 能被 7 和 11 同时整除? 这些整数是什么?
11. 在 100 到 999 之间包含多少个正整数
 - a) 被 7 整除?
 - b) 是奇数?
 - c) 有相同的 3 个十进制数字?
 - d) 不被 4 整除?
 - e) 被 3 或 4 整除?
 - f) 不被 3 也不被 4 整除?
 - g) 被 3 整除但不被 4 整除?
 - h) 被 3 和 4 整除?
12. 有多少个串含有 3 个十进制数字且
 - a) 同一数字不能出现 3 次?
 - b) 以奇数字开始?
 - c) 恰有 2 个数字是 4?
13. 一个委员会由 50 个州构成, 每个州可从州长或两个参议员中选一个参加, 有多少种不同的方式?
14. 用 2 个字母后跟 4 个数字或者 2 个数字后跟 4 个字母可构成多少种车牌?
15. 用 2 个或 3 个字母后跟 2 个或 3 个数字可构成多少种车牌?
16. 由 8 个英语字母可构成多少个串?
 - a) 如果字母可以重复且不包含元音字母
 - b) 如果字母不能重复且不包含元音字母
 - c) 如果字母可以重复且以元音字母开始
 - d) 如果字母不能重复且以元音字母开始
 - e) 如果字母可以重复且包含至少一个元音字母
 - f) 如果字母可以重复且包含恰好一个元音字母
 - g) 如果字母可以重复且以 X 开始并至少包含一个元音字母
 - h) 如果字母可以重复且以 X 开始和结束并至少包含一个元音字母
17. 从 5 元素集到含有下述元素数的集合有多少一对一的函数?
 - a) 4
 - b) 5
 - c) 6
 - d) 7
18. 从集合 $\{1, 2, \dots, n\}$ 到集合 $\{0, 1\}$ 有多少个满足下列条件的函数? 这里的 n 是正整数。
 - a) 是一一对一的
 - b) 对 1 和 n 赋值为 0
 - c) 对恰好一个小于 n 的正整数赋值为 1
19. 从 m 元素集到 n 元素集有多少个部分函数(见 2.3 节的练习 33)? 这里的 m 和 n 是正整数。
20. 如果一个字符串反转后所得结果与原来的字符串一样, 就称它是一个回文。有多少个长为 n 的二进制串是回文?
21. 在一个婚礼上摄影师安排 6 个人在一排拍照, 包含新娘和新郎在内, 如果满足下述条件, 有多少种安排方式?
 - a) 新娘必须在新郎旁边
 - b) 新娘不在新郎旁边
 - c) 新娘在新郎左边的某个位置
22. 有多少个 10 位二进制串以 3 个 0 开始或以 2 个 0 结束?
- **23. 有多少个 8 位二进制串包含 3 个连续的 0 或者 4 个连续的 1?
24. 有多少个不超过 100 的正整数能被 4 或 6 整除?
25. 假定一个计算机系统的口令最少有 8 个、最多有 12 个字符, 其中口令中的每个字符可以是小写英文字母、大写英文字母、数字、或 6 个特殊字符 *、>、<、!、+、= 中的一个。
 - a) 该计算机系统可有多少个不同的口令?
 - b) 有多少个口令含有 6 个特殊字符中的一个?
 - c) 如果一个黑客核对每个可能的口令需要 1 纳秒时间, 他要核对完所有可能的口令需要多长时间?
26. 假定在将来的某个时间世界上的每部电话将被分配一个号码, 这个号码包含一个 1 到 3 位数字的形如 X 、

XX 或 XXX 的国家代码, 后面跟随着一个 10 位数字的形如 NXX-NXX-XXXX 的电话号码(如例 7 所描述的)。在这个编码计划中, 全世界将有多少个不同的有效电话号码?

27. 有多少种不同的方式排列字母 a 、 b 、 c 和 d , 使得 b 不紧跟在 a 的后边?
28. 使用树图确定 $\{3, 7, 9, 11, 24\}$ 的子集数, 使得子集中的元素之和小于 28。
29. a) 假设运动鞋的流行式样对男女都适用。女鞋的大小号码是 6, 7, 8, 9, 男鞋的大小号码是 8, 9, 10, 11 和 12。男鞋有白色和黑色, 而女鞋是白色、红色和黑色。如果一个商店各种大小和颜色的男、女运动鞋必须至少存一双, 用树图确定所需的鞋的数目。
b) 使用计数原理回答(a)中的问题。
30. 从两个任务的求和法则使用数学归纳法证明关于 m 个任务的求和法则。
31. 具有 n 条边的凸多边形有多少条对角线?(如果在多边形内或边界的每两个顶点的连线完全在这个集合之内, 则称为凸多边形。)

3.2 鸽巢原理

3.2.1 引言

有 20 只鸽子要飞往 19 个鸽巢栖息。由于有 20 只鸽子, 而只有 19 个鸽巢, 所以这 19 个鸽巢中至少有 1 个鸽巢里最少栖息着 2 只鸽子。为了说明这个结论是真的, 注意: 如果每个鸽巢中最多栖息着 1 只鸽子的话, 那么最多只有 19 只鸽子有住处, 其中每只鸽子一个巢。这个例子阐述了一个一般原理, 叫做鸽巢原理, 该原理断言: 如果鸽子数比鸽巢数多, 那么一定有一个鸽巢里至少有 2 只鸽子(见图 3-5)。当然, 这个原理除了鸽子和鸽巢外也可以用于其他对象。

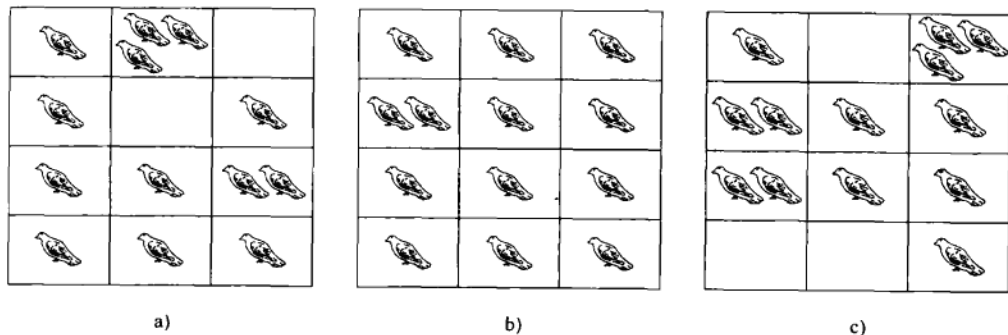


图 3-5 鸽子比鸽巢多

定理 1 鸽巢原理 如果 $k+1$ 个或更多的物体放入 k 个盒子, 那么至少有一个盒子包含了 2 个或更多的物体。

证 假定 k 个盒子中没有一个盒子包含的物体多于 1 个, 那么物体总数至多是 k , 这与至少有 $k+1$ 个物体矛盾。□

鸽巢原理也叫做狄利克雷抽屉原理, 以 19 世纪的德国数学家狄利克雷命名, 他经常在工作中使用这个原理。下面的例子说明了怎样使用鸽巢原理。

例 1 在一组 367 个人中一定至少有 2 个人有相同的生日, 这是由于只有 366 个可能的生日。

例 2 在 27 个英文单词中一定至少有 2 个单词以同一个字母开始, 因为英文字母表中只有 26 个字母。

例 3 如果考试给分是从 0 到 100, 班上必须有多少个学生才能保证在这次期末考试中至少有 2 个学生得到相同的分数?

解 期末考试有 101 个分数。鸽巢原理证明在 102 个学生中一定至少有 2 个学生具有相同的

分数。

鸽巢原理在许多证明中都是有用的工具,有些证明结果是令人意外的,正如图4所给出的。

例4 证明:对每个整数 n ,存在一个数是 n 的倍数,且在它的十进制表示中只出现0和1。

Extra Example

解 令 n 是正整数。考虑 n 个整数 $1, 11, 111, \dots, 11\cdots1$ (在这个数表中,最后一个整数的十进制表示中具有 $n+1$ 个1)。注意到当一个整数被 n 整除时存在 n 个可能的余数。因为这个数表中有 $n+1$ 个整数,由鸽巢原理必有两个整数在除以 n 时有相同的余数。这两个整数之差的十进制表示中只含有0和1,且它能被 n 整除。

3.2.2 广义鸽巢原理

鸽巢原理指出当物体比盒子多时一定至少有2个物体在同一个盒子里。但是当物体数超过盒子数的倍数时可以得出更多的结果。例如,在任意21个十进制数字中一定有3个是相同的。这是由于21个物体被分配到10个盒子里,那么某个盒子的物体一定多于2个。

定理2 广义鸽巢原理 如果 N 个物体放入 k 个盒子,那么至少有一个盒子包含了至少 $\lceil N/k \rceil$ 个物体。

证 假定没有盒子包含了比 $\lceil N/k \rceil - 1$ 多的物体,那么物体总数至多是

$$k\left(\left\lceil \frac{N}{k} \right\rceil - 1\right) < k\left(\left(\frac{N}{k} + 1\right) - 1\right) = N$$

这里用到不等式 $\lceil N/k \rceil < (N/k) + 1$ 。这与存在有总数 N 个物体矛盾。 $\square$

一类普遍的问题是,把一些物体分到 k 个盒子中要使得某个盒子至少含有 r 个物体,求这些物体的最少个数。当有 N 个物体时,广义鸽巢原理告诉我们,只要 $\lceil N/k \rceil \geq r$ 一定有 r 个物体在同一个盒子里。满足 $N/k > r-1$ 的最小正整数,即 $N = k(r-1) + 1$,是满足不等式 $\lceil N/k \rceil \geq r$ 的最小正整数。还可能有更小的 N 值吗?答案是没有。原因是如果我们有 $k(r-1)$ 个物体,我们可以在 k 个盒子中的每个盒子放 $r-1$ 个物体,因此没有一个盒子至少有 r 个物体。

当思考这种问题时下面的想法是有用的,就是在不断地放物体时怎样避免一个盒子至少有 r 个物体出现。为避免把第 r 个物体加到任何一个盒子里,每个盒子最终将以具有 $r-1$ 个物体结束。如果不允许放第 r 个物体到一个盒子里,就没有办法增加下一个物体。

例5至例8说明了怎样使用广义鸽巢原理。

例5 在100个人中至少有 $\lceil 100/12 \rceil = 9$ 个人生在同一个月。

Extra Example

例6 如果有5个可能的成绩 A, B, C, D 和 F ,那么在一个离散数学班里最少要多少个学生才能保证至少6个学生得到相同的分数?

解 为保证至少6个学生得到相同的分数,所需的最少学生数是使得 $\lceil N/5 \rceil = 6$ 的最小整数 N 。这样的最小整数是 $N = 5 \cdot 5 + 1 = 26$ 。如果只有25个学生,可能是每5个学生得到同样的分数,而没有6个学生得到同样的分数。于是,26是保证至少6个学生得到相同的分数所需的最少学生数。

例7 a)从一副标准的52张牌中必须选多少张牌才能保证选出的牌中至少有3张是同样的花色?

b)必须选多少张牌才能保证选出的牌中至少有3张是红心?

解 a)假设存在四个盒子保存四种花色的牌,选中的牌放在同种花色的盒子里。使用广义鸽巢原理,如果选了 N 张牌,那么至少有一个盒子含有至少 $\lceil N/4 \rceil$ 张牌。因此如果 $\lceil N/4 \rceil \geq 3$,我们知道至少选了3张同种花色的牌。使得 $\lceil N/4 \rceil \geq 3$ 的最小的整数 N 是 $N = 2 \cdot 4 + 1 = 9$,所以9张牌就足够了。注意到,如果选8张牌,可能每种花色2张牌。因而必须选9张牌才能保证选出的牌中至少3张是同样的花色。想到这一点的一个好方法就是注意到在选了8张牌以后没有办法避免出现3张同样花色的牌。

b)我们不用广义鸽巢原理回答这个问题,因为我们要保证存在3张红心而不仅仅是3张同样花色的牌。注意到在最坏情况下,选一张红心以前可能选了所有的黑桃、方块、梅花,总共39张牌,下面选的3张牌将都是红心。因此为得到3张红心,可能需要选42张牌。

例8 为保证一个州的2500万个电话有不同的10位电话号码,所需地区代码的最小数是多少?(假定电话号码是 $NXX-NXX-XXXX$ 形式,其中前3位是地区代码, N 表示从2到9包含的十进制数字, X 表示任何十进制数字)。

解 有800万个形如 $NXX-XXXX$ 的不同的电话号码(如3.1节的例8所示)。因此,由广义鸽巢原理,在2500万个电话号码中,一定至少有 $\lceil 25\,000\,000/8\,000\,000 \rceil$ 个同样的电话号码。因而至少需要4个地区代码来保证所有的10位号码是不同的。 ■

例9 尽管没有用到广义鸽巢原理,也用到了类似的原理。

例9 假设计算机科学实验室有15台工作站和10台服务器。可以用一条电缆直接把工作站连到服务器。同一时刻只有一条到服务器的直接连接是有效的。我们想保证在任何时刻任何一组不超过10个工作站可以通过直接连接同时访问不同的服务器。尽管我们可以通过将每台工作站直接连接到每台服务器(使用150条连线)来做到这一点,达到这个目标所需要的最少直接连线的数目是多少?

解 将工作站标记为 $W_1, W_2, \dots, W_{15}$, 服务器标记为 $S_1, S_2, \dots, S_{10}$ 。假设对于 $k=1, 2, \dots, 10$, 我们连接 W_k 到 S_k , 并且 $W_{11}, W_{12}, W_{13}, W_{14}$ 和 W_{15} 中的每个工作站都连接到所有的10个服务器。总共60条直接连线。很清楚,在任何时刻任何一组不超过10个工作站可以通过直接连接同时访问不同的服务器。为看到这一点只要注意到下述事实:如果这个组包含工作站 $W_j (1 \leq j \leq 10)$, 那么 W_j 可以访问服务器 S_j 。对于组里的每个工作站 $W_k (k \geq 11)$, 一定存在不在组里的工作站 $W_j (1 \leq j \leq 10)$ 与之对应, 因此 W_k 可以访问服务器 S_j (这是由于存在多少个不在组里的工作站 $W_j, 1 \leq j \leq 10$, 至少存在同样多的服务器 S_j 可以被其他工作站访问)。

现在假设在工作站和服务器之间直接连线少于60条。那么某个服务器将至多连接 $\lfloor 59/10 \rfloor = 5$ 工作站。(如果所有的服务器连接到至少6个工作站, 那么将存在至少 $6 \cdot 10 = 60$ 条直接连线。)这意味着剩下的9个服务器对于其他10个工作站同时访问不同的服务器就不够用了。因此,至少需要60条直接连线,从而得到答案是60。 ■

3.2.3 巧妙使用鸽巢原理

在鸽巢原理的许多有趣应用中,必须用某种巧妙的方式选择放入盒子的物体。下面将描述这样的一些应用。

例10 在30天的一个月里,某棒球队一天至少打一场比赛,但至多打45场。证明一定有连续的若干天内这个队恰好打了14场。

解 令 a_j 是在这个月的第 j 天或第 j 日之前所打的场数,则 $a_1, a_2, \dots, a_{30}$ 是不同正整数的一个递增序列,其中 $1 \leq a_j \leq 45$ 。从而 $a_1+14, a_2+14, \dots, a_{30}+14$ 也是不同正整数的一个递增序列,其中 $15 \leq a_j+14 \leq 59$ 。

60个正整数 $a_1, a_2, \dots, a_{30}, a_1+14, a_2+14, \dots, a_{30}+14$ 全都小于等于59。因此,由鸽巢原理有两个正整数相等。因为整数 $a_j, j=1, 2, \dots, 30$ 都不相同,并且 $a_j+14, j=1, 2, \dots, 30$ 也不相同,一定存在下标 i 和 j 满足 $a_i = a_j+14$ 。这意味着从第 $j+1$ 天到第 i 天恰好打了14场比赛。 ■

例11 证明在不超过 $2n$ 的任意 $n+1$ 个正整数中一定存在一个正整数被另一个正整数整除。

解 把 $n+1$ 个整数 $a_1, a_2, \dots, a_n$ 中的每一个都写成2的幂与一个奇数的乘积。换句话说,令 $a_j = 2^{k_j} q_j, j=1, 2, \dots, n+1$, 其中 k_j 是非负整数 q_j 是奇数。整数 $q_1, q_2, \dots, q_{n+1}$ 都是小于 $2n$ 的正奇数。因为只存在 n 个小于 $2n$ 的正奇数,由鸽巢原理, $q_1, q_2, \dots, q_{n+1}$ 中必有两个相等。于是,存在整数 i 和 j 使得 $q_i = q_j$ 。令 q_i 与 q_j 的公共值是 q , 那么 $a_i = 2^{k_i} q, a_j = 2^{k_j} q$ 。因而,若 $k_i < k_j$, 则 a_i 整除 a_j ; 若 $k_i > k_j$, 则 a_j 整除 a_i 。 ■

巧妙地应用鸽巢原理证明了在不同整数的序列中存在着确定长度的递增或递减子序列。在给出这个应用之前先回顾某些定义。假定 $a_1, a_2, \dots, a_N$ 是实数序列。它的一个子序列是形为 $a_{i_1}, a_{i_2}, \dots, a_{i_m}$ 的序列, 其中 $1 \leq i_1 < i_2 < \dots < i_m \leq N$ 。因此一个子序列是从初始序列得到的序列, 按

照原来的顺序选取初始序列的某些项,也许要排除其他的项。如果这个序列的每一项都大于它前面的项,就称为严格递增的,如果每一项都小于它前面的项,就称为严格递减的。

定理 3 每个由 n^2+1 个不同实数构成的序列都包含一个长为 $n+1$ 的严格递增子序列或严格递减子序列。

在证明定理 3 之前先给出一个例子。

例 12 序列 8, 11, 9, 1, 4, 6, 12, 10, 5, 7 包含 10 项。注意到 $10=3^2+1$, 存在四个长为 4 的递增子序列, 即 1, 4, 6, 12; 1, 4, 6, 7; 1, 4, 6, 10 和 1, 4, 5, 7。也存在一个长为 4 的递减子序列, 即 11, 9, 6, 5。

现在给出定理的证明。

证 令 $a_1, a_2, \dots, a_{n^2+1}$ 是 n^2+1 个不同实数的序列。和序列中的每一项 a_k 联系着一个有序对, 即 (i_k, d_k) , 其中 i_k 是从 a_k 开始的最长的递增子序列的长度, 且 d_k 是从 a_k 开始的最长的递减子序列的长度。

假定没有长为 $n+1$ 的递增或递减子序列。那么 i_k 和 d_k 都是小于或等于 n 的正整数, $k=1, 2, \dots, n^2+1$ 。因此, 由乘积法则关于 (i_k, d_k) 存在 n^2 个可能的有序对。根据鸽巢原理, n^2+1 个有序对中必有两个相等。换句话说, 存在项 a_i 和 a_t , $s < t$, 使得 $i_s = i_t$ 和 $d_s = d_t$ 。我们将证明这是不可能的。由于序列的项是不同的, 不是 $a_s < a_t$ 就是 $a_s > a_t$ 。如果 $a_s < a_t$, 那么由于 $i_s = i_t$, 那么把 a_t 加到从 a_s 开始的递增子序列前面就构造出一个从 a_s 开始的长为 i_s+1 的递增子序列。从而产生矛盾。类似地, 如果 $a_s > a_t$, 可以证明 d_t 一定大于 d_s , 从而也产生矛盾。

最后的例子说明了怎样把广义鸽巢原理用于组合学的重要部分拉姆齐理论, 它是以英国数学家拉姆齐而命名的。一般地说, 拉姆齐理论可用于处理集合元素的子集分配问题。

例 13 假定一组有 6 个人, 任两个人或者是朋友或者是敌人。证明在这组人中或存在 3 个人彼此都是朋友, 或存在 3 个人彼此都是敌人。

解 令 A 是 6 个人中的一人, 组里其他 5 个人中至少有 3 个人是 A 的朋友, 或至少有 3 个人是 A 的敌人。这可从广义鸽巢原理得出, 因为当 5 个物体分成两个集合时, 其中的一个集合至少有 $\lceil 5/2 \rceil = 3$ 个元素。若是前一种情况, 假定 B, C 和 D 是 A 的朋友。如果这 3 个人中有 2 个人也是朋友, 那么这 2 个人和 A 构成彼此是朋友的 3 人组。否则, B, C 和 D 构成彼此为敌人的 3 人组。对于后一种情况的证明, 当 A 存在 3 个或更多的敌人时可以用类似的方法处理。

拉姆齐数 $R(m, n)$ (其中 m 和 n 是大于或等于 2 的正整数) 表示: 假设晚会上每两个人是朋友或者是敌人, 那么在一个晚会上使得或者有 m 个人两两都是朋友, 或者有 n 个人两两都是敌人所需要的最少人数。例 13 显示 $R(3, 3) \leq 6$ 。在一组 5 个人中, 其中每两个人是朋友或是敌人, 可能没有 3 个人两两是朋友, 也没有 3 个人两两是敌人, 因此我们断言 $R(3, 3) = 6$ 。

可以证明某些关于拉姆齐数的有用的性质, 但是对于大多数拉姆齐数找到精确的值是困难的。根据对称性可以证明 $R(m, n) = R(n, m)$ 。对于每个正整数 $n \geq 2$ 我们也有 $R(2, n) = n$ (见练习 14)。只知道 9 个拉姆齐数 $R(m, n)$ ($3 \leq m \leq n$) 的精确值, 其中包括 $R(4, 4) = 18$ 。对许多其他的拉姆齐数只知道界, 包括 $R(5, 5)$ 在内, 已知它满足 $43 \leq R(5, 5) \leq 49$ 。有兴趣了解更多有关拉姆齐数知识的读者可以参考 [MiRo91] 或 [GrRoSp90]。

练习

- 假定周末不排课, 证明: 在任一组 6 门课中一定有 2 门课安排在同一天上课。
- 抽屉里有一打棕色的短袜和一打黑色的短袜, 全都没有配好对。一个人在黑暗中随机取出一些袜子。
 - 必须取多少只袜子才能保证至少有 2 只袜子是同色的?
 - 必须取多少只袜子才能保证至少有 2 只袜子是黑色的?
- 证明: 在任意 5 个整数中 (不一定是连续的) 有 2 个整数被 4 除的余数相等。
- 设 n 是正整数。证明: 在任意一组 n 个连续的正整数中恰好有 1 个被 n 整除。
- 在一个大学里每个学生来自 50 个州中的一个州, 那么必须有多少个学生注册才能保证至少有 100 个学生

来自同一个州?

- *6. 设 $(x_i, y_i, z_i) (i=1, 2, 3, 4, 5, 6, 7, 8, 9)$ 是 xyz 空间中一组具有整数坐标的 9 个不同的点。证明：至少有一对点的连线中点的坐标是整数。
7. a) 如果从前 8 个正整数中选 5 个整数一定存在一对整数其和等于 9。
b) 如果不是选 5 个而是选 4 个整数，a) 的结论还为真吗?
8. 从集合 $\{1, 2, 3, 4, 5, 6\}$ 中必须选多少个数才能保证其中至少有一对数之和等于 7?
9. 一个公司在仓库储存产品。仓库中的存储柜由它们的通道、在通道中的位置和货架来指定。整个仓库有 50 个通道，每个通道 85 个水平位置，每个位置 5 个货架。公司产品数至少是多少才能使得在同一个存储柜中至少有 2 个产品?
10. 在 25 个学生的离散数学班中，学生有一年级的、二年级的或者三年级的。
a) 证明：这个班至少有 9 个是一年级的，或至少有 9 个是二年级的，或至少有 9 个是三年级的。
b) 证明：这个班至少有 3 个是一年级的，或至少有 19 个是二年级的，或至少有 5 个是三年级的。
11. 构造 16 个正整数的序列，使得它没有 5 项的递增或递减子序列。
- *12. 用伪码描述一个算法产生一个不同整数序列的最大递增或递减子序列。
13. 证明：在任一组 10 个人中(其中任两个人或者是朋友或者是敌人)或存在 3 个人彼此都是朋友，或存在 4 个人彼此都是敌人；并且存在 3 个人彼此是敌人，或存在 4 个人彼此是朋友。
14. 证明：如果 n 是正整数， $n \geq 2$ ，那么拉姆齐数 $R(2, n)$ 等于 n 。
15. 证明：在加利福尼亚州(人口 3600 万)至少有 6 个人姓名的 3 个缩写字母相同并且他们生在一年的同一天(但不一定是同一年)。
16. 一个大学有 38 个不同的时间段来安排课程，如果有 677 门不同的课程，那么需要多少个不同的教室?
17. 一个计算机网络由 6 台计算机组成。每台计算机直接连接到零台或者更多的其他计算机。证明：网络中至少有两台计算机直接连接相同数目的其他计算机。
18. 把 100 台计算机连接到 20 台打印机上，为保证 20 台计算机可以直接访问 20 台不同的打印机，找出至少需要多少条缆线。证明你的答案。
19. 一个摔跤选手是 75 小时之内的冠军。该选手一小时至少赛一场，但总共不超过 125 场。证明：存在着连续的若干个小时使得该选手恰好进行了 24 场比赛。
20. 如果 f 是从 S 到 T 的函数，其中 S 和 T 是有穷集，并且 $m = \lceil |S| / |T| \rceil$ ，那么证明至少存在 S 的 m 个元素映射到 T 的同一个值。即存在 S 中的元素 $s_1, s_2, \dots, s_m$ 使得 $f(s_1) = f(s_2) = \dots = f(s_m)$ 。
- *21. 设 x 是无理数。证明：对于某个不超过 n 的正整数 j ，在 jx 和到 jx 最近的整数之间的差的绝对值小于 $1/n$ 。
- *22. 在这个问题中概述了基于广义鸽巢原理的定理 3 的证明，使用的记号与教科书中的证明一样。
a) 假定 $i_k \leq n$ ， $k=1, 2, \dots, n^2+1$ 。使用广义鸽巢原理证明：存在 $n+1$ 个项 $a_{k_1}, a_{k_2}, \dots, a_{k_{n+1}}$ 满足 $i_{k_1} = i_{k_2} = \dots = i_{k_{n+1}}$ ，其中 $1 \leq k_1 < k_2 < \dots < k_{n+1}$ 。
b) 证明： $a_{k_j} > a_{k_{j-1}}$ ， $j=1, 2, \dots, n$ 。[提示：假定 $a_{k_j} < a_{k_{j-1}}$ ，证明这将推出 $i_{k_j} > i_{k_{j-1}}$ 的矛盾。]
c) 使用(a)和(b)证明：如果没有长为 $n+1$ 的递增子序列，那么一定有同样长的递减子序列。

3.3 排列与组合

3.3.1 引言

许多计数问题都可以通过找到特定大小的集合中不同元素排列的不同方法数来得以解决，其中这些元素的次序是有限制的。许多其他计数问题也可以通过从特定大小的集合元素中选择特定数量元素的方法数来得以解决，其中这些元素的次序是不受限制的。例如，从 5 个学生中选出 3 个学生站成一行照相，有多少种选择方法？从 4 个学生中选出 3 个学生组成一个委员会，有多少种选择方法？本节将开发一些方法来解决此类问题。

3.3.2 排列

先通过解决引言中提出的第一个问题以及一些其他相关问题开始本节内容。

例 1 从 5 个学生中选出 3 个学生站成一行照相，有多少种选择方法？让所有 5 个学生站成一行照相，有多少种排列方法？

Extra Examples

解 首先注意：选择学生时次序是有限制的。从5个学生中选择第一个学生站在一行的第一个位置有5种方法。一旦这个学生被选定之后，则有4种方法选择第二个学生站在一行的第二个位置。当第一和第二个学生都被选定之后，则有3种方法选择第三个学生站在一行的第三个位置。根据乘法原理，共有 $5 \cdot 4 \cdot 3 = 60$ 种方法从5个学生中选出3个学生站成一行来照相。

为了排列所有5个学生站成一行来照相，选择第一个学生时有5种方法，选择第二个学生时有4种方法，第三个学生时有3种方法，第四个学生时有2种方法，第五个学生时有1种方法。因此，共有 $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$ 种方法让所有5个学生站成一行来照相。 ■

例1阐述了不同个体有次序的排列是如何计数的。这也引出了几个术语。

集合中不同元素的排列，是对这些元素一种有序的排列。我们也对集合中某些元素的有序安排感兴趣。对一个集合中 r 个元素的有序安排称为一个 r 排列。

不同个体集合的一个排列是这些个体的一种有序安排。我们也对集合的某些元素的有序安排感兴趣。一个集合的 r 个元素的有序安排叫做 r 排列。

例2 设 $S = \{1, 2, 3\}$ 。3, 1, 2 是 S 的一个排列，3, 2 是 S 的一个2排列。 ■

一个 n 元集的 r 排列数记为 $P(n, r)$ 。我们可以使用乘法法则求出 $P(n, r)$ 。

例3 设 $S = \{1, 2, 3\}$ 。 S 的2排列有如下有序安排： a, b ； a, c ； b, a ； b, c ； c, a ；和 c, b 。因此，具有3个元素的这个集合共有6个2排列。为了说明所有具有3个元素的集合都有6个2排列，注意到：有3种方法选择排列中的第一个元素，有2种方法选择排列中的第二个元素，这是因为第二个元素必须不同于第一个元素。根据乘法法则，有 $P(3, 2) = 3 \cdot 2 = 6$ 。 ■

下面利用乘法法则找出求 $P(n, r)$ 的一个公式，其中 n 和 r 都是任意正整数，且 $1 \leq r \leq n$ 。

定理1 具有 n 个不同元素的集合的 r 排列数是

$$P(n, r) = n(n-1)(n-2)\cdots(n-r+1)$$

证 选择这个排列的第一个元素可以有 n 种方法，因为集合中有 n 个元素。选择排列的第二个元素有 $n-1$ 种方法，由于在使用了为第一个位置挑出的元素之后集合里还留下了 $n-1$ 个元素。类似地，选择第三个元素有 $n-2$ 种方法，依此类推，直到选择第 r 个元素恰好有 $n-(r-1) = n-r+1$ 种方法。因此，由乘法法则，存在

$$n(n-1)(n-2)\cdots(n-r+1)$$

个集合的 r 排列。 □

注意 只要 n 是一个非负整数，就有 $P(n, 0) = 1$ ，这是因为恰好有一种方法来排列0个元素，也就是说，恰好有一个排列中没有元素，即空排列。

下面给出定理1的一个有用的推论。

推论1 如果 n 和 r 都是整数，且 $0 \leq r \leq n$ ，则

$$P(n, r) = \frac{n!}{(n-r)!}$$

证 当 n 和 r 是整数，且 $1 \leq r \leq n$ 时，由定理1有

$$P(n, r) = n(n-1)(n-2)\cdots(n-r+1) = \frac{n!}{(n-r)!}$$

因为只要 n 是非负整数，就有 $\frac{n!}{(n-0)!} = \frac{n!}{n!} = 1$ ，我们知道公式 $P(n, r) = \frac{n!}{(n-r)!}$ 当 $r=0$ 时也成立。 □

由定理1知道，如果 n 是一个正整数的话，则有 $P(n, n) = n!$ 。用一些例子来说明这个结论。

例4 在进入竞赛的100个不同的人中有多少种方法选出一个一等奖得主、一个二等奖得主和一个三等奖得主？

解 不管哪个人得哪个奖，选取3个得奖人的方法数是从100个元素的集合中有序选择3个元素的方法数，即100个元素的集合的3排列数。因此，答案是

$$P(100, 3) = 100 \cdot 99 \cdot 98 = 970\,200$$

例 5 假定有 8 个赛跑运动员。第一名得到一枚金牌，第二名得到一枚银牌，第三名得到一枚铜牌。如果比赛可能出现所有可能的结果，有多少种不同的颁奖方式？

解 颁奖方式就是 8 元素的集合的 3 排列数。因此存在 $P(8, 3) = 8 \cdot 7 \cdot 6 = 336$ 种可能的颁奖方式。

例 6 假定一个女推销员要访问 8 个不同的城市。她的访问必须从某个指定的城市开始，但对其他 7 个城市的访问可以按照任何次序进行。当访问这些城市时，这个女推销员可以有多少种可能的次序？

解 由于第一个城市是确定的，而其他 7 个城市可以是任意的顺序，故城市之间可能的路径数是 7 个元素的排列数。因此，这个女推销员有 $7! = 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 5040$ 种方式选择她的旅行。比如说，如果这个女推销员想要在城市中找出具有最短距离的路径，并且她对每一条可能的路径计算总距离，那么她必须考虑 5040 条路径。

例 7 字母 ABCDEFGH 有多少种排列包含串 ABC？

解 由于字母 ABC 必须成组出现，我们可以通过找 6 个对象，即组 ABC 和单个字母 D、E、F、G 和 H 的排列数得到答案。由于这 6 个对象可以按任何次序出现，因此，存在 $6! = 720$ 种 ABCDEFGH 字母的排列，其中 ABC 成组出现。

3.3.3 组合

现在把注意力转到无序选择个体的计数上来。我们先通过解决本章引言中提出的第二个问题开始本节内容。

例 8 从 4 个学生中选出 3 个学生组成一个委员会，有多少种选择方法？

解 为了回答这个问题，只需从含有 4 个学生的集合中找到具有 3 个元素的子集的个数。我们知道，一共有 4 个这样的子集，每个子集中都有一个不同的学生，因为选择 4 个学生等价于从 4 个学生中选出一个人离开这个集合。这就意味着有 4 种方法选择 3 个学生组成一个委员会，其中这些被学生的次序是无关紧要的。

例 8 阐明了这样一个事实：许多计数问题都可以通过从具有 n 个元素的集合中求得特定大小的子集的个数来得以解决，其中 n 是一个正整数。

 集合元素的一个 r 组合是从这个集合无序选取的 r 个元素。于是，简单地说，一个 r 组合是这个集合的一个 r 个元素的子集。

例 9 设 S 是集合 $\{1, 2, 3, 4\}$ ，那么 $\{1, 3, 4\}$ 是 S 的一个 3 组合。

具有 n 个不同元素集合的 r 组合数记为 $C(n, r)$ 。注意 $C(n, r)$ 也记作 $\binom{n}{r}$ 并且称为二项式系数。在 3.4 节我们将学习这个记号。

例 10 因为 $\{a, b, c, d\}$ 的 2 组合是 $\{a, b\}$ ， $\{a, c\}$ ， $\{a, d\}$ ， $\{b, c\}$ ， $\{b, d\}$ 和 $\{c, d\}$ 6 个子集，有 $C(4, 2) = 6$ 。

可以用关于集合的 r 排列数的公式确定 n 元素的集合的 r 组合数。为此只需要注意到集合的 r 排列可以按下述方法得到：首先构成集合的 r 组合，接着排列这些组合中的元素。下面的定理给出了 $C(n, r)$ 的值，它的证明就是基于这个观察。

定理 2 设 n 是正整数， r 是满足 $0 \leq r \leq n$ 的整数， n 元素的集合的 r 组合数等于

$$C(n, r) = \frac{n!}{r!(n-r)!}$$

证 可以如下得到这个集合的 r 排列。先构成集合的 $C(n, r)$ 个 r 组合，然后以 $P(n, r)$ 种方式排序每个 r 组合中的元素，这可以用 $P(r, r)$ 种方式来做。因此，

$$P(n, r) = C(n, r) \cdot P(r, r)$$

这就推出

$$C(n, r) = \frac{P(n, r)}{P(r, r)} = \frac{n!/(n-r)!}{r!/(r-r)!} = \frac{n!}{r!(n-r)!} \quad \square$$

尽管定理 2 中的公式很清楚, 但对很大的 n 和 r 而言, 这个公式并没有什么用处。其原因是: 在实际计算中, 只能对较小的整数才能求得阶乘的准确值, 而且当用浮点数来计算时, 从定理 2 的公式中得到的结果可能并不是一个整数值。因此, 当计算 $C(n, r)$ 时, 首先注意到: 如果从定理 2 的 $C(n, r)$ 计算公式的分子和分母中都消去 $(n-r)!$ 后, 可以得到

$$C(n, r) = \frac{n!}{r!(n-r)!} = \frac{n(n-1)\cdots(n-r+1)}{r!}$$

因此, 为了计算 $C(n, r)$, 可以从分子和分母中消去分母中所有较大的因式, 再把分子中所有没有消去的项相乘, 然后再除以分母中较小的因式。[如果是用手而不是用机器计算的话, 有必要再在 $n(n-1)\cdots(n-r+1)$ 和 $r!$ 中消去公因数。]注意: 许多计算器中都有一个关于计算 $C(n, r)$ 内置函数, 这些函数可以对相对较小的 n 和 r 求得结果, 许多计算机程序也可以用来求 $C(n, r)$ 的值。[这些函数可能会称为 $chose(n, k)$ 或 $binom(n, k)$ 。]

例 11 说明了当 k 相对于 n 较小时, 以及当 k 接近于 n 时, 如何计算 $C(n, r)$ 。该例子也给出了组合数 $C(n, r)$ 的一个关键的恒等式。

例 11 从一副 52 张标准扑克牌中选出 5 张, 共有多少种不同方法? 从一副 52 张标准扑克牌中选出 47 张, 又有多少种不同方法?

解 因为从 52 张牌中选出 5 张, 这 5 张牌的次序不受限制, 所以不同的选择方法数共有

$$C(52, 5) = \frac{52!}{5!47!}$$

为了计算 $C(52, 5)$, 首先在分子和分母中都消去 $47!$, 得

$$C(52, 5) = \frac{52 \cdot 51 \cdot 50 \cdot 49 \cdot 48}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}$$

上述表达式还可以化简: 首先将分子中的 50 除以分母中的因子 5, 则在分子中得到因子 10, 然后将分子中的 48 除以分母中的因子 4, 则在分子中得到因子 12, 再将分子中的 51 除以分母中的因子 3, 则在分子中得到因子 17, 最后, 将分子中的 52 除以分母中的因子 2, 在分子中得到因子 26。于是得到

$$C(52, 5) = 26 \cdot 17 \cdot 10 \cdot 49 \cdot 12 = 2\,598\,960$$

因此, 从一副 52 张标准扑克牌中选出 5 张, 共有 2 598 960 种不同方法。

注意: 从一副 52 张标准扑克牌中选出 47 张, 不同选择方法数为

$$C(52, 47) = \frac{52!}{47!5!}$$

不用再计算这个值了, 因为 $C(52, 5) = C(52, 47)$ 。(因为在计算它们的公式中, 只有分母中 $5!$ 和 $47!$ 的次序是不同的。)因此有: 从一副 52 张标准扑克牌中选出 47 张, 共有 2 598 960 种不同方法。 ■

在例 11 中, 我们看到 $C(52, 5) = C(52, 47)$ 。这是引理 2 中关于 r 组合数一个有用的恒等式的一个特例。

推论 2 设 n 和 r 是满足 $r \leq n$ 的非负整数, 那么 $C(n, r) = C(n, n-r)$ 。

证 由定理 2 得到

$$C(n, r) = \frac{n!}{r!(n-r)!}$$

$$C(n, n-r) = \frac{n!}{(n-r)![n-(n-r)]!} = \frac{n!}{(n-r)!r!}$$

因此, $C(n, r) = C(n, n-r)$ 。 □

通过证明在推论 1 的等式两边使用了不同的理由对同样的个体进行计数, 也可以证明推论 1。我们在定义 1 描述了这种重要的证明类型。

定义 1 组合证明是一种证明, 在这个证明中使用计数的论述而不使用某些其他的方法如代数技巧来证明一个定理。

可以使用组合证明证出许多涉及二项式系数的恒等式。如果可以显示一个恒等式两边通过不同的方法计数了同样的元素, 那么对这个恒等式就可以使用组合证明。现在提供一个推论 1 的组合证明。

证 设 S 是 n 个元素的集合。 S 的每个具有 r 个元素的子集 A 对应于一个具有 $n-r$ 个元素的子集, 即 $\bar{A}$, 因此 $C(n, r) = C(n, n-r)$ 。 $\square$

例 12 有多少种方式从 10 个选手的网球队中选择 5 个选手外出参加在另一个学校的比赛?

解 答案由 10 元素集合的 5 组合数给出。根据定理 2, 这个组合数是

$$C(10, 5) = \frac{10!}{5!5!} = 252$$

例 13 一组 30 个人被培训作为宇航员去完成首次登陆火星的任务。有多少种方式选出 6 个人的小组来完成这个任务(假设所有的小组成员有同样的工作)?

解 因为不考虑这些人被选的次序, 从 30 个人中选 6 个人的小组的方式数是 30 元素集合的 6 组合数。根据定理 2, 这种组合数是

$$C(30, 6) = \frac{30!}{6!24!} = \frac{30 \cdot 29 \cdot 28 \cdot 27 \cdot 26 \cdot 25}{6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} = 593\,775$$

例 14 有多少个长为 n 的二进制串恰好包含 r 个 1?

解 在长为 n 的二进制串中 r 个 1 的位置构成了集合 $\{1, 2, \dots, n\}$ 的 r 组合。因此, 有 $C(n, r)$ 个长为 n 的二进制串恰好包含 r 个 1。

例 15 为开发学校的离散数学课程要选出一个委员会。如果数学系有 9 个教师, 计算机科学系有 11 个教师。而这个委员会要由 3 个数学系的教师和 4 个计算机科学系的教师组成, 那么有多少种选择方式?

解 由乘积法则, 答案是 9 元素集合的 3 组合数与 11 元素集合的 4 组合数之积。根据定理 2, 选择这个委员会的方式数是

$$C(9, 3) \cdot C(11, 4) = \frac{9!}{3!6!} \cdot \frac{11!}{4!7!} = 84 \cdot 330 = 27\,720$$

练习

- 列出 $\{a, b, c\}$ 的所有排列。
- $\{a, b, c, d, e, f, g\}$ 有多少个排列以 a 结尾?
- 求出下面的每个值

a) $P(6, 3)$	b) $P(6, 5)$	c) $P(8, 1)$
d) $P(8, 5)$	e) $P(8, 8)$	f) $P(10, 9)$
- 求出 9 元素集合的 5 排列数。
- 在一场 12 匹马的赛马中, 如果所有的比赛结果都是可能的, 对于第一名、第二名和第三名有多少种可能性?
- 多少个 10 位二进制串包含

a) 恰好 4 个 1?	b) 至多 4 个 1?	c) 至少 4 个 1?	d) 0 的个数和 1 的个数相等?
--------------	--------------	--------------	--------------------
- 一个组有 n 个男士和 n 个女士。如果把他们的男女相间地排成一排, 有多少种方式?
- 有多少种不同的方式从英语字母表中选择 5 个字母?
- 一个 100 个元素的集合有多少个子集包含的元素多于 2 个?
- 一个硬币被掷 10 次, 每次可能出现头像或者非头像。有多少种可能的结果

a) 包含各种不同的情况?	b) 包含恰好 2 个头像?
c) 至多有 3 个不是头像?	d) 头像和非头像的数目相等?
- 字母 $ABCDEFGH$ 有多少个排列包含

a) 串 BCD ?	b) 串 $CFGA$?	c) 串 BA 和 GF ?	d) 串 ABC 和 DE ?
--------------	---------------	--------------------	---------------------

- e) 串 ABC 和 CDE? f) 串 CBA 和 BED?
12. 有多少种方式使得 8 个男士和 5 个女士站成一排并且没有两个女士彼此相邻? [提示: 先放男士, 然后考虑女士可能的位置.]
13. 把编号为 1, 2, ..., 100 的 100 张票卖给 100 个不同的人来抽奖。有 4 项不同的奖, 包括 1 项大奖(到塔希提岛旅游)。如果满足下面的条件, 有多少种不同的抽奖方式?
- 没有限制。
 - 拿 47 号票的人赢得了大奖。
 - 拿 47 号票的人赢了一项奖。
 - 拿 47 号票的人没赢奖。
 - 拿 19 和 47 号票的人都赢了奖。
 - 拿 19、47 和 73 号票的人都赢了奖。
 - 拿 19、47、73 和 97 号票的人都赢了奖。
 - 拿 19、47、73 和 97 号票的人都没赢奖。
 - 拿 19、47、73 或 97 号票的人赢得了大奖。
 - 拿 19 和 47 号票的人赢了奖, 但拿 73 和 97 号票的人没赢奖。
14. 一个俱乐部有 25 个成员。
- 有多少种方式从中选择 4 个人作为董事会成员。
 - 有多少种方式中选出俱乐部的主席、副主席、书记和司库?
- *15. 用不超过 100 的正整数构成 4 排列, 其中有多少个排列包含 3 个连续的整数 $k, k+1, k+2$?
- 这里的连续指按照整数通常的顺序, 并且这些连续整数可能被排列中的其他整数分开。
 - 这里的连续不但指整数是连续的, 而且它们在排列中的位置也是连续的。
16. 英语字母表中包含 21 个辅音和 5 个元音。由英语字母表的 6 个小写字母可构成多少字符串使得它们包含
- 恰好 1 个元音?
 - 恰好 2 个元音?
 - 至少 1 个元音?
 - 至少 2 个元音?
17. 假定某个系包含 10 名男士和 15 名女士。有多少种方式组成一个 6 人委员会且使得它含有相同数量的男士和女士?
18. 有多少个二进制串恰好包含 8 个 0 和 10 个 1, 如果每个 0 后面紧跟着 1 个 1?
19. 有多少个 10 位二进制串包含至少 3 个 1 和至少 3 个 0?
20. 有多少种方式用 3 个字母后跟 3 个数字组成汽车牌照并且没有字母和数字出现 2 次?
21. 如果允许出现并列名次, 3 匹马参加马赛有多少种结果? (注: 可以 2 匹或 3 匹马并列。)
- *22. 有 6 名运动员参加百米赛跑。如果允许并列名次, 有多少种方式授予 3 块奖牌? (跑得最快的运动员得金牌, 恰好只被一个运动员超过的运动员得银牌, 恰好被 2 个运动员超过的运动员得铜牌。)

3.4 二项式系数

正如在 3.3 节谈到的, 具有 n 个元素的集合的 r 组合数常常记作 $\binom{n}{r}$, 由于这些数出现在二项式的幂 $(a+b)^n$ 的展开式中作为系数, 因此这些数叫做二项式系数。我们将讨论二项式定理, 这个定理将二项式的幂表示成与二项式系数有关的项之和。我们将用组合证明来证明这个定理。我们也将显示怎样用组合证明来建立某些恒等式, 它们是表示二项式系数之间关系的许多不同恒等式中的一部分。

3.4.1 二项式定理

 二项式定理给出了二项式幂的展开式的系数。一个二项式只不过是两项的和, 例如 $x+y$ 。(这些项可以是常数与变量的积, 但这里先不考虑。)例 1 说明了为什么这个定理成立。

例 1 $(x+y)^3$ 的展开式可以使用组合推理而不是用三个项的乘法得到。当 $(x+y)^3 = (x+y)(x+y)(x+y)$ 被展开时, 把所有由第一个和的一项、第二个和的一项与第三个和的一项产生的乘积加起来。从而出现了形如 x^3 、 x^2y 、 xy^2 和 y^3 的项。为得到形如 x^3 的项, 在每个和里必须选择

一个 x , 只有一种方式能做到这一点。因此, 乘积中 x^3 项的系数是 1。为得到形如 x^2y 的项, 必须从三个和中的两个和里选 x (而因此在另一个和里选 y)。于是, 这种项的个数是三个个体的 2 组合数, 即 $\binom{3}{2}$ 。类似地, 形如 xy^2 项的个数是三个和中选一个来提供 x 的方式数 (而另两个和中都要选 y), 有 $\binom{3}{1}$ 种方式能够做到这一点。最后, 得到 y^3 的唯一方式是三个和的每一个都选择 y , 恰好有一种方式能够做到这一点。因此得到

$$\begin{aligned}(x+y)^3 &= (x+y)(x+y)(x+y) = (xx+xy+yx+yy)(x+y) \\ &= xx x + xx y + xy x + xyy + yxx + yxy + yyx + yyy \\ &= x^3 + 3x^2y + 3xy^2 + y^3\end{aligned}$$

现在叙述二项式定理。

定理 1 二项式定理 设 x 和 y 是变量, n 是非负整数, 那么

$$\begin{aligned}(x+y)^n &= \sum_{j=0}^n \binom{n}{j} x^{n-j} y^j \\ &= \binom{n}{0} x^n + \binom{n}{1} x^{n-1} y + \binom{n}{2} x^{n-2} y^2 + \cdots + \binom{n}{n-1} x y^{n-1} + \binom{n}{n} y^n\end{aligned}$$

证 这里给出定理的组合证明。当乘积被展开时其中的项都是下述形式: $x^{n-j} y^j$ ($j=0, 1, 2, \dots, n$)。为计数形如 $x^{n-j} y^j$ 的项数, 观察到必须从 n 个和中选 $n-j$ 个 x (从而乘积中其他的 j 个项都是 y) 才能得到这种项。因此, $x^{n-j} y^j$ 的系数是 $\binom{n}{n-j}$, 它等于 $\binom{n}{j}$, 定理得证。 $\square$

例 2~例 4 说明了二项式定理的应用。

例 2 $(x+y)^4$ 的展开式是什么?

解 由二项式定理得到

$$\begin{aligned}(x+y)^4 &= \sum_{j=0}^4 \binom{4}{j} x^{4-j} y^j \\ &= \binom{4}{0} x^4 + \binom{4}{1} x^3 y + \binom{4}{2} x^2 y^2 + \binom{4}{3} x y^3 + \binom{4}{4} y^4 \\ &= x^4 + 4x^3 y + 6x^2 y^2 + 4x y^3 + y^4\end{aligned}$$

例 3 在 $(x+y)^{25}$ 的展开式中 $x^{12} y^{13}$ 的系数是什么?

解 由二项式定理得到这个系数是

$$\binom{25}{13} = \frac{25!}{13!12!} = 5\,200\,300$$

例 4 在 $(2x-3y)^{25}$ 的展开式中 $x^{12} y^{13}$ 的系数是什么?

解 首先注意到这个表达式等于 $(2x+(-3y))^{25}$ 。由二项式定理, 我们有

$$(2x+(-3y))^{25} = \sum_{j=0}^{25} \binom{25}{j} (2x)^{25-j} (-3y)^j$$

因此, 当 $j=13$ 时得到展开式中 $x^{12} y^{13}$ 的系数, 即

$$\binom{25}{13} 2^{12} (-3)^{13} = -\frac{25!}{13!12!} 2^{12} 3^{13}$$

我们可以用二项式定理证明某些有用的恒等式。正如推论 1、2 和 3 所示。

推论 1 设 n 为非负整数, 那么

$$\sum_{k=0}^n \binom{n}{k} = 2^n$$

证 用二项式定理, 令 $x=1$ 和 $y=1$, 我们有

$$2^n = (1+1)^n = \sum_{k=0}^n \binom{n}{k} 1^k 1^{n-k} = \sum_{k=0}^n \binom{n}{k}$$

这正是所需要的结果。 □

推论 1 也有一个好的组合证明, 我们现在给出这个证明。

证 一个 n 元素集合有 2^n 个不同的子集。每个子集有 0 个元素, 1 个元素, 2 个元素, $\dots$, n 个元素。具有 0 个元素的子集有 $\binom{n}{0}$ 个, 1 个元素的子集有 $\binom{n}{1}$ 个, 2 个元素的子集有 $\binom{n}{2}$ 个, $\dots$, n 个元素的子集有 $\binom{n}{n}$ 个。于是

$$\sum_{k=0}^n \binom{n}{k}$$

计数了 n 元素集合的子集总数。这证明了

$$\sum_{k=0}^n \binom{n}{k} = 2^n$$
□

推论 2 设 n 是正整数, 那么

$$\sum_{k=0}^n (-1)^k \binom{n}{k} = 0$$

证 由二项式定理得出

$$0 = 0^n = ((-1) + 1)^n = \sum_{k=0}^n \binom{n}{k} (-1)^k 1^{n-k} = \sum_{k=0}^n \binom{n}{k} (-1)^k$$

从而证明了推论。 □

注意 推论 2 推出

$$\binom{n}{0} + \binom{n}{2} + \binom{n}{4} + \dots = \binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \dots$$

推论 3 设 n 是非负整数, 那么

$$\sum_{k=0}^n 2^k \binom{n}{k} = 3^n$$

证 注意到这个公式的左边是二项式定理提供的对 $(1+2)^n$ 的展开式。因此, 由二项式定理可以看出

$$(1+2)^n = \sum_{k=0}^n \binom{n}{k} 1^{n-k} 2^k = \sum_{k=0}^n \binom{n}{k} 2^k$$

因此

$$\sum_{k=0}^n 2^k \binom{n}{k} = 3^n$$
□

3.4.2 帕斯卡恒等式和三角形

二项式系数满足许多不同的恒等式。现在我们介绍其中最重要的一些恒等式。

定理 2 帕斯卡恒等式 设 n 和 k 是满足 $n \geq k$ 的正整数, 那么有

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

证 假定 T 是包含 $n+1$ 个元素的集合。令 a 是 T 的一个元素且 $S = T - \{a\}$ 。注意到 T 的包含 k 个元素的子集有 $\binom{n+1}{k}$ 个。然而 T 的含 k 个元素的子集或者包含 a 和 S 中的 $k-1$ 个元素, 或者不包含 a 但包含 S 中的 k 个元素。由于 S 的 $k-1$ 元子集有 $\binom{n}{k-1}$ 个, 故 T 含 a 在内的 k 元子集有

$\binom{n}{k-1}$ 个。又由于 S 的 k 元子集有 $\binom{n}{k}$ 个, 故 T 的不含 a 的 k 元子集有 $\binom{n}{k}$ 个。从而得到

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

□

注意 这里给出了帕斯卡恒等式的一个组合证明。也可以从关于 $\binom{n}{r}$ 的公式通过代数推导来证明这个恒等式(见节末的练习 10)。

注意 对所有整数 n , 可以用帕斯卡恒等式和初始条件 $\binom{n}{0} = \binom{n}{n} = 1$ 递归地定义二项式系数。这些递归定义用于计算二项式系数, 因为使用这些递归定义只需要整数加法。

帕斯卡恒等式是二项式系数以三角形表示的几何排列的基础, 如图 3-6 所示。

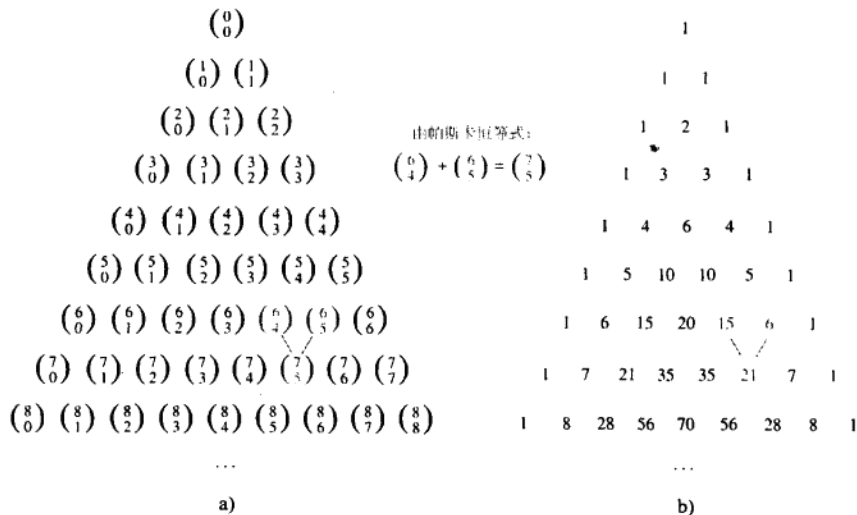


图 3-6 帕斯卡三角形

这个三角形的第 n 行由二项式系数

$$\binom{n}{k}, k = 0, 1, \dots, n$$

组成。这个三角形叫做帕斯卡三角形。帕斯卡恒等式证明, 当这个三角形中两个相邻的二项式系数相加时, 就产生了下一行在这两个系数之间的二项式系数。

3.4.3 其他的二项式系数恒等式

我们从众多二项式系数恒等式中选两个恒等式, 用它们的组合证明来作为本节的结束。

定理 3 范德蒙德恒等式 设 m 、 n 和 r 是非负整数, 其中 r 不超过 m 或 n , 那么

$$\binom{m+n}{r} = \sum_{k=0}^r \binom{m}{r-k} \binom{n}{k}$$

注意 这个恒等式是由 18 世纪的数学家亚历山大·舍费尔·范德蒙德发现的。

证 假定在第一集合中有 m 个项, 第二集合中有 n 个项。从这两个集合的并集中取 r 个元素的方式数是 $\binom{m+n}{r}$ 。从并集中取 r 个元素的另一种方式是先从第一个集合中取 k 个元素, 接着从

第二个集合中取 $r-k$ 个元素, 其中 k 是满足 $0 \leq k \leq r$ 的整数。由乘积法则, 这可以用 $\binom{m}{k} \binom{n}{r-k}$ 种方式完成。所以, 从这个并集中选取 r 个元素的总方式数等于

$$\binom{m+n}{r} = \sum_{k=0}^r \binom{m}{r-k} \binom{n}{k}$$

这就证明了范德蒙德恒等式。 □

推论 4 来自范德蒙德恒等式。

推论 4 如果 n 是一个非负整数, 那么

$$\binom{2n}{n} = \sum_{k=0}^n \binom{n}{k}^2$$

证 在范德蒙德恒等式中令 $m=r=n$ 得到

$$\binom{2n}{n} = \sum_{k=0}^n \binom{n}{n-k} \binom{n}{k} = \sum_{k=0}^n \binom{n}{k}^2$$

最后一步的相等是使用了恒等式 $\binom{n}{k} = \binom{n}{n-k}$ 。 □

我们可以通过计数具有不同性质的二进制串来证明组合恒等式, 如定理 4 的证明所示。

定理 4 设 n 和 r 是非负整数, $r \leq n$, 那么

$$\binom{n+1}{r+1} = \sum_{j=r}^n \binom{n}{j}$$

证 我们使用组合证明。由 3.3 节例 14, 左边 $\binom{n+1}{r+1}$ 计数了 $n+1$ 位长包含了 $r+1$ 个 1 的二进制串。

我们证明, 在具有 $r+1$ 个 1 的二进制串中, 通过考虑与最后一个 1 可能位置的相关情况, 等式右边计数了同样的对象。这最后一个 1 一定出现在位置 $r+1, r+2, \dots$, 或者 $n+1$ 。此外, 如果最后一个 1 出现在第 k 位, 那么一定有 r 个 1 出现在前 $k-1$ 位。因此, 根据 3.3 节例 14, 这样的二进制串有 $\binom{k-1}{r}$ 个, 对所有的 k 求和, 其中 $r+1 \leq k \leq n+1$, 我们发现

$$\sum_{k=r+1}^{n+1} \binom{k-1}{r} = \sum_{j=r}^n \binom{n}{j}$$

个 n 位二进制串恰含有 $r+1$ 个 1。(注意, 最后一步是改变变量 $j=k-1$ 的结果。)由于左边和右边计数了同样的对象, 因此相等。这就完成了证明。 □

练习

1. 求 $(x+y)^4$ 的展开式

a) 使用组合理由, 如例 1 所示。

b) 使用二项式定理。

2. 求 $(x+y)^6$ 的展开式。

3. 在 $(x+y)^{100}$ 的展开式中有多少个项?

4. 在 $(2-x)^{15}$ 中 x^9 的系数是什么?

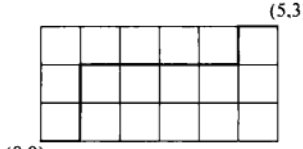
5. 在 $(2x-3y)^{200}$ 中 $x^{101}y^{99}$ 的系数是什么?

*6. 给出一个关于 $(x^2-1/x)^{100}$ 的展开式中 x^k 系数的公式, 其中 k 是一个整数。

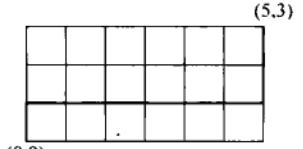
7. 帕斯卡三角形中包含二项式系数 $\binom{9}{k}$ ($0 \leq k \leq 9$) 的行是什么?

8. 证明: 对一切正整数 n 和 k ($0 \leq k \leq n$), $\binom{n}{k} \leq 2^n$ 。

*9. 证明: 如果 n 和 k 是整数, 其中 $1 \leq k \leq n$, 那么 $\binom{n}{k} \leq n^k / 2^{k-1}$ 。

10. 使用关于 $\binom{n}{r}$ 的公式证明帕斯卡恒等式。
11. 证明: 如果 n 和 k 是整数, $1 \leq k \leq n$, 那么 $k \binom{n}{k} = n \binom{n-1}{k-1}$.
- a) 使用组合证明。[提示: 证明恒等式两边计数了从一个 n 元素集合中选 k 个元素, 然后从这个子集中再选 1 个元素的方法。]
- b) 使用基于 3.3 节定理 2 给出的 $\binom{n}{r}$ 公式的代数证明。
12. 证明: 如果 n 和 k 是正整数, 那么 $\binom{n+1}{k} = (n+1) \binom{n}{k-1} / k$ 。
- 使用这个恒等式构造一个二项式系数的归纳的定义。
13. 设 n 是正整数, 证明 $\binom{2n}{n+1} + \binom{2n}{n} = \binom{2n+2}{n+1} / 2$ 。
- *14. 证明 $\sum_{k=0}^r \binom{n+k}{k} = \binom{n+r+1}{r}$ 。其中 n 和 r 是正整数。
- a) 用组合论证
- b) 用帕斯卡恒等式
- *15. 给出关于 $\sum_{k=1}^n k \binom{n}{k} = n2^{n-1}$ 的组合证明。[提示: 以两种方法计数选择一个委员会然后选择这个委员会领导的方式数。]
16. 证明: 一个非空集合具有奇数个元素的子集数与具有偶数个元素的子集数相等。
17. 在这个练习里, 我们将要计数 xy 平面上在原点和 (m, n) 点之间的路径数。这些路径由一系列步构成, 其中每一步是向右或者向上移动一个单位(不允许向左或向下移动)。下图给出了两条这种从 $(0, 0)$ 到 $(5, 3)$ 的路径(用粗线标识)。
- 

(0,0) (5,3)



(0,0) (5,3)
- a) 证明上述每条这种类型的路径可以用由 m 个 0 和 n 个 1 组成的二进制串表示, 其中 0 表示向右移动一个单位, 1 表示向上移动一个单位。
- b) 从 a) 推断存在着 $\binom{m+n}{n}$ 条所求类型的路径。
18. 使用练习 17 证明定理 4。[提示: 计数练习 17 所描述的那种 n 步路径数。每条路径必须在一个 $(n-k, k)$ 点结束, 其中 $k=0, 1, 2, \dots, n$ 。]
19. 使用练习 17 证明练习 14 中的恒等式。[提示: 首先观察到从 $(0, 0)$ 到 $(n+1, r)$ 的路径数等于 $\binom{n+1+r}{r}$ 。其次, 按照开始向上恰好走 k 个单位分别计数每一类路径, 其中 $k=0, 1, 2, \dots, r$, 然后对结果求和。]
- *20. 如果一个序列的前若干项如下列出, 对于它的第 n 项确定一个与二项式系数有关的公式。[提示: 对帕斯卡三角形的观察有助于问题的求解。虽然以这一组给定的项作为开始的序列有无数多个, 但下面列出的每个序列都是所求的那种序列的开始。]
- a) 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, ...
- b) 1, 4, 10, 20, 35, 56, 84, 120, 165, 220, ...
- c) 1, 2, 6, 20, 70, 252, 924, 3432, 12870, 48620, ...
- d) 1, 1, 2, 3, 6, 10, 20, 35, 70, 126, ...
- e) 1, 1, 1, 3, 1, 5, 15, 35, 1, 9, ...
- f) 1, 3, 15, 84, 495, 3003, 18564, 116280, 735471, 4686825, ...

3.5 排列与组合的推广

3.5.1 引言

在许多计数问题里元素可以被重复使用。例如，一个字母或一个数字可以在一个车牌中多次使用。当选一打多纳圈时，每种可以被重复地选择。这与本章前面讨论的计数问题形成对照，因为在那里我们只考虑每个项至多可以使用一次的排列和组合。在这一节我们将介绍怎样求解元素可以多次使用的计数问题。

还有，某些计数问题涉及不可区分的元素。例如，为计数单词 *SUCCESS* 的字母可能被重新排列的方式数，必须考虑相同字母的放置。这又与前面讨论的所有元素都被认为是不同的计数问题大相径庭。在这一节，我们将描述怎样求解某些元素是不可区分的计数问题。

此外，这一节也将解释怎样求解另一类重要的计数问题，即计数把不同的元素放入盒子的方法数的问题。这种问题的一个例子是把扑克牌发给 4 个玩牌人的不同的方式数。

把在本章前面描述的方法与这一节引入的方法一起考虑，就构成一个求解广泛的计数问题的有用的工具箱。当把第 4 章讨论的新方法再加到这个库里，你将能够求解在广泛的研究领域中产生的大多数计数问题。

3.5.2 有重复的排列

当元素允许重复时，使用乘积法则可以很容易地计数排列数，正如例 1 所示。

例 1 用英文字母可以构成多少个 r 位字符串？

解 因为有 26 个字母，且每个字母可以被重复使用，由乘积法则可以看出存在 26^r 个 r 位字符串。 ■

定理 1 给出了当允许重复时一个 n 元素集合的 r 排列数。

定理 1 具有 n 个物体的集合允许重复的 r 排列数是 n^r 。

证 当允许重复时，在 r 排列中对 r 个位置中的每个位置有 n 种方式选择集合的元素，因为对每个选择，所有 n 个物体都是有效的。因此，由乘积法则，当允许重复时存在 n^r 个 r 排列。 □

3.5.3 有重复的组合

考虑下面元素允许重复的组合的实例。

例 2 从包含苹果、橙子和梨的碗里选 4 个水果。如果选择水果的顺序无关，且只关心水果的类型而不管是该类型的哪一个水果，那么当碗中每类水果至少有 4 个时有多少种选法？

解 为了求解这个问题，我们列出选择水果的所有可能的方式。共有 15 种方式：

4 个苹果	4 个橙子	4 个梨
3 个苹果，1 个橙子	3 个苹果，1 个梨	3 个橙子，1 个苹果
3 个橙子，1 个梨	3 个梨，1 个苹果	3 个梨，1 个橙子
2 个苹果，2 个橙子	2 个苹果，2 个梨	2 个橙子，2 个梨
2 个苹果，1 个橙子，1 个梨	2 个橙子，1 个苹果，1 个梨	2 个梨，1 个苹果，1 个橙子

这个解是从 3 个元素的集合{苹果，橙子，梨}中允许重复的 4 组合数。 ■

为求解这种类型的更复杂的计数问题，我们需要计数一个 n 元素集合的 r 组合的一般方法。在例 3 中，我们将给出这一方法。

例 3 从包含 1 美元、2 美元、5 美元、10 美元、20 美元、50 美元及 100 美元的钱袋中选 5 张纸币，有多少种方式？假定不管纸币被选的次序，同种币值的纸币都是不加区别的，并且至少每种纸币有 5 张。

解 因为纸币被选的次序是无关的且 7 种不同类型的纸币都可以选 5 次，问题涉及的是计数从 7 个元素的集合中允许重复的 5 组合数。列出所有的可能性将是乏味的，因为存在许多的解。相反，我们将给出一种方法来计数允许重复的组合数。

假设一个零钱盒子有7个隔间,每个保存一种纸币,如图3-7所示。这些隔间被6块隔板分开,正如图中所画的。每选择1张纸币就在相应的隔间里放置1个标记。图3-8针对选择5张纸币的3种不同方式给出了这种对应,其中的竖线表示6个隔板,星表示5张纸币。

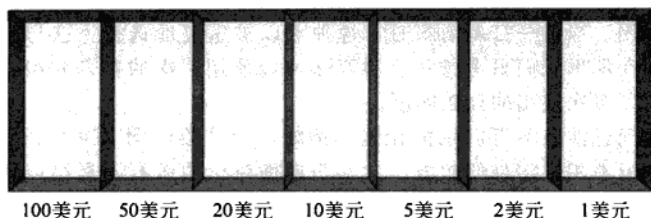


图 3-7 有 7 种类型纸币的零钱盒

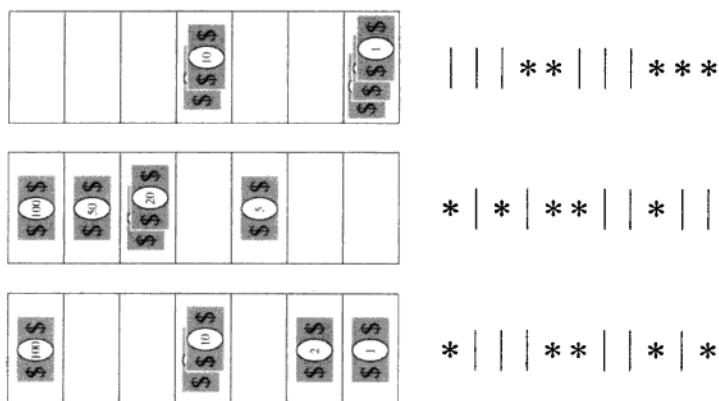


图 3-8 选择 5 张纸币的方式实例

选择 5 张纸币的方法数对应了安排 6 条竖线和 5 颗星的方法数。因此,选择 5 张纸币的方法数就是从 11 个可能的位置选 5 颗星位置的方法数。这对应了从含 11 个物体的集合中无序地选择 5 个物体的方法数,可以有 $C(11, 5)$ 种方式。因此存在

$$C(11, 5) = \frac{11!}{5!6!} = 462$$

种方式从有 7 类纸币的袋中选择 5 张纸币。

定理 2 将这个讨论一般化。

定理 2 n 个元素的集合中允许重复的 r 组合有 $C(n+r-1, r) = C(n+r-1, n-1)$ 个。

证 当允许重复时 n 元素集合的每个 r 组合可以用 $n-1$ 条竖线和 r 颗星的表表示。这 $n-1$ 条竖线用来标记 n 个不同的单元。每当集合的第 i 个元素出现在组合中,第 i 个单元就包含一颗星。例如,4 元素集合的一个 6 组合用 3 条竖线和 6 颗星来表示。这里

$$* * | * || * * *$$

代表了恰包含 2 个第一元素、1 个第二元素、0 个第三元素和 3 个第四元素的组合。

正如我们已经看到的,包含 $n-1$ 条竖线和 r 颗星的每一个不同的表对应了 n 元素集合的允许重复的一个 r 组合。这种表的个数是 $C(n-1+r, r)$, 因为每个表对应了从包含 r 颗星和 $n-1$ 条竖线的 $n-1+r$ 个位置中取 r 个位置来放 r 颗星的一种选择。这种表的个数还等于 $C(n-1+r, n-1)$, 因为每个表对应于取 $n-1$ 个位置来放 $n-1$ 条竖线的一种选择。□

例 4 至例 6 说明怎样使用定理 2。

Example

例 4 设一家甜点店有 4 种不同类型的甜点, 那么从中选 6 块甜点有多少种不同的方式? 假定只关心甜点的类型, 而不管是哪一块甜点或者选择的次序。

解 选择 6 块甜点的方式数是具有 4 类元素集合的 6 组合数。由定理 2 这等于 $C(4+6-1, 6) = C(9, 6)$ 。由于

$$C(9, 6) = C(9, 3) = \frac{9 \cdot 8 \cdot 7}{1 \cdot 2 \cdot 3} = 84$$

因此, 选择 6 块甜点的不同方式数有 84 种。

定理 2 也可以用于求给定线性方程的整数解的个数。这可以由例 5 来说明。

例 5 方程

$$x_1 + x_2 + x_3 = 11$$

有多少个解? 其中 x_1 、 x_2 和 x_3 是非负整数。

解 为计数解的个数, 注意到一个解对应了从 3 元素集合中选 11 个元素的方式, 以使得 x_1 选自第一类, x_2 选自第二类, x_3 选自第三类。因此, 解的个数等于 3 元素集合允许重复的 11 组合数。由定理 2 存在解的个数为

$$C(3+11-1, 11) = C(13, 11) = C(13, 2) = \frac{13 \cdot 12}{1 \cdot 2} = 78$$

当对变元加上限制时, 也可以求出这个方程的解的个数。例如, 当变元是满足 $x_1 \geq 1$ 、 $x_2 \geq 2$ 且 $x_3 \geq 3$ 的整数时, 也可以求出这个方程的解的个数。满足此限制的方程的解对应于 11 个项的选择, 使得项 x_1 取自第一类, 项 x_2 取自第二类, 项 x_3 取自第三类, 并且第一类元素至少取 1 个, 第二类元素至少取 2 个, 第三类元素至少取 3 个。因此, 先选 1 个第一类的元素, 2 个第二类的元素, 3 个第三类的元素; 然后再多选 5 个元素。由定理 2, 可以用

$$C(3+5-1, 5) = C(7, 5) = C(7, 2) = \frac{7 \cdot 6}{1 \cdot 2} = 21$$

种方式做到。于是, 对给定限制的方程存在 21 个解。

例 6 显示了怎样计数在确定变量值时产生的允许重复的组合数, 当每次通过某一类确定的嵌套循环时, 这个变量的值都会增加。

例 6 在下面的伪码被执行后 k 的值是什么?

```

k := 0
for i1 := 1 to n
    for i2 := 1 to i1
        ⋮
    for im := 1 to im-1
        k := k + 1

```

解 k 的初值是 0, 且对于满足

$$1 \leq i_m \leq i_{m-1} \leq \cdots \leq i_1 \leq n$$

的整数序列 $i_1, i_2, \dots, i_m$, 每次通过这个嵌套循环时 k 的值就加 1。这种整数序列的个数是从 $\{1, 2, \dots, n\}$ 中允许重复地选择 m 个整数的方式数。(为看到这一点只需注意到一旦这个整数序列选定以后, 如果我们按非降顺序排列序列中的整数, 这就唯一地确定了一组对 $i_m, i_{m-1}, \dots, i_1$ 的赋值。相反, 每个这样的赋值对应了一个唯一的无序集合。)所以由定理 2 得出在这个代码被执行后 $k = C(n+m-1, m)$ 。

从一个 n 元素集合中, 允许重复和不允许重复地选择 r 个元素, 其有序或无序的选择数的公式在表 3-1 给出。

表 3-1 允许和不允许重复的组合与排列

类 型	允许重复	公 式
r 排列	不	$\frac{n!}{(n-r)!}$
r 组合	不	$\frac{n!}{r! (n-r)!}$
r 排列	是	n^r
r 组合	是	$\frac{(n+r-1)!}{r! (n-1)!}$

3.5.4 具有不可区别物体的集合的排列

在计数问题中某些元素可能是没有区别的。在这种情况下必须小心避免重复计数。考虑例 7。

例 7 重新排序单词 *SUCCESS* 中的字母能构成多少个不同的串？

解 因为 *SUCCESS* 中的某些字母是重复的，答案并不是 7 个字母的排列数。这个单词包含 3 个 S、2 个 C、1 个 U 和 1 个 E。为确定重新排序单词中的字母能构成多少个不同的串，首先，注意到 3 个 S 可以用 $C(7, 3)$ 种不同的方式放在 7 个位置中，剩下 4 个空位。然后可以用 $C(4, 2)$ 种方式放 2 个 C，留下 2 个空位。又可以用 $C(2, 1)$ 种方式放 U，留下 1 个空位。因此，放 E 只有 $C(1, 1)$ 种方式。从而，由乘积法则，产生的不同的串数是

$$\begin{aligned} C(7, 3)C(4, 2)C(2, 1)C(1, 1) &= \frac{7!}{3!4!} \cdot \frac{4!}{2!2!} \cdot \frac{2!}{1!1!} \cdot \frac{1!}{1!0!} \\ &= \frac{7!}{3!2!1!1!} \\ &= 420 \end{aligned}$$

使用和例 7 同样的推理，能够证明定理 3。

定理 3 设类型 1 的相同的物体有 n_1 个，类型 2 的相同的物体有 n_2 个，…，类型 k 的相同的物体有 n_k 个，那么 n 个物体的不同排列数是

$$\frac{n!}{n_1!n_2!\cdots n_k!}$$

证 为确定排列数，首先注意到可以用 $C(n, n_1)$ 种方式在 n 个位置中放类型 1 的 n_1 个物体，剩下 $n - n_1$ 个空位。然后用 $C(n - n_1, n_2)$ 种方式放类型 2 的物体，剩下 $n - n_1 - n_2$ 个空位。继续放类型 3 的物体，…，类型 $k - 1$ 的物体，直到最后可用 $C(n - n_1 - n_2 - \cdots - n_{k-1}, n_k)$ 种方式放类型 k 的物体。因此由乘积法则，不同排列的总数是

$$\begin{aligned} &C(n, n_1)C(n - n_1, n_2)\cdots C(n - n_1 - \cdots - n_{k-1}, n_k) \\ &= \frac{n!}{n_1!(n - n_1)!} \frac{(n - n_1)!}{n_2!(n - n_1 - n_2)!} \cdots \frac{(n - n_1 - \cdots - n_{k-1})!}{n_k!0!} \\ &= \frac{n!}{n_1!n_2!\cdots n_k!} \end{aligned}$$

3.5.5 把物体放入盒子

许多计数问题都可以通过枚举把不同物体放入不同盒子的方式数来解决(这些被放入盒子的物体的次序是无关紧要的)。这些物体既可以是可辨别的，即每个都是不同的，也可以是不可辨别的，即认为每个都是相同的。可辨别的物体有时称为是有标号的，而不可辨别的物体则称为是没有标号的。类似地，盒子也可以是可辨别的，即每个盒子都不同，也可以是不可辨别的，即每个都相同。可辨别的盒子通常称为是有标号的，而不可辨别的盒子则称为是没有标号的。当利用把物体放入盒子的模型来解决计数问题时，需要确定物体是不是有标号的，盒子是不是有标号的。尽管从计数问题的内容中可以明确地做出决定，但计数问题有时是不明确的，这

使我们很难确定究竟使用哪个模型。在这种情况下,最好的办法就是说明一下你做了什么样的假定,并解释为什么你所选择的模型与你所做的假定是不相违背的。

Extra Examples

我们将会看到,计算把物体放入可辨别的盒子的方式数,不管物体是不是可辨别的,这种计数问题都有封闭的计算公式;然而不幸的是,如果要计算把物体放入不可辨别的盒子里的方式数时,不管物体是不是可辨别的,这种计数问题都没有封闭的计算公式。

可辨别的物体与可辨别的盒子 首先考虑把可辨别的物体放入可辨别的盒子时的情况。考虑例8,在该例子中,物体就是扑克牌,盒子就是选手的手。

例8 有多少种方式把52张标准的扑克牌发给4个人使得每个人5张牌?

解 我们将使用乘法法则求解这个问题。开始,第一个人得到5张牌可以有 $C(52, 5)$ 种方式。第二个人得到5张牌可以有 $C(47, 5)$ 种方式,因为只剩下47张牌。第三个人得到5张牌可以有 $C(42, 5)$ 种方式。最后,第四个人得到5张牌可以有 $C(37, 5)$ 种方式。因此,发给4个人每人5张牌的方式总数是

$$\begin{aligned} C(52, 5)C(47, 5)C(42, 5)C(37, 5) &= \frac{52!}{47!5!} \cdot \frac{47!}{42!5!} \cdot \frac{42!}{37!5!} \cdot \frac{37!}{32!5!} \\ &= \frac{52!}{5!5!5!5!32!} \end{aligned}$$

注意 例8的解等于52个物体的排列数,这些物体分成5个不同的类,其中四类,每类有5个相同的物体,第五类有32个物体。可以通过在这种排列和给人发牌之间定义一个一一对应来说明这个等式。为定义这个对应,首先把牌从1到52排序。然后将发给第一个人的牌与分配给第一类物体在排列中的位置对应。类似地,发给第二、第三和第四个人的牌分别与第二、第三、第四类物体所分配的位置对应。没有发给任何人的牌与第五类物体所分配的位置对应。读者应该能够验证这是一个一一对应。

例8是涉及把不同的物体分配到不同的盒子的一个典型的问题。这些不同的物体是52张牌,5个不同的盒子是4个人的手和其余的牌。可以使用下面的定理求解把不同的物体分配到不同的盒子的计数问题。

定理4 把 n 个不同的物体分配到 k 个不同的盒子使得 n_i 个物体放入盒子 i ($i=1, 2, \dots, k$)的方式数等于

$$\frac{n!}{n_1!n_2!\cdots n_k!}$$

定理4可以使用乘法法则证明。详细证明见本节的练习24。它也可以通过在定理3计数的排列和定理4计数的放物体的方法之间建立一一对应来给出证明。

不可辨别的物体与可辨别的盒子 计算将 n 个不可辨别的物体放入 k 个可辨别的盒子的方法数问题,其结果等价于在允许重复计数的情况下,对具有 k 个元素的集合计算 n 组合数的问题。其原因是:在允许重复计数的情况下,具有 k 个元素集合的 n 组合数与将 n 个不可辨别的球放入 k 个可辨别的盒子的方法数之间存在一个一一对应的关系。为了建立这种对应关系,每次将一个球放入第 i 个盒子,则对应于集合中的第 i 个元素被纳入了 n 组合数。

例9 将10个不可辨别的球放入8个可辨别的桶里,共有多少种方法?

解 将10个不可辨别的球放入8个可辨别的桶里的方法数等于在允许重复计数的情况下,从具有8个元素的集合中取出的10组合的个数。因此有

$$C(8+10-1, 10) = C(17, 10) = \frac{17!}{10!7!} = 19\,448$$

这意味着有 $C(n+r-1, n-1)$ 种方法将 r 个不可辨别的球放入 n 个可辨别的盒子。

可辨别的物体与不可辨别的盒子 计算将 n 个可辨别的物体放入 k 个不可辨别的盒子的方式数问题,要比计算将物体(不管物体是不是可辨别的)放入可辨别的盒子的方法数问题都困难。我们

将用一个例子来说明这一点。

例 10 将 4 个不同的雇员安排在 3 间不可辨别的办公室，有多少种方式？其中每间办公室可以安排任意个数的雇员。

解 我们将通过枚举雇员安排在办公室的所有方式来求解该问题。设 A, B, C, D 分别代表 4 个雇员。首先注意：可以把 4 个雇员都安排在同一间办公室；也可以将 3 个雇员安排在同一间办公室，第 4 个雇员安排在另一间办公室；也可以将 2 个雇员安排在同一间办公室，另外 2 个雇员安排在另一间办公室；最后，还可以将 2 个雇员安排在同一间办公室，而另外 2 个雇员各安排一间不同的办公室。上述每一种安排方式都可以用把 A, B, C, D 分成不相交的子集的方式来表示。

恰好有一种方式将所有 4 个雇员都安排在同一间办公室，用 $\{\{A, B, C, D\}\}$ 来表示。恰好有 4 种方式将 3 个雇员安排在同一间办公室，而第 4 个雇员安排在另一间不同的办公室，用 $\{\{A, B, C\}, \{D\}\}, \{\{A, B, D\}, \{C\}\}, \{\{A, C, D\}, \{B\}\}$ 和 $\{\{B, C, D\}, \{A\}\}$ 来表示。恰好有 3 种方式将 2 个雇员安排在同一间办公室，另外 2 个雇员安排在另一间办公室，用 $\{\{A, B\}, \{C, D\}\}, \{\{A, C\}, \{B, D\}\}$ 和 $\{\{A, D\}, \{B, C\}\}$ 来表示。最后，有 6 种方式将 2 个雇员安排在同一间办公室，而另外 2 个雇员各安排一间不同的办公室。分别用 $\{\{A, B\}, \{C\}, \{D\}\}, \{\{A, C\}, \{B\}, \{D\}\}, \{\{A, D\}, \{B\}, \{C\}\}, \{\{B, C\}, \{A\}, \{D\}\}, \{\{B, D\}, \{A\}, \{C\}\}$ 和 $\{\{C, D\}, \{A\}, \{B\}\}$ 来表示。

计算所有的可能性，得到：共有 14 种方式将 4 个不同的雇员安排在 3 间不可辨别的办公室。思考这个问题的另外一种方法是：将要安排的办公室数是多少。注意：将 4 个不同雇员安排在 3 间不可辨别的办公室（没有空办公室）共有 6 种方式，将 4 个不同雇员安排在两间不可辨别的办公室（有一间空办公室）共有 7 种方式，将 4 个不同雇员全安排在同一间办公室共有 1 种方式。 ■

关于计算把 n 个可辨别的物体放入 j 个不可辨别的盒子数问题，我们没有一个简单可用的封闭公式。但是，却有一个相当复杂的计算公式，下面将给出这个公式，如果读者愿意，可以忽略该公式的讨论部分。设 $S(n, j)$ 表示将 n 个可辨别的物体放入 j 个不可辨别的盒子的方式数，其中不允许有空的盒子。数 $S(n, j)$ 称为第二类斯特灵数。例如，例 10 证明了 $S(4, 3) = 6$ ， $S(4, 2) = 7$ 和 $S(4, 1) = 1$ 。我们看到将 n 个可辨别的物体放入 k 个不可辨别的盒子（其中非空的盒子数等于 $k, k-1, \dots, 2$ ，或 1）的方式数等于

$$\sum_{j=1}^k S(n, j)$$

例如，跟踪例 10 的推理过程，将 4 个不同雇员安排在 3 间不可辨别的办公室共有

$$S(4, 1) + S(4, 2) + S(4, 3) = 1 + 7 + 6 = 14$$

种方式。利用容斥原理（见 4.6 节）可以证明：

$$S(n, j) = \frac{1}{j!} \sum_{i=0}^{j-1} (-1)^i \binom{j}{i} (j-i)^n$$

因此，将 n 个不可辨别的物体放入 k 个不可辨别的盒子的方法数等于

$$\sum_{j=1}^k S(n, j) = \sum_{j=1}^k \frac{1}{j!} \sum_{i=0}^{j-1} (-1)^i \binom{j}{i} (j-i)^n$$

（关于第二类斯特灵数的详细信息，可以参考组合数学教材，如 [B607]、[Br99]、[RoTe05] 以及 [MiRo91] 中的第 6 章。）

不可辨别的物体与不可辨别的盒子 有些计数问题可以通过确定将不可辨别的物体放入不可辨别的盒子的方式数而得解决。用一个例子来说明这一原理。

例 11 将同一本书的 6 个副本放到 4 个相同的盒子里，其中每个盒子都能容纳 6 个副本，有多少种不同的方式？

解 我们来枚举所有的放入方式。对每一种放入方式，将按照具有最多副本数的盒子的次序依次列出每个盒子中的副本数，即列出的次序是递减的。那么，放入方式有

6,
 5, 1
 4, 2
 4, 1, 1
 3, 3
 3, 2, 1
 3, 1, 1, 1
 2, 2, 2
 2, 2, 1, 1

例如, 4, 1, 1 表示: 有一个盒子中有 4 份副本、第二个盒子中有 1 份副本、第三个盒子中有 1 份副本(第四个盒子是空的)。因为已经枚举了将 6 个副本放到最多 4 个盒子里的所有方式, 我们知道, 共有 9 种方式来完成这项任务。

将 n 个不可辨别的物体放入 k 个不可辨别的盒子, 等价于将 n 写成最多 k 个非递增正整数的和。如果 $a_1 + a_2 + \cdots + a_j = n$, 其中 $a_1, a_2, \cdots, a_j$ 都是正整数, 且 $a_1 \geq a_2 \geq \cdots \geq a_j$, 那么就说 $a_1, a_2, \cdots, a_j$ 是将正整数 n 划分成 j 个正整数的一个划分。可以看到, 如果 $p_k(n)$ 是将正整数 n 划分成最多 k 个正整数的方式数的话, 那么将 n 个不可辨别的物体放入 k 个不可辨别的盒子里的方式数就是 $p_k(n)$ 。关于这个数, 我们没有更简单的公式来表示它了。

练习

1. 从一个 3 元素集合中允许重复地有序选取 5 个元素有多少种不同的方式?
2. 6 个字母的字符串有多少个?
3. 分配 3 种工作给 5 个雇员, 如果每个雇员可以得到 1 种以上的工作, 那么有多少种不同的分配方式?
4. 从一个 5 元素集合中允许重复地无序选取 3 个元素有多少种不同的方式?
5. 一个百吉饼店有洋葱百吉饼、罂粟子百吉饼、鸡蛋百吉饼、咸味百吉饼、粗制燕麦百吉饼、芝麻百吉饼、葡萄干百吉饼和普通百吉饼, 有多少种方式选择
 - a) 6 个百吉饼?
 - b) 12 个百吉饼?
 - c) 24 个百吉饼?
 - d) 12 个百吉饼, 并且每类至少有 1 个?
 - e) 12 个百吉饼, 并且至少有 3 个鸡蛋百吉饼和不超过 2 个咸味百吉饼?
6. 一个小猪储钱罐包含 100 个相同的 1 美分和 80 个相同的 5 美分硬币, 从中选 8 个硬币有多少种方式?
7. 一个出版商有 3000 本离散数学书, 如果这些书是没有区别的, 那么将这些书存储在 3 个库房有多少种方式?
8. 方程 $x_1 + x_2 + x_3 + x_4 + x_5 = 21$ 有多少个解? 其中 $x_i (i=1, 2, 3, 4, 5)$ 是非负整数, 并且使得
 - a) $x_1 \geq 1$
 - b) $x_i \geq 2, i=1, 2, 3, 4, 5$
 - c) $0 \leq x_1 \leq 10$
 - d) $0 \leq x_1 \leq 3, 1 \leq x_2 < 4, x_3 \geq 15$
9. 有多少 10 位三进制数字(0, 1 或 2)串恰含有 2 个 0、3 个 1 和 5 个 2?
10. 假设一个大家庭有 14 个孩子, 包括 2 组三胞胎、3 组双胞胎以及 2 个单胞胎。这些孩子坐在一排椅子上, 如果相同的三胞胎或双胞胎的孩子不能互相区分, 那么有多少种方式?
11. 把 6 个相同的球放到 9 个不同的箱子里有多少种方法?
12. 把 12 个不同的物体放到 6 个不同的盒子里并且每个盒子有 2 个物体, 有多少种方法?
13. 有多少个小于 1 000 000 的正整数其数字之和等于 19?
14. 一次离散数学的期末考试有 10 道题。如果总分数是 100 且每道题至少 5 分, 那么有多少种方式来分配这些题的分数?
15. 如果被传送的二进制串必须以 1 开始, 必须有另外 3 位 1(使得传送的 1 共有 4 位), 必须包含总共 12 位 0, 必须每个 1 后面至少跟随 2 个 0, 那么有多少个不同的二进制串?
16. 使用 ABRACADABRA 中的所有字母可以构造多少个不同的串?

17. 使用 ORONO 中的某些或全部字母可以构造多少个不同的串?
18. 用 EVERGREEN 中的字母可以构造多少个至少含 7 个字符的串?
19. 一个学生有 3 个芒果、2 个番木瓜和 2 个猕猴桃。如果这个学生每天吃 1 个水果, 并且只考虑水果的类型, 那么有多少种不同的方式吃完这些水果?
20. 有多少种不同的方式在 xyz 空间上从原点 $(0, 0, 0)$ 到达 $(4, 3, 5)$ 点? 这个旅行的每一步是在 x 正方向移动一个单位, y 正方向移动一个单位, 或者 z 正方向移动一个单位。 (x, y, z) 负方向的移动是禁止的, 即不允许回头。
21. 把一副标准的 52 张扑克牌发给 5 个人, 每个人得到 7 张牌, 有多少种方式?
22. 当把一副标准的 52 张牌发给 4 个人时, 若使得每个人有一手包含 1 个 A 的牌, 这种概率是多少?
23. n 本书放在 k 个不同的书架上有多少种方式?
- 如果这些书是同一种书。
 - 如果所有的书都不同, 并且考虑这些书在书架上的位置。
- *24. 通过先把物体放入第一个盒子, 然后把物体放入第二个盒子, $\dots$, 的方法, 使用乘法法则证明定理 4。
- *25. 在这个练习中, 我们将通过在两个集合之间建立一一对应来证明定理 2。这两个集合分别是集合 $S = \{1, 2, \dots, n\}$ 的允许重复的 r 组合的集合和集合 $T = \{1, 2, 3, \dots, n+r-1\}$ 的 r 组合的集合。
- 把 S 的允许重复的 r 组合中的元素排成一个递增序列 $x_1 \leq x_2 \leq \dots \leq x_r$ 。证明: 对这个序列的第 k 项加上 $k-1$ 而构成的序列是严格递增的。断言这个序列由 T 的 r 个不同的元素构成。
 - 证明 a) 所描述的过程在 S 的允许重复的 r 组合的集合与 T 的 r 组合的集合之间定义了一一对应。[提示: 通过把 T 的满足 $1 \leq x_1 < x_2 < \dots < x_r \leq n+r-1$ 的 r 组合 $\{x_1, x_2, \dots, x_r\}$, 与从第 k 个元素减去 $k-1$ 得到的 S 的允许重复的 r 组合相联系, 证明这个对应是可逆的。]
 - 断言存在着 $C(n+r-1, r)$ 个 n 元素集合的允许重复的 r 组合。
26. 有多少种不同的方式将 6 个可辨别的物体放入 4 个不可辨别的盒子, 使得每个盒子里至少有 1 个物体?
27. 有多少种不同的方式将 6 个临时雇员安排到 4 个相同的办公室, 使得每个办公室中至少有 1 个临时雇员?
28. 有多少种不同的方式将 6 个不可辨别的物体放入 4 个不可辨别的盒子, 使得每个盒子里至少有 1 个物体?
29. 有多少种不同的方式将 9 张相同的 DVD 放入 3 个不可辨别的盒子, 使得每个盒子里至少有 2 张 DVD?
30. 有多少种不同的方式将 5 个球放到 3 个盒子里, 要求每个盒子里至少有 1 个球, 如果
- 球与盒都是有标号的?
 - 球是有标号的, 但盒子是没有标号的?
 - 球是没有标号的, 但盒子是有标号的?
 - 球与盒都是没有标号的?
- *31. 假如一个武器巡视员必须对五个不同场所中的每个场所巡视两次, 每天巡视一个场所。巡视员可以自由选择巡视场所的次序, 但他不能连着两天都巡视 X 场所, 因为 X 场所是最可疑的场所。那么, 该巡视员有多少种不同的方式来巡视这些场所?
- *32. 证明多项式定理: 如果 n 是正整数, 则

$$(x_1 + x_2 + \dots + x_m)^n = \sum_{n_1 + n_2 + \dots + n_m = n} C(n; n_1, n_2, \dots, n_m) x_1^{n_1} x_2^{n_2} \dots x_m^{n_m}$$

其中

$$C(n; n_1, n_2, \dots, n_m) = \frac{n!}{n_1! n_2! \dots n_m!}$$

是多项式系数。

33. 求 $(x+y+z)^{10}$ 中的 $x^3 y^2 z^5$ 的系数。

3.6 生成排列和组合

3.6.1 引言

本章前几节已经描述了各种类型的排列和组合的计数方法, 但是有时候需要生成排列和组合, 而不仅仅是计数。考虑下面三个问题。第一, 假设一个销售商必须访问 6 个城市。应该按照什么顺序访问这些城市而使得总的旅行时间最少? 一种方法就是确定 $6! = 720$ 种不同顺序的访问时间并且选择具有最小旅行时间的访问顺序。第二, 假定 6 个数的集合中某些数的和是 100。找出这些数的一种方法就是生成所有 $2^6 = 64$ 个子集并且检查它们的元素和。第三, 假设一个实验室有 95 个雇员, 一个项目需要一组 12 人组成的有 25 种特定技能的雇员。(每个雇员可能有一种或多种技

能)。找出这组雇员的一种方法就是找出所有的12个雇员的小组,然后检查他们是否有所需要的技能。这些例子都说明为了解问题常常需要生成排列和组合。

3.6.2 生成排列

任何 n 元素集合可以与集合 $\{1, 2, 3, \dots, n\}$ 建立一一对应。我们可以如下列出任何 n 元素集合的所有排列:生成 n 个最小正整数的排列,然后用对应的元素替换这些整数。已经建立了许多不同的算法来生成这个集合的 $n!$ 个排列。我们将要描述的算法是以 $\{1, 2, 3, \dots, n\}$ 的排列集合上的字典顺序为基础的。按照这个顺序,如果对于某个 k , $1 \leq k \leq n$, $a_1 = b_1, a_2 = b_2, \dots, a_{k-1} = b_{k-1}$ 并且 $a_k < b_k$, 那么排列 $a_1 a_2 \dots a_n$ 在排列 $b_1 b_2 \dots b_n$ 的前边。换句话说,如果在 n 个最小正整数集合的两个排列不等的第一位置,一个排列的数小于第二个排列的数,那么这个排列按照字典顺序排在第二个排列的前边。

例1 集合 $\{1, 2, 3, 4, 5\}$ 的排列 23415 在排列 23514 的前边,因为这些排列在前两位相同,但第一排列在第三位中的数是4,小于第二排列在第三位中的数5。类似地,排列 41532 在排列 52143 的前边。

生成 $\{1, 2, \dots, n\}$ 的排列的算法基础是从一个给定排列 $a_1 a_2 \dots a_n$ 按照字典顺序构造下一个排列的过程。我们将说明怎样做到这一点。首先假设 $a_{n-1} < a_n$, 交换 a_{n-1} 和 a_n 可得到一个更大的排列。没有其他的排列既大于原来的排列并且又小于这个通过交换 a_{n-1} 与 a_n 得到的排列。例如,在 234156 后面的下一个最大的排列是 234165。另一方面,如果 $a_{n-1} > a_n$, 那么由交换这个排列中的最后两项不可能得到一个更大的排列。看看排列中的最后3个整数,如果 $a_{n-2} < a_{n-1}$, 那么可以重新安排这后3个数而得到下一个最大的排列。 a_{n-1} 和 a_n 中比较小的数大于 a_{n-2} , 先把这个数放在位置 $n-2$, 然后把剩下的那个数和 a_{n-2} 按照递增的顺序放到最后两个位置。例如,在 234165 后面的下一个最大的排列是 234516。

另一方面,如果 $a_{n-2} > a_{n-1}$ (且 $a_{n-1} > a_n$), 那么不可能由安排在这个排列的最后三项而得到更大的排列。基于这个观察,可以描述一个一般的方法,对于给定的排列 $a_1 a_2 \dots a_n$ 依据字典顺序来生成下一个最大的排列。首先,找到整数 a_j 和 a_{j+1} 使得 $a_j < a_{j+1}$ 且

$$a_{j+1} > a_{j+2} > \dots > a_n$$

即在这个排列中的最后一对相邻的整数,使得这个对的第一个整数小于第二个整数。然后,把 $a_{j+1}, a_{j+2}, \dots, a_n$ 中大于 a_j 的最小的整数放到第 j 个位置,再按照递增顺序从位置 $j+1$ 到 n 列出 $a_j, a_{j+1}, a_{j+2}, \dots, a_n$ 中其余的整数,这就得到依照字典顺序的下一个最大的排列。容易看出,没有其他的排列大于排列 $a_1 a_2 \dots a_n$ 而小于这个新生成的排列。(对这一事实的验证留给读者作为练习)。

例2 在 362541 后面按照字典顺序下一个最大的排列是什么?

解 使得 $a_j < a_{j+1}$ 的最后一对整数 a_j 和 a_{j+1} 是 $a_3 = 2$ 和 $a_4 = 5$ 。排列在 2 右边大于 2 的最小整数是 $a_5 = 4$, 因此 4 被放在第三位置。然后整数 2, 5 和 1 依递增顺序放到最后 3 个位置,即这个排列的最后 3 个位置是 125。于是,下一个最大排列是 364125。

为生成整数 $1, 2, 3, \dots, n$ 的 $n!$ 个排列,按照字典顺序由最小的排列,即 $123 \dots n$ 开始,连续施用 $n! - 1$ 次生成下一个最大排列的过程,就得到 n 个最小的整数按字典顺序的所有排列。

例3 按字典顺序生成整数 $1, 2, 3$ 的排列。

解 从 123 开始,由交换 3 和 2 得到下一个排列 132。下一步,因为 $3 > 2$ 和 $1 < 3$, 排列在 132 中的 3 个整数,把 3 和 2 中较小的放到第一个位置,然后按递增顺序把 1 和 3 放到位置 2 和 3 而得到 213。跟着 213 的是 231, 它是由交换 1 和 3 得到的,因为 $1 < 3$ 。下一个最大的排列把 3 放在第一位置,后面是 1 和 2 按递增顺序排列,即 312。最后,交换 1 和 2 得到最后一个排列 321。

算法 1 显示了在给定排列不是最大的排列 $n n-1 n-2 \dots 2 1$ 时,在它的后面按照字典顺序找到下一个最大排列的过程。

算法 1 按字典顺序生成下一个最大排列

procedure *next permutation* ($a_1 a_2 \cdots a_n$: $\{1, 2, \dots, n\}$ 的排列, 不等于 $n n-1 \cdots 2 1$)

$j := n-1$

while $a_j > a_{j+1}$

$j := j-1$

{ j 是使得 $a_j < a_{j+1}$ 的最大下标}

$k := n$

while $a_j > a_k$

$k := k-1$

{ a_k 是在 a_j 右边大于 a_j 的最小整数}

交换 a_j 和 a_k

$r := n$

$s := j+1$

while $r > s$

begin

交换 a_r 和 a_s

$r := r-1$

$s := s+1$

end

{这把在第 j 位后边的排列尾部按递增顺序放置}

3.6.3 生成组合



怎样可以生成一个有穷集的元素的所有组合呢? 由于一个组合仅仅就是一个子集, 我们可以利用在 $\{a_1, a_2, \dots, a_n\}$ 和 n 位二进制串之间的对应。

如果 a_k 在子集中, 对应的二进制串在位置 k 有一个 1; 如果 a_k 不在子集中, 对应的二进制串在位置 k 有一个 0。如果可以列出所有的 n 位二进制串, 那么通过在子集和二进制串之间的对应就可以列出所有的子集。

一个 n 位二进制串也是一个在 0 到 2^n-1 之间的整数的二进制展开式。按照它们的二进制展开式, 作为整数根据递增顺序可以列出这 2^n-1 个二进制串。为生成所有的 n 位二进制展开式, 从具有 n 个 0 的二进制串 $000\cdots 00$ 开始。然后, 继续找下一个最大的展开式, 直到得到 $111\cdots 11$ 为止。在每一步找下一个最大的二进制展开式时, 先确定从右边起第一个不是 1 的位置。然后把这个位置右边的所有的 1 变成 0 并且将这第一个 0 (从右边数) 变成 1。

例 4 找出在 10 0010 0111 后面的下一个最大的二进制串。

解 这个串从右边数不是 1 的第 1 位是从右边起的第 4 位。把这一位变成 1 并且将它后面所有的位变成 0。这就生成了下一个最大的二进制串 10 0010 1000。

生成在 $b_{n-1}b_{n-2}\cdots b_1b_0$ 后面的下一个最大的二进制串的过程给出在算法 2 中。

算法 2 生成下一个最大的二进制串

procedure *next bit string* ($b_{n-1}b_{n-2}\cdots b_1b_0$: 不等于 $11\cdots 11$ 的二进制串)

$i := 0$

while $b_i = 1$

begin

$b_i := 0$

$i := i+1$

end

$b_i := 1$

下面将给出生成集合 $\{1, 2, 3, \dots, n\}$ 的 r 组合的算法。一个 r 组合可以表示成一个序列, 这个序列按照递增的顺序包含了这个子集中的元素。使用在这些序列上的字典顺序可以列出这些 r 组合。在 $a_1 a_2 \dots a_r$ 后面的下一个组合可以按下面的方法得到: 首先, 找到序列中使得 $a_i \neq n-r+i$ 的最后元素 a_i , 然后用 a_i+1 代替 a_i , 且对于 $j=i+1, i+2, \dots, r$ 用 $a_i+j-i+1$ 代替 a_j 。请读者证明这就按字典顺序生成了下一个最大的组合。下面的例子说明了这个过程。

例 5 找出集合 $\{1, 2, 3, 4, 5, 6\}$ 在 $\{1, 2, 5, 6\}$ 后面的下一个最大的4组合。

解 在具有 $a_1=1, a_2=2, a_3=5, a_4=6$ 的项中使得 $a_i \neq 6-4+i$ 的最后的项是 $a_2=2$ 。为得到下一个最大的4组合, 把 a_2 加1得 $a_2=3$ 。然后, 置 $a_3=3+1=4$ 且 $a_4=3+2=5$ 。从而下一个最大的4组合是 $\{1, 3, 4, 5\}$ 。 ■

算法3用伪码给出了这个过程。

算法 3 按字典顺序生成下一个 r 组合

procedure *next r-combination* ($\{a_1, a_2, \dots, a_r\} : \{1, 2, \dots, n\}$ 的满足 $a_1 < a_2 < \dots < a_r$ 的不等于 $\{n-r+1, \dots, n\}$ 的真子集)

$i := r$

while $a_i = n-r+i$

$i := i-1$

$a_i := a_i + 1$

for $j := i+1$ **to** r

$a_j := a_i + j - i$

练习

- 按照字典顺序排列下述 $\{1, 2, 3, 4, 5\}$ 的排列: 43521, 15432, 45321, 23451, 23514, 14532, 21345, 45213, 31452, 31542。
- 找出按照字典顺序跟在下面每一个排列后面的下一个最大的排列。
a) 1432 b) 54123 c) 12453 d) 45231 e) 6714235 f) 31528764
- 使用算法1按照字典顺序生成前4个正整数的24个排列。
- 使用算法3列出集合 $\{1, 2, 3, 4, 5\}$ 的所有的3组合。
- 证明: 算法3按字典顺序生成给定 r 组合后面的下一个最大的 r 组合。
- 列出 $\{1, 2, 3, 4, 5\}$ 的所有3排列。

下面的练习建立了另一个算法来生成 $\{1, 2, 3, \dots, n\}$ 的排列。这个算法是基于整数的康托尔展开。每个小于 $n!$ 的非负整数有一个唯一的康托尔展开式

$$a_1 1! + a_2 2! + \dots + a_{n-1} (n-1)!$$

其中 a_i 是一个不超过 i 的非负整数, $i=1, 2, \dots, n-1$ 。整数 $a_1, a_2, \dots, a_{n-1}$ 叫做这个整数的康托尔数字。

给定 $\{1, 2, \dots, n\}$ 的一个排列。令 a_{k-1} 是排列中在 k 后面且小于 k 的整数个数, $k=2, 3, \dots, n$ 。例如, 在排列43215中, a_1 是在2后面且小于2的整数个数, 所以 $a_1=1$ 。类似地, 对这个例子 $a_2=2, a_3=3$ 且 $a_4=0$ 。考虑从 $\{1, 2, 3, \dots, n\}$ 的排列的集合到小于 $n!$ 的非负整数的集合的函数。这个函数把一个排列映到一个非负整数, 而这个整数把以这种方式定义的 $a_1, a_2, \dots, a_{n-1}$ 作为它的康托尔数字。

*7. 证明: 这里描述的对应是 $\{1, 2, 3, \dots, n\}$ 的排列的集合与小于 $n!$ 的非负整数之间的双射。

关键术语和结果

术语

组合数学: 研究物体安排的科学。

枚举: 物体安排的计数。

树图: 由根、从根出发的分支以及从分支的某些端点出发的其他分支构成的图。

排列：集合元素的一个有序的安排。

r 排列：集合的 r 个元素的一个有序安排。

$P(n, r)$ ： n 元素集合的 r 排列数。

r 组合：集合的 r 个元素的无序选取。

$C(n, r)$ ： n 元素集合的 r 组合数。

$\binom{n}{r}$ (二项式系数)：也是 n 元素集合的 r 组合数。

组合证明：基于计数变量的证明。

帕斯卡三角形：二项式系数的一种表示，其中三角形的第 i 行包含 $\binom{i}{j}$, $j=0, 1, 2, \dots, i$ 。

结果

乘法法则：一种基本的计数技术。这个法则指出，当一个过程由两个子任务构成时，完成这个过程的方式数是完成第一个任务的方式数和完成第一个任务之后再第二个任务的方式数之积。

求和法则：一种基本的计数技术。这个法则指出，如果两个任务不能同时做，那么用这种或那种方式完成任务的总方式数是完成两种任务的方式数之和。

鸽巢原理：当比 k 多的物体放到 k 个盒子时，一定存在一个盒子包含了至少 2 个物体。

广义鸽巢原理：当 N 个物体放入 k 个盒子时，一定存在一个盒子包含了至少 $\lceil N/k \rceil$ 个物体。

$$P(n, r) = \frac{n!}{(n-r)!}$$

$$C(n, r) = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

$$\text{帕斯卡恒等式: } \binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

$$\text{二项式定理: } (x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k$$

当允许重复时，一个 n 元素集合有 n^r 个 r 排列。

当允许重复时，一个 n 元素集合有 $C(n+r-1, r)$ 个 r 组合。

如果类型 i 的不可区分的物体有 n_i 个, $i=1, 2, 3, \dots, k$, 那么 n 个物体的排列有 $n! / (n_1! n_2! \dots n_k!)$ 个。

生成集合 $\{1, 2, \dots, n\}$ 的排列的算法。

复习题

- 解释怎样用求和与乘法法则找出长度不超过 10 的二进制串的个数。
- 解释怎样找出长度不超过 10 且至少有 1 位 0 的二进制串的个数。
- 怎样用乘法法则找出从 m 元素集合到 n 元素集合的函数个数？
 - 从一个 5 元素集合到一个 10 元素集合存在多少个函数？
 - 怎样用乘法法则找出从 m 元素集合到 n 元素集合的一对一函数的个数？
 - 从一个 5 元素集合到一个 10 元素集合存在多少个一对一函数？
 - 从一个 5 元素集合到一个 10 元素集合存在多少个映上(注：满射)的函数？
- 如果首先赢 4 个球的队就能取胜，你怎样找出两个队加赛的所有可能的结果数？
- 你怎样找出以 101 开始或以 010 结束的 10 位二进制串数？
- 叙述鸽巢原理。
 - 解释怎样用鸽巢原理证明在 11 个整数中至少 2 个整数的最后一位相同？
- 叙述广义鸽巢原理。
 - 解释怎样用广义鸽巢原理证明在 91 个整数中有 10 个整数的最后一位数字相同？

8. a) 一个 n 元素集合的 r 排列和 r 组合的区别是什么?
 b) 推导一个与 n 元素集合的 r 组合数及 r 排列数有关的等式。
 c) 有多少种方式从一班 25 个学生中选 6 个学生参加一个委员会?
 d) 有多少种方式从一班 25 个学生中选 6 个学生担任委员会中不同的常务委员?
9. a) 什么是帕斯卡三角形?
 b) 在帕斯卡三角形中的一行是怎样从它的上一行产生的?
10. 什么是恒等式的组合证明? 这样的证明与代数证明有什么不同?
11. 解释怎样用组合论证证明帕斯卡恒等式。
12. a) 叙述二项式定理。
 b) 解释怎样用组合论证证明二项式定理。
 c) 求在 $(2x+5y)^{201}$ 的展开式中 $x^{100}y^{101}$ 项的系数。
13. a) 解释怎样找出与从 n 个物体允许重复地无序选取 r 个物体的方法数有关的公式。
 b) 如果同种类型的物体是不加区分的, 那么从 5 种不同类型的物体中选择 12 个物体有多少种方式?
 c) 从这 5 种不同类型的物体中选择 12 个物体, 如果第一类物体必须至少 3 个, 那么有多少种方式?
 d) 从这 5 种不同类型的物体中选择 12 个物体, 如果第一类物体不多于 4 个, 那么有多少种方式?
 e) 从这 5 种不同类型的物体中选择 12 个物体, 如果第一类物体必须至少 2 个, 但是第二类物体不超过 3 个, 那么有多少种方式?
14. a) 设 n 和 r 是正整数, 解释为什么方程 $x_1+x_2+\cdots+x_n=r$ 的解的个数等于 n 元素集合的允许重复的 r 组合数。这里的 x_i 是非负整数, $i=1, 2, 3, \cdots, n$ 。
 b) 方程 $x_1+x_2+x_3+x_4=17$ 有多少个非负整数解?
 c) b) 的方程有多少个正整数解?
15. a) n 个物体有 k 种不同的类型, 其中类型 1 有 n_1 个无区别的物体, 类型 2 有 n_2 个无区别的物体, $\cdots$, 类型 k 有 n_k 个无区别的物体, 推导一个与这些物体的排列数有关的公式。
 b) 有多少种方式来排序单词 *INDISCREETNESS* 的字母?
16. 描述一个算法来生成 n 个最小正整数集合的所有排列。
17. a) 把 52 张标准的扑克牌发给 6 个人, 每人 5 张牌, 有多少种方式?
 b) 有多少种方式把 n 个有区别的物体分配给 k 个有区别的盒子且使得第 i 个盒子含有 n_i 个物体?
18. 描述一个算法来生成 n 个最小正整数集合的所有的组合。

补充练习

1. 从 10 个不同的项中选 6 项有多少种方式?
 a) 若这些项是有序选择的并且不允许重复。
 b) 若这些项是有序选择的并且允许重复。
 c) 若这些项是无序选择的并且不允许重复。
 d) 若这些项是无序选择的并且允许重复。
2. 从 6 个不同的项中选 10 项有多少种方式?
 a) 若这些项是有序选择的并且不允许重复。
 b) 若这些项是有序选择的并且允许重复。
 c) 若这些项是无序选择的并且不允许重复。
 d) 若这些项是无序选择的并且允许重复。
3. 一个考试包含 100 个真假判断题。如果答案可以空缺, 一个学生回答这些考题可能有多少种不同的方式?
4. 有多少个 10 位二进制串以 000 开始或以 111 结束?
5. 字母表 $\{a, b, c\}$ 上有多少个 10 位字符串恰有 3 个 a 或恰有 4 个 b ?
6. 一个校园电话系统的内部电话号码由 5 个数字组成, 且第一个数字不等于 0。在这个系统中可以分配多少个不同的电话号码?
7. 一个冰激凌屋有 28 种不同味道的冰激凌、8 种不同的果汁和 12 种配料。
 a) 如果每种味道的可以不止 1 勺, 并且不考虑次序, 那么取 3 勺冰激凌放在一个盘中有多少种不同的方式?
 b) 如果一个小圣代包含 1 勺冰激凌、1 种果汁和 1 种配料, 那么有多少种不同的小圣代?
 c) 如果一个大圣代包含 3 勺冰激凌、2 种果汁和 3 种配料。其中每种味道的冰激凌可以不止 1 个并且不考虑次序, 每种果汁只能用 1 次且不考虑次序, 同时每种配料也只能用 1 次并且不考虑次序。那么有多少种不同的大圣代?

8. 有多少个小于 1000 的正整数
- a) 恰有 3 个十进制数字? b) 有奇数个十进制数字?
- c) 至少有 1 个十进制数字等于 9? d) 没有奇数个十进制数字?
- e) 有两个连续的十进制数字等于 5? f) 是回文(即正读和倒读是一样的)?
9. 当用十进制记法写出从 1 到 1000 的数时, 有多少个下面的数字被用到?
- a) 0 b) 1 c) 2 d) 9
10. 有黄道十二宫, 需要有多少人才能保证其中至少 6 个人在同一宫?
11. 一个幸运卡甜点公司制作 213 种不同的幸运卡。一个学生使用这个公司的幸运卡在餐馆吃饭。如果这个学生没有获得 4 张同样的幸运卡, 那么他在这个餐馆吃饭最多可以吃多少次?
12. 为保证至少 2 个人人生在一周的同一天和同一个月(可以不在同一年), 那么需要多少人?
13. 证明: 在 10 个不超过 50 的正整数集合中至少有 2 个不同的 5 元子集有同样的和。
14. 一包棒球卡有 20 张。如果总共有 550 种不同的卡, 那么需要买多少包卡才能保证其中的 2 张卡是一样的。
15. a) 从一副牌中需要选多少张牌才能保证至少选中 2 张 A?
- b) 从一副牌中需要选多少张牌才能保证至少选中 2 张 A 和 2 种点数?
- c) 从一副牌中需要选多少张牌才能保证至少有 2 张同样点数的牌?
- d) 从一副牌中需要选多少张牌才能保证至少有 2 张不同点数的牌?
- *16. 证明: 在任何 $n+1$ 个不超过 $2n$ 的正整数中必存在 2 个数互素。
- *17. 证明: 在 m 个整数的序列中存在若干个连续的整数其和可被 m 整除。
18. 证明: 如果放 5 个点在边长为 2 的正方形中, 那么其中至少有 2 个点的距离不超过 $\sqrt{2}$ 。
19. 证明: 一个有理数的十进制展开式一定从某一点出现循环。
20. 曾经有一种计算机病毒通过感染的邮件信息感染了一台微机, 该病毒在这台微机的邮箱中向 100 个邮件地址都发送了自身的副本。那么, 当该微机将感染的邮件信息发送 5 次之后, 它所感染的不同计算机的最大数量是多少?
21. 有多少种方式从 20 种多纳圈中选 12 个多纳圈?
- a) 如果没有 2 个多纳圈是同种的? b) 如果所有的多纳圈都是同种的?
- c) 如果不加限制? d) 如果至少有 2 种多纳圈?
- e) 如果必须至少有 6 个蓝莓馅的多纳圈? f) 如果至多有 6 个蓝莓馅的多纳圈?
22. 求 n , 如果
- a) $P(n, 2) = 110$ b) $P(n, n) = 5040$ c) $P(n, 4) = 12P(n, 2)$
23. 求 n , 如果
- a) $C(n, 2) = 45$ b) $C(n, 3) = P(n, 2)$ c) $C(n, 5) = C(n, 2)$
24. 证明: 如果 n 和 r 是非负整数且 $n \geq r$, 则
- $$P(n+1, r) = P(n, r)(n+1)/(n+1-r)$$
- *25. 设 S 是 n 元素集合, 存在多少个有序对 (A, B) 使得 A 和 B 是 S 的子集且 $A \subseteq B$? [提示: 证明 S 中的每个元素属于 A , $B-A$ 或 $S-B$.]
26. 通过构造在集合的具有偶数个元素的子集与具有奇数个元素的子集之间的对应, 给出关于 5.4 节推论 2 的组合证明。[提示: 取定集合的一个元素 a , 如下构造对应: 如果 a 不在子集中就把它放到子集中; 如果 a 在子集中就把它从子集中取出。]
27. 设 n 和 r 是非负整数且 $r < n$ 。证明
- $$C(n, r-1) = C(n+2, r+1) - 2C(n+1, r+1) + C(n, r+1)$$
28. 使用数学归纳法证明 $\sum_{j=2}^n C(j, 2) = C(n+1, 3)$, 其中 n 是大于 1 的整数。
29. 证明: 如果 n 是整数, 则
- $$\sum_{k=0}^n 3^k \binom{n}{k} = 4^n$$
30. 在这个练习中我们将推导一个关于 n 个最小正整数的平方和的公式。我们将用两种方式计数三元组 (i, j, k) 的个数, 其中 i, j 和 k 是整数且满足 $0 \leq i < k$, $0 \leq j < k$, $1 \leq k \leq n$ 。

a) 证明: 对于给定的 k 存在 k^2 个这样的三元组, 因此有 $\sum_{k=1}^n k^2$ 个这样的三元组。

b) 证明: 具有 $0 \leq i < j < k$ 的三元组个数和 $0 \leq j < i < k$ 的三元组个数都等于 $C(n+1, 3)$ 。

c) 证明: 具有 $0 \leq i = j < k$ 的三元组个数等于 $C(n+1, 2)$ 。

d) 把 a)、b) 和 c) 组合起来得出

$$\begin{aligned}\sum_{k=1}^n k^2 &= 2C(n+1, 3) + C(n+1, 2) \\ &= n(n+1)(2n+1)/6\end{aligned}$$

*31. 设 $n \geq 4$, 有多少个 n 位二进制串恰好 01 在其中出现两次?

32. 设集合 S 和子集族 $A_1, A_2, \dots, A_n$, 其中每个子集含有 d 个元素, $d \geq 2$ 。如果可以把两种不同的颜色分配给 S 的元素, 每个元素一种颜色, 且使得每个子集 A_i 都包含了两种颜色的元素, 则称这个子集族是可 2 涂色的。设 $m(d)$ 是最大的正整数, 使得对于每个子集族, 如果子集数小于 $m(d)$, 且每个子集含 d 个元素, 就是可 2 涂色的。

a) 证明: 具有 $2d-1$ 个元素的集合 S 的所有 d 子集构成的子集族不是可 2 涂色的。

b) 证明: $m(2)=3$ 。

**c) 证明: $m(3)=7$ 。[提示: 证明 $\{1, 3, 5\}, \{1, 2, 6\}, \{1, 4, 7\}, \{2, 3, 4\}, \{2, 5, 7\}, \{3, 6, 7\}, \{4, 5, 6\}$ 不是可 2 涂色的。然后证明所有具有 3 个元素的 6 个集合的集合族每个都是可 2 涂色的。]

33. 一个教授为一次离散数学考试出了 20 道多选题, 每道题可能的答案为 a, b, c 或 d 。如果具有答案 a, b, c 和 d 的试题数分别为 8, 3, 4 和 5, 且试题可以用任意的顺序安排, 那么可能有多少种不同的答案。

34. 8 个人围圆桌就座有多少种不同的安排? 其中如果一种安排通过旋转能从另一种安排得到, 那么就认为这两种安排是一样的。

35. 把 24 个学生分给 5 个指导教师有多少种方式?

36. 一蒲式耳包含 20 个无区别的 Delicious 苹果、20 个无区别的 Macintosh 苹果和 20 个无区别的 Granny Smith 苹果, 从中选 12 个苹果, 如果每类至少选 3 个有多少种方式?

37. 方程 $x_1 + x_2 + x_3 = 17$ 有多少个非负整数解?

a) 若 $x_1 > 1, x_2 > 2, x_3 > 3$

b) 若 $x_1 < 6, x_3 > 5$

c) 若 $x_1 < 4, x_2 < 3, x_3 > 5$

38. 使用单词 PEPPERCORN 的所有字母构成字符串。

a) 可以构成多少个不同的字符串?

b) 其中有多少字符串以 P 开始和结束?

c) 在多少个字符串中有 3 个连续的 P ?

39. 10 元素集合有多少个子集

a) 少于 5 个元素?

b) 多于 7 个元素?

c) 有奇数个元素?

40. 一个交通逃逸事故的证人告诉警察, 肇事汽车的车牌包含 3 个字母后面跟着 3 个数字, 以字母 AS 开始且包含数字 1 和 2。有多少不同的车牌符合这个描述?

41. 有多少种方式把 n 个相同的物体放入 m 个不同的容器而使得没有一个容器是空的?

42. 6 个男孩和 8 个女孩坐在一排椅子上, 如果没有两个男孩相邻, 有多少种方式?

43. 将 6 个物体放入 5 个盒子中有多少种方式, 如果

a) 物体与盒子都是有标号的?

b) 物体是有标号的, 但盒子是没有标号的?

c) 物体是没有标号的, 但盒子是有标号的?

d) 物体与盒子都是没有标号的?

44. 将 5 个物体放入 6 个盒子中有多少种方式, 如果

a) 物体与盒子都是有标号的?

b) 物体是有标号的, 但盒子是没有标号的?

c) 物体是没有标号的, 但盒子是有标号的?

d) 物体与盒子都是没有标号的?

45. 设计一个算法生成一个有穷集的所有允许重复的 r 排列。

46. 设计一个算法生成一个有穷集的所有允许重复的 r 组合。

*47. 证明: 如果 m, n 为整数, $m \geq 3, n \geq 3$, 那么 $R(m, n) \leq R(m, n-1) + R(m-1, n)$ 。

*48. 在一组 6 个人中, 每两个人是朋友或者是敌人, 证明在这组人中不存在 3 个人两两是朋友, 也不存在 4 个人两两是敌人。从而证明了 $R(3, 4) \geq 7$ 。

计算机题目

按下述给定的输入和输出写程序。

1. 给定正整数 n 和不超过 n 的非负整数, 找出 n 元素集合的 r 排列数和 r 组合数。
2. 给定正整数 n 和 r , 找出 n 元素集合的允许重复的 r 排列数和允许重复的 r 组合数。
3. 给定正整数列, 找出这个序列的最长的递增和递减子序列。
- *4. 给定方程 $x_1 + x_2 + \cdots + x_n = C$, 其中 C 是一个常数, $x_1, x_2, \cdots, x_n$ 是非负整数, 列出所有的解。
5. 给定正整数 n , 按字典顺序列出集合 $\{1, 2, 3, \cdots, n\}$ 的所有的排列。
6. 给定正整数 n 和不超过 n 的非负整数 r , 按字典顺序列出集合 $\{1, 2, 3, \cdots, n\}$ 的所有的 r 组合。
7. 给定正整数 n 和不超过 n 的非负整数 r , 按字典顺序列出集合 $\{1, 2, 3, \cdots, n\}$ 的所有的 r 排列。
8. 给定正整数 n , 列出集合 $\{1, 2, 3, \cdots, n\}$ 的所有的组合。
9. 给定正整数 n 和 r , 列出集合 $\{1, 2, 3, \cdots, n\}$ 的允许重复的所有 r 排列。
10. 给定正整数 n 和 r , 列出集合 $\{1, 2, 3, \cdots, n\}$ 的允许重复的所有 r 组合。

计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 当两个队加时赛时, 赢的队是 9 分中首先得 5 分、11 分中首先得 6 分、13 分中首先得 7 分和 15 分中首先得 8 分的队。找出加时赛的可能的结果数。
2. 哪些二项式系数是奇数? 你能根据数的特征给出一个猜想吗?
3. 目前还不知道二项式系数 $C(2n, n)$ 是否一定被一个素数的平方整除, 也不知道当 n 增长时在 $C(2n, n)$ 的素数分解中最大的指数是否无界增长。通过对于尽可能多的正整数 n 找 $C(2n, n)$ 的分解式中素数的最小和最大的幂来探索这个问题。
4. 尽量找出更多的小于 200 的奇数 n , 使得 $C(n, \lfloor n/2 \rfloor)$ 不能被一个素数的平方整除。根据你的证据给出一个臆测公式。
- *5. 对每个小于 100 的整数, 确定 $C(2n, n)$ 是否能被 3 整除。根据 n 的三位展开基中的数字, 你能臆测一个公式来告诉我们: 关于哪个整数 n , 二项式系数 $C(2n, n)$ 能被 3 整除吗?
6. 生成 8 元素集合的所有的排列。
7. 生成 9 元素集合的所有的 6 排列。
8. 生成 8 元素集合的所有的组合。
9. 生成 7 元素集合允许重复的所有 5 组合。

写作题目

用本书以外的资料, 按下列要求写成短文。

1. 描述狄利克雷和其他的数学家对鸽巢原理的早期应用。
2. 讨论扩充目前电话编码计划的方式以适合对更多电话号码飞速增长的需求。(看看你是否能够找到某些来自电信产业的建议。) 对你要讨论的每个新的编码计划说明怎样找到它所支持的不同电话号码的个数。
3. 本书描述了许多组合恒等式。找一找关于这种恒等式的资料, 并且描述除了本书引入之外的其他重要的组合恒等式。给出其中某些恒等式的有代表性的证明, 包括组合证明。
4. 描述在统计力学中的质点分布所使用的不同的模型, 包括麦克斯韦-玻尔兹曼、玻色-爱因斯坦和费米-狄拉克(Fermi-Dirac)统计, 在每种情况下描述模型中使用的计数技术。
5. 定义第一类斯特林数并且描述它们的某些性质以及所满足的恒等式。
6. 定义第二类斯特林数并且描述它们的某些性质以及所满足的恒等式。
7. 定义拉姆齐数, 叙述和证明显示它们存在的拉姆齐定理, 并且描述目前已知的有关拉姆齐数的结果。
8. 描述生成 n 元素集合所有排列的其他算法, 这些算法不是在 3.6 节给出的算法。把这些算法的计算复杂性与书上和 3.6 节练习所描述算法的计算复杂性进行比较。
9. 至少描述一种方法生成一个正整数 n 的所有的剖分。

高级计数技术

许多计数问题用第 3 章讨论的方法是不容易求解的。一个问题是：有多少个 n 位二进制串不包含两个连续的 0？为求解这个问题，令 a_n 是这种 n 位二进制串数，可以证明 $a_{n+1} = a_n + a_{n-1}$ 。这个等式叫做递推关系，它和初始条件 $a_1 = 2$ 和 $a_2 = 3$ 确定了序列 $\{a_n\}$ 。此外，从这个与序列的项有关的等式可以找到 a_n 的显式公式。正如我们将要看到的，可以用一种类似的技术来求解许多不同的计数问题。

我们也将看到，可以用形式幂级数（也叫做生成函数）来求解许多计数问题，其中 x 的幂的系数代表我们感兴趣的序列的项。除了求解计数问题，生成函数还可用于求解递推关系以及证明组合恒等式。

许多其他类型的计数问题不能使用第 3 章所讨论的技术求解，例如：有多少种方式把 7 项工作分给 3 个雇员而使得每个雇员至少得到一项工作？有多少个素数小于 1000？可以用计数集合并集中的元素数来求解这两个问题。我们将开发一种叫做容斥原理的技术来计数在集合并集中的元素个数，并且将说明怎样用这种技术求解计数问题。

可以用本章学到的技术与第 3 章的基本技术一起求解许多计数问题。

4.1 递推关系基础

4.1.1 引言

一群细菌的数目每小时增加一倍。如果开始有 5 个细菌，在 n 小时末将有多少个细菌？为求解这个问题，令 a_n 是 n 小时末的细菌数。因为细菌数每小时增加一倍，只要 n 是正整数，关系 $a_n = 2a_{n-1}$ 就成立。对所有的非负整数 n ，这个关系和初始条件 $a_0 = 5$ 一起唯一地确定了 a_n 。利用这一信息可找出关于 a_n 的公式。

某些计数问题不能用第 3 章给出的技术求解，但可以通过找到序列的项之间的关系，如在涉及细菌的问题中的关系，即递推关系来求解。我们将研究各种能用递推关系构造模型的计数问题。我们也将在本节和 4.2 节建立一些方法，针对满足某类递推关系的序列，求出序列的项的显式公式。

4.1.2 递推关系

一个序列的递归定义指定了一个或多个初始的项以及一个由前项确定后项的规则。可以用递归定义来求解计数问题。对于递归定义，这个从某些前项求后项的规则就叫做递推关系。

定义 1 关于序列 $\{a_n\}$ 的递推关系是一个等式，它把 a_n 用序列中在 a_n 前面的一项或多项即 $a_0, a_1, \dots, a_{n-1}$ 来表示，这里 $n \geq n_0$ ， n_0 是一个非负整数。如果一个序列的项满足递推关系，这个序列就叫做递推关系的解。

在递归和递推关系之间存在重要的联系，关于这一点我们在本章的后边再做研究。递归算法依据较小规模的同一问题的一个或者多个实例的解得到规模为 n 的问题的解。因此，当我们分析一个递归算法的复杂性时，我们得到一个递推关系，它把求解规模为 n 的问题需要的运算次数用求解一个或多个小规模实例的同一问题所需要的运算次数来表示。

例 1 令 $\{a_n\}$ 是一个序列，它满足递推关系 $a_n = a_{n-1} - a_{n-2}$ ， $n = 2, 3, 4, \dots$ ，且 $a_0 = 3$ ， $a_1 = 5$ ，那么 a_2 和 a_3 是什么？

解 从递推关系可以看出, $a_2 = a_1 - a_0 = 5 - 3 = 2$ 且 $a_3 = a_2 - a_1 = 2 - 5 = -3$ 。 ■

例2 确定序列 $\{a_n\}$ 是否为递推关系 $a_n = 2a_{n-1} - a_{n-2}$, $n=2, 3, 4, \dots$ 的解。这里 $a_n = 3n$, n 是非负整数。对 $a_n = 2^n$ 和 $a_n = 5$ 也回答同一个问题。

解 假设对每一个非负整数 n , $a_n = 3n$, 那么对于 $n \geq 2$, 可以看出 $2a_{n-1} - a_{n-2} = 2[3(n-1)] - 3(n-2) = 3n = a_n$ 。于是, $\{a_n\}$ 是该递推关系的解, 其中 $a_n = 3n$ 。

假设对每一个非负整数 n , $a_n = 2^n$ 。注意到 $a_0 = 1$, $a_1 = 2$, $a_2 = 4$ 。因为 $2a_1 - a_0 = 2 \cdot 2 - 1 = 3 \neq a_2$, 不难看出序列 $\{a_n\}$ 不是该递推关系的解, 其中 $a_n = 2^n$ 。

假设对每一个非负整数 n , $a_n = 5$, 那么对于 $n \geq 2$ 有 $a_n = 2a_{n-1} - a_{n-2} = 2 \cdot 5 - 5 = 5 = a_n$ 。因此, $\{a_n\}$, $a_n = 5$ 是该递推关系的解。 ■

序列的初始条件说明了在递推关系起作用的首项之前的那些项。例如, 例1中的 $a_0 = 3$ 和 $a_1 = 5$ 是初始条件。递推关系和初始条件唯一地确定了一个序列。这是由于一个递推关系和初始条件一起提供了这个序列的递归定义。只要使用足够多次, 序列的任何一项都可以从初始条件开始通过递推关系求出。但是对于某些特定类型的序列, 可以有更好的方法通过它的递推关系和初始条件来计算它的项。我们将在本节和4.2节讨论这些方法。

4.1.3 用递推关系构造模型

我们可以使用递推关系构造各种问题的模型, 例如找复利, 计数岛上的兔子, 确定汉诺塔难题的移动次数, 以及计数具有确定性质的二进位串。

例3 复利 假设一个人在银行的储蓄账户上存了10 000美元, 年复利是11%。那么在30年后账上将有多少钱?

解 为求解这个问题, 令 P_n 表示 n 年后账上的钱数。因为 n 年后账上的钱等于在 $n-1$ 年后账上的钱加上第 n 年的利息, 易见序列 $\{P_n\}$ 满足递推关系

$$P_n = P_{n-1} + 0.11P_{n-1} = (1.11)P_{n-1}$$

初始条件是 $P_0 = 10\,000$ 。

我们可以使用迭代法找到关于 P_n 的公式。注意

$$P_1 = (1.11)P_0$$

$$P_2 = (1.11)P_1 = (1.11)^2 P_0$$

$$P_3 = (1.11)P_2 = (1.11)^3 P_0$$

⋮

$$P_n = (1.11)P_{n-1} = (1.11)^n P_0$$

当代入初始条件 $P_0 = 10\,000$, 就得到公式 $P_n = (1.11)^n 10\,000$ 。我们可以使用数学归纳法验证它的正确性。公式对 $n=0$ 是正确的, 这是初始条件的直接结果。假定 $P_n = (1.11)^n 10\,000$, 那么由递推关系和归纳假设,

$$P_{n+1} = (1.11)P_n = (1.11)(1.11)^n 10\,000 = (1.11)^{n+1} 10\,000$$

这证明了对 P_n 的显式公式是正确的。

将 $n=30$ 代入公式 $P_n = (1.11)^n 10\,000$ 就证明了在30年后账上包含

$$P_{30} = (1.11)^{30} 10\,000 = 228\,922.97 \text{ 美元}$$

例4 说明了怎样用递推关系建立关于岛上兔子数的模型。

例4 兔子和斐波那契数 考虑下面的问题, 它是由里奥那多·底·比萨, 也就是斐波那契, 于13世纪在《算书》(*Liber abaci*)一书中提出来的。一对刚出生的兔子(一公一母)被放到岛上。每对兔子出生后两个月才开始繁殖后代。如图4-1所示, 在出生两个月以后, 每对兔子在每个月都将繁殖一对新的兔子。假定兔子不会死去, 找出 n 个月后关于岛上兔子对数的递推关系。

解 用 f_n 表示 n 个月后的兔子对数。我们将证明 f_n , $n=1, 2, 3, \dots$ 是斐波那契序列的项。

可以用递推关系建立兔子数的模型。在第1个月末, 岛上的兔子对数是 $f_1 = 1$ 。由于这对兔子

新生的对数 (至少两个月大)	已有的对数 (比两个月小)	月	新生的 对数	已有的 对数	总对数
		1	0	1	1
		2	0	1	1
		3	1	1	2
	 	4	1	2	3
 	  	5	2	3	5
  	   	6	3	5	8

图 4-1 岛上的兔子

在第2个月没有繁殖, 因此 $f_2=1$ 。为找到 n 个月后的兔子对数, 要把前一个月岛上的对数 f_{n-1} 加上新生的对数, 而这个数等于 f_{n-2} , 因为每对两个月大的兔子都生出一对新兔子。

因此, 序列 $\{f_n\}$ 满足递推关系

$$f_n = f_{n-1} + f_{n-2}, \quad n \geq 3$$

和初始条件 $f_1=1$ 和 $f_2=1$ 。由于这个递推关系和初始条件唯一地确定了这个序列, 因此 n 个月后岛上的兔子对数由第 n 个斐波那契数给出。 ■

例 5 涉及一个著名的难题。

例 5 汉诺塔 19 世纪后期由法国数学家埃德沃德·卢卡斯发明的一个流行的游戏叫做汉诺塔, 它由安装在一个板上的 3 根柱子和若干大小不同的盘子构成。开始时, 这些盘子按照大小的次序放在第一根柱子上, 使得大盘子在底下(如图 4-2 所示)。游戏的规则是: 每一次把 1 个盘子从一根柱子移动到另一根柱子, 但是不允许这个盘子放在比它小的盘子上面。游戏的目标是把所有的盘子按照大小的次序都放到第二根柱子上, 并且将最大的盘子放在底部。

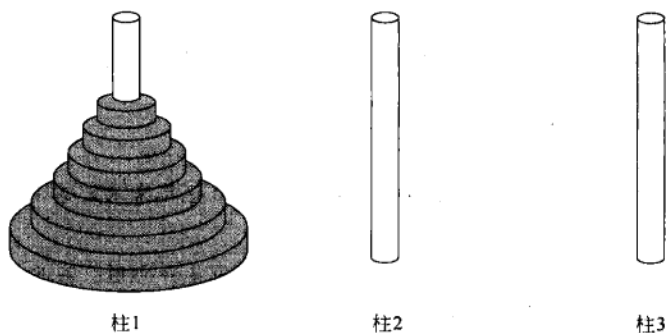


图 4-2 汉诺塔的初始位置

令 H_n 表示解 n 个盘子的汉诺塔问题所需要的移动次数。建立一个关于序列 $\{H_n\}$ 的递推关系。

解 开始 n 个盘子在柱 1。按照游戏规则我们可以用 H_{n-1} 次移动将上边的 $n-1$ 个盘子移到柱 3(图 4-3 说明了此刻的柱子和盘子)。在这些移动中保留最大的盘子不动。然后, 我们用一次移动将最大的盘子移到第二根柱子上。我们可以再使用 H_{n-1} 次移动将柱 3 上的 $n-1$ 个盘子移到柱 2。

把它们放到最大的盘子上面, 这个最大的盘子一直放在柱2的底部。容易看出, 使用更少的步数是不可能求解这个难题的。这就证明了

$$H_n = 2H_{n-1} + 1$$

初始条件是 $H_1 = 1$, 因为依照规则一个盘子可以用1次移动从柱1移到柱2。

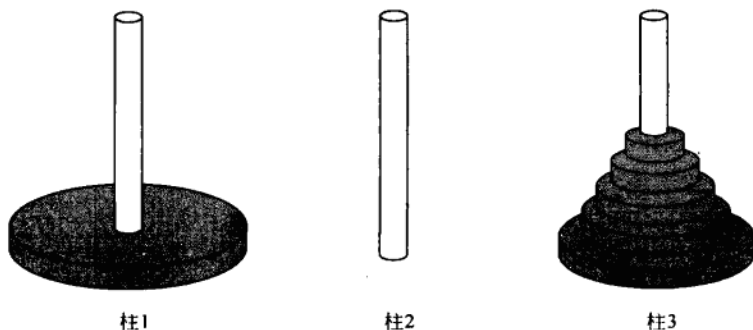


图 4-3 汉诺塔的一个中间位置

我们可以使用迭代方法求解这个递推关系。

$$\begin{aligned} H_n &= 2H_{n-1} + 1 \\ &= 2(2H_{n-2} + 1) + 1 = 2^2 H_{n-2} + 2 + 1 \\ &= 2^2(2H_{n-3} + 1) + 2 + 1 = 2^3 H_{n-3} + 2^2 + 2 + 1 \\ &\vdots \\ &= 2^{n-1} H_1 + 2^{n-2} + 2^{n-3} + \cdots + 2 + 1 \\ &= 2^{n-1} + 2^{n-2} + \cdots + 2 + 1 \\ &= 2^n - 1 \end{aligned}$$

为了用序列前面的项表示 H_n , 我们重复地用到这个递推关系。在倒数第二个等式中用了初始条件 $H_1 = 1$ 。最后一个等式是基于等比序列的求和公式得出的, 这个公式可以在 2.4 节的定理 1 中找到。

用迭代方法找出了具有初始条件 $H_1 = 1$ 的递推关系 $H_n = 2H_{n-1} + 1$ 的解。这个公式可以用数学归纳法证明。证明留给读者作为节后的练习。

关于这个难题有一个古老的传说, 在汉诺有一座塔, 那里的僧侣按照这个游戏的规则从一个柱子到另一个柱子移动 64 个金盘子, 据说当他们结束游戏时世界就到了末日。如果这些僧侣 1 秒钟移动 1 个盘子, 这个世界将在他们开始多久以后终结?

根据这个显式公式, 僧侣需要

$$2^{64} - 1 = 18\,446\,744\,073\,709\,551\,615$$

次移动来搬这些盘子。每次移动需要 1 秒钟, 他们将用 5000 亿年来求解这个难题, 因此这个世界的寿命应该相当长。 ■

注意 许多人研究了源自例 5 所述汉诺塔难题的各种问题。某些问题用到更多的柱子, 某些问题允许同样大小的盘子, 某些问题对盘子的移动类型加以限制。一个最古老和最有趣的问题是雷夫难题^①, 它是 1907 年由亨利·杜德尼在他的《坎特伯雷谜题》(Canterbury Puzzle)一书中提出来的。这个难题是雷夫出的, 他让一个朝圣者把一堆各种大小的奶酪从 4 个凳子中的一个移到另一个, 移动中不允许把直径较大的奶酪放在较小的奶酪上面。如果用柱子和盘子的概念来表述雷夫难题, 除了使用 4 根柱子之外,

① 雷夫, Reve, 更常见的拼写是 reeve, 这个词在古代是指地方长官(governor)。

其他和汉诺塔的规则一样。你可能会奇怪没有人能够确定求解 n 个盘子的雷夫难题所需要的最少移动次数。但是, 存在一个猜想, 至今已经超过 50 年了。这个猜想认为所需移动的最少次数等于由弗雷姆(Frame)和斯图尔特(Stewart)在 1939 年发明的算法所使用的移动次数。(更详细的信息参见节末的练习 27~30 和 [St94]。)

例 6 说明了怎样用递推关系计数具有指定长度和某种性质的二进制串。

例 6 对于不含 2 个连续 0 的 n 位二进制串的个数找出递推关系和初始条件。有多少个这样的 5 位二进制串?

解 设 a_n 表示不含 2 个连续 0 的 n 位二进制串数。为得到一个关于 $\{a_n\}$ 的递推关系, 由求和法则, 不含 2 个连续 0 的 n 位二进制串的个数等于以 0 结尾的这种二进制串数加上以 1 结尾的这种二进制串数。我们将假定 $n \geq 3$, 使得二进制串至少有 3 位。

精确地说, 不含 2 个连续 0 并以 1 结尾的 n 位二进制串就是在不含 2 个连续 0 的 $n-1$ 位二进制串的尾部加上一个 1。因此存在 a_{n-1} 个这样的二进制串。

不含 2 个连续 0 并以 0 结尾的 n 位二进制串在它们的 $n-1$ 位必须是 1; 否则它们就将以 2 个 0 结尾。因而, 精确地说, 不含 2 个连续 0 并以 0 结尾的 n 位二进制串就是在不含 2 个连续 0 的 $n-2$ 位二进制串的尾部加上 10。因此存在 a_{n-2} 个这样的二进制串。

如图 4-4 所示, 可以断言对于 $n \geq 3$ 有

$$a_n = a_{n-1} + a_{n-2}$$

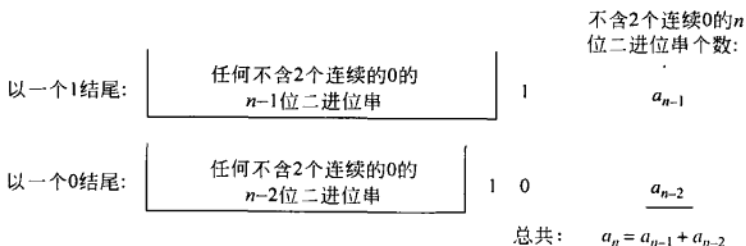


图 4-4 计数不含 2 个连续的 0 的 n 位二进制串

初始条件是 $a_1 = 2$, 因为 1 位的二进制串是 0 和 1, 没有连续的 2 个 0。而 $a_2 = 3$, 因为 2 位的二进制串中满足条件的是 01, 10 和 11。使用 3 次递推关系就可得到 a_5

$$a_3 = a_2 + a_1 = 3 + 2 = 5$$

$$a_4 = a_3 + a_2 = 5 + 3 = 8$$

$$a_5 = a_4 + a_3 = 8 + 5 = 13$$

注意 $\{a_n\}$ 和斐波那契序列满足同样的递推关系。因为 $a_1 = f_3$ 且 $a_2 = f_4$, 从而有 $a_n = f_{n+2}$ 。

例 7 说明怎样用递推关系建立编码字数的模型, 这种编码字是通过确定的正确性检测所允许的。

例 7 编码字的枚举 一个计算机系统把一个十进制数字串作为一个编码字, 如果它包含偶数个 0, 就是有效的。例如, 1 230 407 869 是有效的, 而 120 987 045 608 不是有效的。设 a_n 是有效的 n 位编码字的个数。找出一个关于 a_n 的递推关系。

解 注意到 $a_1 = 9$, 因为存在 10 个 1 位十进制数字串, 并且只有一个, 即串 0 是无效的。通过考虑怎样由 $n-1$ 位的数字串构成一个 n 位有效数字串, 就可以推导出关于这个序列的递推关系。从少 1 位数字的串构成 n 位有效数字串有两种方式。

第一种, 在一个 $n-1$ 位的有效数字串后面加上一个非 0 的数字就可以得到一个 n 位的有效数字串。加这个数字的方式有 9 种。因此用这种方法构成 n 位有效数字串的方式有 $9a_{n-1}$ 种。

第二种, 在一个无效的 $n-1$ 位数字串后面加上一个 0 就可以得到 n 位有效的数字串。(这将产生具有偶数个 0 的串, 因为无效的 $n-1$ 位数字串有奇数个 0。)这样做的方式数等于无效的 $n-1$ 位数字串的个数。因为存在 10^{n-1} 个 $n-1$ 位数字串, 其中有 a_{n-1} 个是有效的, 通过在无效的 $n-1$ 位数字串后面加上一个 0 就得到 $10^{n-1} - a_{n-1}$ 个 n 位的有效数字串。

因为所有的 n 位有效数字串都用这两种方式之一产生, 从而存在

$$\begin{aligned} a_n &= 9a_{n-1} + (10^{n-1} - a_{n-1}) \\ &= 8a_{n-1} + 10^{n-1} \end{aligned}$$

个 n 位有效数字串。

例 8 中的递推关系在许多不同的场合都可以见到。

例 8 求关于 C_n 的递推关系, 其中 C_n 是通过在 $n+1$ 个数 $x_0 \cdot x_1 \cdot x_2 \cdots x_n$ 的乘积中加括号来规定乘法的次序的方式数。例如, $C_3 = 5$, 因为对 $x_0 \cdot x_1 \cdot x_2 \cdot x_3$ 有 5 种加括号的方式来规定乘法的次序:

$$\begin{aligned} &((x_0 \cdot x_1) \cdot x_2) \cdot x_3 \quad (x_0 \cdot (x_1 \cdot x_2)) \cdot x_3 \quad (x_0 \cdot x_1) \cdot (x_2 \cdot x_3) \\ &x_0 \cdot ((x_1 \cdot x_2) \cdot x_3) \quad x_0 \cdot (x_1 \cdot (x_2 \cdot x_3)) \end{aligned}$$

解 为求关于 C_n 的递推关系, 我们注意到无论怎样在 $x_0 \cdot x_1 \cdot x_2 \cdots x_n$ 中插入括号总有一个“ $\cdot$ ”运算符留在所有括号的外边, 即执行最后一次乘法的运算符。[例如, 在 $(x_0 \cdot (x_1 \cdot x_2)) \cdot x_3$ 中的最后一个运算符, 在 $(x_0 \cdot x_1) \cdot (x_2 \cdot x_3)$ 中的第二个运算符。]这个最后的运算符出现在 $n+1$ 个数中的两个数之间, 比如说 x_k 和 x_{k+1} 之间。当最后的运算符出现在 x_k 和 x_{k+1} 之间时, 存在 $C_k C_{n-k-1}$ 种方式插入括号来确定 $n+1$ 个数被乘的次序, 因为有 C_k 种方式在乘积 $x_0 \cdot x_1 \cdots x_k$ 中插入括号来确定这 $k+1$ 个数的乘法次序, 且有 C_{n-k-1} 种方式在乘积 $x_{k+1} \cdot x_{k+2} \cdots x_n$ 中插入括号来确定这 $n-k$ 个数的乘法次序。由于这个最后的运算符可能出现在 $n+1$ 个数的任两个数之间, 所以

$$\begin{aligned} C_n &= C_0 C_{n-1} + C_1 C_{n-2} + \cdots + C_{n-2} C_1 + C_{n-1} C_0 \\ &= \sum_{k=0}^{n-1} C_k C_{n-k-1} \end{aligned}$$

注意到初始条件是 $C_0 = 1$ 和 $C_1 = 1$ 。这个递推关系可以用生成函数的方法求解, 这种方法将在 4.4 节讨论。可以证明 $C_n = C(2n, n)/(n+1)$ 。(见节末的练习 20。)

序列 $\{C_n\}$ 是卡特兰 (Catalan) 数的序列。这个序列是除上例之外的许多不同计数问题的解(细节见 [MiRo91] 或 [Ro84a] 中有关卡特兰数的一章)。

练习

1. 求给定序列的前 5 项, 每个序列由下面的递推关系和初始条件定义。

a) $a_n = 6a_{n-1}$, $a_0 = 2$

b) $a_n = a_{n-1}^2$, $a_1 = 2$

c) $a_n = a_{n-1} + 3a_{n-2}$, $a_0 = 1$, $a_1 = 2$

d) $a_n = na_{n-1} + n^2 a_{n-2}$, $a_0 = 1$, $a_1 = 1$

e) $a_n = a_{n-1} + a_{n-3}$, $a_0 = 1$, $a_1 = 2$, $a_2 = 0$

2. 令 $a_n = 2^n + 5 \cdot 3^n$, $n = 0, 1, 2, \dots$

a) 求 a_0 , a_1 , a_2 , a_3 和 a_4 。

b) 证明 $a_2 = 5a_1 - 6a_0$, $a_3 = 5a_2 - 6a_1$, $a_4 = 5a_3 - 6a_2$ 。

c) 证明: 对一切整数 n , $n \geq 2$, $a_n = 5a_{n-1} - 6a_{n-2}$ 。

3. 序列 $\{a_n\}$ 是递推关系 $a_n = 8a_{n-1} - 16a_{n-2}$ 的解吗? 如果

a) $a_n = 0$

b) $a_n = 1$

c) $a_n = 2^n$

d) $a_n = 4^n$

e) $a_n = n4^n$

f) $a_n = 2 \cdot 4^n + 3n \cdot 4^n$

g) $a_n = (-4)^n$

h) $a_n = n^2 4^n$

4. 证明序列 $\{a_n\}$ 是递推关系 $a_n = a_{n-1} + 2a_{n-2} + 2n - 9$ 的解, 如果

a) $a_n = -n + 2$

b) $a_n = 5(-1)^n - n + 2$

c) $a_n = 3(-1)^n + 2^n - n + 2$

d) $a_n = 7 \cdot 2^n - n + 2$

5. 用例 5 中用到的迭代方法求下面每个递推关系和初始条件的解。

a) $a_n = 3a_{n-1}$, $a_0 = 2$

b) $a_n = a_{n-1} + 2$, $a_0 = 3$

c) $a_n = a_{n-1} + n$, $a_0 = 1$

d) $a_n = a_{n-1} + 2n + 3$, $a_0 = 4$

- e) $a_n = 2a_{n-1} - 1, a_0 = 1$ f) $a_n = 3a_{n-1} + 1, a_0 = 1$ g) $a_n = na_{n-1}, a_0 = 5$ h) $a_n = 2na_{n-1}, a_0 = 1$
6. 假设一群细菌的数目每小时增长为 3 倍。
 - a) 建立关于 n 小时后细菌数的递推关系。
 - b) 如果初始的群体有 100 个细菌, 那么 10 小时后将有多少个细菌?
 7. 一个工厂逐月增长地定做体育赛车。在第 1 个月只做了 1 辆, 在第 2 个月做了 2 辆, 照此下去, 到第 n 个月做了 n 辆。
 - a) 对这个工厂前 n 个月生产的赛车数构造一个递推关系。
 - b) 在第一年生产了多少辆赛车?
 - c) 求出这个工厂在前 n 个月生产赛车数的显式公式。
 8. 借 5000 美元, 利率 7%, 如果每个月还款 100 美元, 找出关于第 k 个月末欠款余额 $B(k)$ 的递推关系。[提示: 将 $B(k)$ 用 $B(k-1)$ 项表示, 月息是 $(0.07/12)B(k-1)$ 。]
 9. 用数学归纳法验证在例 5 导出的求解汉诺塔难题所需移动次数的公式。
 10. 一台出售邮票簿的售货机只接受 1 美元硬币、1 美元纸币以及 5 美元纸币。
 - a) 找出与放 n 美元到这台售货机的方式数有关的递推关系, 这里要考虑硬币和纸币放入的次序。
 - b) 初始条件是什么?
 - c) 一本邮票簿需 10 美元, 有多少种付款方式?
 11. a) 求与包含 2 个连续 0 的 n 位二进制串的个数有关的递推关系。
 - b) 初始条件是什么?
 - c) 包含 2 个连续 0 的 7 位二进制串有多少个?
 12. a) 求与不包含 3 个连续 0 的 n 位二进制串的个数有关的递推关系。
 - b) 初始条件是什么?
 - c) 不包含 3 个连续 0 的 7 位二进制串有多少个?
 13. a) 一个人爬阶梯, 如果每次可以上 1 或 2 阶, 求与爬 n 步阶梯的方式数有关的递推关系。
 - b) 初始条件是什么?
 - c) 这个人爬 8 步阶梯上飞机有多少种方式?
 14. a) 求与不包含 2 个连续 0 的 n 位三进制串的个数有关的递推关系。
 - b) 初始条件是什么?
 - c) 不包含 2 个连续 0 的 6 位三进制串有多少个?
 - *15. a) 求与不包含 2 个连续 0 或 2 个连续 1 的 n 位三进制串的个数有关的递推关系。
 - b) 初始条件是什么?
 - c) 不包含 2 个连续 0 或 2 个连续 1 的 6 位三进制串有多少个?
 - *16. a) 求与不包含连续的相同符号的 n 位三进制串的个数有关的递推关系。
 - b) 初始条件是什么?
 - c) 不包含连续的相同符号的 6 位三进制串有多少个?
 17. 信息通过信道传送要使用两个信号。一个信号的传送需要 1 微秒, 而另一个信号的传送需要 2 微秒。
 - a) 求与在 n 微秒内发送的不同信息数有关的递推关系, 其中信息由这两个信号的序列构成, 并且信息中的每个信号后面都紧跟着下一个信号。
 - b) 初始条件是什么?
 - c) 用这两个信号在 10 微秒内可以发送多少条不同的信息?
 18. a) 如果 R_n 是一个平面被 n 条直线划分出的区域个数, 其中没有两条直线是平行的也没有 3 条直线交于一点, 找出由 R_n 满足的递推关系。
 - b) 使用迭代求出 R_n 。
 - *19. a) 找出由 S_n 满足的递推关系, 其中 S_n 是三维空间被 n 个平面分成的区域数, 如果每 3 个平面交于一点, 但没有 4 个平面交于一点。
 - b) 使用迭代求出 S_n 。
 20. 包含偶数个 0 的 7 位二进制串有多少个?
 21. a) 用地砖铺一条人行道, 地砖是红色、绿色或灰色的。如果没有两块红砖相邻且同色的地砖是不加区别的, 找出与用 n 块砖铺一条路的方式数有关的递推关系。

b) 对于(a)中的递推关系有什么初始条件?

c) 用7块砖铺一条在(a)中所描述的路有多少种方式?

*22. 设 $S(m, n)$ 表示从 m 元素集到 n 元素集的映上函数个数。证明 $S(m, n)$ 满足递推关系

$$S(m, n) = n^m - \sum_{k=1}^{n-1} C(n, k) S(m, k)$$

其中 $m \geq n$ 且 $n > 1$, 初始条件是 $S(m, 1) = 1$ 。

23. a) 使用在例8所建立的递推关系确定 C_5 , 即为确定相乘的次序在6个数的乘积中加括号的方式数。

b) 使用在例8的解答中所提到的关于 C_5 的封闭公式检验你在(b)得到的结果。

 练习 24~26 是格雷厄姆·克努斯(Graham Knuth)和帕塔什尼克(Patashnik)在[GrKnPa94]所描述的约瑟夫问题(Josephus Problem)的一种变形。这个问题来源于历史学家弗拉维乌斯·约瑟夫的一本账。41个犹太叛民在一世纪犹太-罗马战争期间被罗马人追赶逃入山洞, 约瑟夫是这群人中的一个。这些叛民宁愿死也不愿被俘。他们决定围成一个圆圈并且围着这个圆圈重复数数, 每数到3就杀掉这个位置的人而留下其他的人。但是约瑟夫和另一个叛民不愿意这样被杀掉。他们确定了应该站的位置是最后两个活下来的叛民的位置。我们考虑的问题开始时有 n 个人, 记为1到 n , 站成一个圆圈。每一步, 每第2个仍旧活着的人将被排除, 直到只剩下一个人为止。我们把生还的人数记作 $J(n)$ 。

24. 对每个正整数 n 的值, $1 \leq n \leq 16$, 确定 $J(n)$ 的值。

25. 对于 $n \geq 1$, 证明 $J(n)$ 满足递推关系 $J(2n) = 2J(n) - 1$ 和 $J(2n+1) = 2J(n) + 1$, 且 $J(1) = 1$ 。

26. 根据你关于 $J(n)$ 的公式确定 $J(100)$, $J(1000)$ 和 $J(10\,000)$ 。

练习 27~30 涉及雷夫难题, 即具有4个柱和 n 个盘子的汉诺塔变形问题。在给出这些练习之前, 我们描述一个弗雷姆-斯图尔特(Frame-Stewart)算法, 它把盘子从柱1移到柱4并且没有较大的盘子放在较小的盘子上面。给定盘子数 n 作为输入, 这个算法依赖于一个整数 k 的选择, $1 \leq k \leq n$ 。当只有一个盘子时, 把它从柱1移到柱4, 然后算法停止。对于 $n > 1$, 算法递归地使用下面的3步。首先使用所有的4根柱递归地把最小的 $n-k$ 个盘子从柱1移到柱2。下一步使用汉诺塔问题的三根柱算法, 不使用放 $n-k$ 个最小盘子的柱, 把 k 个最大的盘子递归地从柱1移到柱4。最后, 使用所有4根柱递归地将 $n-k$ 个最小的盘子移到柱4。弗雷姆和斯图尔特证明, 使用他们的算法, 为了达到最少的移动次数, 应该选择 k 使得 n 是不超过第 k 个三角形数 $t_k = k(k+1)/2$ 的最小的正整数, 即 $t_{k-1} < n \leq t_k$ 。有一个未被证实的猜想, 称为弗雷姆猜想, 就是不管盘子怎样移动, 该算法对于求解这个难题所需要的移动次数最少。

27. 证明: 具有4个盘子的雷夫难题最少可以使用9次移动求解。

28. 证明: 如果 $R(n)$ 是由弗雷姆-斯图尔特算法求解具有 n 个盘子的雷夫难题所使用的移动次数, 这里选择 k 是满足 $n \leq k(k+1)/2$ 的最小的整数, 那么 $R(n)$ 满足递推关系 $R(n) = 2R(n-k) + 2^k - 1$ 和 $R(0) = 0$, $R(1) = 1$ 。

*29. 证明: 如果 k 如练习28所选, 那么 $R(n) = \sum_{i=1}^k i2^{i-1} - (t_k - n)2^{k-1}$ 。

*30. 证明: $R(n)$ 是 $O(\sqrt{n}2^{\sqrt{n}})$ 。

设 $\{a_n\}$ 是实数序列, 这个序列的后向差分递归地定义如下:

第一差分 ∇a_n 是

$$\nabla a_n = a_n - a_{n-1}$$

从 $\nabla^k a_n$ 得到第 $k+1$ 差分 $\nabla^{k+1} a_n$, 即

$$\nabla^{k+1} a_n = \nabla^k a_n - \nabla^k a_{n-1}$$

31. 证明: $a_{n-2} = a_n - 2\nabla a_n + \nabla^2 a_n$ 。

32. 用 $a_n, \nabla a_n, \nabla^2 a_n$ 的项表示递推关系 $a_n = a_{n-1} + a_{n-2}$ 。

4.2 求解线性递推关系

4.2.1 引言

 各种各样的递推关系出现在模型里。某些递推关系可以用迭代或者其他的特别技术求解。但是, 有一类重要的递推关系可以用一种系统的方法明确地求解。在这种递推关

系中, 序列的项由它的前项的线性组合来表示。

定义 1 一个常系数的 k 阶线性齐次递推关系是形如

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$$

的递推关系, 其中 $c_1, c_2, \dots, c_k$ 是实数, $c_k \neq 0$ 。

这个定义中的递推关系是线性的, 因为它的右边是序列前项的倍数之和。这个递推关系是齐次的, 因为所出现的各项都是 a_j 的倍数。序列各项的系数都是常数而不是依赖于 n 的函数。阶为 k 是因为 a_n 由序列前面的 k 项来表示。

根据数学归纳法第二原理, 满足这个定义的递推关系的序列由这个递推关系和 k 个初始条件

$$a_0 = C_0, a_1 = C_1, \dots, a_{k-1} = C_{k-1}$$

唯一地确定。

例 1 递推关系 $P_n = (1, 11)P_{n-1}$ 是 1 阶的线性齐次递推关系。递推关系 $f_n = f_{n-1} + f_{n-2}$ 是 2 阶的线性齐次递推关系。递推关系 $a_n = a_{n-5}$ 是 5 阶的线性齐次递推关系。

下面是一些常系数的但不是线性齐次递推关系的例子。

例 2 递推关系 $a_n = a_{n-1} + a_{n-2}^2$ 不是线性的。递推关系 $H_n = 2H_{n-1} + 1$ 不是齐次的。递推关系 $B_n = nB_{n-1}$ 不是常系数的。

研究线性齐次递推关系有两个理由。第一, 在建立问题的模型时经常出现这种递推关系。第二, 它们可以用系统的方法求解。

4.2.2 求解常系数线性齐次递推关系

求解常系数线性齐次递推关系的基本方法是寻找形如 $a_n = r^n$ 的解, 其中 r 是常数。注意 $a_n = r^n$ 是递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$ 的解, 当且仅当

$$r^n = c_1 r^{n-1} + c_2 r^{n-2} + \cdots + c_k r^{n-k}$$

当等式的两边除以 r^{n-k} 并且从左边减去右边时, 我们得到等价的方程

$$r^k - c_1 r^{k-1} - c_2 r^{k-2} - \cdots - c_{k-1} r - c_k = 0$$

因此, 序列 $\{a_n\}$ 以 $a_n = r^n$ 作为解, 当且仅当 r 是这后一个方程的解。这个方程叫做该递推关系的特征方程。方程的解叫做该递推关系的特征根。正如我们将要看到的, 可以用这些特征根给出这种递推关系的所有解的显式公式。

我们首先看一个 2 阶常系数线性齐次递推关系的处理结果。然后, 叙述相应的阶可能大于 2 的一般性结果。由于得到一般性结果所需要的证明比较复杂, 本书不再赘述。

我们现在回到 2 阶线性齐次递推关系。首先, 考虑存在两个不等的特征根的情况。

定理 1 设 c_1 和 c_2 是实数。假设 $r^2 - c_1 r - c_2 = 0$ 有两个不等的根 r_1 和 r_2 , 那么序列 $\{a_n\}$ 是递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 的解, 当且仅当 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$, $n = 0, 1, 2, \dots$, 其中 α_1 和 α_2 是常数。

证 证明这个定理必须做两件事。首先, 必须证明如果 r_1 和 r_2 是特征方程的根, 并且 α_1 和 α_2 是常数, 那么序列 $\{a_n\}$ ($a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$) 是递推关系的解。第二, 必须证明如果序列 $\{a_n\}$ 是解, 那么对某个常数 α_1 和 α_2 有 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ 。

现在我们将证明如果 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$, 那么序列 $\{a_n\}$ 是递推关系的解。因为 r_1 和 r_2 是 $r^2 - c_1 r - c_2 = 0$ 的根, 从而 $r_1^2 = c_1 r_1 + c_2$, $r_2^2 = c_1 r_2 + c_2$ 。

从这些等式可以看出

$$\begin{aligned} c_1 a_{n-1} + c_2 a_{n-2} &= c_1 (\alpha_1 r_1^{n-1} + \alpha_2 r_2^{n-1}) + c_2 (\alpha_1 r_1^{n-2} + \alpha_2 r_2^{n-2}) \\ &= \alpha_1 r_1^{n-2} (c_1 r_1 + c_2) + \alpha_2 r_2^{n-2} (c_1 r_2 + c_2) \\ &= \alpha_1 r_1^{n-2} r_1^2 + \alpha_2 r_2^{n-2} r_2^2 \\ &= \alpha_1 r_1^n + \alpha_2 r_2^n \\ &= a_n \end{aligned}$$

这证明了序列 $\{a_n\}$ 以 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ 作为递推关系的解。

为证明递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 的每一个解 $\{a_n\}$ 都有形式 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$, $n=0, 1, 2, \dots$, α_1 和 α_2 为某个常数。假设 $\{a_n\}$ 是递推关系的解, 初始条件是 $a_0 = C_0$, $a_1 = C_1$ 。下面证明存在常数 α_1 和 α_2 使得具有 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ 的序列 $\{a_n\}$ 满足同样的初始条件。这要求

$$a_0 = C_0 = \alpha_1 + \alpha_2$$

$$a_1 = C_1 = \alpha_1 r_1 + \alpha_2 r_2$$

我们可以求解这两个关于 α_1 和 α_2 的方程。从第一个方程得到 $\alpha_2 = C_0 - \alpha_1$ 。把它代入第二个方程得

$$C_1 = \alpha_1 r_1 + (C_0 - \alpha_1) r_2$$

因此,

$$C_1 = \alpha_1 (r_1 - r_2) + C_0 r_2$$

这说明了

$$\alpha_1 = \frac{(C_1 - C_0 r_2)}{r_1 - r_2}$$

和

$$\alpha_2 = C_0 - \alpha_1 = C_0 - \frac{(C_1 - C_0 r_2)}{r_1 - r_2} = \frac{C_0 r_1 - C_1}{r_1 - r_2}$$

这里关于 α_1 和 α_2 的表达式依赖于 $r_1 \neq r_2$ 的事实。(当 $r_1 = r_2$ 时, 这个定理不成立。)因此, 由于这两个 α_1 和 α_2 的值, 具有 $\alpha_1 r_1^n + \alpha_2 r_2^n$ 的序列 $\{a_n\}$ 满足这两个初始条件。因为这个递推关系和这些初始条件唯一地确定了这个序列, 从而得出 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ 。□

常数线性齐次递推关系的特征根可能是复数。定理 1(和本节后面的定理)在这种情况下仍旧适用。具有复数特征根的递推关系在本书中不进行讨论。熟悉复数的读者可以做节末的练习 20。

例 3 和例 4 说明怎样用定理 1 求解递推关系。

例 3 什么是如下递推关系的解? 其中 $a_0 = 2$ 和 $a_1 = 7$ 。

$$a_n = a_{n-1} + 2a_{n-2}$$

解 可用定理 1 求解这个问题。递推关系的特征方程是 $r^2 - r - 2 = 0$ 。它的根是 $r = 2$ 和 $r = -1$ 。因此, 序列 $\{a_n\}$ 是递推关系的解当且仅当

$$a_n = \alpha_1 2^n + \alpha_2 (-1)^n$$

α_1 和 α_2 是常数。由初始条件, 得

$$a_0 = 2 = \alpha_1 + \alpha_2$$

$$a_1 = 7 = \alpha_1 \cdot 2 + \alpha_2 \cdot (-1)$$

求解这两个等式得 $\alpha_1 = 3$ 和 $\alpha_2 = -1$ 。因此, 关于这个递推关系和初始条件的解是序列 $\{a_n\}$, 其中

$$a_n = 3 \cdot 2^n - (-1)^n$$

例 4 找一个关于斐波那契数的显式公式。

解 斐波那契数的序列满足递推关系 $f_n = f_{n-1} + f_{n-2}$ 和初始条件 $f_0 = 0$ 及 $f_1 = 1$ 。特征方程 $r^2 - r - 1 = 0$ 的根是 $r_1 = (1 + \sqrt{5})/2$ 和 $r_2 = (1 - \sqrt{5})/2$ 。因此, 从定理 1 得到斐波那契数由

$$f_n = \alpha_1 \left(\frac{1 + \sqrt{5}}{2} \right)^n + \alpha_2 \left(\frac{1 - \sqrt{5}}{2} \right)^n$$

给出, 其中 α_1 和 α_2 为常数。可用初始条件 $f_0 = 0$ 和 $f_1 = 1$ 确定这些常数。我们有

$$f_0 = \alpha_1 + \alpha_2 = 0$$

$$f_1 = \alpha_1 \left(\frac{1 + \sqrt{5}}{2} \right) + \alpha_2 \left(\frac{1 - \sqrt{5}}{2} \right) = 1$$

对这些关于 α_1 和 α_2 的联立方程的解是

$$\alpha_1 = 1/\sqrt{5}, \quad \alpha_2 = -1/\sqrt{5}$$

于是, 斐波那契数由下面的式子给出:

$$f_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^n$$

当存在二重特征根时定理1不再适用。如果发生这种情况,当 r_0 是特征方程的一个二重根时那么 $a_n = nr_0^n$ 是递推关系的另一个解。定理2说明了怎样处理这种情况。

定理2 设 c_1 和 c_2 是实数, $c_2 \neq 0$ 。假设 $r^2 - c_1 r - c_2 = 0$ 只有一个根 r_0 。序列 $\{a_n\}$ 是递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 的解,当且仅当 $a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$, $n=0, 1, 2, \dots$, 其中 α_1 和 α_2 是常数。

定理2的证明留作节末的练习。例5说明了这个定理的应用。

例5 具有初始条件 $a_0=1$ 和 $a_1=6$ 的递推关系

$$a_n = 6a_{n-1} - 9a_{n-2}$$

的解是什么?

解 $r^2 - 6r + 9 = 0$ 唯一的根是 $r=3$ 。因此,这个递推关系的解是:

$$a_n = \alpha_1 3^n + \alpha_2 n 3^n$$

其中 α_1 和 α_2 是常数。使用初始条件得

$$a_0 = 1 = \alpha_1$$

$$a_1 = 6 = \alpha_1 \cdot 3 + \alpha_2 \cdot 3$$

求解这两个方程得 $\alpha_1=1$ 和 $\alpha_2=1$ 。从而,这个具有给定初始条件的递推关系的解是

$$a_n = 3^n + n 3^n$$

我们现在叙述这个关于常系数线性齐次递推关系的解的一般性结果,这里的阶可以大于2且假定特征方程有不等根。这个结果的证明留给读者作为练习。

定理3 设 $c_1, c_2, \dots, c_k$ 是实数。假设特征方程

$$r^k - c_1 r^{k-1} - \dots - c_k = 0$$

有 k 个不等的根 $r_1, r_2, \dots, r_k$ 。那么序列 $\{a_n\}$ 是递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$$

的解,当且仅当

$$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \dots + \alpha_k r_k^n$$

$n=0, 1, 2, \dots$, 其中 $\alpha_1, \alpha_2, \dots, \alpha_k$ 是常数。

我们用例6说明定理的使用。

例6 求出具有初始条件 $a_0=2$, $a_1=5$ 和 $a_2=15$ 的递推关系

$$a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$$

的解。

解 这个递推关系的特征多项式是

$$r^3 - 6r^2 + 11r - 6$$

因为 $r^3 - 6r^2 + 11r - 6 = (r-1)(r-2)(r-3)$, 所以特征根是 $r=1$, $r=2$ 和 $r=3$ 。因此,递推关系的解的形式是

$$a_n = \alpha_1 \cdot 1^n + \alpha_2 \cdot 2^n + \alpha_3 \cdot 3^n$$

为找到常数 α_1, α_2 以及 α_3 , 使用初始条件得

$$a_0 = 2 = \alpha_1 + \alpha_2 + \alpha_3$$

$$a_1 = 5 = \alpha_1 + \alpha_2 \cdot 2 + \alpha_3 \cdot 3$$

$$a_2 = 15 = \alpha_1 + \alpha_2 \cdot 4 + \alpha_3 \cdot 9$$

当求解这三个关于 α_1, α_2 和 α_3 的联立方程时,得到 $\alpha_1=1$, $\alpha_2=-1$ 且 $\alpha_3=2$ 。于是,这个递推关系和给定初始条件的唯一解是满足

$$a_n = 1 - 2^n + 2 \cdot 3^n$$

的序列 $\{a_n\}$ 。

我们现在叙述关于常系数线性齐次递推关系的最一般化的结果,这里允许特征方程有重根。要点是对于特征方程的每个根 r ,通解是形如 $P(n)r^n$ 的项之和,其中 $P(n)$ 是 $m-1$ 次多项式,而

m 是这个根的重数。我们把证明作为一个挑战性的练习留给读者。

定理 4 设 $c_1, c_2, \dots, c_k$ 是实数, 假设特征方程

$$r^k - c_1 r^{k-1} - \dots - c_k = 0$$

有 t 个不等的根 $r_1, r_2, \dots, r_t$, 其重数分别为 $m_1, m_2, \dots, m_t$, 满足 $m_i \geq 1, i=1, 2, \dots, t$, 且 $m_1 + m_2 + \dots + m_t = k$ 。那么序列 $\{a_n\}$ 是递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$$

的解, 当且仅当

$$\begin{aligned} a_n = & (a_{1,0} + a_{1,1}n + \dots + a_{1,m_1-1}n^{m_1-1})r_1^n \\ & + (a_{2,0} + a_{2,1}n + \dots + a_{2,m_2-1}n^{m_2-1})r_2^n \\ & + \dots + (a_{t,0} + a_{t,1}n + \dots + a_{t,m_t-1}n^{m_t-1})r_t^n \end{aligned}$$

$n=0, 1, 2, \dots$, 其中 $a_{i,j}$ 是常数, $1 \leq i \leq t$ 且 $0 \leq j \leq m_i - 1$ 。

例 7 说明在特征方程有重根时怎样用定理 4 求一个线性齐次递推关系的通解形式。

例 7 假设线性齐次递推关系的特征方程的根是 2, 2, 2, 5, 5 和 9 (即有 3 个根, 根 2 的重数为 3, 根 5 的重数为 2, 根 9 的重数为 1)。那么通解形式是什么?

解 由定理 4, 解的一般形式是

$$(a_{1,0} + a_{1,1}n + a_{1,2}n^2)2^n + (a_{2,0} + a_{2,1}n)5^n + a_{3,0}9^n$$

我们现在说明在特征方程有 3 重根时如何用定理 4 求解常系数线性齐次递推关系。

例 8 找出具有初始条件 $a_0=1$, $a_1=-2$ 和 $a_2=-1$ 的递推关系

$$a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$$

的解。

解 这个递推关系的特征方程是

$$r^3 + 3r^2 + 3r + 1 = 0$$

因为 $r^3 + 3r^2 + 3r + 1 = (r+1)^3$, 特征方程只有一个 3 重根 $r=-1$ 。由定理 4, 这个递推关系的解是下述形式

$$a_n = a_{1,0}(-1)^n + a_{1,1}n(-1)^n + a_{1,2}n^2(-1)^n$$

为求出常数 $a_{1,0}$, $a_{1,1}$ 和 $a_{1,2}$, 使用初始条件, 得到

$$a_0 = 1 = a_{1,0}$$

$$a_1 = -2 = -a_{1,0} - a_{1,1} - a_{1,2}$$

$$a_2 = -1 = a_{1,0} + 2a_{1,1} + 4a_{1,2}$$

这 3 个方程的联立解是 $a_{1,0}=1$, $a_{1,1}=3$, $a_{1,2}=-2$ 。于是, 这个递推关系和给定初始条件的唯一解是序列 $\{a_n\}$, 其中

$$a_n = (1 + 3n - 2n^2)(-1)^n$$

4.2.3 常系数线性非齐次的递推关系

我们已经知道如何求解常系数线性齐次的递推关系。是否有一种相对简单的技术来求解如像 $a_n = 3a_{n-1} + 2n$ 的常系数线性但是非齐次的递推关系呢? 我们将看到, 仅仅对某些特定类型的递推关系存在肯定的回答。

递推关系 $a_n = 3a_{n-1} + 2n$ 是一个常系数线性非齐次递推关系, 即形如

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + F(n)$$

的递推关系的例子, 其中 $c_1, c_2, \dots, c_k$ 是实数, $F(n)$ 是只依赖于 n 且不恒为 0 的函数。递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$$

叫做相伴的齐次递推关系。它在非齐次递推关系的解中起了重要的作用。

例 9 递推关系 $a_n = a_{n-1} + 2^n$, $a_n = a_{n-1} + a_{n-2} + n^2 + n + 1$, $a_n = 3a_{n-1} + n3^n$ 和 $a_n = a_{n-1} + a_{n-2} + a_{n-3} + n!$ 是常系数线性非齐次递推关系。相伴的线性齐次递推关系分别是 $a_n = a_{n-1}$, $a_n = a_{n-1} + a_{n-2}$, $a_n = 3a_{n-1}$ 和 $a_n = a_{n-1} + a_{n-2} + a_{n-3}$ 。

关于常系数线性非齐次递推关系的一个关键事实是, 每个解都是一个特解与相伴的线性齐次递推关系的一个解之和, 正如定理 5 所述。

定理 5 如果 $\{a_n^{(p)}\}$ 是常系数非齐次线性递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)$$

的一个特解, 那么每个解都是 $\{a_n^{(p)} + a_n^{(h)}\}$ 的形式, 其中 $\{a_n^{(h)}\}$ 是相伴的齐次递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$$

的一个解。

证 由于 $\{a_n^{(p)}\}$ 是非齐次递推关系的特解, 我们知道

$$a_n^{(p)} = c_1 a_{n-1}^{(p)} + c_2 a_{n-2}^{(p)} + \cdots + c_k a_{n-k}^{(p)} + F(n)$$

现在假设 $\{b_n\}$ 是常系数非齐次递推关系的第二个解, 使得

$$b_n = c_1 b_{n-1} + c_2 b_{n-2} + \cdots + c_k b_{n-k} + F(n)$$

从第二个等式减去第一个等式得

$$b_n - a_n^{(p)} = c_1 (b_{n-1} - a_{n-1}^{(p)}) + c_2 (b_{n-2} - a_{n-2}^{(p)}) + \cdots + c_k (b_{n-k} - a_{n-k}^{(p)})$$

从而得到 $\{b_n - a_n^{(p)}\}$ 是相伴的线性齐次递推关系的一个解, 比如说是 $\{a_n^{(h)}\}$ 。因此, 对所有的 n 有 $b_n = a_n^{(p)} + a_n^{(h)}$ 。 $\square$

由定理 5, 我们看到求解常系数非齐次递推关系的关键是找一个特解。然后每个解都是这个特解与相伴的齐次递推关系的一个解之和。尽管不存在对每个函数 $F(n)$ 都有效的一般性方法来求这种解, 但有某些技术对特定的函数类型 $F(n)$, 例如多项式函数与常数的幂函数有效。例 10 和例 11 就说明了这一点。

例 10 求递推关系 $a_n = 3a_{n-1} + 2n$ 的所有的解。具有 $a_1 = 3$ 的解是什么?

解 为求解这个常系数线性非齐次递推关系, 我们需要求解与它相伴的线性齐次方程并且找到一个关于给定非齐次方程的特解。相伴的线性齐次方程是 $a_n = 3a_{n-1}$ 。它的解是 $a_n^{(h)} = \alpha 3^n$, 其中 α 是常数。

我们现在找一个特解。因为 $F(n) = 2n$ 是 n 的 1 次多项式, 解的一个合理的尝试就是 n 的线性函数, 比如说 $p_n = cn + d$, 其中 c 和 d 是常数。为确定是否存在这种形式的解, 假设 $p_n = cn + d$ 是一个这样的解。那么方程 $a_n = 3a_{n-1} + 2n$ 就变成 $cn + d = 3(c(n-1) + d) + 2n$ 。化简和归并同类项得 $(2+2c)n + (2d-3c) = 0$ 。从而, $cn + d$ 是一个解, 当且仅当 $2+2c=0$ 和 $2d-3c=0$ 。这说明 $cn + d$ 是一个解当且仅当 $c = -1$ 和 $d = -3/2$ 。因而, $a_n^{(p)} = -n - 3/2$ 是一个特解。

根据定理 5, 所有的解都是下述形式

$$a_n = a_n^{(p)} + a_n^{(h)} = -n - 3/2 + \alpha \cdot 3^n$$

其中 α 是常数。

为找出具有 $a_1 = 3$ 的解, 在得到的通解公式中令 $n=1$ 。我们有 $3 = -1 - 3/2 + 3\alpha$, 这就推出 $\alpha = 11/6$ 。我们要找的解是 $a_n = -n - 3/2 + (11/6)3^n$ 。 $\blacksquare$

例 11 求出下述递推关系

$$a_n = 5a_{n-1} - 6a_{n-2} + 7^n$$

的所有的解。

解 这是一个线性非齐次递推关系。它的相伴的齐次递推关系

$$a_n = 5a_{n-1} - 6a_{n-2}$$

的解是 $a_n^{(h)} = \alpha_1 \cdot 3^n + \alpha_2 \cdot 2^n$, 其中 α_1 和 α_2 是常数。因为 $F(n) = 7^n$, 一个合理的解是 $a_n^{(p)} = C \cdot 7^n$, 其中 C 是常数。把这些项代入递推关系得 $C \cdot 7^n = 5C \cdot 7^{n-1} - 6C \cdot 7^{n-2} + 7^n$ 。提出公因式 7^{n-2} , 这个等式变成 $49C = 35C - 6C + 49$, 从而推出 $20C = 49$ 或 $C = 49/20$ 。于是, $a_n^{(p)} = (49/20)7^n$ 是特解。由定理 5, 所有的解都有下述形式

$$a_n = \alpha_1 \cdot 3^n + \alpha_2 \cdot 2^n + (49/20)7^n$$

在例 10 和例 11 中, 我们凭经验猜想了一个特定形式的解。在两种情况下, 我们都能找到特

解。这并不是偶然的。每当 $F(n)$ 是 n 的多项式和一个常数的 n 次幂之积时, 我们就能知道一个特解恰好是什么形式, 正如定理 6 所述。定理 6 的证明作为一个挑战性的练习留给读者。

定理 6 假设 $\{a_n\}$ 满足线性非齐次递推关系

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)$$

其中 $c_1, c_2, \dots, c_k$ 是实数, 且

$$F(n) = (b_r n^r + b_{r-1} n^{r-1} + \cdots + b_1 n + b_0) s^n$$

其中 $b_0, b_1, \dots, b_r$ 和 s 是实数。当 s 不是相伴的线性齐次递推关系的特征方程的根时, 存在一个下述形式的特解

$$(p_r n^r + p_{r-1} n^{r-1} + \cdots + p_1 n + p_0) s^n$$

当 s 是特征方程的根且它的重数是 m 时, 存在一个下述形式的特解

$$n^m (p_r n^r + p_{r-1} n^{r-1} + \cdots + p_1 n + p_0) s^n$$

注意当 s 是相伴的线性齐次递推关系的特征方程的 m 重根时, 因式 n^m 保证给出的特解不是相伴的线性齐次递推关系的一个解。我们下面给出一个例子说明定理 6 所提供的特解形式。

例 12 当 $F(n)=3^n$, $F(n)=n3^n$, $F(n)=n^2 2^n$ 和 $F(n)=(n^2+1)3^n$ 时, 线性非齐次递推关系 $a_n=6a_{n-1}-9a_{n-2}+F(n)$ 的特解有什么形式?

解 相伴的线性齐次递推关系是 $a_n=6a_{n-1}-9a_{n-2}$ 。它的特征方程 $r^2-6r+9=(r-3)^2=0$ 有一个 2 重的单根 3。 $F(n)$ 的形式为 $P(n)s^n$, 其中 $P(n)$ 是一个多项式, s 是一个常数。为应用定理 6, 我们需要知道 s 是否是这个特征方程的根。

由于 $s=3$ 是重数 $m=2$ 的根而 $s=2$ 不是根, 定理 6 告诉我们如果 $F(n)=3^n$, 特解的形式是 $p_0 n^2 3^n$; 如果 $F(n)=n3^n$, 特解的形式是 $n^2(p_1 n + p_0)3^n$; 如果 $F(n)=n^2 2^n$, 特解的形式是 $(p_2 n^2 + p_1 n + p_0)2^n$; 如果 $F(n)=(n^2+1)3^n$, 特解的形式是 $n^2(p_2 n^2 + p_1 n + p_0)3^n$ 。■

在求解定理 6 所谈的那种类型的递推关系时, 若 $s=1$ 一定要小心处理。特别是把定理用于 $F(n)=b_r n^r + b_{r-1} n^{r-1} + \cdots + b_1 n + b_0$, 参数 s 取值 $s=1$ 时(尽管项 1^n 没有明确地出现)的情况。根据这个定理, 解的形式就依赖于是否 1 是相伴的线性齐次递推关系的特征方程的根。这将在例 13 中说明, 它说明了怎样用定理 6 找出前 n 个正整数之和的公式。

例 13 设 a_n 是前 n 个正整数的和, 即

$$a_n = \sum_{k=1}^n k$$

注意, a_n 满足线性非齐次递推关系

$$a_n = a_{n-1} + n$$

(为从前 $n-1$ 个正整数的和 a_{n-1} 得到前 n 个正整数的和 a_n , 只需加上 n 即可)。注意初始条件是 $a_1=1$ 。

对于 a_n , 相伴的线性齐次递推关系是

$$a_n = a_{n-1}$$

这个齐次递推关系的解是 $a_n^{(h)} = c(1)^n = c$, 其中 c 是一个常数。为找到 $a_n = a_{n-1} + n$ 的所有的解, 我们仅需要找一个特解。由定理 6, 由于 $F(n)=n=n(1)^n$ 且 $s=1$ 是相伴的线性齐次递推关系的特征方程的 1 阶根, 存在一个形如 $n(p_1 n + p_0) = p_1 n^2 + p_0 n$ 的特解。

把它代入递推关系得到 $p_1 n^2 + p_0 n = p_1 (n-1)^2 + p_0 (n-1) + n$ 。化简得 $n(2p_1 - 1) + (p_0 - p_1) = 0$, 这意味着 $2p_1 - 1 = 0$ 和 $p_0 - p_1 = 0$, 即 $p_0 = p_1 = 1/2$ 。因此

$$a_n^{(p)} = \frac{n^2}{2} + \frac{n}{2} = \frac{n(n+1)}{2}$$

是一个特解。所以, 原递推关系 $a_n = a_{n-1} + n$ 的所有的解由 $a_n = a_n^{(h)} + a_n^{(p)} = c + n(n+1)/2$ 给出。由于 $a_1=1$, 我们有 $1 = a_1 = c + 1 \cdot 2/2 = c + 1$, 故 $c=0$ 。从而 $a_n = n(n+1)/2$ 。(这和 2.4 节表 2-5 给出的以及前面导出的公式一样。) ■

练习

1. 确定下面哪些是常系数线性齐次递推关系, 如果是, 求它们的阶。

$$a) a_n = 3a_{n-1} + 4a_{n-2} + 5a_{n-3}$$

$$b) a_n = 2na_{n-1} + a_{n-2}$$

$$c) a_n = a_{n-1} + a_{n-4}$$

$$d) a_n = a_{n-1} + 2$$

$$e) a_n = a_{n-1}^2 + a_{n-2}$$

$$f) a_n = a_{n-2}$$

$$g) a_n = a_{n-1} + n$$

2. 求解下述具有给定初始条件的递推关系。

$$a) a_n = 2a_{n-1}, n \geq 1, a_0 = 3$$

$$b) a_n = a_{n-1}, n \geq 1, a_0 = 2$$

$$c) a_n = 5a_{n-1} - 6a_{n-2}, n \geq 2, a_0 = 1, a_1 = 0$$

$$d) a_n = 4a_{n-1} - 4a_{n-2}, n \geq 2, a_0 = 6, a_1 = 8$$

$$e) a_n = -4a_{n-1} - 4a_{n-2}, n \geq 2, a_0 = 0, a_1 = 1$$

$$f) a_n = 4a_{n-2}, n \geq 2, a_0 = 0, a_1 = 4$$

$$g) a_n = a_{n-2}/4, n \geq 2, a_0 = 1, a_1 = 0$$

3. 使用 4.1 节练习 17 描述的两个信号在 n 微秒内可以传送多少不同的信息?

4. 使用 1×2 和 2×2 的块铺满一块 $2 \times n$ 的长方形板, 有多少种方式?

5. 年初把一笔 100 000 美元的钱存入一个投资基金。在每年的最后一天得到两份红利。第一份红利是当年账上钱数的 20%。第二份红利是前一年账上钱数的 45%。

a) 如果不允许取钱, 找出一个关于 $\{P_n\}$ 的递推关系, 其中 P_n 是第 n 年末账上的钱数。

b) 如果不允许取钱, n 年以后账上有多少钱?

6. 卢卡斯数满足递推关系

$$L_n = L_{n-1} + L_{n-2}$$

和初始条件 $L_0 = 2$ 和 $L_1 = 1$ 。

a) 证明 $L_n = f_{n-1} + f_{n+1}$, $n = 2, 3, \dots$, 其中 f_n 是第 n 个斐波那契数。

b) 求出卢卡斯数的显式公式。

7. 求解 $a_n = 7a_{n-2} + 6a_{n-3}$, $a_0 = 9$, $a_1 = 10$, $a_2 = 32$ 。

8. 求解 $a_n = 2a_{n-1} + 5a_{n-2} - 6a_{n-3}$, $a_0 = 7$, $a_1 = -4$, $a_2 = 8$ 。

9. 证明下述涉及斐波那契数和二项式系数的恒等式:

$$f_{n+1} = C(n, 0) + C(n-1, 1) + \dots + C(n-k, k)$$

其中 n 是正整数且 $k = \lfloor n/2 \rfloor$ 。[提示: 设 $a_n = C(n, 0) + C(n-1, 1) + \dots + C(n-k, k)$ 。证明序列 $\{a_n\}$ 和斐波那契序列满足的递推关系和初始条件一样。]

10. 求解递推关系 $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$, $a_0 = 5$, $a_1 = -9$, $a_2 = 15$ 。

11. 如果线性齐次递推关系的特征方程的根是 1, 1, 1, 1, -2, -2, -2, 3, 3, -4, 那么它的解的一般形式是什么?

12. 考虑非齐次线性递推关系 $a_n = 3a_{n-1} + 2^n$ 。

a) 证明 $a_n = -2^{n+1}$ 是这个递推关系的一个解。

b) 使用定理 5 找出这个递推关系的所有解。

c) 找出具有 $a_0 = 1$ 的解。

13. a) 确定常数 A 和 B 的值, 使得 $a_n = An + B$ 是递推关系 $a_n = 2a_{n-1} + n + 5$ 的一个解。

b) 使用定理 5 找出这个递推关系的所有解。

c) 找出这个递推关系具有 $a_0 = 4$ 的解。

14. 什么是线性非齐次递推关系 $a_n = 8a_{n-2} - 16a_{n-4} + F(n)$ 的特解的一般形式? 如果

$$a) F(n) = n^3$$

$$b) F(n) = (-2)^n$$

$$c) F(n) = n2^n$$

$$d) F(n) = n^2 4^n$$

$$e) F(n) = (n^2 - 2)(-2)^n$$

$$f) F(n) = n^4 2^n$$

$$g) F(n) = 2$$

15. a) 找出递推关系 $a_n = 2a_{n-1} + 3^n$ 的所有解。

b) 找出(a)中的递推关系具有初始条件 $a_1 = 5$ 的解。

16. 找出递推关系 $a_n = 5a_{n-1} - 6a_{n-2} + 2^n + 3n$ 的所有解。[提示: 找形如 $qn2^n + p_1n + p_2$ 的特解, 其中 q, p_1, p_2 是常数。]

17. 找出递推关系 $a_n = 4a_{n-1} - 4a_{n-2} + (n+1)2^n$ 的所有解。

18. 找出递推关系 $a_n = 4a_{n-1} - 3a_{n-2} + 2^n + n + 3$ 的具有 $a_0 = 1, a_1 = 4$ 的解。

19. 设 a_n 是前 n 个三角形数的和, 即 $a_n = \sum_{k=1}^n t_k$, 其中 $t_k = k(k+1)/2$. 证明 $\{a_n\}$ 满足线性非齐次递推关系 $a_n = a_{n-1} + n(n+1)/2$ 和初始条件 $a_1 = 1$. 使用定理 6 求解这个递推关系以确定关于 a_n 的公式.
- *20. a) 求线性齐次递推关系 $a_n = a_{n-4}$ 的特征根. (注: 这些根包含复数.)
 b) 求(a)的递推关系具有 $a_0 = 1, a_1 = 0, a_2 = -1$ 和 $a_3 = 1$ 的解.
- *21. a) 用例 4 找到的关于第 n 个斐波那契数 f_n 的公式证明 f_n 是最接近

$$\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n$$

的整数.

- b) 确定对哪些 n 有 f_n 大于

$$\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n$$

对哪些 n 有 f_n 小于

$$\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n$$

22. 假设留在岛上的每对遗传工程培育的兔子在一个月大时生出 2 对新兔子, 在两个月大和以后的每个月都生出 6 对新兔子. 没有兔子死去, 也没有兔子从岛上离开.
 a) 一对新生的兔子留在岛上, 求出与 n 个月后岛上兔子对数有关的递推关系.
 b) 通过求解(a)中的递推关系确定一对新生的兔子留在岛上 n 个月以后岛上的兔子对数.
23. 在一个充满活力的新软件公司, 一个新女雇员的初始工资为 50 000 美元, 公司允诺每年年底她的工资将是她前一年工资的 2 倍, 并且她在公司的每年都将额外增加 10 000 美元.
 a) 构造一个与被雇用的第 n 年她的工资数有关的递推关系.
 b) 求解这个递推关系找出她被雇用的第 n 年的工资.
- **24. 证明定理 4.
25. 求解具有初始条件 $T(1)=6$ 的递推关系 $T(n) = nT^2(n/2)$. [提示: 令 $n=2^k$, 然后做替换 $a_k = \log T(2^k)$ 以得到一个线性非齐次的递推关系.]

4.3 分治算法和递推关系

4.3.1 引言

许多递归算法把一个给定输入的问题划分成一个或多个小问题. 连续施用这种划分直到可以很快地找到这些较小问题的解. 例如, 在执行一个二分检索时把对一个元素在表中的搜索归约为对该元素在长度减半的表中的搜索. 我们继续施用这种分解直到只剩下一个元素. 当我们使用归并排序算法排序一个整数的数表时, 我们将这个表划分成相等大小的两半并且分别排序每一半. 然后将两个排好序的半个数表归并. 这种类型的递归算法的另一个例子就是整数乘法的过程, 它将两个整数相乘的问题分解成三组位数减半的整数相乘. 这种分解连续施用直到只剩下一位的整数为止. 这些过程叫做分治算法. 这是因为它们将一个问题的划分成较小规模的同一问题的一个或多个实例, 然后用这些小问题的解来处理这个问题以找到初始问题的解, 这当中也许需要一些额外的工作.

这一节将说明怎样用递推关系分析分治算法的计算复杂性. 我们将用这些递推关系估计许多不同的分治算法(包括我们在本节引入的算法)所使用的运算次数.

4.3.2 分治递推关系

假设一个递归算法把一个规模为 n 的问题分成 a 个子问题, 其中每个子问题的规模是 n/b (为简单起见, 假设 n 是 b 的倍数; 实际上, 较小的问题的规模常常是小于等于或者大于等于 n/b 的最近的整数). 此外, 假设在把子问题的解组合成原来问题的解的算法处理步中需要总量为 $g(n)$ 的额外的运算. 那么, 如果 $f(n)$ 表示求解规模为 n 的问题所需的运算数, 则得出 f 满足递推关系

$$f(n) = af(n/b) + g(n)$$

这就叫做分治递推关系。

首先我们将建立一些可用于研究某些重要算法的复杂性的分治递推关系。然后将说明怎样用这些分治递推关系估计这些算法的复杂性。

例1 二分搜索 当 n 是偶数时, 二分检索算法把对某个元素在长度为 n 的搜索序列中的搜索转变成对同一元素在长度为 $n/2$ 的搜索序列中的二分检索。(因此, 规模为 n 的问题已经被分解成规模为 $n/2$ 的问题。)为执行这个分解需要 2 次比较(一次是为了确定要用到表的哪一半, 另一次是为了确定表是否还有项留下来)。所以, 如果 $f(n)$ 是在规模为 n 的搜索序列中搜索一个元素所需要的比较次数, 那么当 n 是偶数时 $f(n) = f(n/2) + 2$ 。 ■

例2 找一个序列的最大和最小 考虑下面的查找序列 $a_1, a_2, \dots, a_n$ 中最小和最大元素的算法。如果 $n=1$, 那么 a_1 就是最大和最小的元素。如果 $n>1$, 把这个序列分成两个序列, 或者两者有同样多的元素, 或者一个集合比另一个集合多一个元素。问题就归约成查找两个较小序列的最大和最小元素。比较两个较小集合的最大和最小元素, 从而得到整体的最大和最小元素, 原问题的解就得到了。

设 $f(n)$ 是找 n 元素集合的最小和最大元素所需要的总的比较次数。我们已经说明了当 n 是偶数时一个规模为 n 的问题可以归约成两个规模为 $n/2$ 的问题, 这里要使用 2 次比较, 一次是比较两个集合的最小元素, 而另一次是比较两个集合的最大元素。当 n 是偶数时就得到递推关系 $f(n) = 2f(n/2) + 2$ 。 ■

例3 归并排序 归并排序算法把一个具有 n 个项(其中 n 为偶数)的待排序的表划分成两个表, 每个表 $n/2$ 个元素, 并且用少于 n 次的比较将两个排好序的表归并成一个排好序的表。因此, 通过归并排序算法排序 n 个元素的表用到的比较次数小于 $M(n)$, 其中函数 $M(n)$ 满足分治递推关系

$$M(n) = 2M(n/2) + n$$

例4 整数的快速乘法 令人惊讶的是存在许多比整数乘法的传统算法更有效的算法。这里描述的一个有效的算法就用到了分治技术。这个快速的乘法算法开始把每个 $2n$ 位的二进制整数分成两块, 每块 n 位。然后, 原来 $2n$ 位的二进制整数的乘法被分解成 3 个 n 位二进制数的乘法, 还要加上移位和加法。

假设 a 和 b 是两个整数的 $2n$ 位的二进制表达式(为了使得它们等长, 如果需要的话, 可在这些表达式前面加上若干个 0)。令

$$a = (a_{2n-1} a_{2n-2} \dots a_1 a_0)_2, \quad b = (b_{2n-1} b_{2n-2} \dots b_1 b_0)_2$$

令

$$a = 2^n A_1 + A_0, \quad b = 2^n B_1 + B_0$$

其中

$$A_1 = (a_{2n-1} \dots a_{n+1} a_n)_2, \quad A_0 = (a_{n-1} \dots a_1 a_0)_2$$

$$B_1 = (b_{2n-1} \dots b_{n+1} b_n)_2, \quad B_0 = (b_{n-1} \dots b_1 b_0)_2$$

快速整数乘法算法是基于恒等式

$$ab = (2^{2n} + 2^n)A_1 B_1 + 2^n(A_1 - A_0)(B_0 - B_1) + (2^n + 1)A_0 B_0$$

关于这个恒等式的一个重要的事实就是, 它证明了两个 $2n$ 位整数的乘法可以用 3 个 n 位整数的乘法加上加法、减法以及移位来实现。这证明了如果 $f(n)$ 是两个 n 位整数相乘所需的按位运算的总数, 那么

$$f(2n) = 3f(n) + Cn$$

这个等式成立的理由如下: 3 次 n 位整数的乘法可以使用 $3f(n)$ 次按位运算实现。每次加法、减法和移位使用的运算次数是 n 位运算的常数倍, 而 Cn 表示由这些运算用到的总的按位运算数。 ■

例5 快速矩阵乘法 使用矩阵乘法的定义进行两个 $n \times n$ 矩阵相乘需要 n^3 次乘法和 $n^2(n-1)$ 次加法。因此, 按照这种方法计算两个 $n \times n$ 矩阵之积需要 $O(n^3)$ 次运算(乘法和加法)。令人惊讶的是, 对于两个 $n \times n$ 矩阵相乘存在更有效的分治算法。这个由 V. 斯特拉森于

1969年提出的算法当 n 是偶数时将两个 $n \times n$ 矩阵的相乘归约为两个 $(n/2) \times (n/2)$ 矩阵的7次相乘和 $(n/2) \times (n/2)$ 矩阵的15次相加。(要了解这个算法的细节见[CoLeRiSt01]。)于是,如果 $f(n)$ 是用到的运算(乘法和加法)次数,那么当 n 是偶数时有

$$f(n) = 7f(n/2) + 15n^2/4$$

正如例1~例5所示,在许多不同的情况中都出现了形如 $f(n) = af(n/b) + g(n)$ 的递推关系。可以对满足这种递推关系的函数的阶作出估计。假设 f 满足这个递推关系,其中 n 可被 b 整除。令 $n = b^k$,其中 k 是一个正整数。那么

$$\begin{aligned} f(n) &= af(n/b) + g(n) \\ &= a^2 f(n/b^2) + ag(n/b) + g(n) \\ &= a^3 f(n/b^3) + a^2 g(n/b^2) + ag(n/b) + g(n) \\ &\vdots \\ &= a^k f(n/b^k) + \sum_{j=0}^{k-1} a^j g(n/b^j) \end{aligned}$$

由于 $n/b^k = 1$,从而有

$$f(n) = a^k f(1) + \sum_{j=0}^{k-1} a^j g(n/b^j)$$

我们可以使用这个关于 $f(n)$ 的等式估计满足分治关系的函数的阶。

定理1 设 f 是满足递推关系

$$f(n) = af(n/b) + c$$

的增函数,其中 n 被 b 整除, $a \geq 1$, b 是大于1的整数, c 是一个正实数。那么

$$f(n) \text{ 是 } \begin{cases} O(n^{\log_b a}) & \text{如果 } a > 1 \\ O(\log n) & \text{如果 } a = 1 \end{cases}$$

进一步,当 $n = b^k$,其中 k 是正整数,

$$f(n) = C_1 n^{\log_b a} + C_2$$

其中 $C_1 = f(1) + c/(a-1)$ 且 $C_2 = -c/(a-1)$ 。

证 首先令 $n = b^k$ 。由定理前面的讨论得到的关于 $f(n)$ 的表达式和 $g(n) = c$,我们有

$$f(n) = a^k f(1) + \sum_{j=0}^{k-1} a^j c = a^k f(1) + c \sum_{j=0}^{k-1} a^j$$

先考虑 $a=1$ 的情况。那么有

$$f(n) = f(1) + ck$$

由于 $n = b^k$,我们有 $k = \log_b n$,于是

$$f(n) = f(1) + c \log_b n$$

当 n 不是 b 的幂时,对某个正整数 k 有 $b^k < n < b^{k+1}$ 。由于 f 是增加的,故 $f(n) \leq f(b^{k+1}) = f(1) + c(k+1) = (f(1) + c) + ck \leq (f(1) + c) + c \log_b n$ 。因此,当 $a=1$ 时在两种情况下 $f(n)$ 都是 $O(\log n)$ 。

现在假设 $a > 1$ 。首先假定 $n = b^k$, k 是正整数。由几何级数的求和公式得

$$\begin{aligned} f(n) &= a^k f(1) + c(a^k - 1)/(a - 1) \\ &= a^k [f(1) + c/(a - 1)] - c/(a - 1) \\ &= C_1 n^{\log_b a} + C_2 \end{aligned}$$

这是由于 $a^k = a^{\log_b n} = n^{\log_b a}$,其中 $C_1 = f(1) + c/(a-1)$ 且 $C_2 = -c/(a-1)$ 。

现在假设 n 不是 b 的幂。那么 $b^k < n < b^{k+1}$,其中 k 是一个非负整数。由于 f 是增加的,

$$\begin{aligned} f(n) &\leq f(b^{k+1}) = C_1 a^{k+1} + C_2 \\ &\leq (C_1 a) a^{\log_b n} + C_2 \\ &\leq (C_1 a) n^{\log_b a} + C_2 \end{aligned}$$

这是由于 $k \leq \log_b n < k+1$ 。

于是, $f(n)$ 是 $O(n^{\log_a a})$ 。 □

例 6~9 说明怎样使用定理 1。

例 6 设 $f(n) = 5f(n/2) + 3$ 且 $f(1) = 7$, 求 $f(2^k)$, 其中 k 是一个正整数。如果 f 是一个增函数, 请估计 $f(n)$ 。

 **解** 从定理 1 的证明, 考虑 $a=5$, $b=2$, $c=3$, 我们看到如果 $n=2^k$, 那么

$$\begin{aligned} f(n) &= a^k [f(1) + c/(a-1)] + [-c/(a-1)] \\ &= 5^k [7 + (3/4)] - 3/4 \\ &= 5^k (31/4) - 3/4 \end{aligned}$$

又如果 $f(n)$ 是增加的, 定理 1 证明了 $f(n)$ 是 $O(n^{\log_a a}) = O(n^{\log_5 5})$ 。 ■

我们可以使用定理 1 估计二分检索算法和例 2 查找序列的最小和最大元素的算法的计算复杂性。

例 7 估计二分检索使用的比较次数。

解 在例 1 中证明了当 n 是偶数时 $f(n) = f(n/2) + 2$, 其中 $f(n)$ 是在规模为 n 的序列中实现一个二分检索需要的比较次数。因此得出 $f(n)$ 是 $O(\log n)$ 。 ■

例 8 估计用例 2 给定的算法查找序列的最大和最小元素所使用的比较次数。

解 在例 2 我们证明了当 n 是偶数时 $f(n) = 2f(n/2) + 2$, 其中 f 是算法需要的比较次数。于是, 由定理 1 得到 $f(n) = O(n^{\log_2 2}) = O(n)$ 。 ■

我们现在叙述一个更一般、更复杂的定理, 定理 1 是它的特例。这个定理(或者更强的版本)有时称作主定理(Master Theorem), 因为它在分析许多重要的分治算法的复杂性中很有用。

定理 2 主定理 设 f 是满足递推关系

$$f(n) = af(n/b) + cn^d$$

的增函数, 其中 $n=b^k$, k 是一个正整数, $a \geq 1$, b 是大于 1 的整数, c 和 d 是实数, 满足 c 是正的且 b 是非负的。那么

$$f(n) \text{ 是 } \begin{cases} O(n^d) & \text{如果 } a < b^d \\ O(n^d \log n) & \text{如果 } a = b^d \\ O(n^{\log_a a}) & \text{如果 } a > b^d \end{cases}$$

定理 2 的证明留给读者作为本节末的练习 11~13。

例 9 归并排序的复杂性 在例 3 我们解释了用归并排序来排序 n 个元素的表使用的比较次数少于 $M(n)$, 其中 $M(n) = 2M(n/2) + n$ 。根据主定理(定理 2)我们找到 $M(n)$ 是 $O(n \log n)$ 。 ■

例 10 估计使用例 4 描述的快速乘法算法做两个 n 位整数相乘所需要的按位运算次数。

解 例 4 证明了当 n 是偶数时 $f(n) = 3f(n/2) + Cn$, 其中 $f(n)$ 是使用快速乘法算法做两个 n 位整数相乘所需的按位运算次数。于是, 由定理 2 得到 $f(n)$ 是 $O(n^{\log_3 3})$ 。注意 $\log_3 3 \sim 1.6$ 。因为通常的乘法算法使用 $O(n^2)$ 次按位运算, 所以对于足够大的整数, 包括实际应用中出现的大整数, 快速乘法算法在时间复杂性方面比通常的算法有了本质的改进。 ■

例 11 估计使用例 5 的矩阵乘法算法做两个 $n \times n$ 矩阵相乘所需要的乘法和加法次数。

解 令 $f(n)$ 表示使用例 5 提到的算法做两个 $n \times n$ 矩阵相乘所需的加法和乘法次数。当 n 是偶数时我们有 $f(n) = 7f(n/2) + 15n^2/4$ 。于是由定理 2 得到 $f(n)$ 是 $O(n^{\log_7 7})$ 。注意 $\log_7 7 \sim 2.8$ 。由于通常的两个 $n \times n$ 矩阵相乘的算法要用 $O(n^3)$ 次加法和乘法, 显然, 对足够大的整数 n , 包括出现在许多实际应用中的大整数, 这个算法比通常的算法在时间复杂性方面更加有效。 ■

最接近点对问题 我们在结束这一节之前引入一个来自于计算几何的分治算法, 计算几何是离散数学的一部分, 专注于求解几何问题的算法。

 **例 12 最接近点对问题** 平面上有 n 个点 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ 的集合, 其中两点 (x_i, y_i) 和 (x_j, y_j) 之间的距离是通常的欧几里得距离

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

考虑确定其中最接近的点对的问题。这个问题见于许多应用中，例如确定在某航空控制中心管理的特定高度的空间最近的一对飞机。怎样以一种有效的方式找到这个最近的点对？

解 为解决这个问题，可以首先确定每对点的距离，然后找到这些距离的最小值。但是，这种方法需要 $O(n^2)$ 次的距离计算和比较，因为存在 $C(n, 2) = n(n-1)/2$ 个点对。不过存在一个精致的分治算法，对于 n 个点可以用 $O(n \log n)$ 次的距离计算和比较求解这个最接近点对问题。我们在这里描述的算法是归功于米凯尔·萨莫斯(见[PrSa85])。

为了简单起见，假设 $n = 2^k$ ，其中 k 是正整数。(我们避免某些当 n 不是 2 的幂时必需的技术上的考虑)。当 $n = 2$ 时，只有一对点；在这两个点之间的距离就是最小距离。

算法的递归部分将问题划分成两个子问题，每个涉及一半的点。第一步使用归并排序算法，根据这些点的 x 坐标对点进行排序。这需要 $O(n \log n)$ 次运算。一旦根据它们的 x 坐标对这些点排好序，我们画一条垂线 ℓ 将 n 个点分成两部分，左半部分和右半部分大小相等，每部分包含 $n/2$ 个点，如图 4-5 所示。(如果有任何点落到划分线 ℓ 上，必要的话，我们把它们分在这两部分里。)在后面的递归步骤我们不再需要根据 x 坐标排序，因为我们可以从所有的点中选择对应的排序子集。这个选择是可以用 $O(n)$ 次比较完成的任务。

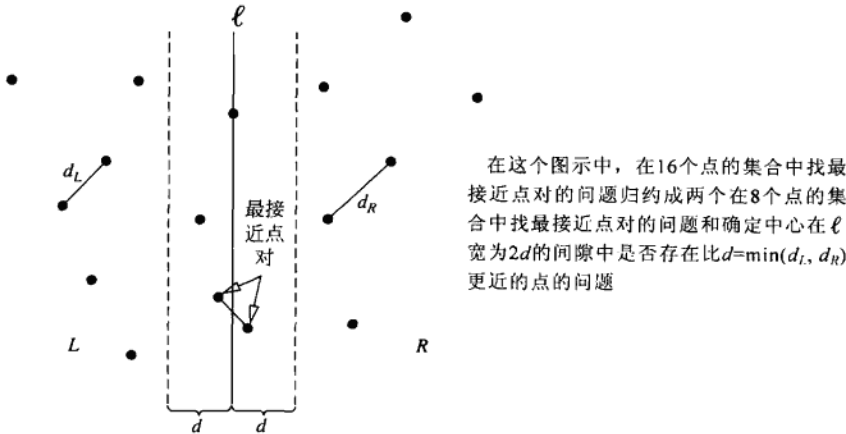


图 4-5 求解最接近点对问题的算法的递归步

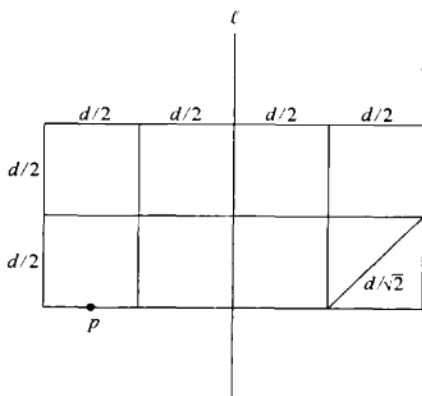
最接近的点对的位置有三种可能：(1) 它们两点都在左部区域 L ，(2) 它们两点都在右部区域 R ，(3) 一点在左部区域且另一点在右部区域。递归地使用这个算法计算 d_L 和 d_R ，其中 d_L 是在左部区域的点之间的最小距离， d_R 是在右部区域的点之间的最小距离，令 $d = \min(d_L, d_R)$ 。为了成功地将在原始集合找最接近点对的问题划分成在两个区域分别找最短距离的问题，我们必须处理算法的治之部分，这要求我们考虑最接近的点处在不同的区域的情况，即一点在 L 中，另一点在 R 中。因为存在一对点距离为 d ，或者它们都在 R 中，或者它们都在 L 中。对于分在不同区域的最接近的点，要求其距离一定小于 d 。

如果一点在左边区域，一点在右边区域且处在小于 d 的距离内，这些点一定位于宽度 $2d$ 的以 ℓ 作为其中心的垂直间隙中。(否则，这些点的距离一定大于它们的 x 坐标之差，而这个距离将超过 d 。)为了检查在这个间隙中的点，我们对它们进行排序并按照 y 坐标递增的顺序把它们列出来。这可以使用归并排序用 $O(n \log n)$ 次运算做到，并且只需在算法开始时做一次，而不是在每个递归步做。在每个递归步，我们从已经按照其 y 坐标排好序的所有点的集合，构造在这个区域内的根据其 y 坐标排序的点的子集，这可以用 $O(n)$ 次比较做到。

从间隙中具有最小 y 坐标的一个点开始，我们连续地检查间隙中的每个点，计算这个点与

隙中所有其他具有较大 y 坐标且与这个点的距离小于 d 的点之间的距离。注意到为检查点 p ，我们只需要考虑在 p 和下述矩形中的一组点之间的距离。这个矩形的高为 d ，宽为 $2d$ ， p 在它的底边上，并且它的垂直边与 l 的距离为 d 。

我们可以证明在这个点集中至多存在 8 个点，其中包含 p 在内(或者在这个 $2d \times d$ 的矩形的边上)。为了看到这一点，注意到在图 4-6 所示的 8 个 $d/2 \times d/2$ 的正方形中，每个正方形内部至多可能存在一个点。这是由于在一个正方形的边上或者内部最远距离的点是到对角线的长度 $d/\sqrt{2}$ (使用勾股定理可以得到)，这个距离小于 d ，并且每个这样的正方形是完全处在左区域或者右区域内。这意味着在这一步我们至多只需要与 d 比较 7 个距离，这些距离是在 p 和矩形内部或者边上 7 个或者更少的其他的点之间。



包含 p 在内，至多 8 个点可能处在中心在 l 的 $2d \times d$ 的矩形内或者边上，因为 8 个 $(d/2) \times (d/2)$ 的正方形中，每个内部或边上至多可能存在一个点。

图 4-6 说明对间隙中的每个点至多需要考虑另外 7 个点

由于在宽为 $2d$ 的间隙中的总点数不超过 n (集合中的总点数)，至多需要与 d 比较 $7n$ 个距离以找到点之间的最小距离。即只存在 $7n$ 个距离可能小于 d 。因此，一旦用归并排序按照这些点的 x 坐标和 y 坐标对它们进行排序以后，我们发现求解最接近点对问题需要的比较次数不超过满足递推关系

$$f(n) = 2f(n/2) + 7n$$

的增函数 $f(n)$ ，其中 $f(2)=1$ 。根据定理 2，得到 $f(n)$ 是 $O(n \log n)$ 。用归并排序算法根据点的 x 坐标和 y 坐标对点做两次排序，做每次排序用 $O(n \log n)$ 次比较，在算法的 $O(\log n)$ 个步中的每一步，这些坐标的排序子集每次可以用 $O(n)$ 次比较得到。因此，这个最接近点对问题可以用 $O(n \log n)$ 次比较求解。

练习

- 在 64 个元素的集合中，做二分检索需要多少次比较？
- 使用快速乘法算法乘 $(1110)_2$ 和 $(1010)_2$ 。
- 确定在例 4 中的常数 C 的值，并且使用它估计用快速乘法算法做两个 64 位二进制整数相乘所需要的按位运算次数。
- 假设当 n 被 3 整除时有 $f(n) = f(n/3) + 1$ 和 $f(1) = 1$ ，求
 - $f(3)$
 - $f(27)$
 - $f(729)$
- 假设当 n 被 5 整除时有 $f(n) = f(n/5) + 3n^2$ 和 $f(1) = 4$ ，求
 - $f(5)$
 - $f(125)$
 - $f(3125)$
- 假设 n 个投票人为不同的候选人(可能存在多于 2 个候选人)进入某个办公室投票，选票作为一个序列的元素。如果一个人得到的选票超过半数他就赢得竞选。
 - 设计一个分治算法确定是否一个候选人得到半数选票，如果是，则确定这个候选人是谁。[提示：设 n 为偶数，并且将选票序列划分成两个序列，每个序列具有 $n/2$ 个元素。注意到如果对于两个半长序列每一个都没有得到一半以上的选票，那么一个人就不可能得到所有选票的一半以上。]

b) 使用主定理估计你在(a)中设计的算法所需要的比较次数。

7. 设函数 f 满足递推关系 $f(n) = 2f(\sqrt{n}) + 1$, 其中 n 是大于 1 的完全平方数且 $f(2)=1$ 。

a) 求 $f(16)$ 。

b) 求关于 $f(n)$ 的大 O 估计。[提示: 做替换 $m = \log n$ 。]

**8. 这个练习涉及求 n 个实数序列的连续项的最大和问题。当所有的项都是正数时, 所有项之和就给出了答案, 但是当某些项是负数时情况就比较复杂了。例如, 序列 $-2, 3, -1, 6, -7, 4$ 的连续项的最大和是 $3 + (-1) + 6 = 8$ 。(这个练习基于 [Be86])。

a) 使用伪码描述一个求解该问题的算法, 这个算法依次找从第一项开始的连续项之和, 从第二项开始的连续项之和, 等等, 并在算法执行时记录当前找到的最大和。

b) 依照所做的计算的次数与比较次数确定在(a)中算法的计算复杂性。

c) 设计一个分治算法求解这个问题。[提示: 假设序列中有偶数个项, 把这个序列分成两半。解释当连续项的最大和包含了在两个半序列的项时怎样处理这种情况。]

d) 使用(c)中的算法求下面每个序列的连续项的最大和: $-2, 4, -1, 3, 5, -6, 1, 2; 4, 1, -3, 7, -1, -5, 3, -2; -1, 6, 3, -4, -5, 8, -1, 7$ 。

e) 对于(c)中的分治算法使用的求和次数和比较次数求一个递推关系。

f) 使用主定理估计这个分治算法的计算复杂性。依照计算复杂性把这个算法和(a)中的算法做比较, 结果如何?

9. 使用例 12 描述的求最接近点对的算法, 使用点之间的欧几里得距离, 求下述点 $(1, 2), (1, 6), (2, 4), (2, 8), (3, 1), (3, 6), (3, 10), (4, 3), (5, 1), (5, 5), (5, 9), (6, 7), (7, 1), (7, 4), (7, 9), (8, 6)$ 的最接近点对。

10. 如果两点间的距离定义是 $d((x_i, y_i), (x_j, y_j)) = \max(|x_i - x_j|, |y_i - y_j|)$, 使用例 12 描述的算法中用的那些合理的步骤并加以改变, 构造一个求两点之间的最小距离的算法。

在练习 11~13 中, 假设 f 是一个满足递推关系 $f(n) = af(n/b) + cn^d$ 的增函数, $a \geq 1$, b 是大于 1 的整数, c 和 d 是正实数。这些练习提供一个关于定理 2 的证明。

*11. 证明: 如果 $a = b^d$ 且 n 是 b 的幂, 那么 $f(n) = f(1)n^d + cn^d \log_b n$ 。

*12. 证明: 如果 $a \neq b^d$ 且 n 是 b 的幂, 那么 $f(n) = C_1 n^d + C_2 n^{\log_b a}$, 其中

$$C_1 = b^d c / (b^d - a) \text{ 且 } C_2 = f(1) + b^d c / (a - b^d)$$

13. 使用练习 12 证明: 如果 $a > b^d$, 那么 $f(n)$ 是 $O(n^{\log_b a})$ 。

4.4 生成函数

4.4.1 引言

表示序列的一种有效方法就是生成函数, 它把序列的项作为一个形式幂级数中变量 x 的幂的系数。可以用生成函数求解许多类型的计数问题, 例如在各种限制下选取或分配不同种类物体的方式数, 使用不同面额的硬币换一美元的方式数等。也可以用生成函数求解递推关系。它先把关于序列的项的递推关系转换成涉及生成函数的方程, 然后求解这个方程并找出关于这个生成函数的封闭形式。从这个封闭形式可以找到生成函数的幂级数的系数, 从而求解了原来的递推关系。生成函数也可以利用函数之间相对简单的关系来证明组合恒等式, 由于这些关系可以转换成涉及序列的项的恒等式。生成函数是有用的工具, 除了本节描述的内容以外, 可以用它来研究序列的许多性质, 例如建立关于序列的项的渐进公式。

我们从序列的生成函数的定义开始。

定义 1 实数序列 $a_0, a_1, \dots, a_k, \dots$ 的生成函数是无穷级数

$$G(x) = a_0 + a_1 x + \dots + a_k x^k + \dots = \sum_{k=0}^{\infty} a_k x^k$$

注意 定义 1 给出的 $\{a_k\}$ 的生成函数有时叫做 $\{a_k\}$ 的普通的生成函数, 以和这个序列的其他类型的生成函数相区别。

例 1 序列 $\{a_k\}$ 具有 $a_k = 3$, $a_k = k+1$ 和 $a_k = 2^k$ 的生成函数分别是 $\sum_{k=0}^{\infty} 3x^k$, $\sum_{k=0}^{\infty} (k+1)x^k$, $\sum_{k=0}^{\infty} 2^k x^k$.

我们通过置 $a_{n+1}=0$, $a_{n+2}=0$, 依此下去, 把一个有限序列 $a_0, a_1, \dots, a_n$ 扩充成一个无限序列, 就可以定义一个实数的有限序列的生成函数。这个无限序列 $\{a_n\}$ 的生成函数 $G(x)$ 是一个 n 次多项式, 因为当 $j>n$ 时没有形如 $a_j x^j$ 的项出现, 即

$$G(x) = a_0 + a_1 x + \dots + a_n x^n$$

例 2 序列 1, 1, 1, 1, 1, 1 的生成函数是什么?

解 1, 1, 1, 1, 1, 1 的生成函数是

$$1 + x + x^2 + x^3 + x^4 + x^5$$

由 2.4 节的定理 1 有

$$(x^6 - 1)/(x - 1) = 1 + x + x^2 + x^3 + x^4 + x^5$$

因此, $G(x) = (x^6 - 1)/(x - 1)$ 是序列 1, 1, 1, 1, 1, 1 的生成函数。

例 3 设 m 是正整数。令 $a_k = C(m, k)$, $k=0, 1, 2, \dots, m$ 。那么序列 $a_0, a_1, \dots, a_m$ 的生成函数是什么?

解 这个序列的生成函数是

$$G(x) = C(m, 0) + C(m, 1)x + C(m, 2)x^2 + \dots + C(m, m)x^m$$

二项式定理证明 $G(x) = (1+x)^m$ 。

4.4.2 关于幂级数的有用事实

当用生成函数求解计数问题时, 通常将它们考虑成形式幂级数。这里忽略了这些级数的收敛问题。但是为了应用某些微积分的结果, 考虑幂级数对什么 x 收敛有时是很重要的。在我们的讨论中将不涉及收敛性问题。熟悉微积分的读者如果想了解所涉及级数的收敛性的细节, 可以参阅有关这方面内容的教科书。

现在我们将叙述某些与无穷级数有关的重要事实, 这些将在研究生成函数时用到。这些事实的讨论和相关的结果都可以在微积分教科书中找到。

例 4 函数 $f(x) = 1/(1-x)$ 是序列 1, 1, 1, 1, $\dots$ 的生成函数, 因为对 $|x| < 1$ 有

$$1/(1-x) = 1 + x + x^2 + \dots$$

例 5 函数 $f(x) = 1/(1-ax)$ 是序列 1, a , a^2 , a^3 , $\dots$ 的生成函数, 因为当 $|ax| < 1$ 或等价地说对 $|x| < 1/|a|$, $a \neq 0$ 有

$$1/(1-ax) = 1 + ax + a^2 x^2 + \dots$$

我们也需要了解两个生成函数是怎样相加和相乘的。这些结果的证明也可以在微积分教科书中找到。

定理 1 令 $f(x) = \sum_{k=0}^{\infty} a_k x^k$, $g(x) = \sum_{k=0}^{\infty} b_k x^k$, 那么

$$f(x) + g(x) = \sum_{k=0}^{\infty} (a_k + b_k) x^k$$

$$f(x)g(x) = \sum_{k=0}^{\infty} \left(\sum_{j=0}^k a_j b_{k-j} \right) x^k$$

注意 正如本节所考虑的所有级数一样, 定理 1 只有当幂级数在一个区间内收敛时才有效。但是, 生成函数的定理并不局限于这种级数。在级数不收敛的情况下, 定理 1 中的命题可以看成是生成函数和与积的定义。

我们将在例 6 中说明怎样使用定理 1。

例6 设 $f(x)=1/(1-x)^2$. 用例4求出表达式 $f(x)=\sum_{k=0}^{\infty} a_k x^k$ 中的系数 $a_0, a_1, a_2, \dots$.

解 由例4看出

$$1/(1-x) = 1 + x + x^2 + x^3 + \dots$$

因此, 由定理1有

$$1/(1-x)^2 = \sum_{k=0}^{\infty} \left(\sum_{j=0}^k 1 \right) x^k = \sum_{k=0}^{\infty} (k+1) x^k$$

注意 这一结果也可以通过微分从例4导出. 从已知生成函数的恒等式产生新的恒等式的一种有用的技术就是求导.

为了用生成函数求解许多重要的计数问题, 我们需要在指数不是正整数的情况下应用二项式定理. 在叙述广义二项式定理之前, 我们需要定义广义二项式系数.

定义2 设 u 是实数且 k 是非负整数, 那么广义二项式系数 $\binom{u}{k}$ 定义为

$$\binom{u}{k} = \begin{cases} u(u-1)\cdots(u-k+1)/k! & \text{若 } k > 0 \\ 1 & \text{若 } k = 0 \end{cases}$$

例7 求广义二项式系数 $\binom{-2}{3}$ 和 $\binom{1/2}{3}$ 的值.

解 在定义2中取 $u=-2$ 和 $k=3$ 得

$$\binom{-2}{3} = \frac{(-2)(-3)(-4)}{3!} = -4$$

类似地取 $u=1/2$ 和 $k=3$ 得

$$\begin{aligned} \binom{1/2}{3} &= \frac{(1/2)(1/2-1)(1/2-2)}{3!} \\ &= (1/2)(-1/2)(-3/2)/6 \\ &= 1/16 \end{aligned}$$

当上边的参数是负整数时, 例8对广义二项式系数提供了一个有用的公式. 我们后面的讨论中会用到它.

例8 当上边的参数是负整数时, 广义二项式系数可以用通常的二项式系数的项表示. 为此只需注意到

$$\begin{aligned} \binom{-n}{r} &= \frac{(-n)(-n-1)\cdots(-n-r+1)}{r!} && \text{由广义二项式系数定义} \\ &= \frac{(-1)^r n(n+1)\cdots(n+r-1)}{r!} && \text{从分子的每一项中提取因子 } -1 \\ &= \frac{(-1)^r (n+r-1)(n+r-2)\cdots n}{r!} && \text{由乘法的交换律} \\ &= \frac{(-1)^r (n+r-1)!}{r!(n-1)!} && \text{分子和分母同时乘以 } (n-1)! \\ &= (-1)^r \binom{n+r-1}{r} && \text{由二项式系数的定义} \\ &= (-1)^r C(n+r-1, r) && \text{使用另外一种二项式系数符号表示} \end{aligned}$$

我们现在叙述广义二项式定理.

定理2 广义二项式定理 设 x 是实数, $|x| < 1$, u 是实数, 那么

$$(1+x)^u = \sum_{k=0}^{\infty} \binom{u}{k} x^k$$

可以使用麦克劳林级数的理论证明定理2, 我们将这个证明留给熟悉这部分微积分的读者完成。

注意 当 u 是正整数时, 广义二项式定理就归约到 3.4 节提出的二项式定理, 因为如果 $k > u$, 在这种情况下 $\binom{u}{k} = 0$ 。

例9说明了当指数是负整数时定理2的应用。

例9 当 n 是正整数时, 使用广义二项式定理求 $(1+x)^{-n}$ 和 $(1-x)^{-n}$ 的生成函数。

解 由广义二项式定理得

$$(1+x)^{-n} = \sum_{k=0}^{\infty} \binom{-n}{k} x^k$$

使用例8所提供的关于 $\binom{-n}{k}$ 的简单公式得到

$$(1+x)^{-n} = \sum_{k=0}^{\infty} (-1)^k C(n+k-1, k) x^k$$

用 $-x$ 代替 x 得到

$$(1-x)^{-n} = \sum_{k=0}^{\infty} C(n+k-1, k) x^k$$

表4-1归纳了一些经常出现的有用的生成函数。

表4-1 有用的生成函数

$G(x)$	a_k
$(1+x)^n = \sum_{k=0}^n C(n, k) x^k$ $= 1 + C(n, 1)x + C(n, 2)x^2 + \cdots + x^n$	$C(n, k)$
$(1+ax)^n = \sum_{k=0}^n C(n, k) a^k x^k$ $= 1 + C(n, 1)ax + C(n, 2)a^2x^2 + \cdots + a^n x^n$	$C(n, k) a^k$
$(1+x^r)^n = \sum_{k=0}^n C(n, k) x^{rk}$ $= 1 + C(n, 1)x^r + C(n, 2)x^{2r} + \cdots + x^{rn}$	当 r/k 时为 $C(n, k/r)$, 否则为 0
$\frac{1-x^{n+1}}{1-x} = \sum_{k=0}^n x^k = 1 + x + x^2 + \cdots + x^n$	当 $k \leq n$ 时为 1, 否则为 0
$\frac{1}{1-x} = \sum_{k=0}^{\infty} x^k = 1 + x + x^2 + \cdots$	1
$\frac{1}{1-ax} = \sum_{k=0}^{\infty} a^k x^k = 1 + ax + a^2 x^2 + \cdots$	a^k
$\frac{1}{1-x^r} = \sum_{k=0}^{\infty} x^{rk} = 1 + x^r + x^{2r} + \cdots$	当 r/k 时为 1, 否则为 0
$\frac{1}{(1-x)^2} = \sum_{k=0}^{\infty} (k+1)x^k = 1 + 2x + 3x^2 + \cdots$	$k+1$
$\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty} C(n+k-1, k) x^k$ $= 1 + C(n, 1)x + C(n+1, 2)x^2 + \cdots$	$C(n+k-1, k) = C(n+k-1, n-1)$
$\frac{1}{(1+x)^n} = \sum_{k=0}^{\infty} C(n+k-1, k) (-1)^k x^k$ $= 1 - C(n, 1)x + C(n+1, 2)x^2 - \cdots$	$(-1)^k C(n+k-1, k) = (-1)^k C(n+k-1, n-1)$

(续)

$G(x)$	a_k
$\frac{1}{(1-ax)^n} = \sum_{k=0}^{\infty} C(n+k-1, k) a^k x^k$ $= 1 + C(n, 1)ax + C(n+1, 2)a^2x^2 + \cdots$	$C(n+k-1, k)a^k = C(n+k-1, n-1)a^k$
$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$	$1/k!$
$\ln(1+x) = \sum_{k=0}^{\infty} \frac{(-1)^{k+1}}{k} x^k = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots$	$(-1)^{k+1}/k$

注：在大多数微积分的书讨论幂级数的部分可以找到关于最后两个生成函数的级数。

4.4.3 计数问题与生成函数

生成函数可以用于求解各种计数问题。特别地，它们可以用于计数各种类型的组合数。在第3章，我们开发了一些技术计数 n 元素集合的允许重复的 r 组合数，在这些组合中可能存在某些附加的约束。这种问题与计数形如

$$e_1 + e_2 + \cdots + e_n = C$$

方程的解是等价的，其中 C 是常数，每个 e_i 是可能具有某些约束的非负整数。也可以用生成函数求解这种类型的计数问题，正如例10~例12所示。

例10 求

$$e_1 + e_2 + e_3 = 17$$

的解的个数，其中 e_1, e_2, e_3 是非负整数，满足 $2 \leq e_1 \leq 5, 3 \leq e_2 \leq 6, 4 \leq e_3 \leq 7$ 。

解 具有上述限制的解的个数是

$$(x^2 + x^3 + x^4 + x^5)(x^3 + x^4 + x^5 + x^6)(x^4 + x^5 + x^6 + x^7)$$

的展开式中 x^{17} 的系数。这是因为我们在乘积中得到等于 x^{17} 的项是通过在第一个和中取项 x^4 ，在第二个和中取项 x^6 ，在第三个和中取项 x^7 ，其中幂指数 e_1, e_2 和 e_3 满足方程 $e_1 + e_2 + e_3 = 17$ 和给定的限制。

不难看出在这个乘积中的 x^{17} 的系数是3。因此，存在3个解。（注意，计算这个系数与枚举方程的具有给定约束的所有解几乎要做同样多的工作。但是，正如我们将看到的，这里说明的方法常常可以用于求解各种各样的具有特殊规则的计数问题。此外，可以用计算机代数系统做这种计算。）

例11 把8块相同的饼干分给3个不同的孩子，如果每个孩子至少接受2块饼干并且不超过4块饼干，那么有多少种不同的分配方式？

解 因为每个孩子至少接受2块饼干且不超过4块饼干，在关于序列 $\{C_n\}$ 的生成函数中对每个孩子存在一个等于

$$(x^2 + x^3 + x^4)$$

的因式，其中 c_n 是分配 n 块饼干的方式数。因为存在3个孩子，这个生成函数是

$$(x^2 + x^3 + x^4)^3$$

我们要求这个乘积中的 x^8 的系数。理由就是在展开式中 x^8 的项对应于选3项的方式数，其中每个因式选1项且指数加起来等于8。此外，来自第一、第二和第三个因式的项的指数分别是第一、第二和第三个孩子接受的饼干数。通过计算说明这个系数等于6。于是存在6种方式分配饼干使得每个孩子至少接受2块，但是不超过4块饼干。

例12 把价值1美元、2美元和5美元的代币插入售货机为价值 r 美元的某种物品付款，使用生成函数确定在代币插入是有序的和无序的两种情况下付款的方式数。（例如为一种价值3美元的物品付款，当不考虑代币插入的次序时存在2种方式：插入3个1美元的代币或1个1美元和1个2美元的代币。当考虑代币插入的次序时有3种方式：插入3个1美元的代币，插入1个1美元代

币然后1个2美元的代币,插入1个2美元代币然后1个1美元代币。)

解 在不考虑代币插入次序的情况下,我们所关心的就是为产生 r 美元的总数所使用的每种代币的数目。因为可以使用任意多个1美元的代币、任意多个2美元的代币和任意多个5美元的代币,答案就是在生成函数

$$(1+x+x^2+x^3+\cdots)(1+x^2+x^4+x^6+\cdots)(1+x^5+x^{10}+x^{15}+\cdots)$$

中的 x^r 的系数。(这个乘积中的第一个因式表示所使用的1美元代币,第二个表示所使用的2美元代币,第三个表示所使用的5美元代币。)例如,用1美元、2美元和5美元为一个价值7美元的物品付款的方式数由展开式中 x^7 的系数给出,结果等于6。

当考虑代币插入的次序时,插入恰好 n 个代币产生 r 美元总数的方式数是在

$$(x+x^2+x^5)^n$$

中的 x^r 的系数,因为这 n 个代币中的每一个可能是1美元代币、2美元代币或5美元代币。又由于可以插入的代币不限数量,因此当考虑代币插入的次序时,使用1美元、2美元或5美元代币产生 r 美元的方式数是在

$$\begin{aligned} 1+(x+x^2+x^5)+(x+x^2+x^5)^2+\cdots &= \frac{1}{1-(x+x^2+x^5)} \\ &= \frac{1}{1-x-x^2-x^5} \end{aligned}$$

中 x^r 的系数。这里我们把插入0个代币、1个代币、2个代币、3个代币等方式数加起来,同时我们使用了恒等式 $1/(1-x)=1+x+x^2+\cdots$ 且用 $x+x^2+x^5$ 代替了 x 。例如,用1美元、2美元和5美元的代币为一个价值7美元的物品付款,当考虑使用代币的次序时,方式数是这个展开式中 x^7 的系数,等于26。[为看到这个系数等于26,要把 $(x+x^2+x^5)^k$ 的展开式中 x^7 的系数加起来,其中 $2 \leq k \leq 7$ 。这项工作可以用大量的手工计算完成,也可以使用一个计算机代数系统来完成。] ■

例13说明了当求解带不同假设的问题时生成函数具有的多功能性。

例13 假设已经建立了二项式定理,使用生成函数找出 n 元素集合的 k 组合数。

解 集合的 n 个元素中的每一个元素都对生成函数 $f(x) = \sum_{k=0}^n a_k x^k$ 贡献了项 $(1+x)$ 。因此 $f(x)$ 是关于 $\{a_k\}$ 的生成函数,其中 a_k 表示 n 元素集合的 k 组合数。于是,

$$f(x) = (1+x)^n$$

但是由二项式定理,我们有

$$f(x) = \sum_{k=0}^n \binom{n}{k} x^k$$

其中

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

于是, $C(n, k)$, n 元素集合的 k 组合数是

$$\frac{n!}{k!(n-k)!}$$

注意 在3.4节我们使用了关于 n 元素集合的 r 组合数的公式证明了二项式定理。这些例子说明也可以用数学归纳法证明二项式定理,再用二项式定理推导关于 n 元素集合的 r 组合数的公式。

例14 使用生成函数找出当元素允许重复时 n 元素集合的 r 组合数公式。

解 设 $G(x)$ 是关于序列 $\{a_r\}$ 的生成函数,其中 a_r 等于 n 元素集合的允许重复的 r 组合数,

$G(x) = \sum_{r=0}^{\infty} a_r x^r$, 即当我们构成允许重复的 r 组合时,对 n 元素集合的元素的选择不受限制,这 n 个元素中的每一个元素都对 $G(x)$ 的乘积展开式贡献了 $(1+x+x^2+x^3+\cdots)$ 这个因式。这是由于当

一个 r 组合被构成时(要选总共 r 个元素), 每个元素都可以被选择 0 次、1 次、2 次、3 次, 等等。因为集合中存在 n 个元素, 且每一个都对 $G(x)$ 贡献了相同的因式, 从而有

$$G(x) = (1 + x + x^2 + \cdots)^n$$

只要 $|x| < 1$, 我们就有 $1 + x + x^2 + \cdots = 1/(1-x)$, 所以

$$G(x) = 1/(1-x)^n = (1-x)^{-n}$$

使用广义二项式定理(定理 2), 得到

$$(1-x)^{-n} = (1+(-x))^{-n} = \sum_{r=0}^{\infty} \binom{-n}{r} (-x)^r$$

当 r 是正整数时, n 元素集合的允许重复的 r 组合数就是这个和式中的 x^r 的系数。因此, 使用例 8 我们求出 a_r 等于

$$\begin{aligned} \binom{-n}{r} (-1)^r &= (-1)^r C(n+r-1, r) \cdot (-1)^r \\ &= C(n+r-1, r) \end{aligned}$$

注意, 例 14 的结果与我们在 3.5 节定理 2 所叙述的结果一样。

例 15 使用生成函数求出从 n 类不同的物体中选择 r 个物体并且每类物体至少选 1 个的方式数。

解 因为我们需要每类物体至少选 1 个, 这 n 个类中的每类物体都对序列 $\{a_r\}$ 的生成函数 $G(x)$ 贡献了因式 $(x+x^2+x^3+\cdots)$ 其中 a_r 是从 n 类不同的物体中选择 r 个物体并且每类物体至少选 1 个的方式数。因此,

$$G(x) = (x + x^2 + x^3 + \cdots)^n = x^n (1 + x + x^2 + \cdots)^n = x^n / (1-x)^n$$

使用广义二项式定理和例 8, 有

$$\begin{aligned} G(x) &= x^n / (1-x)^n \\ &= x^n \cdot (1-x)^{-n} \\ &= x^n \sum_{r=0}^{\infty} \binom{-n}{r} (-x)^r \\ &= x^n \sum_{r=0}^{\infty} (-1)^r C(n+r-1, r) (-1)^r x^r \\ &= \sum_{r=0}^{\infty} C(n+r-1, r) x^{n+r} \\ &= \sum_{t=n}^{\infty} C(t-1, t-n) x^t \\ &= \sum_{r=n}^{\infty} C(r-1, r-n) x^r \end{aligned}$$

在倒数第二个等式我们令 $t=n+r$ 以使得当 $r=0$ 时 $t=n$ 且 $n+r-1=t-1$, 从而对求和进行了移位, 然后在最后的等式中用 r 替换 t 作为和的下标而回到了初始的记号。因此, 从 n 类不同的物体中选择 r 个物体, 如果每类物体必须至少选 1 个时存在 $C(r-1, r-n)$ 种方式。

4.4.4 使用生成函数求解递推关系

我们可以通过找相关的生成函数的显式公式来求解关于递推关系和初始条件的解。这可以用例 16 和例 17 来说明。

例 16 求解递推关系 $a_k = 3a_{k-1}$, $k=1, 2, 3, \cdots$ 且初始条件 $a_0=2$ 。

解 设 $G(x)$ 是序列 $\{a_k\}$ 的生成函数, 即 $G(x) = \sum_{k=0}^{\infty} a_k x^k$ 。首先注意到

$$xG(x) = \sum_{k=0}^{\infty} a_k x^{k+1} = \sum_{k=1}^{\infty} a_{k-1} x^k$$

使用递推关系有

$$\begin{aligned} G(x) - 3xG(x) &= \sum_{k=0}^{\infty} a_k x^k - 3 \sum_{k=1}^{\infty} a_{k-1} x^k \\ &= a_0 + \sum_{k=1}^{\infty} (a_k - 3a_{k-1}) x^k \\ &= 2 \end{aligned}$$

因为 $a_0 = 2$ 且 $a_k = 3a_{k-1}$, 于是

$$G(x) - 3xG(x) = (1 - 3x)G(x) = 2$$

求解 $G(x)$ 得 $G(x) = 2/(1 - 3x)$ 。使用表 6-1 的恒等式 $1/(1 - ax) = \sum_{k=0}^{\infty} a^k x^k$, 我们有

$$G(x) = 2 \sum_{k=0}^{\infty} 3^k x^k = \sum_{k=0}^{\infty} 2 \cdot 3^k x^k$$

于是, $a_k = 2 \cdot 3^k$ 。

例 17 设一个有效的编码字是一个包含偶数个 0 的十进制数字串。令 a_n 表示 n 位有效编码字的个数。在 4.1 节的例 7 中我们证明了序列 $\{a_n\}$ 满足递推关系

$$a_n = 8a_{n-1} + 10^{n-1}$$

且初始条件 $a_1 = 9$ 。使用生成函数找出关于 a_n 的显式公式。

解 为了简化关于生成函数的推导, 我们通过置 $a_0 = 1$ 将序列扩充, 当把这个值赋给 a_0 并且使用递推关系就得到 $a_1 = 8a_0 + 10^0 = 8 + 1 = 9$, 这与初始条件一致。(由于存在一个长为 0 的编码字——空串, 这也是有意义的。)

用 x^n 乘递推关系的两边得

$$a_n x^n = 8a_{n-1} x^n + 10^{n-1} x^n$$

设 $G(x) = \sum_{n=0}^{\infty} a_n x^n$ 是序列 $a_0, a_1, a_2, \dots$ 的生成函数, 从 $n=1$ 开始对上面的等式两边求和, 得到

$$\begin{aligned} G(x) - 1 &= \sum_{n=1}^{\infty} a_n x^n = \sum_{n=1}^{\infty} (8a_{n-1} x^n + 10^{n-1} x^n) \\ &= 8 \sum_{n=1}^{\infty} a_{n-1} x^n + \sum_{n=1}^{\infty} 10^{n-1} x^n \\ &= 8x \sum_{n=1}^{\infty} a_{n-1} x^{n-1} + x \sum_{n=1}^{\infty} 10^{n-1} x^{n-1} \\ &= 8x \sum_{n=0}^{\infty} a_n x^n + x \sum_{n=0}^{\infty} 10^n x^n \\ &= 8xG(x) + x/(1 - 10x) \end{aligned}$$

其中我们已经使用了例 5 对第二个和进行求值。因此有

$$G(x) - 1 = 8xG(x) + x/(1 - 10x)$$

求解 $G(x)$ 得

$$G(x) = \frac{1 - 9x}{(1 - 8x)(1 - 10x)}$$

把等式的右边展开成部分分式(正如在微积分中研究有理函数的积分时所做的)得到

$$G(x) = \frac{1}{2} \left(\frac{1}{1 - 8x} + \frac{1}{1 - 10x} \right)$$

两次使用例 5(一次设 $a=8$, 一次设 $a=10$) 得

$$\begin{aligned} G(x) &= \frac{1}{2} \left(\sum_{n=0}^{\infty} 8^n x^n + \sum_{n=0}^{\infty} 10^n x^n \right) \\ &= \sum_{n=0}^{\infty} \frac{1}{2} (8^n + 10^n) x^n \end{aligned}$$

于是,证明了

$$a_n = \frac{1}{2}(8^n + 10^n)$$

4.4.5 使用生成函数证明恒等式

在第3章我们已经看到怎样使用组合证明方法来建立组合恒等式。这里将显示这种恒等式,还有关于广义二项式系数的恒等式,都可以使用生成函数来证明。有时候生成函数的方法比其他方法更简单,特别是用生成函数的封闭形式比使用序列本身更能简化证明过程。我们用例18说明怎样用生成函数证明恒等式。

例18 使用生成函数证明

$$\sum_{k=0}^n C(n, k)^2 = C(2n, n)$$

其中 n 是正整数。

解 首先注意到根据二项式定理 $C(2n, n)$ 是 $(1+x)^{2n}$ 中 x^n 的系数。而我們也有

$$\begin{aligned}(1+x)^{2n} &= [(1+x)^n]^2 \\ &= [C(n, 0) + C(n, 1)x + C(n, 2)x^2 + \cdots + C(n, n)x^n]^2\end{aligned}$$

在这个展开式中 x^n 的系数是 $C(n, 0)C(n, n) + C(n, 1)C(n, n-1) + C(n, 2)C(n, n-2) + \cdots + C(n, n)C(n, 0)$ 。因为 $C(n, n-k) = C(n, k)$, 它等于 $\sum_{k=0}^n C(n, k)^2$ 。由于 $C(2n, n)$ 和 $\sum_{k=0}^n C(n, k)^2$ 都表示 $(1+x)^{2n}$ 中 x^n 的系数, 它们一定是相等的。

本节末的练习22要求用生成函数来证明帕斯卡恒等式和范德蒙德恒等式。

练习

1. 求关于有穷序列 2, 2, 2, 2, 2, 2 的生成函数。

在练习2~4中, 封闭形式意味着代数表达式不仅包括所列数值的和, 还包括省略部分。

2. 求关于下面每个序列生成函数的封闭形式。(用最明显的选择设定每个序列的通项形式。)

a) 0, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, ...

b) 0, 0, 0, 1, 1, 1, 1, 1, 1, ...

c) 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, ...

d) 2, 4, 8, 16, 32, 64, 128, 256, ...

e) $\binom{7}{0}, \binom{7}{1}, \binom{7}{2}, \dots, \binom{7}{7}, 0, 0, 0, 0, 0, \dots$

f) 2, -2, 2, -2, 2, -2, 2, -2, ...

g) 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, ...

h) 0, 0, 0, 1, 2, 3, 4, ...

3. 求关于序列 $\{a_n\}$ 的生成函数的封闭形式, 其中

a) $a_n = 5$, 对所有的 $n = 0, 1, 2, \dots$

b) $a_n = 3^n$, 对所有的 $n = 0, 1, 2, \dots$

c) $a_n = 2$, 对 $n = 3, 4, 5, \dots$ 且 $a_0 = a_1 = a_2 = 0$

d) $a_n = 2n + 3$, 对所有的 $n = 0, 1, 2, \dots$

e) $a_n = \binom{8}{n}$, 对所有的 $n = 0, 1, 2, \dots$

f) $a_n = \binom{n+4}{n}$, 对所有的 $n = 0, 1, 2, \dots$

4. 对于下面每一个生成函数给出关于它所确定序列的封闭公式。

a) $(3x-4)^3$

b) $(x^3+1)^3$

c) $1/(1-5x)$

d) $x^3/(1+3x)$

e) $x^2 + 3x + 7 + 1/(1-x^2)$

f) $(x^4/(1-x^4)) - x^3 - x^2 - x - 1$

g) $x^2/(1-x)^2$

h) $2e^{2x}$

5. 求出下面每个函数的幂级数中 x^{10} 的系数。

a) $(1+x^5+x^{10}+x^{15}+\dots)^3$

b) $(x^3+x^4+x^5+x^6+x^7+\dots)^3$

c) $(x^4+x^5+x^6)(x^3+x^4+x^5+x^6+x^7)(1+x+x^2+x^3+x^4+\dots)$

d) $(x^2+x^4+x^6+x^8+\dots)(x^3+x^6+x^9+\dots)(x^4+x^8+x^{12}+\dots)$

e) $(1+x^2+x^4+x^6+x^8+\dots)(1+x^4+x^8+x^{12}+\dots)(1+x^6+x^{12}+x^{18}+\dots)$

6. 求出下面每个函数的幂级数中 x^{10} 的系数。
 a) $1/(1-2x)$ b) $1/(1+x)^2$ c) $1/(1-x)^3$ d) $1/(1+2x)^4$ e) $x^4/(1-3x)^3$
7. 把 10 个相同的球分给 4 个孩子, 如果每个孩子至少得到 2 个球, 使用生成函数确定不同的分法数。
8. 把 15 个相同的动物玩具分给 6 个孩子使得每个孩子至少得到 1 个但不超过 3 个, 使用生成函数确定不同的分法数。
9. 把 25 个相同的多纳圈分给 4 个警官使得每个警官至少得到 3 个但不超过 7 个, 有多少种方式?
10. 求序列 $\{c_k\}$ 的生成函数, 其中 c_k 是使用 1 美元、2 美元、5 美元和 10 美元纸币换 k 美元的方法数。
11. 对 $(1+x+x^2+x^3+\cdots)^3$ 展开式中 x^4 的系数给出组合解释。使用这个解释求出这个数。
12. a) 什么是关于 $\{a_k\}$ 的生成函数? 这里的 a_k 是 $x_1+x_2+x_3=k$ 的解的个数, 其中 x_1, x_2 和 x_3 是满足 $x_1 \geq 2, 0 \leq x_2 \leq 3, 2 \leq x_3 \leq 5$ 的整数。
 b) 使用(a)的答案求 a_6 。
13. 解释怎样使用生成函数找到用 3 分、4 分和 20 分的邮票在信封上贴满 r 分邮费的方式数。
 a) 假设不考虑贴邮票的次序。
 b) 假设邮票贴成一行并且考虑贴的次序。
 c) 当不考虑贴邮票的次序时, 使用(a)的答案确定用 3 分、4 分和 20 分的邮票在信封上贴满 46 分邮费的方式数。(建议使用计算机代数程序。)
 d) 当考虑贴邮票的次序时, 使用(b)的答案确定用 3 分、4 分和 20 分的邮票在信封上贴满一行 46 分邮费的方式数。(建议使用计算机代数程序。)
14. 使用生成函数(如果需要的话, 也使用计算机代数程序)求出换 1 美元的方式数。
 a) 用 10 美分和 25 美分。 b) 用 5 美分、10 美分和 25 美分。
 c) 用 1 美分、10 美分和 25 美分。 d) 用 1 美分、5 美分、10 美分和 25 美分。
15. 使用生成函数求出换 100 美元的方式数。
 a) 用 10 美元、20 美元和 50 美元纸币。
 b) 用 5 美元、10 美元、20 美元和 50 美元纸币。
 c) 用 5 美元、10 美元、20 美元和 50 美元纸币, 并且每种纸币至少使用 1 张。
 d) 用 5 美元、10 美元和 20 美元纸币, 并且每种纸币至少使用 1 张但不超过 4 张。
16. 如果 $G(x)$ 是关于序列 $\{a_k\}$ 的生成函数, 那么关于下述每个序列的生成函数是什么?
 a) $0, 0, 0, a_3, a_4, a_5, \dots$ b) $a_0, 0, a_1, 0, a_2, 0, \dots$
 c) $0, 0, 0, 0, a_0, a_1, a_2, \dots$ d) $a_0, 2a_1, 4a_2, 8a_3, 16a_4, \dots$
 e) $a_0, a_1/2, a_2/3, a_3/4, \dots$ [提示: 这里需要微积分。]
 f) $a_0, a_0+a_1, a_0+a_1+a_2, a_0+a_1+a_2+a_3, \dots$
17. 使用生成函数求解递推关系 $a_k = 3a_{k-1} + 2$, 初始条件 $a_0 = 1$ 。
18. 使用生成函数求解递推关系 $a_k = 5a_{k-1} - 6a_{k-2}$, 初始条件 $a_0 = 6$ 和 $a_1 = 30$ 。
19. 使用生成函数求解递推关系 $a_k = 4a_{k-1} - 4a_{k-2} + k^2$, 初始条件 $a_0 = 2$ 和 $a_1 = 5$ 。
20. 使用生成函数找出关于斐波那契数的显式公式。
- *21. (需要微积分) 设 $\{C_n\}$ 是卡特朗数的序列, 即具有初值 $C_0 = C_1 = 1$ 的递推关系 $C_n = \sum_{k=0}^{n-1} C_k C_{n-1-k}$ 的解(见 4.1 节例 8)。
 证明: 如果 $G(x)$ 是关于卡特朗数的序列的生成函数, 那么 $xG(x)^2 - G(x) + 1 = 0$ 。(使用初始条件) 断定 $G(x) = (1 - \sqrt{1-4x})/(2x)$ 。
22. 使用生成函数证明范德蒙恒等式: $C(m+n, r) = \sum_{k=0}^r C(m, r-k)C(n, k)$, 其中 m, n 和 r 是非负整数, 且 r 不超过 m 或 n 。[提示: 看 $(1+x)^{m+n} = (1+x)^m(1+x)^n$ 两边的 x^r 的系数。]
 关于序列 $\{a_n\}$ 的指数生成函数是级数

$$\sum_{n=0}^{\infty} \frac{a_n}{n!} x^n$$

例如, 关于序列 $1, 1, 1, \dots$ 的指数生成函数是 $\sum_{n=0}^{\infty} x^n/n! = e^x$ 。(你将发现这个级数在下面的练习中很有

用。) 注意, e^x 是关于序列 $1, 1, 1/2!, 1/3!, 1/4!, \dots$ 的普通生成函数。

23. 求一个关于序列 $\{a_n\}$ 的指数生成函数的封闭形式, 其中

$$a) a_n = 2$$

$$b) a_n = (-1)^n$$

$$c) a_n = 3^n$$

$$d) a_n = n+1$$

$$e) a_n = 1/(n+1)$$

24. 求以下述函数为指数生成函数的序列。

$$a) f(x) = e^{-x}$$

$$b) f(x) = 3x^{2x}$$

$$c) f(x) = e^{3x} - 3e^{2x}$$

$$d) f(x) = (1-x) + e^{-2x}$$

$$e) f(x) = e^{-2x} - (1/(1-x))$$

$$f) f(x) = e^{-3x} - (1+x) + (1/(1-2x))$$

$$g) f(x) = e^{x^2}$$

25. 一个编码系统用八进制(基为 8)数字串对信息编码。一个编码字是有效的, 当且仅当它包含偶数个 7。

a) 求一个关于 n 位长有效编码字个数的线性非齐次递推关系。初始条件是什么?

b) 使用 4.2 节的定理 6 解这个递推关系。

c) 用生成函数解这个递推关系。

在研究整数 n 的不同类型的剖分数时生成函数是很有用的。一个正整数的剖分是把这个整数写成正整数之和, 和中的整数允许重复并且不考虑次序。例如, 5 的剖分(不加限制)是 $1+1+1+1+1$, $1+1+1+2$, $1+1+3$, $1+2+2$, $1+4$, $2+3$ 和 5。练习 26~28 说明了这种应用。

26. 证明: 在 $1/((1-x)(1-x^2)(1-x^3)\dots)$ 的形式幂级数展开式中 x^n 的系数 $p(n)$ 等于 n 的剖分数。

27. 证明: 在 $(1+x)(1+x^2)(1+x^3)\dots$ 的形式幂级数展开式中 x^n 的系数 $p_d(n)$ 等于 n 剖分成不相等的整数(即把 n 写成正整数之和)的方式数, 其中不管这些整数的次序但不允许重复。

28. 证明: 如果 n 是正整数, 那么 n 剖分成不相等的整数的方式数等于 n 剖分成允许重复的奇整数的方式数; 即 $p_o(n) = p_d(n)$ 。[提示: 证明关于 $p_o(n)$ 和 $p_d(n)$ 的生成函数相等。]

29. (需要微积分) 证明如果 G_X 是随机变量 X 的概率生成函数, 使得 $X(s)$ 对于所有的 $s \in S$ 是非负整数, 那么

$$a) G_X(1) = 1$$

$$b) E(X) = G'_X(1)$$

$$c) V(X) = G''_X(1) + G'_X(1) - G'_X(1)^2$$

4.5 容斥

4.5.1 引言

一个离散数学班包含 30 个女生和 50 个二年级学生。在这个班里有多少个女生或二年级学生? 如果没有更多的信息, 这个问题是没法求解的。把女生数和二年级学生数加起来不一定能得出正确的结果, 因为二年级的女生可能被计数了两次。这个事实说明, 在班里的女生或二年级学生数是班里的女生数与二年级学生数之和减去二年级的女生数。在 3.1 节曾经介绍过求解这种计数问题的技术。这里我们将把在那一节引入的思想加以推广, 以求解更为广泛的计数问题。

4.5.2 容斥原理

两个有穷集的并集中存在多少个元素? 在 2.2 节中证明了两个集合 A 和 B 的并集中的元素数是这些集合的元素数之和减去其交集的元素数, 即

$$|A \cup B| = |A| + |B| - |A \cap B|$$

正如我们在 3.1 节证明的, 这个关于两个集合并集中元素数的公式在计数问题中是很有用的。例 1~例 3 进一步说明了这个公式的用处。

例 1 一个离散数学班包含 25 个计算机科学专业的学生、13 个数学专业的学生和 8 个同时主修数学和计算机科学两个专业的学生。如果每个学生主修数学专业、计算机科学专业或者同时主修这两个专业, 那么班里有多少个学生?

解 设 A 是这个班里计算机科学专业的学生的集合, B 是这个班里数学专业的学生的集合, 那么 $A \cap B$ 是班里主修数学和计算机科学两个专业的学生的集合。因为这个班的每个学生或者主修计算机科学, 或者主修数学(或者同时主修两个专业), 从而得到这个班里的学生数是 $|A \cup B|$ 。于是

$$\begin{aligned} |A \cup B| &= |A| + |B| - |A \cap B| \\ &= 25 + 13 - 8 \\ &= 30 \end{aligned}$$

因此, 这个班有 30 个学生。这个计算在图 4-7 中说明。

例 2 有多少个不超过 1000 的正整数可以被 7 或 11 整除?

解 设 A 是不超过 1000 且可被 7 整除的正整数的集合, B 是不超过 1000 且可被 11 整除的正整数的集合, 那么 $A \cup B$ 是不超过 1000 且可被 7 或 11 整除的正整数的集合, $A \cap B$ 是不超过 1000 且可被 7 和 11 同时整除的正整数的集合。我们知道在不超过 1000 的正整数中有 $\lfloor 1000/7 \rfloor$ 个整数可被 7 整除, 并且有 $\lfloor 1000/11 \rfloor$ 个整数可被 11 整除。由于 7 和 11 是互素的, 被 7 和 11 同时整除的整数就是被 $7 \cdot 11$ 整除的整数。因此, 有 $\lfloor 1000/(11 \cdot 7) \rfloor$ 个不超过 1000 的正整数可被 7 和 11 同时整除。于是存在

$$\begin{aligned} |A \cup B| &= |A| + |B| - |A \cap B| \\ &= \left\lfloor \frac{1000}{7} \right\rfloor + \left\lfloor \frac{1000}{11} \right\rfloor - \left\lfloor \frac{1000}{7 \cdot 11} \right\rfloor \\ &= 142 + 90 - 12 \\ &= 220 \end{aligned}$$

个正整数不超过 1000 且可被 7 或 11 整除。这个计算在图 4-8 中说明。

$$|A \cup B| = |A| + |B| - |A \cap B| = 25 + 13 - 8 = 30$$

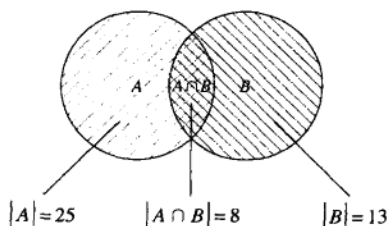


图 4-7 离散数学班的学生的集合

$$|A \cup B| = |A| + |B| - |A \cap B| = 142 + 90 - 12 = 220$$

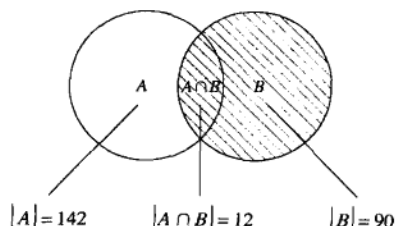


图 4-8 不超过 1000 的可被 7 或 11 整除的正整数的集合

例 3 说明怎样求有穷全集中两个集合的并集之外的元素数。

例 3 假设你们学校有 1807 个新生。其中 453 人选了一门计算机科学课, 567 人选了一门数学课, 299 人同时选了计算机科学课和数学课。有多少学生既没有选计算机科学课也没有选数学课?

解 为找出既没有选数学课也没有选计算机科学课的新生数, 就要从新生总数中减去选了其中一门课的学生数。设 A 是选了一门计算机课的所有新生的集合, 且 B 是选了一门数学课的所有新生的集合。于是 $|A| = 453$, $|B| = 567$, 且 $|A \cap B| = 299$ 。选了一门计算机科学或数学课的学生数是

$$\begin{aligned} |A \cup B| &= |A| + |B| - |A \cap B| \\ &= 453 + 567 - 299 \\ &= 721 \end{aligned}$$

因此, 存在 $1807 - 721 = 1086$ 个新生既没有选计算机科学课也没有选数学课。

在本节的后面将说明怎样求有限个集合的并集中的元素数。这个结果叫做容斥原理。设 n 是任意正整数, 在考虑 n 个集合的并集之前, 先推导与 3 个集合 A , B 和 C 的并集中的元素数有关的公式。为推导这个公式, 首先注意到以下事实: $|A| + |B| + |C|$ 对 3 个集合中那些恰好在其中 1 个集合的元素只计数了 1 次, 恰好在其中 2 个集合的元素计数了 2 次, 恰好在其中 3 个集合的元素计数了 3 次。这个结果在图 4-9a 中说明。

为了去掉在多个集合中元素的重复计数, 减去这 3 个集合中的每 2 个集合的交集的元素数, 得到

$$|A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C|$$

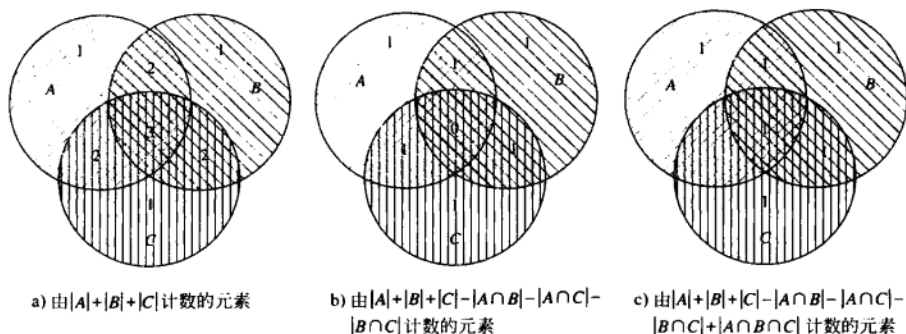


图 4-9 求关于 3 个集合的并集中元素数的公式

这个表达式对恰好出现在其中 1 个集合的元素仍旧计数 1 次。对恰好出现在其中 2 个集合的元素也计数 1 次，因为 2 个集合的交集有 3 个，而这种元素只出现在其中之一。但是，那些出现在 3 个集合的元素将被这个表达式计数 0 次，因为它们将会出现在所有的两两相交的 3 个交集中。这个结果显示在图 4-9b 中。

为了纠正这个漏计，还要加上 3 个集合交集的元素数。这个最后的表达式对每个元素计数了 1 次，不管它是在 1 个、2 个还是在 3 个集合里。于是

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

这个公式显示在图 4-9c 中。

例 4 说明了怎样使用这个公式。

例 4 1232 个学生选了西班牙语课，879 个学生选了法语课，114 个学生选了俄语课。103 个学生选了西班牙语和法语课，23 个学生选了西班牙语和俄语课，14 个学生选了法语和俄语课。如果 2092 个学生至少在西班牙语、法语和俄语课中选 1 门，有多少个学生选了所有这 3 门语言课？

解 设 S 是选西班牙语课的学生集合， F 是选法语课的学生集合， R 是选俄语课的学生集合。那么

$$\begin{aligned} |S| &= 1232, & |F| &= 879, & |R| &= 114 \\ |S \cap F| &= 103, & |S \cap R| &= 23, & |F \cap R| &= 14 \end{aligned}$$

且

$$|S \cup F \cup R| = 2092$$

把这些等式代入下面的等式

$$|S \cup F \cup R| = |S| + |F| + |R| - |S \cap F| - |S \cap R| - |F \cap R| + |S \cap F \cap R|$$

得

$$2092 = 1232 + 879 + 114 - 103 - 23 - 14 + |S \cap F \cap R|$$

求解上式得出 $|S \cap F \cap R| = 7$ 。因此有 7 个学生同时选了西班牙语、法语和俄语课。这个结果在图 4-10 中说明。

我们将叙述和证明容斥原理，它将告诉我们在有限个有穷集的并集中有多少个元素。

定理 1 容斥原理 设 $A_1, A_2, \dots, A_n$ 是有穷集。那么

$$\begin{aligned} |A_1 \cup A_2 \cup \dots \cup A_n| &= \sum_{1 \leq i \leq n} |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| \\ &+ \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| - \dots + (-1)^{n+1} |A_1 \cap A_2 \cap \dots \cap A_n| \end{aligned}$$

证 我们将通过证明并集中的每个元素在等式右边恰好被计数 1 次来证明这个公式。假设 a 恰好是 $A_1, A_2, \dots, A_n$ 中 r 个集合的成员，其中 $1 \leq r \leq n$ 。这个元素被 $\sum |A_i|$ 计数了 $C(r, 1)$

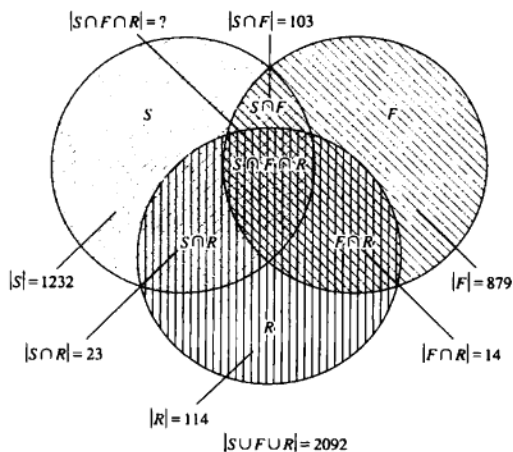


图 4-10 选了西班牙语、法语和俄语课程的学生集合

次，被 $\sum |A_i \cap A_j|$ 计数了 $C(r, 2)$ 次。一般说来，它被涉及 m 个 A_i 集合的求和计数了 $C(r, m)$ 次。于是，这个元素恰好被等式右边的表达式计数了

$$C(r, 1) - C(r, 2) + C(r, 3) - \cdots + (-1)^{r+1} C(r, r)$$

次。我们的目标是求出这个值。由 3.4 节的推论 2，我们有

$$C(r, 0) - C(r, 1) + C(r, 2) - \cdots + (-1)^r C(r, r) = 0$$

于是

$$1 = C(r, 0) = C(r, 1) - C(r, 2) + \cdots + (-1)^{r+1} C(r, r)$$

因此，并集中的每个元素在等式右边的表达式中恰好被计数 1 次。这就证明了容斥原理。□

对于每个正整数 n ，容斥原理对于 n 个集合并集的元素数给出了一个公式。对于 n 个集合的集合族的每一个非空子集的交，在这个公式中都存在一项计数了它的元素。因此在这个公式中有 $2^n - 1$ 项。

例 5 对于 4 个集合的并集中的元素数给出一个公式。

解 容斥原理显示

$$\begin{aligned} |A_1 \cup A_2 \cup A_3 \cup A_4| &= |A_1| + |A_2| + |A_3| + |A_4| \\ &\quad - |A_1 \cap A_2| - |A_1 \cap A_3| - |A_1 \cap A_4| - |A_2 \cap A_3| - |A_2 \cap A_4| - |A_3 \cap A_4| \\ &\quad + |A_1 \cap A_2 \cap A_3| + |A_1 \cap A_2 \cap A_4| + |A_1 \cap A_3 \cap A_4| + |A_2 \cap A_3 \cap A_4| \\ &\quad - |A_1 \cap A_2 \cap A_3 \cap A_4| \end{aligned}$$

注意，这个公式包含 15 个不同的项，对于 $\{A_1, A_2, A_3, A_4\}$ 的每个非空子集有一项。■

练习

- 在 $A_1 \cup A_2$ 中存在多少个元素？如果在 A_1 中存在 12 个元素，在 A_2 中存在 18 个元素，并且
 - $A_1 \cap A_2 = \emptyset$
 - $|A_1 \cap A_2| = 1$
 - $|A_1 \cap A_2| = 6$
 - $A_1 \subseteq A_2$
- 一个调查显示在美国 96% 的家庭至少有 1 台电视机，98% 的家庭有电话，95% 的家庭有电话且至少有 1 台电视机。在美国有百分之几的家庭既没有电话也没有电视机？
- 求 $A_1 \cup A_2 \cup A_3$ 中的元素数，如果每个集合有 100 个元素并且
 - 这些集合是两两不交的。
 - 每对集合中存在 50 个公共元素并且没有元素在所有这 3 个集合里。
 - 每对集合中存在 50 个公共元素并且有 25 个元素在所有这 3 个集合里。
 - 这些集合是相等的。

4. 一个学校有 2504 个计算机科学专业的学生, 其中 1876 人选修了 Pascal, 999 人选修了 Fortran, 345 人选修了 C, 876 人选修了 Pascal 和 Fortran, 231 人选修了 Fortran 和 C, 290 人选修了 Pascal 和 C。如果 189 个学生选了 Fortran, Pascal 和 C, 那么 2504 个学生中有多少学生没选这 3 门程序设计语言课的任何一门?
5. 一个学校有 507, 292, 312 和 344 个学生分别选了微积分、离散数学、数据结构或程序设计语言课, 且有 14 人选了微积分和数据结构课, 213 人选了微积分和程序设计语言课, 211 人选了离散数学和数据结构课, 43 人选了离散数学和程序设计语言课, 没有学生同时选微积分和离散数学课, 也没有学生同时选数据结构和程序设计语言课。问有多少学生在微积分、离散数学、数据结构或程序设计语言中选了课?
6. 求不超过 100 且是奇数或平方数的正整数个数。
7. 有多少 8 位二进制串不包含 6 个连续的 0?
8. 在 10 个十进制数字的排列中有多少个以 3 个数字 987 开始, 在第 5 和第 6 位包含数字 45, 且最后 3 位是 123?
9. 有 4 个集合, 如果这些集合分别有 50, 60, 70 和 80 个元素, 每一对集合有 5 个公共元素, 每 3 个集合有 1 个公共元素, 并且没有元素在所有的 4 个集合里。问在这 4 个集合的并集中有多少个元素?
10. 根据容斥原理写出关于 5 个集合并集元素数的显式公式。
11. 有 6 个集合, 如果知道其中任 3 个集合都是不相交的, 根据容斥原理写出关于这 6 个集合并集元素数的显式公式。
12. 设 E_1, E_2 和 E_3 是样本空间 S 的 3 个事件。求一个关于 $E_1 \cup E_2 \cup E_3$ 的概率的公式。
13. 从 1 到 100(含 1 和 100 在内)不允许重复地随机取 4 个数, 求所有的都是奇数、所有的都被 3 整除或所有的都被 5 整除的概率。
14. 一个样本空间有 5 个事件, 如果其中没有 4 个事件同时出现, 求关于这 5 个事件的并的概率公式。
15. 求一个样本空间中 n 个事件的并的概率公式。

4.6 容斥原理的应用

4.6.1 引言

可以使用容斥原理求解许多计数问题。例如, 我们可以使用这个原理找出小于某个正整数的素数个数。通过计数从一个有穷集到另一个有穷集的映上函数的个数, 能够求解许多问题, 而容斥原理就可以用来求出这种函数的个数。也可以使用容斥原理求解著名的帽子认领问题。帽子认领问题是: 一个招待随机地将帽子发还存放帽子的人, 求没有人取回自己的帽子的概率。

4.6.2 容斥原理的另一种形式

容斥原理有另一种表述形式, 它在计数问题中是很有用的。特别地, 这种形式可以用于求解在一个集合中的元素数, 使得这些元素不具有 n 个性质 $P_1, P_2, \dots, P_n$ 中的任何一条性质。

设 A_i 是具有性质 P_i 的元素的子集。具有所有这些性质 $P_{i_1}, P_{i_2}, \dots, P_{i_k}$ 的元素数将记作 $N(P_{i_1} P_{i_2} \dots P_{i_k})$ 。用集合的术语写这些等式, 有

$$|A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_k}| = N(P_{i_1} P_{i_2} \dots P_{i_k})$$

如果不具有 n 个性质 $P_1, P_2, \dots, P_n$ 中的任何一条的元素数记作 $N(P'_1 P'_2 \dots P'_n)$ 并且集合中的元素数记作 N , 那么有

$$N(P'_1 P'_2 \dots P'_n) = N - |A_1 \cup A_2 \cup \dots \cup A_n|$$

由容斥原理, 有

$$N(P'_1 P'_2 \dots P'_n) = N - \sum_{1 \leq i \leq n} N(P_i) + \sum_{1 \leq i < j \leq n} N(P_i P_j) - \sum_{1 \leq i < j < k \leq n} N(P_i P_j P_k) + \dots + (-1)^n N(P_1 P_2 \dots P_n)$$

例 1 说明怎样使用容斥原理确定具有约束条件的方程的整数解的个数。

例 1 $x_1 + x_2 + x_3 = 11$ 有多少个整数解? 其中 x_1, x_2 和 x_3 是非负整数, 且 $x_1 \leq 3, x_2 \leq 4, x_3 \leq 6$ 。

解 为使用容斥原理, 令解的性质 P_1 为 $x_1 > 3$, 性质 P_2 为 $x_2 > 4$, 性质 P_3 为 $x_3 > 6$ 。满足不

等式 $x_1 \leq 3$, $x_2 \leq 4$ 以及 $x_3 \leq 6$ 的解的个数是

$$N(P_1'P_2'P_3') = N - N(P_1) - N(P_2) - N(P_3) + N(P_1P_2) \\ + N(P_1P_3) + N(P_2P_3) - N(P_1P_2P_3)$$

使用与 3.5 节的例 5 相同的技术, 得

$$N = \text{解的总数} = C(3+11-1, 11) = 78$$

$$N(P_1) = (\text{具有 } x_1 \geq 4 \text{ 的解数}) = C(3+7-1, 7) = C(9, 7) = 36$$

$$N(P_2) = (\text{具有 } x_2 \geq 5 \text{ 的解数}) = C(3+6-1, 6) = C(8, 6) = 28$$

$$N(P_3) = (\text{具有 } x_3 \geq 7 \text{ 的解数}) = C(3+4-1, 4) = C(6, 4) = 15$$

$$N(P_1P_2) = (\text{具有 } x_1 \geq 4 \text{ 且 } x_2 \geq 5 \text{ 的解数}) = C(3+2-1, 2) = C(4, 2) = 6$$

$$N(P_1P_3) = (\text{具有 } x_1 \geq 4 \text{ 且 } x_3 \geq 7 \text{ 的解数}) = C(3+0-1, 0) = 1$$

$$N(P_2P_3) = (\text{具有 } x_2 \geq 5 \text{ 且 } x_3 \geq 7 \text{ 的解数}) = 0$$

$$N(P_1P_2P_3) = (\text{具有 } x_1 \geq 4, x_2 \geq 5 \text{ 且 } x_3 \geq 7 \text{ 的解数}) = 0$$

把这些等式代入关于 $N(P_1'P_2'P_3')$ 的公式, 证明满足 $x_1 \leq 3$, $x_2 \leq 4$ 以及 $x_3 \leq 6$ 的解的个数等于

$$N(P_1'P_2'P_3') = 78 - 36 - 28 - 15 + 6 + 1 + 0 - 0 = 6$$

4.6.3 埃拉托色尼筛

可以用容斥原理找出不超过一个给定正整数的素数个数。一个合数可以被一个不超过它的平方根的素数整除。因此, 为找出不超过 100 的素数个数, 首先注意到不超过 100 的合数一定有一个不超过 10 的素因子。由于小于 10 的素数只有 2, 3, 5 和 7, 因此不超过 100 的素数就是这 4 个素数以及那些大于 1 和不超过 100 且不被 2, 3, 5 或 7 整除的正整数。为应用容斥原理, 令 P_1 是一个整数被 2 整除的性质, P_2 是一个整数被 3 整除的性质, P_3 是一个整数被 5 整除的性质, P_4 是一个整数被 7 整除的性质。于是, 不超过 100 的素数个数是

$$4 + N(P_1'P_2'P_3'P_4')$$

由于存在 99 个比 1 大且不超过 100 的正整数, 容斥原理显示

$$N(P_1'P_2'P_3'P_4') = 99 - N(P_1) - N(P_2) - N(P_3) - N(P_4) \\ + N(P_1P_2) + N(P_1P_3) + N(P_1P_4) + N(P_2P_3) + N(P_2P_4) + N(P_3P_4) \\ - N(P_1P_2P_3) - N(P_1P_2P_4) - N(P_1P_3P_4) - N(P_2P_3P_4) \\ + N(P_1P_2P_3P_4)$$

不超过 100 (且大于 1) 并被 $\{2, 3, 5, 7\}$ 的子集中的所有素数整除的正整数个数是 $\lfloor 100/N \rfloor$, 其中 N 是这个子集中的素数之积。(这是由于任意两个素数都没有公因子。)因此

$$N(P_1'P_2'P_3'P_4') = 99 - \left\lfloor \frac{100}{2} \right\rfloor - \left\lfloor \frac{100}{3} \right\rfloor - \left\lfloor \frac{100}{5} \right\rfloor - \left\lfloor \frac{100}{7} \right\rfloor + \left\lfloor \frac{100}{2 \cdot 3} \right\rfloor + \left\lfloor \frac{100}{2 \cdot 5} \right\rfloor + \left\lfloor \frac{100}{2 \cdot 7} \right\rfloor \\ + \left\lfloor \frac{100}{3 \cdot 5} \right\rfloor + \left\lfloor \frac{100}{3 \cdot 7} \right\rfloor + \left\lfloor \frac{100}{5 \cdot 7} \right\rfloor - \left\lfloor \frac{100}{2 \cdot 3 \cdot 5} \right\rfloor - \left\lfloor \frac{100}{2 \cdot 3 \cdot 7} \right\rfloor - \left\lfloor \frac{100}{2 \cdot 5 \cdot 7} \right\rfloor \\ - \left\lfloor \frac{100}{3 \cdot 5 \cdot 7} \right\rfloor + \left\lfloor \frac{100}{2 \cdot 3 \cdot 5 \cdot 7} \right\rfloor \\ = 99 - 50 - 33 - 20 - 14 + 16 + 10 + 7 + 6 + 4 + 2 - 3 - 2 - 1 - 0 + 0 \\ = 21$$

于是存在 $4+21=25$ 个不超过 100 的素数。

可以用埃拉托色尼筛求不超过一个给定正整数的所有的素数。例如, 可用下面的过程找不超过 100 的所有素数。首先, 保留 2 而将其余那些被 2 整除的整数删除。因为 3 是保留下来的第一个大于 2 的整数, 除 3 之外, 删除其余那些被 3 整除的整数。因为 5 是在 3 后面下一个留下来的整数, 除 5 之外删除其余那些被 5 整除的整数。下一个留下的整数是 7, 因此留下 7, 删除其余那些被 7 整除的整数。由于所有不超过 100 的合数被 2, 3, 5 或 7 整除, 那么所有留下来的大于 1 的数是素数。在表 4-2 中, 4 个子表显示了每一步删除的整数, 在第一个表中得到下划线的是除 2 之外其余被 2 整除的整数, 在第二个表中得到下划线的是除 3 之外其余被 3 整除的整数。

在第三个表中得到下划线的是除5之外其余被5整除的整数,在第四个表中得到下划线的是除7之外其余被7整除的整数。没划线的整数是不超过100的素数。

表 4-2 埃拉托色尼筛

除 2 之外其余被 2 整除的整数得到下划线											除 3 之外其余被 3 整除的整数得到下划线										
1	2	3	<u>4</u>	5	<u>6</u>	7	<u>8</u>	9	<u>10</u>		1	2	3	<u>4</u>	5	<u>6</u>	7	<u>8</u>	9	<u>10</u>	
11	<u>12</u>	13	<u>14</u>	15	<u>16</u>	17	<u>18</u>	19	<u>20</u>		11	<u>12</u>	13	<u>14</u>	<u>15</u>	<u>16</u>	17	<u>18</u>	19	<u>20</u>	
21	<u>22</u>	23	<u>24</u>	25	<u>26</u>	27	<u>28</u>	29	<u>30</u>		21	<u>22</u>	23	<u>24</u>	25	<u>26</u>	<u>27</u>	<u>28</u>	29	<u>30</u>	
31	<u>32</u>	33	<u>34</u>	35	<u>36</u>	37	<u>38</u>	39	<u>40</u>		31	<u>32</u>	<u>33</u>	<u>34</u>	35	<u>36</u>	37	<u>38</u>	39	<u>40</u>	
41	<u>42</u>	43	<u>44</u>	45	<u>46</u>	47	<u>48</u>	49	<u>50</u>		41	<u>42</u>	43	<u>44</u>	<u>45</u>	<u>46</u>	47	<u>48</u>	49	<u>50</u>	
51	<u>52</u>	53	<u>54</u>	55	<u>56</u>	57	<u>58</u>	59	<u>60</u>		51	<u>52</u>	53	<u>54</u>	55	<u>56</u>	<u>57</u>	<u>58</u>	59	<u>60</u>	
61	<u>62</u>	63	<u>64</u>	65	<u>66</u>	67	<u>68</u>	69	<u>70</u>		61	<u>62</u>	63	<u>64</u>	65	<u>66</u>	67	<u>68</u>	69	<u>70</u>	
71	<u>72</u>	73	<u>74</u>	75	<u>76</u>	77	<u>78</u>	79	<u>80</u>		71	<u>72</u>	73	<u>74</u>	<u>75</u>	<u>76</u>	77	<u>78</u>	79	<u>80</u>	
81	<u>82</u>	83	<u>84</u>	85	<u>86</u>	87	<u>88</u>	89	<u>90</u>		81	<u>82</u>	83	<u>84</u>	85	<u>86</u>	<u>87</u>	<u>88</u>	89	<u>90</u>	
91	<u>92</u>	93	<u>94</u>	95	<u>96</u>	97	<u>98</u>	99	<u>100</u>		91	<u>92</u>	<u>93</u>	94	95	<u>96</u>	97	<u>98</u>	99	<u>100</u>	

除 5 之外其余被 5 整除的整数得到下划线											除 7 之外其余被 7 整除的整数得到下划线										
1	2	3	<u>4</u>	5	<u>6</u>	7	<u>8</u>	9	<u>10</u>		1	2	3	<u>4</u>	5	<u>6</u>	7	<u>8</u>	9	<u>10</u>	
11	<u>12</u>	13	<u>14</u>	<u>15</u>	<u>16</u>	17	<u>18</u>	19	<u>20</u>		11	<u>12</u>	13	<u>14</u>	<u>15</u>	<u>16</u>	17	<u>18</u>	19	<u>20</u>	
<u>21</u>	<u>22</u>	23	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	29	<u>30</u>		21	<u>22</u>	23	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	29	<u>30</u>	
31	<u>32</u>	<u>33</u>	<u>34</u>	35	<u>36</u>	37	<u>38</u>	39	<u>40</u>		31	<u>32</u>	<u>33</u>	<u>34</u>	<u>35</u>	<u>36</u>	37	<u>38</u>	39	<u>40</u>	
41	<u>42</u>	43	<u>44</u>	<u>45</u>	<u>46</u>	47	<u>48</u>	49	<u>50</u>		41	<u>42</u>	43	<u>44</u>	<u>45</u>	<u>46</u>	47	<u>48</u>	49	<u>50</u>	
<u>51</u>	<u>52</u>	53	<u>54</u>	<u>55</u>	<u>56</u>	<u>57</u>	<u>58</u>	59	<u>60</u>		51	<u>52</u>	53	<u>54</u>	55	<u>56</u>	<u>57</u>	<u>58</u>	59	<u>60</u>	
61	<u>62</u>	63	<u>64</u>	65	<u>66</u>	67	<u>68</u>	69	<u>70</u>		61	<u>62</u>	<u>63</u>	<u>64</u>	<u>65</u>	<u>66</u>	67	<u>68</u>	69	<u>70</u>	
71	<u>72</u>	73	<u>74</u>	<u>75</u>	<u>76</u>	77	<u>78</u>	79	<u>80</u>		71	<u>72</u>	73	<u>74</u>	<u>75</u>	<u>76</u>	77	<u>78</u>	79	<u>80</u>	
<u>81</u>	<u>82</u>	83	<u>84</u>	<u>85</u>	<u>86</u>	87	<u>88</u>	89	<u>90</u>		81	<u>82</u>	83	<u>84</u>	85	<u>86</u>	<u>87</u>	<u>88</u>	89	<u>90</u>	
91	<u>92</u>	<u>93</u>	<u>94</u>	95	<u>96</u>	97	<u>98</u>	99	<u>100</u>		91	<u>92</u>	<u>93</u>	<u>94</u>	95	<u>96</u>	97	<u>98</u>	99	<u>100</u>	

4.6.4 映上函数的个数

也可以用容斥原理确定从 m 元素集到 n 元素集的映上函数的个数。首先考虑例2。

例2 从6元素集到3元素集有多少个映上函数?

解 假定在陪域中的元素是 b_1, b_2, b_3 。设 P_1, P_2, P_3 分别是 b_1, b_2, b_3 不在函数值域中的性质。注意到一个函数是映上的当且仅当没有性质 P_1, P_2 和 P_3 。根据容斥原理得到6元素集到3元素集的映上函数个数是

$$N(P_1'P_2'P_3') = N - [N(P_1) + N(P_2) + N(P_3)] \\ + [N(P_1P_2) + N(P_1P_3) + N(P_2P_3)] - N(P_1P_2P_3)$$

其中 N 是从6元素集到3元素集的函数总数。我们将对等式右边的每一项求值。

由3.1节的例5得出 $N = 3^6$ 。注意到 $N(P_i)$ 是值域中不含 b_i 的函数个数。所以,对于定义域中的每个元素的函数值有2种选择,从而得到 $N(P_i) = 2^6$ 。此外,这种项有 $C(3, 1)$ 个。注意到 $N(P_iP_j)$ 是值域中不含 b_i 和 b_j 的函数个数。所以,对于定义域中的每个元素的函数值只有1种选择。从而得到 $N(P_iP_j) = 1^6 = 1$ 。此外,这种项有 $C(3, 2)$ 个。还有,注意到 $N(P_1P_2P_3) = 0$, 因为这个项是值域中不含 b_1, b_2 和 b_3 的函数个数。很显然,没有这样的函数。于是,从6元素集到3元素集的映上函数个数是

$$3^6 - C(3, 1)2^6 + C(3, 2)1^6 = 729 - 192 + 3 = 540$$

现在叙述从 m 元素集到 n 元素集的映上函数个数的一般性结果。这个结果的证明留给读者作为练习。

定理 1 设 m 和 n 是正整数满足 $m \geq n$ 。那么存在

$$n^m - C(n, 1)(n-1)^m + C(n, 2)(n-2)^m - \cdots + (-1)^{n-1} C(n, n-1) \cdot 1^m$$

个从 m 元素集到 n 元素集的映上函数。

从 m 个元素的集到 n 个元素的集的映上函数是这样一种对应方式：它把定义域中的 m 个元素分配到 n 个不可辨别的盒子中，使得每个盒子都不是空的，然后将陪域中的 n 个元素中的每一个元素都与一个盒子相对应。这意味着从具有 m 个元素的集到具有 n 个元素的集的映上函数的个数，等于把 m 个可辨别的物体分配到 n 个不可辨别的盒子中，使得每个盒子都不空时的方法数乘以具有 n 个元素的集的排列数。因此，从 m 个元素的集到 n 个元素的集的映上函数的个数为 $n! S(m, n)$ ，其中 $S(m, n)$ 是 3.5 节中定义的第二类斯特林数。这意味着我们可以用定理 1 来推导 3.5 节所给的关于 $S(m, n)$ 的公式。（关于第二类斯特林数更详细的信息，可参见 [MiRo91] 中的第 6 章。）

下面给出定理 1 的另一个应用的实例。

例 3 把 5 项工作分给 4 个不同的雇员，如果每个雇员至少分配 1 项工作，问有多少种方式？

解 把工作分配看作从 5 个工作集到 4 个雇员集的函数。每个雇员至少得到 1 项工作的分配对应于从工作集到雇员集的映上函数。因此，由定理 1 存在

$$4^5 - C(4, 1)3^5 + C(4, 2)2^5 - C(4, 3)1^5 = 1024 - 972 + 192 - 4 = 240$$

种方式来分配工作并使得每个雇员至少得到 1 项工作。 ■

4.6.5 错位排列

下面将用容斥原理计数排列 n 个物体并使得没有一个物体在它的初始位置上的方式数。考虑下面的例子。

例 4 帽子认领问题 在一个餐厅里，一个新的雇员寄存 n 个人的帽子时忘记把寄存号放在帽子上。当顾客取回他们的帽子时，这个雇员从剩下的帽子中随机选择发给他们。问没有一个人收到自己的帽子的概率是多少？ ■

注意 答案就是重新排列帽子使得没有帽子在它的初始位置上的方式数除以 n 个帽子的排列数 $n!$ 。在我们找出排列 n 个物体并使得没有一个物体在它的初始位置上的方式数后再考虑这个例子。



一个错位排列是使得没有一个物体在它的初始位置上的排列。为求解例 4 中的问题我们需要确定 n 个物体的错位排列数。

例 5 排列 21453 是 12345 的一个错位排列，因为没有数在它的初始位置上。但是，21543 不是 12345 的错位排列，因为 4 留在它的初始位置上。 ■

设 D_n 表示 n 个物体的错位排列数。例如， $D_3 = 2$ ，因为 123 的错位排列是 231 和 312。我们将使用容斥原理对所有的正整数 n 求 D_n 。

定理 2 n 元素集的错位排列数是

$$D_n = n! \left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!} \right]$$

证 如果排列保持元素 i 不变，就设排列有性质 P_i 。错位排列的个数就是对 $i=1, 2, \dots, n$ ，没有性质 P_i 的排列数，或

$$D_n = N(P'_1 P'_2 \cdots P'_n)$$

使用容斥原理得到

$$D_n = N - \sum_i N(P_i) + \sum_{i < j} N(P_i P_j) - \sum_{i < j < k} N(P_i P_j P_k) + \cdots + (-1)^n N(P_1 P_2 \cdots P_n)$$

其中 N 是 n 个元素的排列数。这个等式说明，所有的元素都发生变化的排列数，等于排列的总数减去至少保持 1 个元素不变的排列数，加上至少保持 2 元素不变的排列数，减去至少保持 3 个元素不变的排列数，等等。现在找出在等式右边出现的所有的量。

首先注意到 $N=n!$, 因为 N 仅仅就是 n 个元素排列的总数。还有, $N(P_i)=(n-1)!$, 这是由乘积法则得到的, 因为 $N(P_i)$ 是保持元素 i 不变的排列数, 因而第 i 个位置是确定的, 但是其余的每个位置可以放任意元素。类似地,

$$N(P_i P_j) = (n-2)!$$

因为这是保持元素 i 和 j 不变的排列数, 但是其余 $(n-2)$ 个元素的位置可以被任意地安排。一般说来有

$$N(P_{i_1} P_{i_2} \cdots P_{i_m}) = (n-m)!$$

因为这是保持元素 $i_1, i_2, \dots, i_m$ 不变的排列数, 但是其他 $(n-m)$ 个元素的位置可以被任意安排。由于存在 $C(n, m)$ 种方式从 n 个元素中选择 m 个, 从而有

$$\sum_{1 \leq i \leq n} N(P_i) = C(n, 1)(n-1)!$$

$$\sum_{1 \leq i < j \leq n} N(P_i P_j) = C(n, 2)(n-2)!$$

一般地, 有

$$\sum_{1 \leq i_1 < i_2 < \cdots < i_m \leq n} N(P_{i_1} P_{i_2} \cdots P_{i_m}) = C(n, m)(n-m)!$$

所以, 把这些等式代入关于 D_n 的公式得

$$D_n = n! - C(n, 1)(n-1)! + C(n, 2)(n-2)! - \cdots + (-1)^n C(n, n)(n-n)!$$

$$= n! - \frac{n!}{1!(n-1)!}(n-1)! + \frac{n!}{2!(n-2)!}(n-2)! - \cdots + (-1)^n \frac{n!}{n!0!}0!$$

化简这个表达式得

$$D_n = n! \left[1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!} \right]$$

□

现在对于给定的正整数 n 求 D_n 就简单了。例如, 使用定理 2 得

$$D_3 = 3! \left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} \right] = 6 \left(1 - 1 + \frac{1}{2} - \frac{1}{6} \right) = 2$$

正如我们前面所看到的。

现在可以给出例 4 中的问题的解。

解 没有一个人收到自己的帽子的概率是 $D_n/n!$ 。由定理 2, 这个概率是

$$\frac{D_n}{n!} = 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!}$$

对于 $2 \leq n \leq 7$, 这个概率的值在表 4-3 中给出。

表 4-3 错位排列的概率

n	2	3	4	5	6	7
$D_n/n!$	0.500 00	0.333 33	0.375 00	0.366 67	0.368 06	0.367 86

使用微积分的方法可以证明

$$e^{-1} = 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!} + \cdots \sim 0.368$$

因为这是一个项趋向于 0 的交错级数, 当 n 无限增长时, 没有一个人取回自己的帽子的概率趋于 $e^{-1} \sim 0.368$ 。事实上, 可以证明这个概率与 e^{-1} 的差在 $1/(n+1)!$ 之内。 ■

练习

- 假设 1 蒲式耳 100 个苹果中 20 个有虫, 15 个有擦伤。只有没虫也没擦伤的苹果才可以卖。如果 10 个擦伤的苹果有虫, 那么 100 个苹果中有多少个可以卖?
- 方程 $x_1 + x_2 + x_3 = 13$ 有多少个解? 其中 x_1, x_2, x_3 是小于 6 的非负整数。
- 使用容斥原理求小于 200 的素数的个数。
- 有多少小于 10 000 的正整数不是一个整数的 2 次或更高次幂?

5. 有多少种方式把 6 个不同的玩具分给 3 个不同的孩子并使得每个孩子至少得到 1 个玩具?
6. 有多少种方式把 7 项不同的工作分给 4 个不同的雇员, 使得每个雇员至少得到 1 项工作, 并且把最困难的工作分给最好的雇员?
7. 一个 7 元素集合有多少个错位排列?
8. 一个把信放入信袋的机器发生了故障并且随机把信放入信袋中。在一组 100 封信中发生下面事件的概率是多少?
 - a) 没有信放对了信袋。
 - b) 恰好 1 封信放对了信袋。
 - c) 恰好 98 封信放对了信袋。
 - d) 恰好 99 封信放对了信袋。
 - e) 所有的信都放对了信袋。
- *9. 有多少种方式安排数字 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 使得没有偶数在它的初始位置上?
10. 对哪些正整数 n , 错位排列数 D_n 是偶数?
- *11. 当 n 的素因子分解式是

$$n = p_1^{a_1} p_2^{a_2} \cdots p_m^{a_m}$$

时, 使用容斥原理推导一个关于 $\phi(n)$ 的公式。

12. 有多少个 $\{1, 2, 3, 4, 5, 6\}$ 的错位排列的前 3 位是取某种次序的整数 1, 2, 3?
13. 证明定理 1。

关键术语和结果

术语

递推关系: 一个公式, 它把序列除了某些初始项以外的项, 表示成这个序列前面的一个或若干个项的函数。

递推关系的初始条件: 满足递推关系的序列在该关系起作用之前的某些项的值。

常系数线性齐次递推关系: 一个递推关系, 除了初始项之外, 它把序列的项表示成前面项的线性组合。

常系数线性齐次递推关系的特征根: 与常系数线性齐次递推关系相关的多项式的根。

常系数线性非齐次递推关系: 一个递推关系, 除了初始项之外, 它把序列的项表示成前面项的线性组合加上一个仅仅依赖于序标的不恒为 0 的函数。

分治算法: 求解问题的一种算法, 求解中递归地把问题划分成固定数目的较小的同种类型的问题。

序列的生成函数: 用序列的第 n 项作为 x^n 的系数的形式幂级数。

埃拉托色尼筛: 找出小于一个给定正整数的素数的过程。

错位排列: 使得没有物体在它的初始位置上的排列。

结果

两个有穷集合并集的元素数公式:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

三个有穷集合并集的元素数公式:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

容斥原理:

$$\begin{aligned} |A_1 \cup A_2 \cup \cdots \cup A_n| &= \sum_{1 \leq i \leq n} |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| \\ &\quad + \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| \\ &\quad - \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n| \end{aligned}$$

从 m 元素集到 n 元素集的映上函数个数:

$$n^m - C(n, 1)(n-1)^m + C(n, 2)(n-2)^m - \cdots + (-1)^{n-1} C(n, n-1) \cdot 1^m$$

n 个物体的错位排列数:

$$D_n = n! \left[1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!} \right]$$

复习题

1. a) 什么是递推关系?
b) 如果在一个获利 9% 的账上储蓄 1 000 000 美元, 求与 n 年后账上钱数有关的递推关系。
2. 解释怎样用斐波那契数求解关于兔子的斐波那契问题。
3. a) 找出与求解汉诺塔难题的所需步数有关的递推关系。
b) 显示怎样使用迭代来求解这个递推关系。
4. a) 解释怎样找一个与不包含两个连续的 1 的 n 位二进制串个数有关的递推关系。
b) 描述另一个计数问题使得它的解满足同一个递推关系。
5. 定义一个 k 阶的线性齐次递推关系。
6. a) 解释怎样求解二阶线性齐次递推关系。
b) 如果 $a_0 = 3$, $a_1 = 15$, 对于 $n \geq 2$, 求解递推关系 $a_n = 13a_{n-1} - 22a_{n-2}$ 。
c) 如果 $a_0 = 3$, $a_1 = 35$, 对于 $n \geq 2$, 求解递推关系 $a_n = 14a_{n-1} - 49a_{n-2}$ 。
7. a) 如果 $f(n)$ 满足分治递推关系 $f(n) = af(n/b) + g(n)$, 这里 b 整除正整数 n , 解释怎样求 $f(b^k)$, 其中 k 是正整数。
b) 如果 $f(n) = 3f(n/4) + 5n/4$ 且 $f(1) = 7$, 求 $f(256)$ 。
8. a) 对于用二分搜索在表中找一个数所用的比较次数推导一个分治的递推关系。
b) 从你在(a)给出的分治递推关系使用 4.3 节中的定理 1, 对于二分搜索所用的比较次数给出一个大 O 估计。
9. a) 给出一个关于 3 个集合并集元素数的公式。
b) 解释为什么这个公式是有效的。
c) 解释怎样使用(a)的公式求不超过 1000 且能被 6, 10 或 15 整除的正整数个数。
d) 解释怎样使用(a)的公式求方程 $x_1 + x_2 + x_3 + x_4 = 22$ 的非负整数解的个数, 其中 $x_1 < 8$, $x_2 < 6$, $x_3 < 5$ 。
10. a) 给出一个关于 4 个集合并集元素数的公式。解释为什么它是有效的。
b) 假设 A_1, A_2, A_3 和 A_4 每个集合含 25 个元素, 其中任意 2 个集合的交含 5 个元素, 任 3 个集合的交含 2 个元素, 所有 4 个集合含 1 个公共元素。问在这 4 个集合的并集中有多少个元素?
11. a) 叙述容斥原理。
b) 概述这个原理的证明。
12. 解释怎样使用容斥原理计数从 m 元素集合到 n 元素集合的映上函数的个数。
13. a) 怎样计数把 m 项工作分给 n 个雇员并使得每个雇员至少得到一项工作的方案数?
b) 把 7 项工作分给 3 个雇员并使得每个雇员至少得到一项工作有多少种方案?
14. 解释怎样使用容斥原理计数不超过正整数 n 的素数个数。
15. a) 定义一个错位排列。
b) 一个寄存帽子的人给 n 个人发还帽子, 使得没有人得到自己帽子的方式的计数为什么和 n 个物体的错位排列数一样?
c) 解释怎样计数 n 个物体的错位排列数。

补充练习

1. 一个 10 人小组开始一系列的通信活动, 每个人把这封信寄给另外 4 个人。每个收到信的人再把这封信寄给另外的 4 个人。
a) 如果没有人收到的信多于 1 封, 求与这个通信活动的第 n 步寄出信数有关的递推关系。
b) 在(a)中递推关系的初始条件是什么?
c) 在这个通信活动的第 n 步寄出了多少封信?
2. 一个核反应堆产生了 18 克放射性同位素。每小时放射性同位素衰变 1%。
a) 对 n 小时后留下的同位素量建立一个递推关系。
b) 对于(a)的递推关系, 初始条件是什么?

c) 求解这个递推关系。

3. 美国政府每小时印1美元纸币超过10 000张, 5美元纸币超过4000张, 10美元纸币超过3000张, 20美元纸币超过2500张, 50美元纸币超过1000张, 100美元纸币与前一小时的张数一样。在初始时刻每种钱币有1000张。

a) 建立一个关于第 n 小时总钱数的递推关系。

b) (a)中递推关系的初始条件是什么?

c) 求解这个第 n 小时总钱数的递推关系。

d) 建立一个关于前 n 小时总钱数的递推关系。

e) 求解这个关于前 n 小时总钱数的递推关系。

4. 每个前一小时已经存在的细菌在每小时都分裂出2个新的细菌, 并且所有的细菌只有2小时的生命。假设这群细菌开始时有100个新细菌。

a) 建立关于 n 小时后存在细菌数目的递推关系。

b) 这个递推关系的解是什么?

c) 什么时候这群细菌的个数将超过100万个?

5. 使用两个不同的信号在通信信道发送信息。传送一个信号需要2微秒, 传送另一个信号要3微秒。一个信息的每个信号后紧跟着下一个信号。

a) 求与在 n 微秒中可以发送的不同信号数有关的递推关系。

b) 对于(a)的递推关系, 初始条件是什么?

c) 在12微秒内可以发送多少个不同的信息?

6. 一个小邮局只有4分、6分和10分邮票。如果考虑邮票使用的次序, 求与这些邮票构成 n 分邮资的方式数有关的递推关系。这个递推关系的初始条件是什么?

7. 使用在练习6描述的规则, 有多少种方式构成下述邮资?

a) 12分

b) 14分

c) 18分

d) 22分

8. 求具有 $a_0=1$ 和 $b_0=2$ 的联立方程组

$$a_n = a_{n-1} + b_{n-1}$$

$$b_n = a_{n-1} - b_{n-1}$$

9. 如果 $a_0=1$ 且 $a_1=2$, 求解递推关系 $a_n = a_{n-1}^2 / a_{n-2}$ 。[提示: 两边取对数得到关于序列 $\log a_n$ 的递推关系, $n=0, 1, 2, \dots$]

- *10. 如果 $a_0=2$ 和 $a_1=2$, 求解递推关系 $a_n = a_{n-1}^3 a_{n-2}^2$ 。(见练习9的提示。)

11. 如果 $a_0=2$, $a_1=4$ 和 $a_2=8$, 求解递推关系 $a_n = 3a_{n-1} - 3a_{n-2} + a_{n-3} + 1$ 。

12. 如果 $a_0=2$, $a_1=2$ 和 $a_2=4$, 求解递推关系 $a_n = 3a_{n-1} - 3a_{n-2} + a_{n-3}$ 。

- *13. 假设在4.1节的例4中, 一对兔子在繁殖2次以后就离开这个岛。求与第 n 个月中的岛上兔子对数有关的递推关系。

14. 当 n 可被5整除时求解递推关系 $f(n) = 3f(n/5) + 2n^k$, 其中 $n=5^k$, k 是正整数, $f(1)=1$ 。

15. 如果 f 是增函数, 估计练习14中 f 的大小。

16. 找出与下述算法所使用比较次数有关的递推关系: 通过把 n 个数的序列递归地划分成两个子序列找出最大和第二大的元素, 在每一步划分时要求这两个子序列项数相等或一个子序列比另一个子序列多一项。当子序列达到2项时停止。

17. 估计练习16描述的算法所使用的比较次数。

设 $\{a_n\}$ 是实数序列。这个序列的前向差分定义如下: 第一个前向差分是 $\Delta a_n = a_{n+1} - a_n$; 第 $k+1$ 个前向差分 $\Delta^{k+1} a_n$ 是通过 $\Delta^{k+1} a_n = \Delta^k a_{n+1} - \Delta^k a_n$ 由 $\Delta^k a_n$ 得到的。

18. 求 $\Delta^k a_n$, 其中

a) $a_n = 3$

b) $a_n = 4n + 7$

c) $a_n = n^2 + n + 1$

19. 设 $a_n = 3n^3 + n + 2$, 求 $\Delta^k a_n$, 其中 k 等于

a) 2

b) 3

c) 4

- *20. 假设 $a_n = P(n)$, 其中 P 是 d 次多项式。证明: 对所有的非负整数 n , $\Delta^{d+1} a_n = 0$ 。

21. 令 $\{a_n\}$ 和 $\{b_n\}$ 是实数序列。证明

$$\Delta(a_n b_n) = a_{n+1}(\Delta b_n) + b_n(\Delta a_n)$$

22. 证明: 如果 $F(x)$ 和 $G(x)$ 分别是序列 $\{a_k\}$ 和 $\{b_k\}$ 的生成函数, 且 c 和 d 是实数, 那么 $(cF + dG)(x)$ 是 $\{ca_k + db_k\}$ 的生成函数。

23. (需要微积分)这个练习说明了怎样使用生成函数求解递推关系 $(n+1)a_{n+1} = a_n + (1/n!)$, $n \geq 0$, 初始条件 $a_0 = 1$.
- a) 设 $G(x)$ 是关于 $\{a_n\}$ 的生成函数。证明 $G'(x) = G(x) + e^x$ 且 $G(0) = 1$ 。
- b) 由(a)证明 $(e^{-x}G(x))' = 1$, 且断定 $G(x) = xe^x + e^x$ 。
- c) 使用(b)找出关于 a_n 的封闭公式。
24. 假设在离散数学班的第一次考试中 14 个学生得 A, 第二次考试中 18 个得 A。如果 22 个学生在第一或第二次考试得 A, 有多少学生两次考试都得 A?
25. 在蒙默思郡(英国威尔士郡原郡名)323 个农场至少有马、牛或羊其中的 1 种。如果 224 个农场有马, 85 个有牛, 57 个有羊, 18 个农场 3 种家畜全有, 那么有多少个农场恰好有这 3 种家畜中的 2 种?
26. 查询某学院关于学生记录的数据库得到下述数据: 学院有 2175 个学生, 其中 1675 个不是一年级学生, 1074 个学生选了微积分, 444 个学生选了离散数学, 607 个不是一年级学生且选了微积分, 350 个学生选了微积分和离散数学, 201 个不是一年级学生且选了离散数学, 143 个不是一年级学生并且选了微积分和离散数学。所有这些对查询的回答都是正确的吗?
27. 某大学数学学院的学生可以选择下述一个或多个方向作为主修方向: 应用数学(AM)、纯粹数学(PM)、运筹学(OR)及计算机科学(CS)。如果包括同时主修在内, 主修 AM 的有 23 个学生; 主修 PM 的有 17 个学生; 主修 OR 的 44 个; 主修 CS 的 63 个; 主修 AM 与 PM 的 5 个; 主修 AM 与 CS 的 8 个; 主修 AM 与 OR 的 4 个; 主修 PM 与 CS 的 6 个; 主修 PM 与 OR 的 5 个; 主修 OR 与 CS 的 14 个; 主修 PM, OR 与 CS 的 2 个; 主修 AM, OR 与 CS 的 2 个; 主修 PM, AM 与 OR 的 1 个; 主修 PM, AM 与 CS 的 1 个; 还有 1 个主修所有 4 个方向。问这个学院有多少学生?
28. 当使用容斥原理表示 7 个集合的并集中的元素数时, 如果其中没有 6 个或更多的集合含有公共元素, 那么需要多少项?
29. 方程 $x_1 + x_2 + x_3 = 20$, $2 < x_1 < 6$, $6 < x_2 < 10$, $0 < x_3 < 5$ 有多少个正整数解?
30. 有多少小于 1 000 000 的正整数
- a) 能够被 2, 3 或 5 整除? b) 不被 7, 11 或 13 整除? c) 被 3 整除但不被 7 整除?
31. 有多少小于 200 的正整数是
- a) 整数的 2 次或更高次幂? b) 整数的 2 次或更高次幂, 或是素数?
- c) 不被一个大于 1 的整数的平方整除? d) 不被一个大于 1 的整数的立方整除?
- e) 不被 3 个或更多的素数整除?
- *32. 把 6 个不同的工作分给 3 个不同的雇员, 如果最难的工作分给最有经验的雇员并且最容易的工作分给最缺乏经验的雇员, 那么有多少种分法?
33. 由寄存帽子的人随机发还给 n 个人帽子, 那么恰好一个人拿到自己帽子的概率是多少?
34. 有多少个 6 位二进制串不包含 4 个连续的 1?
35. 一个 6 位二进制串包含至少 4 个 1 的概率是多少?

计算机题目

按下述给定的输入和输出写出程序。

- 给定正整数 n , 列出汉诺塔难题从一根柱子到另一根柱子依照游戏规则移动 n 个盘子需要的所有移动。
- 给定正整数 n 和整数 k , $1 \leq k \leq n$, 列出富雷姆-斯图尔特算法(见 4.1 节练习 27 前面的说明)依照游戏规则用 4 根柱子从一根柱子到另一根柱子移动 n 个盘子需要的所有移动。
- 给定正整数 n , 列出不包含连续 2 个 0 的所有的 n 位二进制序列。
- 给定正整数 n , 写出在 $n+1$ 个变量的乘积中加括号的所有方式。
- 给定递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 和正整数 k , 其中 c_1 和 c_2 是实数, 初始条件为 $a_0 = C_0$ 及 $a_1 = C_1$, 使用迭代求 a_k 。
- 给定递推关系 $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ 和初始条件 $a_0 = C_0$ 及 $a_1 = C_1$, 确定唯一的解。
- 给定形如 $f(n) = af(n/b) + c$ 的递推关系, 其中 a 是实数, b 是正整数, c 是实数, k 是正整数, 使用迭代求 $f(b^k)$ 。
- 给定 3 个集合的交集的元素个数, 每两个集合的交集的元素个数, 和每个集合中的元素个数, 求其并集中的元素个数。

9. 给定正整数 n , 求在 n 个集合的并集中元素个数的公式。
10. 给定正整数 m 和 n , 求从 m 元素集到 n 元素集的映上函数个数。
11. 给定正整数 n , 列出集合 $\{1, 2, 3, \dots, n\}$ 的所有错位排列。

计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 求 f_{100} , f_{500} 和 f_{1000} 的精确值, 其中 f_n 是斐波那契数。
2. 求比 1 000 000 大、比 1 000 000 000 大和比 1 000 000 000 000 大的最小的斐波那契数。
3. 求尽可能多的同为素数的斐波那契数, 目前还不知道是否存在无限多个这样的数。
4. 写出求解 10 个盘子的汉诺塔难题所需要的所有的移动。
5. 按照雷夫难题的规则, 用 4 根柱子从一根柱到另一根柱移动 20 个盘子, 写出使用弗雷姆-斯图尔特算法需要的所有移动。
6. 通过下面的方法验证求解 n 个盘子的雷夫难题的弗雷姆猜想: 对于尽可能多的整数 n , 证明这个难题不可能使用比具有最优选择 k 的弗雷姆-斯图尔特算法还要少的移动来求解。
7. 计算对于各种整数 n , 包括 16, 64, 256 和 1024, 使用在 4.3 节描述的快速乘法和整数相乘的标准算法做两个 n 位整数相乘所需要的运算次数。
8. 计算对于各种整数 n , 包括 4, 16, 64 和 128, 使用在 4.3 节描述的快速矩阵乘法和矩阵相乘的标准算法做两个 $n \times n$ 矩阵相乘所需要的运算次数。
9. 使用埃拉托色尼筛求不超过 1000 的所有素数。
10. 使用在 4.6 节描述的求不超过 100 的素数个数的方法求不超过 10 000 的素数个数。
11. 列出 $\{1, 2, 3, 4, 5, 6, 7, 8\}$ 的所有的错位排列。
12. 对所有不超过 20 的正整数 n 计算 n 个物体的一个排列是错位排列的概率, 并确定这些概率逼近 e 的速度。

写作题目

用本书以外的资料, 按下列要求写成短文。

1. 找出斐波那契发表的关于兔子数模型难题的原始材料。讨论斐波那契提出的这个问题和其他问题, 并且给出关于斐波那契本人的某些信息。
2. 解释斐波那契数怎样在其他应用中出现, 如叶序、植物叶片排列的研究、镜子反射的研究等。
3. 描述汉诺塔难题的各种不同的变形问题, 包括多于 3 个柱子的(包括课本和练习中讨论的雷夫难题在内), 盘子移动受限制的, 以及允许有同样大小盘子的。关于求解每种变形问题所要求的移动次数有什么已知的结论?
4. 尽可能多地讨论出现卡塔兰数的不同的问题。
5. 描述派尔克发现的与一次谎话搜索相关的乌拉姆问题的解。
6. 讨论派尔克发现的与多次谎话搜索相关的乌拉姆问题的变种, 关于这个问题你还知道什么?
7. 讨论求平面上点集的凸包的分治算法。
8. 查询幸运数(lucky number)的定义。解释怎样使用类似于埃拉托色尼筛的技术找出幸运数。求所有小于 1000 的幸运数。
9. 描述在数论中使用的筛法。使用这种方法已经得到了哪些结果?
10. 查询古代法国纸牌相遇游戏的规则。描述这些规则并且描述皮埃尔·雷蒙德·蒙特莫特关于“相遇问题”的论文。
11. 描述怎样使用指数生成函数求解各种计数问题。
12. 描述计数的 Polyá 理论和可使用这个理论求解的计数问题的种类。
13. 管家问题是求安排 n 对夫妇围圆桌就座的方法数, 使得就座时男女相间并且没有丈夫和妻子相邻。解释怎样用卢卡斯(E. Lucas)方法求解这个问题。
14. 解释怎样使用棋盘多项式(rook polynomial)求解计数问题。

关 系

在许多情况下集合的元素之间都存在某种关系。每天我们都要涉及各种关系，例如在一个商行和它的电话号码之间的关系，雇员与其工资之间的关系，一个人与一个亲属之间的关系，等等。在数学中我们研究的关系，如一个正整数与被它除的一个正整数、一个整数与和它模 5 同余的一个整数、一个实数与一个比它大的实数之间的关系，以及一个实数 x 和它的函数值 $f(x)$ ，等等。在计算机科学中常常出现的关系，如一个程序与它所使用的一个变量、一种计算机语言与这个语言的一个有效语句之间的关系等。

集合的元素之间的关系被表示成一种结构，这种结构叫做关系。它其实是集合间的笛卡儿积的一个子集。可以用关系来求解问题，例如，确定在一个网络中的哪两个城市之间开通航线，为一个复杂课题的不同阶段的工作找一个可行的执行次序，或者产生一个有用的方式以便在计算机数据库中存储信息。

在某些计算机语言中，一个变量名仅是它的前 31 个字母有效。此关系由前 31 个字母相同的字符串有序对组成，这是一种被称为等价关系的特殊关系。等价关系在数学和计算机科学中均有体现。本章将要学习等价关系和其他几种特殊关系。

5.1 关系及其性质

5.1.1 引言

 可以用两个相关元素构成的有序对来表达两个集合的元素之间的关系，这是一种最直接的方式。为此，有序对的集合就叫做二元关系。在这一节，我们引入用于描述二元关系的基本术语。在这一章的后面，我们将使用关系来求解涉及通信网络、项目调度以及识别集合中具有共同性质的元素等问题。

定义 1 设 A 和 B 是集合，一个从 A 到 B 的二元关系是 $A \times B$ 的子集。

换句话说，一个从 A 到 B 的二元关系是有序对的集合 R ，其中每个有序对的第一个元素取自 A 并且第二个元素取自 B 。我们使用记号 aRb 表示 $(a, b) \in R$ ， $a \not R b$ 表示 $(a, b) \notin R$ 。当 (a, b) 属于 R 时称 a 与 b 有关系 R 。

二元关系表示了两个集合的元素之间的关系。在本章的后面我们将引入 n 元关系，它表示了三个以上集合中元素之间的关系。当不发生混淆时我们将省去二元这个词。

例 1~例 3 说明了关系的概念。

例 1 设 A 是你们学校的学生的集合， B 是课程的集合。令 R 是由 (a, b) 对构成的关系，其中 a 是选修课程 b 的学生。例如，如果 Jason Goodfriend 和 Deborah Sherman 选修 CS518，即离散数学，有序对 $(\text{Jason Goodfriend}, \text{CS518})$ 和 $(\text{Deborah Sherman}, \text{CS518})$ 属于 R 。如果 Jason Goodfriend 也选修 CS510，即数据结构，那么有序对 $(\text{Jason Goodfriend}, \text{CS510})$ 也属于 R 。但是，如果 Deborah Sherman 没有选修 CS510，那么有序对 $(\text{Deborah Sherman}, \text{CS510})$ 不在 R 中。

注意如果一个学生目前没有选修任何课程，那么在 R 中没有以这个学生为第一元素的对。类似地，如果一门课程目前没有开设，那么在 R 中也没有以这门课程作为第二个元素的对。 ■

例 2 设 A 是所有城市的集合， B 是美国的 50 个州的集合。如下定义关系 R ：如果城市 a 是在州 b ，则 (a, b) 属于 R 。例如， $(\text{Boulder}, \text{科罗拉多州})$ ， $(\text{Bangor}, \text{缅因州})$ ， $(\text{Ann Arbor}, \text{密歇根州})$ ， $(\text{Cupertino}, \text{加利福尼亚州})$ 和 $(\text{Red Bank}, \text{新泽西州})$ 是在 R 中。 ■

例3 设 $A=\{0, 1, 2\}$, $B=\{a, b\}$, 那么 $\{(0, a), (0, b), (1, a), (2, b)\}$ 是从 A 到 B 的关系。这意味着, 比如说有 $0Ra$, 但没有 $1Rb$, 关系可以用图来表示, 如图 5-1 所示, 用箭头来表示有序对。另一种表示关系的方式就是用一张表, 这也在图 5-1 给出。在 5.3 节我们将更详细地讨论关系的表示。

5.1.2 函数作为关系

一个从集合 A 到集合 B 的函数 f (如 2.3 节的定义) 对于 A 中的每个元素都指定 B 中一个唯一的元素。 f 的图示是使得 $b=f(a)$ 的有序对 (a, b) 的集合。由于 f 的图示是 $A \times B$ 的子集, 它就是一个从 A 到 B 的关系。此外, 函数的图示有下述性质: A 的每个元素恰好是图中一个有序对的第一元素。

相反, 如果 R 是从 A 到 B 的关系, 并且使得 A 中的每个元素是 R 中恰好一个有序对的第一元素, 那么 R 的图示就可以定义一个函数。只要对 A 的每个元素指定唯一的元素 $b \in B$ 使得 $(a, b) \in R$ 就可以做到。

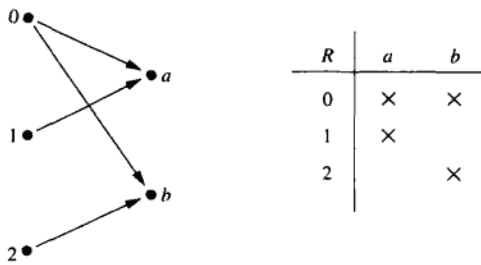


图 5-1 在例 3 的关系 R 中的有序对

可以用关系表达在集合 A 和集合 B 之间的一对多的相关性, 其中 A 的一个元素可以与 B 中多个元素有关。函数表示了这样一种关系, 对于 A 中的每个元素恰好只有一个 B 中的元素与之相关。

关系是函数的推广, 可以用关系表示集合之间的更为广泛的联系。

5.1.3 集合的关系

集合 A 到它自身的关系是特别令人感兴趣的。

定义 2 集合 A 的关系是从 A 到 A 的关系。

换句话说, 集合 A 的关系是 $A \times A$ 的子集。

例 4 设 A 是集合 $\{1, 2, 3, 4\}$, A 上的关系 $R = \{(a, b) \mid a \text{ 整除 } b\}$ 中有哪些有序对?

解 (a, b) 在 R 中, 当且仅当 a 和 b 是不超过 4 的正整数且 a 整除 b , 我们看到

$$R = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 4), (3, 3), (4, 4)\}$$

在这个关系中的有序对的图和表的表示都在图 5-2 中给出。

例 5 给出了某些整数集合上的关系的实例。

例 5 考虑下面这些整数集合上的关系:

$$R_1 = \{(a, b) \mid a \leq b\}$$

$$R_2 = \{(a, b) \mid a > b\}$$

$$R_3 = \{(a, b) \mid a = b \text{ 或 } a = -b\}$$

$$R_4 = \{(a, b) \mid a = b\}$$

$$R_5 = \{(a, b) \mid a = b + 1\}$$

$$R_6 = \{(a, b) \mid a + b \leq 3\}$$

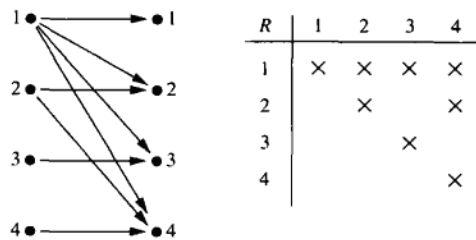


图 5-2 例 4 中关系 R 的有序对

其中, 哪些关系包含了有序对 $(1, 1)$, $(1, 2)$, $(2, 1)$, $(1, -1)$ 以及 $(2, 2)$?

注意 和例 1~例 4 的关系不同, 这些是无穷集合上的关系。

解 有序对 $(1, 1)$ 在 R_1 , R_3 , R_4 和 R_6 中; 有序对 $(1, 2)$ 在 R_1 和 R_6 中; 有序对 $(2, 1)$ 在 R_2 , R_5 和 R_6 中; $(1, -1)$ 在 R_2 , R_3 和 R_6 中; 最后, 有序对 $(2, 2)$ 在 R_1 , R_3 和 R_4 中。

不难确定有穷集上的关系个数, 因为集合 A 上的关系仅仅是 $A \times A$ 的子集。

例 6 n 元素集合有多少个关系?

解 集合 A 上的关系是 $A \times A$ 的子集。因为当 A 是 n 元素集合时 $A \times A$ 有 n^2 个元素, 并且 m

个元素的集合有 2^m 个子集, 故 $A \times A$ 的子集有 2^n 个。于是 n 元素集合有 2^n 个关系。例如, 在集合 $\{a, b, c\}$ 上存在 $2^{3^2} = 2^9 = 512$ 个关系。

5.1.4 关系的性质

有若干个用于把集合上的关系分类的性质。这里我们只介绍其中最重要的性质。

在某些关系中一个元素总是与自己相关。例如, 设 R 是所有的人的集合上的关系, 若 x 和 y 有相同的母亲和相同的父亲, 那么 (x, y) 属于 R 。于是对于每个人 x , 有 xRx 。

定义 3 如果对每个元素 $a \in A$ 有 $(a, a) \in R$, 那么集合 A 上的关系 R 叫做自反的。

注意 我们看到如果 $\forall a((a, a) \in R)$, 那么集合上的关系是自反的, 这里的论域是 A 中所有元素的集合。

我们看到如果 A 的每个元素都关系到它自己, A 上的关系就是自反的。例 7~9 说明了自反关系的概念。

例 7 考虑下面 $\{1, 2, 3, 4\}$ 上的关系

$$R_1 = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 4), (4, 1), (4, 4)\}$$

$$R_2 = \{(1, 1), (1, 2), (2, 1)\}$$

$$R_3 = \{(1, 1), (1, 2), (1, 4), (2, 1), (2, 2), (3, 3), (4, 1), (4, 4)\}$$

$$R_4 = \{(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3)\}$$

$$R_5 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\}$$

$$R_6 = \{(3, 4)\}$$

其中哪些是自反的?

解 关系 R_1 和 R_3 是自反的, 因为它们都包含了所有形如 (a, a) 的对, 即 $(1, 1)$, $(2, 2)$, $(3, 3)$ 和 $(4, 4)$ 。其他的不是自反的, 因为它们不包含所有这些有序对。特别地, R_2 , R_4 和 R_6 不是自反的, 因为 $(3, 3)$ 都不在这些关系里。

例 8 例 5 中哪些关系是自反的?

解 这个例子中的自反关系是 R_1 (因为对每个整数 a 有, $a \leq a$), R_3 和 R_4 。对于这个例子中其他的关系容易找到不在关系中的形如 (a, a) 的对 (留给读者作为练习)。

例 9 正整数集合上的“整除”关系是自反的吗?

解 因为只要 a 是正整数就有 $a | a$, “整除”关系是自反的。(注意, 如果我们将正整数集替换为所有整数集, 则“整除”关系不是自反的, 因为 0 不能整除 0。)

在某些关系中一个元素与第二个元素相关, 当且仅当第二个元素也与第一个元素相关。比如一个关系由 (x, y) 对构成, 其中 x 和 y 是你校的学生并且他们至少学一门公共课程。这个关系就有这种性质。而某些关系有另一种性质, 即如果一个元素与第二个元素相关, 那么第二个元素就不与第一个元素相关。比如一个关系由 (x, y) 对构成, 其中 x 和 y 是你校的学生, 且 x 比 y 的平均成绩高。这个关系就有后一种性质。

定义 4 对于 $a, b \in A$, 如果只要 $(a, b) \in R$ 就有 $(b, a) \in R$, 则集合 A 上的关系 R 叫做对称的。如果对 $a, b \in A$, 仅当 $a = b$ 时 $(a, b) \in R$ 和 $(b, a) \in R$, 则集合 A 上的关系 R 叫做反对称的。

注意 使用量词, 可以看到如果 $\forall a \forall b((a, b) \in R \rightarrow (b, a) \in R)$, 则 A 上的关系 R 是对称的。类似地, 如果 $\forall a \forall b((a, b) \in R \wedge (b, a) \in R \rightarrow (a = b))$, 则 A 上的关系 R 是反对称的。

就是说, 关系 R 是对称的, 当且仅当如果 a 与 b 相关则 b 与 a 就相关。关系 R 是反对称的, 当且仅当不存在由不同元素 a 和 b 构成的有序对使得 a 与 b 相关并且 b 与 a 也相关。对称与反对称的概念不是对立的, 因为一个关系可以同时有这两种性质或者两种性质都没有。一个关系如果包含了某些形如 (a, b) 的对, 其中 $a \neq b$, 这个关系就不可能同时是对称和反对称的。

注意 尽管从统计数据中可以看出, 对称的或反对称的关系相对不多, 但许多重要的关系都具有两种性质其中之一(见练习 21)。

例 10 例 7 中的哪些关系是对称的? 哪些是反对称的?

解 关系 R_2 和 R_3 是对称的, 因为在这些关系中, 只要 (a, b) 属于关系就有 (b, a) 属于关系。对于关系 R_2 , 唯一需要检查的就是 $(1, 2)$ 和 $(2, 1)$ 属于这个关系。对于 R_3 , 必须要检查 $(1, 2)$ 和 $(2, 1)$ 属于这个关系, 还有 $(1, 4)$ 和 $(4, 1)$ 也属于这个关系。读者应该能够验证其他的系统中没有一个是反对称的, 只需找到一个有序对 (a, b) , 使得它在关系中但 (b, a) 不在关系中。

R_1 、 R_5 和 R_6 都是反对称的。其中每一个关系都不存在这样的有序对, 即它由元素 a 和 b 构成, 使得 $a \neq b$, 但 (a, b) 和 (b, a) 都属于这个关系。读者只需找到有序对 (a, b) 满足 $a \neq b$, 但 (a, b) 和 (b, a) 都属于这个关系, 就能够验证其他系统中没有一个是反对称的。 ■

例 11 例 5 中的哪些关系是对称的? 哪些是反对称的?

解 关系 R_3 、 R_4 和 R_6 是对称的。 R_3 是对称的, 因为如果 $a=b$ 或 $a=-b$, 那么就有 $b=a$ 或 $b=-a$ 。 R_4 是对称的, 因为 $a=b$ 推出 $b=a$ 。 R_6 是对称的, 因为 $a+b \leq 3$ 推出 $b+a \leq 3$ 。读者应该能验证其他关系没有一个是反对称的。

关系 R_1 、 R_2 、 R_5 和 R_7 是反对称的。 R_1 是反对称的, 因为不等式 $a \leq b$ 和 $b \leq a$ 推出 $a=b$ 。 R_2 是反对称的, 因为 $a > b$ 和 $b > a$ 是不可能的。 R_5 是反对称的, 因为两个元素相对于 R_5 相关, 当且仅当它们是相等的。 R_7 是反对称的, 因为 $a=b+1$ 和 $b=a+1$ 是不可能的。读者应该能验证其他关系没有一个是反对称的。 ■

例 12 正整数集合上的整除关系是对称的吗? 是反对称的吗?

解 这个关系不是对称的, 因为 $1 \mid 2$, 但 $2 \nmid 1$ 。它是反对称的, 因为如果 a 和 b 是正整数且 $a \mid b$ 和 $b \mid a$, 那么 $a=b$ (这个验证留给读者作为练习)。 ■

设 R 是有序对 (x, y) 构成的关系, 其中 x 与 y 是你校的学生, 且 x 比 y 得到更多的学分。假设 x 与 y 相关并且 y 与 z 相关, 这意味着 x 比 y 得到更多的学分并且 y 比 z 得到更多的学分。可以断言 x 比 z 得到更多的学分, 因此 x 与 z 相关。我们证明了 R 有传递性, 这个性质定义如下。

定义 5 如果对于 $a, b, c \in A$, $(a, b) \in R$ 并且 $(b, c) \in R$ 则 $(a, c) \in R$, 那么集合 A 上的关系 R 叫做传递的。

注意 使用量词, 如果 $\forall a \forall b \forall c ((a, b) \in R \wedge (b, c) \in R) \rightarrow (a, c) \in R$, 集合 A 上的关系就叫做传递的。

例 13 例 7 中的关系哪些是传递的?

解 R_4 、 R_5 和 R_6 是传递的。对于这些关系, 我们可以通过验证若 (a, b) 和 (b, c) 属于这个关系则 (a, c) 也属于这个关系来证明每个关系都是传递的。例如, R_4 是传递的, 因为只有 $(3, 2)$ 和 $(2, 1)$, $(4, 2)$ 和 $(2, 1)$, $(4, 3)$ 和 $(3, 1)$, 以及 $(4, 3)$ 和 $(3, 2)$ 是这种有序对, 而 $(3, 1)$, $(4, 1)$ 和 $(4, 2)$ 属于 R_4 。读者应该能验证 R_5 和 R_6 也是传递的。

R_1 不是传递的, 因为 $(3, 4)$ 和 $(4, 1)$ 属于 R_1 , 但 $(3, 1)$ 不属于 R_1 。 R_2 不是传递的, 因为 $(2, 1)$ 和 $(1, 2)$ 属于 R_2 , 但 $(2, 2)$ 不属于 R_2 。 R_3 不是传递的, 因为 $(4, 1)$ 和 $(1, 2)$ 属于 R_3 , 但 $(4, 2)$ 不属于 R_3 。 ■

例 14 例 5 中的哪些关系是传递的?

解 关系 R_1 、 R_2 、 R_3 和 R_4 是传递的。 R_1 是传递的, 因为 $a \leq b$ 和 $b \leq c$ 推出 $a \leq c$ 。 R_2 是传递的, 因为 $a > b$ 和 $b > c$ 推出 $a > c$ 。 R_3 是传递的, 因为 $a = \pm b$ 和 $b = \pm c$ 推出 $a = \pm c$ 。正如读者应该验证的, 显然 R_4 也是传递的。 R_5 不是传递的, 因为 $(2, 1)$ 和 $(1, 0)$ 属于 R_5 , 但 $(2, 0)$ 不属于 R_5 。 R_6 不是传递的, 因为 $(2, 1)$ 和 $(1, 2)$ 属于 R_6 , 但 $(2, 2)$ 不属于 R_6 。 ■

例 15 正整数集合上的“整除”关系是传递的吗?

解 假设 a 整除 b 且 b 整除 c , 那么存在正整数 k 和 l 使得 $b=ak$ 和 $c=bl$, 因此 $c=a(kl)$, 即 a 整除 c 。从而证明了这个关系是传递的。 ■

可以使用计数技术确定具有特殊性质的关系的个数。由此可以得知：这个性质在 n 元素集合的所有关系的集合上有多普遍。

例 16 n 元素集合上有多少个自反的关系？

解 A 上的关系 R 是 $A \times A$ 的子集。因此，要通过指定 $A \times A$ 中 n^2 个有序对中的每一个是否在 R 中来确定关系。因而，如果 R 是自反的，对于 $a \in A$ ， n 个有序对 (a, a) 中的每一个都必须在 R 中。其他 $n(n-1)$ 个形如 (a, b) 的有序对， $a \neq b$ ，可能在也可能不在 R 中。因此，由计数的乘积法则，存在 $2^{n(n-1)}$ 个自反的关系。[这就是选择具有 $a \neq b$ 的每个元素 (a, b) 是否属于 R 的方式数。]

n 元素集合上的对称关系和反对称关系数可以用和例 16 类似的推理得出(见本节末的练习 21)。计数 n 元素集合上的传递关系数的问题已超出本书的范围。

5.1.5 关系的组合

因为从 A 到 B 的关系是 $A \times B$ 的子集，可以按照两个集合组合的任何方式来组合两个从 A 到 B 的关系。考虑例 17 至例 19。

例 17 设 $A = \{1, 2, 3\}$ 和 $B = \{1, 2, 3, 4\}$ 。组合关系 $R_1 = \{(1, 1), (2, 2), (3, 3)\}$ 和 $R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4)\}$ 可以得到

$$R_1 \cup R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (3, 3)\}$$

$$R_1 \cap R_2 = \{(1, 1)\}$$

$$R_1 - R_2 = \{(2, 2), (3, 3)\}$$

$$R_2 - R_1 = \{(1, 2), (1, 3), (1, 4)\}$$

例 18 设 A 和 B 分别是学校的所有学生和所有课程的集合。假设 R_1 由所有有序对 (a, b) 组成，其中 a 是选修课程 b 的学生。 R_2 由所有的有序对 (a, b) 构成，其中课程 b 是 a 的必修课。关系 $R_1 \cup R_2$ ， $R_1 \cap R_2$ ， $R_1 \oplus R_2$ ， $R_1 - R_2$ 和 $R_2 - R_1$ 是什么？

解 关系 $R_1 \cup R_2$ 由所有的有序对 (a, b) 组成，其中 a 是一个学生，课程 b 要么是他的选修课，要么是他的必修课。 $R_1 \cap R_2$ 是有序对 (a, b) 的集合，其中 a 是一个学生，他选修了课程 b 并且课程 b 也是 a 的必修课。 $R_1 \oplus R_2$ 由所有的有序对 (a, b) 组成，其中学生 a 已经选修了课程 b 但课程 b 不是 a 的必修课，或者课程 b 是 a 的必修课，但是 a 没有选修它。 $R_1 - R_2$ 是所有对 (a, b) 的集合，其中 a 已经选修了课程 b ，但 b 不是 a 的必修课，即 b 是 a 的选修课。 $R_2 - R_1$ 是所有对 (a, b) 的集合，其中 b 是 a 的必修课，但 a 没有选修它。

例 19 设 R_1 是实数集合上的“小于”关系， R_2 是实数集合上的“大于”关系，即 $R_1 = \{(x, y) \mid x < y\}$ 和 $R_2 = \{(x, y) \mid x > y\}$ 。 $R_1 \cup R_2$ ， $R_1 \cap R_2$ ， $R_1 - R_2$ ， $R_2 - R_1$ ， $R_1 \oplus R_2$ 是什么？

解 注意到 $(x, y) \in R_1 \cup R_2$ 当且仅当 $(x, y) \in R_1$ 或 $(x, y) \in R_2$ ，因此 $(x, y) \in R_1 \cup R_2$ 当且仅当 $x < y$ 或 $x > y$ 。由于条件 $x < y$ 或 $x > y$ 与条件 $x \neq y$ 一样，因此 $R_1 \cup R_2 = \{(x, y) \mid x \neq y\}$ 。换句话说，“小于”关系与“大于”关系的并是“不相等”关系。

其次，注意到一个对 (x, y) 同时属于 R_1 和 R_2 是不可能的，因为 $x < y$ 并且 $x > y$ 是不可能的。从而得到 $R_1 \cap R_2 = \emptyset$ 。也可以看到 $R_1 - R_2 = R_1$ ， $R_2 - R_1 = R_2$ ， $R_1 \oplus R_2 = R_1 \cup R_2 - R_1 \cap R_2 = \{(x, y) \mid x \neq y\}$ 。

组合关系有另一种方式，这种方式和函数的复合相似。

定义 6 设 R 是从集合 A 到集合 B 的关系， S 是从集合 B 到集合 C 的关系。 R 和 S 的合成是由有序对 (a, c) 构成的关系，其中 $a \in A$ ， $c \in C$ ，并且对于它们存在一个元素 $b \in B$ 使得 $(a, b) \in R$ 和 $(b, c) \in S$ 。我们用 $S \circ R$ 表示 R 与 S 的合成。

计算两个关系的合成需要我们找出这些元素，它们既是第一个关系中的有序对的第二个元素，也是第二个关系中的有序对的第一个元素。正如例 20 和例 21 所示。

⊖ 原书此处为 $R_1 \cup R_2 - R_1 \cap R_2$ ，译文根据本书第 1 章的定义加了括号。——译者注

例 20 R 是从 $\{1, 2, 3\}$ 到 $\{1, 2, 3, 4\}$ 的关系且 $R = \{(1, 1), (1, 4), (2, 3), (3, 1), (3, 4)\}$, S 是从 $\{1, 2, 3, 4\}$ 到 $\{0, 1, 2\}$ 的关系且 $S = \{(1, 0), (2, 0), (3, 1), (3, 2), (4, 1)\}$, R 与 S 的合成是什么?

解 $S \circ R$ 是由所有 R 中的有序对和 S 中的有序对构成的, 其中在 R 中有序对的第二元素与 S 中有序对的第一元素相同。例如, R 中的有序对 $(2, 3)$ 和 S 中的有序对 $(3, 1)$ 产生了 $S \circ R$ 中的有序对 $(2, 1)$ 。计算所有在 $R \circ S$ 中的有序对, 我们得到

$$S \circ R = \{(1, 0), (1, 1), (2, 1), (2, 2), (3, 0), (3, 1)\}$$

例 21 双亲关系与自身的合成 设 R 是所有人集合上的双亲关系, 即如果 a 是 b 的父母, 则 $(a, b) \in R$ 。当且仅当存在一个人 b 使得 a 是 b 的父母且 b 是 c 的父母。换句话说, $(a, c) \in R \circ R$ 当且仅当 a 是 c 的祖父母或外祖父母。

从两个关系合成的定义可以递归地来定义关系 R 的幂。

定义 7 设 R 是集合 A 上的关系。幂 R^n , $n=1, 2, 3, \dots$, 递归地定义为

$$R^1 = R \quad \text{和} \quad R^{n+1} = R^n \circ R$$

这个定义证明了 $R^2 = R \circ R$, $R^3 = R^2 \circ R = (R \circ R) \circ R$, 等等。

例 22 设 $R = \{(1, 1), (2, 1), (3, 2), (4, 3)\}$, 求幂 R^n , $n=2, 3, 4, \dots$ 。

解 因为 $R^2 = R \circ R$, 我们得到 $R^2 = \{(1, 1), (2, 1), (3, 1), (4, 2)\}$ 。进一步, 因为 $R^3 = R^2 \circ R$, $R^3 = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$ 。其他的计算证明了 R^4 和 R^3 一样, 因此 $R^4 = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$ 。从而对 $n=5, 6, 7, \dots$, 有 $R^n = R^3$ 。读者应该能够验证这个结果。

下面的定理证明一个传递关系的幂是该关系的子集。5.4 节将要用到这一结果。

定理 1 集合 A 上的关系 R 是传递的, 当且仅当对 $n=1, 2, 3$, 有 $R^n \subseteq R$ 。

证 首先证明定理的充分条件。假设对 $n=1, 2, 3$, 有 $R^n \subseteq R$ 。特别地, 有 $R^2 \subseteq R$ 。为证明 R 的传递性, 注意到如果 $(a, b) \in R$ 并且 $(b, c) \in R$, 根据合成定义就有 $(a, c) \in R^2$ 。因为 $R^2 \subseteq R$, 这就意味着 $(a, c) \in R$ 。因此 R 是传递的。

我们将使用数学归纳法证明定理的必要条件。对于 $n=1$, 定理的这个结果是显而易见的。

假定 $R^n \subseteq R$, 其中 n 是一个正整数, 这是归纳假设。为完成归纳步骤, 必须证明这将推出 R^{n+1} 也是 R 的子集。为证明这一点, 假设 $(a, b) \in R^{n+1}$, 那么因为 $R^{n+1} = R^n \circ R$, 存在元素 $x \in A$ 使得 $(a, x) \in R$ 并且 $(x, b) \in R^n$ 。由归纳假设, 即 $R^n \subseteq R$, 推出 $(x, b) \in R$ 。下一步, 因为 R 是传递的, 以及 $(a, x) \in R$ 和 $(x, b) \in R$, 得到 $(a, b) \in R$ 。这就证明了 $R^{n+1} \subseteq R$, 从而完成了证明。□

练习

- 列出从 $A = \{0, 1, 2, 3, 4\}$ 到 $B = \{0, 1, 2, 3\}$ 的关系 R 中的有序对, 其中 $(a, b) \in R$, 当且仅当
 - $a = b$
 - $a + b = 4$
 - $a > b$
 - $a \mid b$
 - $\gcd(a, b) = 1$
 - $\text{lcm}(a, b) = 2$
- 对集合 $\{1, 2, 3, 4\}$ 上的如下每一个关系, 确定它是否是自反的, 是否是对称的, 是否是反对称的, 是否是传递的。
 - $\{(2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)\}$
 - $\{(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (4, 4)\}$
 - $\{(2, 4), (4, 2)\}$
 - $\{(1, 2), (2, 3), (3, 4)\}$
 - $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$
 - $\{(1, 3), (1, 4), (2, 3), (2, 4), (3, 1), (3, 4)\}$
- 确定所有 Web 页上的关系 R 是否为自反的、对称的、反对称的和传递的, 其中 $(a, b) \in R$, 当且仅当
 - 每个访问 Web 页 a 的人也访问了 Web 页 b 。
 - 在 Web 页 a 和 b 上没有公共链接。
 - 在 Web 页 a 和 b 上至少有一条公共链接。
 - 存在一个 Web 页包含了到 Web 页 a 和 Web 页 b 的链接。

4. 确定所有整数集合上的关系 R 是否是自反的、对称的、反对称的和传递的, 其中 $(x, y) \in R$, 当且仅当
- a) $x \neq y$ b) $xy \geq 1$ c) $x = y + 1$ 或 $x = y - 1$ d) $x \equiv y \pmod{7}$
- e) x 是 y 的倍数 f) x 与 y 都是负的或都是非负的 g) $x = y^2$ h) $x \geq y^2$

如果对于每个 $a \in A$, 有 $(a, a) \notin R$, 那么集合 A 上的关系 R 是反自反的, 即如果没有 A 中的元素与自己有关系, 关系 R 就是反自反的。

5. 练习 2 的哪些关系是反自反的?
6. 练习 3 的哪些关系是反自反的?
7. 集合上的关系可能既不是自反的也不是反自反的吗?
8. 给出在所有人的集合上一个反自反关系的例子。

一个关系 R 叫做非对称的, 如果 $(a, b) \in R$ 推出 $(b, a) \notin R$ 。

9. 使用量词表达一个关系是非对称的含义。
10. 从 m 元素集合到 n 元素集合有多少个不同的关系?

■ 设 R 是从集合 A 到集合 B 的关系。从 B 到 A 的逆关系是有序对的集合 $\{(b, a) \mid (a, b) \in R\}$, 记作 R^{-1} , 补关系 $\bar{R}$ 是有序对的集合 $\{(a, b) \mid (a, b) \notin R\}$ 。

11. 设 R 是正整数集合上的关系 $R = \{(a, b) \mid a \text{ 整除 } b\}$, 求
- a) R^{-1} b) $\bar{R}$
12. 设从 A 到 B 的函数 f 是一一对应的。令 R 是和 f 的图相等的关系, 即 $R = \{(a, f(a)) \mid a \in A\}$ 。逆关系 R^{-1} 是什么?
13. 设 A 是你校的学生集合, B 是学校图书馆的书的集合。设 R_1 和 R_2 都是有序对 (a, b) 构成的关系, 在 R_1 中, 学生 a 修一门课程需要读书 b , 在 R_2 中, 学生 a 已经读过书 b 。描述在下面每个关系中的有序对。
- a) $R_1 \cup R_2$ b) $R_1 \cap R_2$ c) $R_1 \oplus R_2$ d) $R_1 - R_2$ e) $R_2 - R_1$
14. 设 R 是由人的集合上的有序对 (a, b) 组成的关系, 其中 a 是 b 的父母。设 S 是由人的集合上的有序对 (a, b) 组成的关系, 其中 a 是 b 的兄弟姐妹。 $S \circ R$ 和 $R \circ S$ 是什么关系?

练习 15~16 涉及的都是实数集合上的关系:

$R_1 = \{(a, b) \in \mathbb{R}^2 \mid a > b\}$, “大于”关系

$R_2 = \{(a, b) \in \mathbb{R}^2 \mid a \geq b\}$, “大于或等于”关系

$R_3 = \{(a, b) \in \mathbb{R}^2 \mid a < b\}$, “小于”关系

$R_4 = \{(a, b) \in \mathbb{R}^2 \mid a \leq b\}$, “小于或等于”关系

$R_5 = \{(a, b) \in \mathbb{R}^2 \mid a = b\}$, “等于”关系

$R_6 = \{(a, b) \in \mathbb{R}^2 \mid a \neq b\}$, “不等于”关系

15. 求
- a) $R_2 \cup R_4$ b) $R_3 \cup R_5$ c) $R_3 \cap R_5$ d) $R_4 \cap R_6$
- e) $R_3 - R_6$ f) $R_6 - R_3$ g) $R_2 \oplus R_5$ h) $R_3 \oplus R_5$
16. 求
- a) $R_2 \circ R_1$ b) $R_2 \circ R_2$ c) $R_3 \circ R_5$ d) $R_4 \circ R_1$
- e) $R_5 \circ R_3$ f) $R_3 \circ R_6$ g) $R_4 \circ R_6$ h) $R_6 \circ R_6$
17. 设 R 是具有博士学位的人的集合上的关系, 要求 $(a, b) \in R$, 当且仅当 a 是 b 的导师。什么情况下一个有序对 (a, b) 在 R^2 中? 什么情况下一个有序对 (a, b) 在 R^n 中? 这里 n 是正整数。(注意每个具有博士学位的人有一个导师。)
18. 设 R_1 和 R_2 分别是整数集合上的“模 3 同余”和“模 4 同余”关系, 即 $R_1 = \{(a, b) \mid a \equiv b \pmod{3}\}$ 和 $R_2 = \{(a, b) \mid a \equiv b \pmod{4}\}$ 。求
- a) $R_1 \cup R_2$ b) $R_1 \cap R_2$ c) $R_1 - R_2$ d) $R_2 - R_1$ e) $R_1 \oplus R_2$
19. $\{0, 1\}$ 上的 16 个不同的关系中有多少个包含了对 $(0, 1)$?
20. a) 在集合 $\{a, b, c, d\}$ 上有多少个关系?
- b) 在集合 $\{a, b, c, d\}$ 上有多少个关系包含对 (a, a) ?
- *21. n 元素集合上有多少个关系是
- a) 对称的? b) 反对称的? c) 非对称的? d) 反自反的? e) 自反的和对称的?

f) 既不是自反的也不是反自反的?

22. 找出在下面定理证明中的错误。

“定理”: 设 R 是集合 A 上的对称和传递的关系, 则 R 是自反的。

“证明”: 设 $a \in A$ 。取元素 $b \in A$ 使得 $(a, b) \in R$ 。由于 R 是对称的, 因而有 $(b, a) \in R$ 。现在使用传递性, 由 $(a, b) \in R$ 和 $(b, a) \in R$ 可以得出结论 $(a, a) \in R$ 。

23. 证明: 集合 A 上的关系 R 是对称的, 当且仅当 $R = R^{-1}$, 其中 R^{-1} 是 R 的逆关系。

24. 证明: 集合 A 上的关系 R 是自反的, 当且仅当逆关系 R^{-1} 是自反的。

25. 设 R 是自反的和传递的关系, 证明对所有的正整数 n , $R^n = R$ 。

26. 设 R 是集合 A 上的自反关系, 证明对所有的正整数 n , R^n 也是自反的。

27. 假设关系 R 是反自反的, R^2 一定是反自反的吗? 对你的答案给出理由。

5.2 n 元关系及其应用

5.2.1 引言

在两个以上集合的元素中常常会产生某种关系。例如, 学生的姓名、学生的专业以及学生的平均学分绩点之间的关系。类似地, 一个航班的航空公司、航班号、出发地、目的地、起飞时间和到达时间等也有一种关系。在数学中也有这种关系。例如, 有 3 个整数, 其中第一个整数比第二个整数大, 而第二个整数比第三个整数大。另一个例子是直线上的点之间的关系, 即当第二个点在第一和第三个点之间时, 这三个点有关系。

本节我们将研究两个以上集合的元素之间的关系。这种关系叫做 n 元关系。可以用这种关系表示计算机的数据库。这种表示有助于回答对数据库中所存信息的查询, 例如: 哪个航班在午夜 3 点到 4 点之间降落在 O'Hare 机场? 你们学校的二年级学生哪些是主修数学或计算机科学的, 并且平均学分绩点大于 3.0? 公司的哪些雇员为这个公司工作不到 5 年但所赚的钱已经超过 50 000 美元?

5.2.2 n 元关系

我们从建立关系数据库理论的基本定义开始。

定义 1 设 $A_1, A_2, \dots, A_n$ 是集合。在这些集合上的 n 元关系是 $A_1 \times A_2 \times \dots \times A_n$ 的子集。这些集合 $A_1, A_2, \dots, A_n$ 叫做关系的域, n 叫做它的阶。

例 1 R 是 $\mathbf{N} \times \mathbf{N} \times \mathbf{N}$ 上的三元组 (a, b, c) 构成的关系, 其中 a, b, c 是满足 $a < b < c$ 的整数。那么 $(1, 2, 3) \in R$, 但 $(2, 4, 3) \notin R$ 。这个关系的阶是 3。它的域都等于整数集合。 ■

例 2 设 R 是 $\mathbf{Z} \times \mathbf{Z} \times \mathbf{Z}$ 上的三元组 (a, b, c) 构成的关系, 其中的 a, b, c 构成等差数列, 即 $(a, b, c) \in R$, 当且仅当存在一个整数 k , 使得 $b = a + k, c = a + 2k$, 或者 $b - a = k, c - b = k$ 。可知 $(1, 3, 5) \in R$, 因为 $3 = 1 + 2, 5 = 1 + 2 \cdot 2$, 但是 $(2, 5, 9) \notin R$, 因为 $5 - 2 = 3$, 而 $9 - 5 = 4$ 。这个关系的阶为 3, 且它的所有域均等于整数集。 ■

例 3 设 R 是 $\mathbf{Z} \times \mathbf{Z} \times \mathbf{Z}^+$ 上的三元组 (a, b, m) 构成的关系, 其中的 a, b, m 都是整数, 且满足 $m \geq 1$, 和 $a \equiv b \pmod{m}$ 。则 $(8, 2, 3), (-1, 9, 5)$ 和 $(14, 0, 7)$ 都属于 R , 但 $(7, 2, 3), (-2, -8, 5)$ 和 $(11, 0, 6)$ 都不属于 R , 因为 $8 \not\equiv 2 \pmod{3}, -1 \not\equiv 9 \pmod{5}$ 和 $14 \not\equiv 0 \pmod{7}$, 而 $7 \not\equiv 2 \pmod{3}, -2 \not\equiv -8 \pmod{5}$ 和 $11 \not\equiv 0 \pmod{6}$ 。这个关系的阶为 3, 且它的前两个域是全部整数集而第三个域为正整数集。 ■

例 4 设 R 是由 5 元组 (A, N, S, D, T) 构成的表示飞机航班的关系, 其中 A 是航空公司, N 是航班号, S 是出发地, D 是目的地, T 是起飞时间。例如, 如果 Nadir 直达航空公司在 15:00 有从 Newark 到 Bangor 的 963 航班, 那么 $(\text{Nadir}, 963, \text{Newark}, \text{Bangor}, 15:00)$ 属于 R 。这个关系的阶是 5, 并且它的域是所有航空公司的集合、航班号的集合、城市的集合、城市的集合以及时间的集合。 ■

5.2.3 数据库和关系

操作数据库信息所需要的时间依赖于这些信息是怎样存储的。插入和删除记录, 更新记录, 检索记录以及从一个重叠的数据库中组合记录的操作, 在一个大型数据库中每

天要执行几百万次。由于这些操作的重要性，已经开发了数据库表示的各种方法。我们将讨论其中的一种基于关系概念的方法，叫做关系数据模型。

数据库由记录组成，这些记录是由字段构成的 n 元组。这些字段是 n 元组的数据项。例如，学生记录的数据库可以由包含学生的姓名、学号、专业、平均学分绩点(GPA)的字段构成。关系数据模型把一个记录的数据库表示成一个 n 元关系。于是，学生记录可以被表示成形如(学生姓名，学号，专业，GPA)的 4 元组。6 个记录的一个数据库样本是：

- (Ackermann, 231455, 计算机科学, 3.88)
- (Adams, 888323, 物理学, 3.45)
- (Chou, 102147, 计算机科学, 3.49)
- (Goodfriend, 453876, 数学, 3.45)
- (Rao, 678543, 数学, 3.90)
- (Stevens, 786576, 哲学, 2.99)

用于表示数据库的关系叫做表，因为这些关系常常用表来给出。这些表的每个列对应于数据库的一个属性。例如，同样的学生数据库显示在表 5-1。这个数据库的属性是学生姓名、学号、专业和平均学分绩点(GPA)。

表 5-1 学生

学生姓名	学号	专业	GPA
Ackermann	231 455	计算机科学	3.88
Adams	888 323	物理学	3.45
Chou	102 147	计算机科学	3.49
Goodfriend	453 876	数学	3.45
Rao	678 543	数学	3.90
Stevens	786 576	心理学	2.99

当 n 元组的某个域的值能够确定这个 n 元组时， n 元关系的这个域就叫做主关键字。这就是说当关系中没有两个 n 元组在这个域有相同的值时，这个域就是主关键字。

常常要从数据库增加或删除记录。由于这一点，一个域是主关键字的性质是随时间而改变的。所以，一个主关键字应该选择那种无论数据库怎样改变都能继续存在的字段。一个关系当前含有的所有 n 元组称为该关系的外延。数据库更持久的内容，包括它的名字和属性，则被称作数据库的内涵。选择主关键字的时候，应当选择那种能够为本数据库所有可能的外延充当主关键字的。要做到这一点，就必须认真考察数据库的内涵，以便预知可能在外延中出现的 n 元组集。

例 5 假设将来不再增加 n 元组，对于表 5-1 所示的 n 元关系，哪个域是主关键字？

解 因为在这个表中对应每个学生的姓名只有一个 4 元组，学生姓名的域是主关键字。类似地，在这个表中学号是唯一的，学号的域也是主关键字。但是，所学专业的域不是主关键字，因为包含同样的专业的 4 元组有多个。平均学分绩点的域也不是主关键字，因为有 2 个 4 元组包含了同样的 GPA。

在一个 n 元关系中域的组合也可以唯一地标识 n 元组。当一组域的值确定了一个关系中的 n 元组时，这些域的笛卡儿积就叫做复合关键字。

例 6 对于表 5-1 中的 n 元关系，假设不增加 n 元组，所学专业的域与 GPA 的域的笛卡儿积是复合关键字吗？

解 这个表中没有两个 4 元组有同样的专业和同样的 GPA，因此这个笛卡儿积是一个复合关键字。

因为主关键字和复合关键字用于唯一地标识数据库中的记录，当新的记录被加到这个数据库时关键字要保持有效是非常重要的。因此，应该做检测以保证在这个或这些相应的字段中每个新

记录与表中所有其他的记录不同。例如,使用学号作为学生记录的关键字是有意义的,因为没有两个学生有同样的学号。一个大学不应该使用姓名字段作为关键字,因为有可能两个学生有同样的姓名(如 John Smith)。

5.2.4 n 元关系的运算

n 元关系存在可以用于构成新的 n 元关系的不同的运算。综合使用这些运算能够回答对数据库的满足特定条件的所有元组的查询。

一个 n 元关系最基本的运算是在这个 n 元关系中确定满足特定条件的所有 n 元组。例如,我们可能想在学生记录的数据库找出主修计算机科学专业的所有学生的记录。可能我们想在同一个数据库找出所有平均学分绩点在 3.5 以上的学生。也可能我们想在这个数据库找出所有主修计算机科学专业的平均学分绩点在 3.5 以上的学生。为完成这些任务我们使用了选择运算。

定义 2 设 R 是 n 元关系, C 是 R 中元素可能满足的一个条件。那么选择运算 s_C 将 n 元关系 R 映射到 R 中满足条件 C 的所有 n 元组构成的 n 元关系。

例 7 为了找出表 5-1 所示的 n 元关系 R 中主修计算机科学专业的学生的记录,我们使用运算 s_{C_1} , 其中 C_1 是条件专业 = “计算机科学”。结果是两个 4 元组 (Ackermann, 231455, 计算机科学, 3.88) 和 (Chou, 102147, 计算机科学, 3.49)。类似地,为了在这个数据库找出平均学分绩点在 3.5 以上的学生记录,我们使用运算 s_{C_2} , 其中 C_2 是条件 $GPA > 3.5$ 。结果是两个 4 元组 (Ackermann, 231455, 计算机科学, 3.88) 和 (Rao, 678543, 数学, 3.90)。最后,为找出主修计算机科学专业的 GPA 在 3.5 以上的学生记录,我们使用运算 s_{C_3} , 其中 C_3 是条件 (专业 = “计算机科学” $\wedge$ $GPA > 3.5$)。结果由一个 4 元组 (Ackermann, 231455, 计算机科学, 3.88) 构成。

通过投影删去关系中每条记录相同的域可以构建一个新的 n 元关系。

定义 3 投影 $P_{i_1, i_2, \dots, i_m}$ 将 n 元组 $(a_1, a_2, \dots, a_n)$ 映到 m 元组 $(a_{i_1}, a_{i_2}, \dots, a_{i_m})$, 其中 $m \leq n$ 。

换句话说,投影 $P_{i_1, i_2, \dots, i_m}$ 删除了 n 元组的 $n - m$ 个分量,保留了第 $i_1, i_2, \dots, i_m$ 个分量。

例 8 当投影 $P_{1,3}$ 施用到 4 元组 (2, 3, 0, 4), (Jane Doe, 234111001, 地理学, 3.14) 以及 (a_1, a_2, a_3, a_4) 时结果是什么?

解 $P_{1,3}$ 把这些 4 元组分别映到 (2, 0), (Jane Doe, 地理学) 和 (a_1, a_3) 。

例 9 说明了怎样使用投影来产生新关系。

例 9 当投影 $P_{1,4}$ 施用于表 5-1 中的关系时结果是什么?

解 当施用投影 $P_{1,4}$ 时,表的第二列和第三列被删除,得到了表示学生姓名和平均学分绩点的对。表 5-2 给出了这个投影的结果。

当一个投影施用到一个关系的表中时,有可能使行变少。当关系中的某些 n 元组在投影的 m 个分量中每个分量的值都相同,只在被删除的分量有不同的值时,就会出现这种情况。例如,考虑下面的例子。

例 10 当投影 $P_{1,2}$ 施用到表 5-3 的关系时得到什么表?

解 表 5-4 给出了当投影 $P_{1,2}$ 施用到表 5-3 时得到的关系。注意在施用了这个投影后行数减少。

表 5-2 GPA

学生姓名	GPA
Ackermann	3.88
Adams	3.45
Chou	3.49
Goodfriend	3.45
Rao	3.90
Stevens	2.99

表 5-3 注册

学生	专业	课程
Glauser	生物学	BI 290
Glauser	生物学	MS 475
Glauser	生物学	PY 410
Marcus	数学	MS 511
Marcus	数学	MS 603
Marcus	数学	CS 322
Miller	计算机科学	MS 575
Miller	计算机科学	CS 455

当两个表分享着某些相同的字段时连接运算将这两个表组合成一个表。例如，一个表包含了航空公司、航班号和通道的字段，另一个表包含了航班号、通道和起飞时间的字段。可以将这两个表组合成一个包含航空公司、航班号、通道和起飞时间字段的表。

定义 4 设 R 是 m 元关系且 S 是 n 元关系，其中 $p \leq m$ 和 $p \leq n$ 。连接 $J_p(R, S)$ 是 $m+n-p$ 元关系，它包含了所有的 $(m+n-p)$ 元组 $(a_1, a_2, \dots, a_{m-p}, c_1, c_2, \dots, c_p, b_1, b_2, \dots, b_{n-p})$ ，其中 m 元组 $(a_1, a_2, \dots, a_{m-p}, c_1, c_2, \dots, c_p)$ 属于 R 且 n 元组 $(c_1, c_2, \dots, c_p, b_1, b_2, \dots, b_{n-p})$ 属于 S 。

换句话说，连接运算 J_p 从两个关系产生一个新的关系，它把第一个关系的所有 m 元组和第二个关系的所有 n 元组组合起来，其中 m 元组的后 p 个分量与 n 元组的前 p 个分量相同。

例 11 当用连接运算 J_2 组合表 5-5 和表 5-6 中的关系时所得结果关系是什么？

解 J_2 产生的关系在表 5-7 中。

表 5-4 专业

学生	专业
Glauser	生物学
Marcus	数学
Miller	计算机科学

表 5-5 教学课程

教授	系	课号
Cruz	动物学	335
Cruz	动物学	412
Farber	心理学	501
Farber	心理学	617
Grammer	物理学	544
Grammer	物理学	551
Rosen	计算机科学	518
Rosen	数学	575

表 5-6 教室安排

系	课号	教室	时间
计算机科学	518	N521	2:00P. M.
数学	575	N502	3:00P. M.
数学	611	N521	4:00P. M.
物理学	544	B505	4:00P. M.
心理学	501	A100	3:00P. M.
心理学	617	A110	11:00A. M.
动物学	335	A100	9:00A. M.
动物学	412	A100	8:00A. M.

表 5-7 教学安排

教授	系	课号	教室	时间
Cruz	动物学	335	A100	9:00A. M.
Cruz	动物学	412	A100	8:00A. M.
Farber	心理学	501	A100	3:00P. M.
Farber	心理学	617	A110	11:00A. M.
Grammer	物理学	544	B505	4:00P. M.
Rosen	计算机科学	518	N521	2:00P. M.
Rosen	数学	575	N502	3:00P. M.

从已知关系产生新关系的运算除了投影和连接运算以外还有其他运算。对这些运算的描述可以在关于数据库理论的书中找到。

5.2.5 SQL



数据库查询语言 SQL(结构化查询语言 Structured Query Language 的缩写)可以用来执行本节所描述的运算。例 12 说明了 SQL 命令是怎样与 n 元关系上的运算相关的。

例 12 我们将说明怎样用 SQL 来表达查询，这是通过显示怎样使用 SQL 从表 5-8 中做一次关于航班的查询来说明的。SQL 语句

```
SELECT Departure_time      (选择起飞时间)
FROM Flights               (从航班)
WHERE destination = 'Detroit' (当目的地 = '底特律')
```

是用于在航班数据库找出投影 P_5 ，它能够(在起飞时间属性上)选出 5 元组使得这些 5 元组满足条件: $\text{destination} = \text{'Detroit'}$ 。输出是一张以底特律为目的地的包含航班时间的表, 即 08:10, 08:47 和 09:44。SQL 使用 FROM 子句识别查询施用到的 n 元关系, WHERE 子句说明选择运算的条件, 而 SELECT 子句说明将被使用的投影运算。(小心: SQL 使用 SELECT 表示一个投影, 而不是一个选择运算。这是一个令人遗憾的术语不一致的例子。)

表 5-8 航班

航空公司	航班号	通道	目的地	起飞时间
Nadir	122	34	底特律	08:10
Acme	221	22	丹佛	08:17
Acme	122	33	安克雷奇	08:22
Acme	323	34	檀香山	08:30
Nadir	199	13	底特律	08:47
Acme	222	22	丹佛	09:10
Nadir	322	34	底特律	09:44

例 13 说明 SQL 怎样做涉及多个表的查询。

例 13 SQL 语句

```
SELECT professor, time           (选择教授, 时间)
FROM teaching_assignments, class_schedule (从教学课程, 教室安排)
WHERE department = 'mathematics' (当系 = '数学')
```

用于找出在数据库(如表 5-7 所示)中的 5 元组的投影 $P_{1,5}$, 这个数据库是分别在表 5-5 中的教学课程和表 5-6 中的教室安排, 数据库满足 $\text{department} = \text{'mathematics'}$ 条件下的连接 J_2 。输出由单一的 2 元组 (Rosen, 3:00PM) 构成。这里的 SQL FROM 子句用于找出两个不同数据库的连接。

本节我们仅仅接触到关系数据库的基本概念。更多的信息可以在 [AhUl95] 中找到。

练习

- 列出关系 $\{(a, b, c) \mid a, b \text{ 和 } c \text{ 是整数且 } 0 < a < b < c < 5\}$ 中的三元组。
- 列出表 5-8 关系中的 5 元组。
- 假设不增加新的 n 元组, 对于表 5-8 中的数据库找出一个由两个字段构成的复合关键字, 其中一个字段是航空公司。
 - 学号可能是主关键字吗?
 - 姓名可能是主关键字吗?
 - 电话号码可能是主关键字吗?
- 3 元关系中的 3 元组表示了一个学生数据库中的下述属性: 学号、姓名、电话号码。
 - 对这个关系确定一个主关键字。
 - 在什么条件下(姓名、住址)是复合关键字?
 - 在什么条件下(姓名、住址、城市)是复合关键字?
- 5 元关系中的 5 元组表示了美国所有人的下述属性: 姓名、社会保险号、住址、城市、州。
 - 对这个关系确定一个主关键字。
 - 在什么条件下(姓名、住址)是复合关键字?
 - 在什么条件下(姓名、住址、城市)是复合关键字?
- 设 C 是条件: 目的地=底特律, 当你施用选择运算 s_C 到表 5-8 的数据库时, 可以得到什么?
- 设 C 是条件: $(\text{航空公司} = \text{Nadir}) \vee (\text{目的地} = \text{丹佛})$, 当你施用选择运算 s_C 到表 8-8 的数据库时, 可以得到什么?
- 哪个投影映射用于删除一个 6 元组的第一、第二和第四个分量?
- 给出施用投影 $P_{1,4}$ 到表 5-8 以后得到的表。
- 构造把连接运算 J_2 施用到表 5-9 和表 5-10 的关系中所得到的表。

表 5-9 零件需求

供货商	零件号	项目
23	1092	1
23	1101	3
23	9048	4
31	4975	3
31	3477	2
32	6984	4
32	9191	2
33	1001	1

表 5-10 零件库存

零件号	项目	数量	颜色代码
1001	1	14	8
1092	1	2	2
1101	3	1	1
3477	2	25	2
4975	3	6	2
6984	4	10	1
9048	4	12	2
9191	2	80	4

11. 证明：如果 C_1 和 C_2 是 n 元关系 R 的元素可能满足的条件，那么 $s_{C_1}(s_{C_2}(R)) = s_{C_2}(s_{C_1}(R))$ 。
12. 证明：如果 C 是 n 元关系 R 和 S 的元素可能满足的条件，那么 $s_C(R \cap S) = s_C(R) \cap s_C(S)$ 。
13. 证明：如果 R 和 S 是 n 元关系，那么 $P_{i_1, i_2, \dots, i_m}(R \cup S) = P_{i_1, i_2, \dots, i_m}(R) \cup P_{i_1, i_2, \dots, i_m}(S)$ 。
14. 给出一个例子证明：如果 R 和 S 是两个 n 元关系，那么 $P_{i_1, i_2, \dots, i_m}(R - S)$ 可能与 $P_{i_1, i_2, \dots, i_m}(R) - P_{i_1, i_2, \dots, i_m}(S)$ 不同。
15. a) 与下述用 SQL 语句表示的查询相对应的运算是什么？
- ```
SELECT Supplier, Project
FROM Part_needs, part_inventory
WHERE Quantity <= 10
```

(选择供货商，项目)  
(从零件需求，零件库存)  
(当数量 ≤ 10)
- b) 给定表 5-9 和表 5-10 的数据库作为输入，这个查询的输出是什么？
16. 试确定例 3 中的关系是否有一个主关键字。

5.3 关系的表示

5.3.1 引言

在本节及本章的剩余部分要研究的所有关系均为二元关系。因此，在这些内容中出现的“关系”一词都表示二元关系。有多种方式表示有穷集之间的关系。正如已经看到的，一种方式是列出它的有序对；另一种方式是使用表，如 5.1 节所示。本节将讨论另外两种表示关系的方式：一种方式是使用 0-1 矩阵；另一种方式是使用称为有向图的图形表达方式，这种方法将在本节偏后一些进行讨论。

一般说来，矩阵适用于计算机程序中关系的表示。另一方面，人们常常发现使用有向图来表示关系对理解这些关系的性质是很有用的。

5.3.2 用矩阵表示关系

可以用 0-1 矩阵来表示一个有穷集之间的关系。假设  $R$  是从  $A = \{a_1, a_2, \dots, a_m\}$  到  $B = \{b_1, b_2, \dots, b_n\}$  的关系。(这里集合  $A$  和集合  $B$  的元素已经按照某一特定的但是任意的次序列出。此外，当  $A=B$  时我们对于  $A$  和  $B$  使用同样的排序。)关系  $R$  可以用矩阵  $M_R = [m_{ij}]$  来表示，其中

$$m_{ij} = \begin{cases} 1 & \text{如果 } (a_i, b_j) \in R \\ 0 & \text{如果 } (a_i, b_j) \notin R \end{cases}$$

换句话说，当  $a_i$  和  $b_j$  有关系时表示  $R$  的 0-1 矩阵的  $(i, j)$  项是 1，当  $a_i$  和  $b_j$  没关系时在这个位置是 0(这种表示依赖于对  $A$  和  $B$  使用的排序)。

下面的例子说明用矩阵来表示关系。

例 1 假设  $A = \{1, 2, 3\}$ ， $B = \{1, 2\}$ 。令  $R$  是从  $A$  到  $B$  的关系，且如果  $a \in A$ ， $b \in B$ ， $a > b$ ，则  $R$  包含  $(a, b)$ 。如果  $a_1 = 1$ ， $a_2 = 2$ ， $a_3 = 3$ ， $b_1 = 1$ ， $b_2 = 2$ ，表示  $R$  的矩阵是什么？

解 因为  $R = \{(2, 1), (3, 1), (3, 2)\}$ ，关于  $R$  的矩阵是

$$\mathbf{M}_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$\mathbf{M}_R$  中的 1 说明了有序对  $(2, 1)$ ,  $(3, 1)$  和  $(3, 2)$  属于  $R$ , 0 说明了没有其他的有序对属于  $R$ 。 ■

**例 2** 设  $A = \{a_1, a_2, a_3\}$ ,  $B = \{b_1, b_2, b_3, b_4, b_5\}$ 。哪些有序对在下面的矩阵所表示的关系  $R$  中?

$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

**解** 因为  $R$  由使得  $m_{ij}=1$  的有序对  $(a_i, b_j)$  构成, 因而

$$R = \{(a_1, b_2), (a_2, b_1), (a_2, b_3), (a_2, b_4), (a_3, b_1), (a_3, b_3), (a_3, b_5)\}$$

集合上的关系的矩阵是一个方阵, 可以用这个矩阵确定关系是否有某种性质。  $A$  上的关系是自反的, 如果对于每个  $a \in A$  有  $(a, a) \in R$ 。于是,  $R$  是自反的, 当且仅当对  $i=1, 2, \dots, n$ ,  $(a_i, a_i) \in R$ 。也就有,  $R$  是自反的, 当且仅当对  $i=1, 2, \dots, n$ ,  $m_{ii}=1$ 。换句话说, 如果如图 5-3 所示,  $\mathbf{M}_R$  的主对角线的所有元素都等于 1, 那么  $R$  是自反的。

如果  $(a, b) \in R$  推出  $(b, a) \in R$ , 那么关系  $R$  是对称的。因此集合  $A = \{a_1, a_2, \dots, a_n\}$  上的关系  $R$  是对称的, 当且仅当只要  $(a_i, a_j) \in R$  就有  $(a_j, a_i) \in R$ 。用矩阵  $\mathbf{M}_R$  的术语来说,  $R$  是对称的当且仅当只要  $m_{ij}=1$  就有  $m_{ji}=1$ 。这也意味着只要  $m_{ij}=0$  就有  $m_{ji}=0$ 。因此  $R$  是对称的, 当且仅当对所有的整数对  $i, j$  (其中  $i=1, 2, \dots, n, j=1, 2, \dots, n$ ) 都有  $m_{ij}=m_{ji}$ 。我们看到  $R$  是对称的当且仅当

$$\mathbf{M}_R = (\mathbf{M}_R)'$$

即  $\mathbf{M}_R$  是对称矩阵。对称关系的矩阵形式如图 5-4a 所示。

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

图 5-3 关于自反关系的 0-1 矩阵

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

a) 对称的

b) 反对称的

图 5-4 关于对称和反对称关系的 0-1 矩阵

关系  $R$  是反对称的, 当且仅当  $(a, b) \in R$  和  $(b, a) \in R$  推出  $a=b$ 。因此, 反对称关系的矩阵有下述性质, 即如果  $m_{ij}=1, i \neq j$ , 则  $m_{ji}=0$ 。或者换句话说, 当  $i \neq j$  时,  $m_{ij}=0$  或  $m_{ji}=0$ 。反对称关系的矩阵形式显示在图 5-4b 中。

**例 3** 假设集合上的关系  $R$  由矩阵

$$\mathbf{M}_R = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

表示,  $R$  是自反的、对称的和反对称的吗?

**解** 因为这个矩阵所有的对角线元素都等于 1,  $R$  是自反的。又由于  $\mathbf{M}_R$  是对称的, 因此  $R$  是对称的。也容易看出  $R$  不是反对称的。 ■

可以用布尔运算加和乘找到两个关系的并和交的矩阵表示。假设集合  $A$  上的关系  $R_1$  和  $R_2$  分别由矩阵  $\mathbf{M}_{R_1}$  和  $\mathbf{M}_{R_2}$  来表示。  $\mathbf{M}_{R_1}$  或  $\mathbf{M}_{R_2}$  在某个位置为 1, 则关系的并的矩阵相应位置有 1。  $\mathbf{M}_{R_1}$  和  $\mathbf{M}_{R_2}$  在某个位置同时为 1, 则关系的交的矩阵相应位置有 1。于是, 这些关系的并和交的矩阵表示是

$$\mathbf{M}_{R_1 \cup R_2} = \mathbf{M}_{R_1} \vee \mathbf{M}_{R_2}, \quad \mathbf{M}_{R_1 \cap R_2} = \mathbf{M}_{R_1} \wedge \mathbf{M}_{R_2}$$

**例 4** 假设集合  $A$  上的关系  $R_1$  和  $R_2$  由下述矩阵表示,  $R_1 \cup R_2$  和  $R_1 \cap R_2$  的矩阵表示是什么?

$$\mathbf{M}_{R_1} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{M}_{R_2} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

**解** 这两个关系的矩阵是

$$\begin{aligned} \mathbf{M}_{R_1 \cup R_2} &= \mathbf{M}_{R_1} \vee \mathbf{M}_{R_2} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \\ \mathbf{M}_{R_1 \cap R_2} &= \mathbf{M}_{R_1} \wedge \mathbf{M}_{R_2} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \blacksquare \end{aligned}$$

现在我们来考虑怎样确定关系合成的矩阵。这个矩阵可以通过关系矩阵的布尔积得到。特别地, 假设  $R$  是从  $A$  到  $B$  的关系且  $S$  是从  $B$  到  $C$  的关系。又假设  $A$ 、 $B$  和  $C$  分别有  $m$ 、 $n$  和  $p$  个元素。令关于  $S \circ R$ ,  $R$  和  $S$  的 0-1 矩阵分别为  $\mathbf{M}_{S \circ R} = [t_{ij}]$ ,  $\mathbf{M}_R = [r_{ij}]$ ,  $\mathbf{M}_S = [s_{ij}]$  (这些矩阵的大小分别为  $m \times p$ ,  $m \times n$  和  $n \times p$ ) 有序对  $(a_i, c_j)$  属于  $S \circ R$  当且仅当存在元素  $b_k$  使得  $(a_i, b_k)$  属于  $R$  并且  $(b_k, c_j)$  属于  $S$ 。从而  $t_{ij} = 1$ , 当且仅当对某个  $k$  有  $r_{ik} = s_{kj} = 1$ 。由布尔积的定义, 这意味着

$$\mathbf{M}_{S \circ R} = \mathbf{M}_R \odot \mathbf{M}_S$$

**例 5** 找出关系  $S \circ R$  的矩阵, 其中表示  $R$  和  $S$  的矩阵是

$$\mathbf{M}_R = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{M}_S = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

**解** 关于  $S \circ R$  的矩阵是

$$\mathbf{M}_{S \circ R} = \mathbf{M}_R \odot \mathbf{M}_S = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \blacksquare$$

表示两个关系合成的矩阵可以用来求  $\mathbf{M}_{R^n}$  的矩阵, 特别地, 由布尔幂的定义有

$$\mathbf{M}_{R^n} = \mathbf{M}_R^{[n]}$$

节末的练习 16 要求证明这个公式。

**例 6** 求关系  $R^2$  的矩阵表示, 其中  $R$  的矩阵表示是

$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

**解**  $R^2$  的矩阵是

$$\mathbf{M}_{R^2} = \mathbf{M}_R^{[2]} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \blacksquare$$

### 5.3.3 用图表示关系

我们已经显示了一个关系可以通过列出它所有的有序对或使用 0-1 矩阵来表示。还有一种重要的表示关系的方法就是图。把集合的每个元素表示成一个点, 每个有序对表示成一条弧, 弧上的箭头标明了弧的方向。当我们把一个有穷集上的关系看做一个有向图时, 就可以使用这种图形表示。

**定义 1** 一个有向图由顶点(或结点)集  $V$  和边(或弧)集  $E$  组成, 其中边集是  $V$  中元素的有序对的集合。顶点  $a$  叫做边  $(a, b)$  的始点, 而顶点  $b$  叫做这条边的终点。

形如  $(a, a)$  的边用一条从顶点  $a$  到自身的弧表示。这种边叫做环。



**例7** 具有顶点  $a, b, c$  和  $d$ , 边  $(a, b), (a, d), (b, b), (b, d), (c, a), (c, b)$  和  $(d, b)$  的有向图给在图 5-5 中。

集合  $A$  上的关系  $R$  表示成一个有向图, 这个图以  $A$  的元素作为顶点, 以有序对  $(a, b)$  作为边, 其中  $(a, b) \in R$ 。这就在集合  $A$  上的关系和以  $A$  作为顶点集的有向图之间构成了一一对应。于是, 每一个关于关系的论述对应着一个关于有向图的论述, 反之亦然。有向图给出了一个关于关系信息的可见显示。因此, 也常常用图研究关系及其性质。(注意从集合  $A$  到集合  $B$  的关系可以用一个有向的图示来表示, 其中对  $A$  中的每个元素存在一个顶点,  $B$  中的每个元素存在一个顶点, 如 5.1 节所示。然而, 当  $A=B$  时, 那种图示表示与这里描述的有向图表示不同, 它对关系信息的显示要少得多。)例 8~例 10 说明了怎样用有向图来表示关系。

**例8** 集合  $\{1, 2, 3, 4\}$  上的关系

$$R = \{(1, 1), (1, 3), (2, 1), (2, 3), (2, 4), (3, 1), (3, 2), (4, 1)\}$$

的有向图显示在图 5-6 中。

**例9** 图 5-7 中的有向图所表示的关系  $R$  中的有序对是什么?

**解** 关系中的有序对  $(x, y)$  是

$$R = \{(1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (3, 1), (3, 3), (4, 1), (4, 3)\}$$

每个有序对对应了有向图的一条边,  $(2, 2)$  和  $(3, 3)$  对应了环。

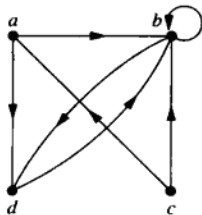


图 5-5 一个有向图

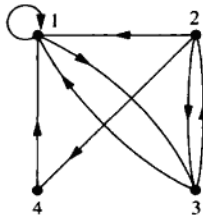


图 5-6 关系  $R$  的有向图

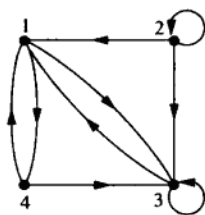


图 5-7 关系  $R$  的有向图

表示关系的有向图可以用来确定关系是否有各种性质。例如, 一个关系是自反的, 当且仅当有向图的每个顶点都有环, 从而每个形如  $(x, x)$  的有序对都出现在关系中。一个关系是对称的, 当且仅当对有向图不同顶点之间的每一条边都存在一条方向相反的边, 从而只要  $(x, y)$  在关系中就有  $(y, x)$  在关系中。类似地, 一个关系是反对称的, 当且仅当在不同的两个顶点之间不存在两条方向相反的边。最后, 一个关系是传递的, 当且仅当只要存在一条从顶点  $x$  到顶点  $y$  的边和一条顶点  $y$  到顶点  $z$  的边, 就有一条从顶点  $x$  到顶点  $z$  的边(完成一个三角形, 其中每条边都是具有正确方向的有向边)。

**注意** 对称关系可以用无向图表示, 这个图中的边没有方向。我们将在第 6 章研究无向图。

**例10** 对于图 5-8 所示的有向图的关系, 确定它们是否为自反的、对称的、反对称的和传递的。

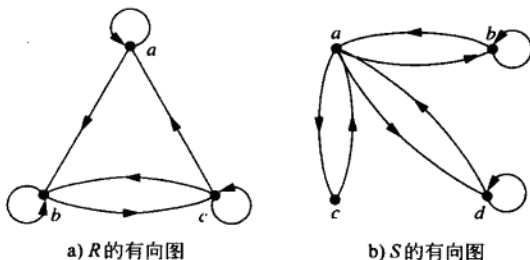


图 5-8 关系  $R$  和  $S$  的有向图

**解** 因为  $R$  的有向图的每个顶点都有环, 因此它是自反的。 $R$  既不是对称的也不是反对称的, 因为存在一条从  $a$  到  $b$  的边, 但没有从  $b$  到  $a$  的边, 并且在连接  $b$  和  $c$  的两个方向都有边。最后,  $R$  不是传递的, 因为从  $a$  到  $b$  有边, 从  $b$  到  $c$  有边, 但是从  $a$  到  $c$  没有边。

因为在有向图  $S$  的所有顶点没出现环, 因此关系不是自反的。因为不同顶点之间的每条边都伴随一条方向相反的边, 因此它是对称的但不是反对称的。不难看出  $S$  不是传递的, 因为  $(c, a)$  和  $(a, b)$  属于  $S$  但  $(c, b)$  不属于  $S$ 。 ■

### 练习

1. 用矩阵表示下面每个  $\{1, 2, 3\}$  上的关系(按增序列出这个集合的元素)。

a)  $\{(1, 1), (1, 2), (1, 3)\}$

b)  $\{(1, 2), (2, 1), (2, 2), (3, 3)\}$

c)  $\{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}$

d)  $\{(1, 3), (3, 1)\}$

2. 列出和下面矩阵对应的  $\{1, 2, 3\}$  上的关系中的有序对(其中行和列对应于按增序列出的整数)。

a)  $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

b)  $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

c)  $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

3. 怎样用表示集合  $A$  上关系  $R$  的有向图确定这个关系是否是反自反的?

4. 确定练习 2 中的矩阵所表示的关系是否为自反的、反自反的、对称的、反对称的和传递的。

5.  $R$  是包含了前 100 个正整数的集合  $A = \{1, 2, \dots, 100\}$  上的关系, 如果  $R$  满足下述条件, 那么表示  $R$  的矩阵有多少个非 0 的元素?

a)  $\{(a, b) \mid a > b\}$

b)  $\{(a, b) \mid a \neq b\}$

c)  $\{(a, b) \mid a = b + 1\}$

d)  $\{(a, b) \mid a = 1\}$

e)  $\{(a, b) \mid ab = 1\}$

6. 当  $R$  是有穷集  $A$  上的关系时, 怎样从表示  $R$  的关系矩阵找到这个关系的补  $\bar{R}$  的矩阵?

7. 设  $R$  是矩阵

$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

所表示的关系, 求表示下述关系的矩阵。

a)  $R^{-1}$

b)  $\bar{R}$

c)  $R^2$

8. 设  $R$  是矩阵

$$\mathbf{M}_R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$
 表示的关系, 求表示下述关系的矩阵

a)  $R^2$

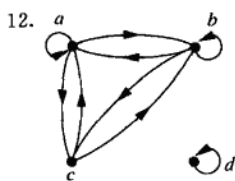
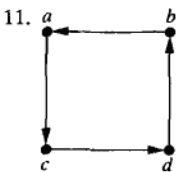
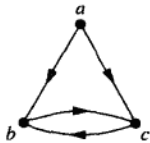
b)  $R^3$

c)  $R^4$

9. 设  $R$  是  $n$  元素集合上的关系。如果在表示  $R$  的矩阵  $\mathbf{M}_R$  中存在  $k$  个非 0 的元素, 那么在表示  $R$  的补  $\bar{R}$  的矩阵  $\mathbf{M}_{\bar{R}}$  中存在多少个非 0 元素?

在练习 10~12 列出由下述有向图所表示的关系中的有序对。

10.



13. 怎样用有穷集  $A$  上关系  $R$  的有向图确定关系是否是非对称的?

14. 确定练习 10~11 所示的有向图表示的关系是否为自反的、反自反的、对称的、反对称的和传递的。

15. 设  $R$  是集合  $A$  上的关系, 解释怎样用表示  $R$  的有向图得到表示关系的逆  $R^{-1}$  的有向图?

16. 证明: 如果  $\mathbf{M}_R$  是表示关系  $R$  的矩阵, 那么  $\mathbf{M}_R^{[n]}$  是表示关系  $R^n$  的矩阵。

## 5.4 关系的闭包

### 5.4.1 引言

一个计算机网络在波士顿、芝加哥、丹佛、底特律、纽约和圣迭戈设有数据中心。从波士顿到芝加哥，波士顿到底特律，芝加哥到底特律，底特律到丹佛和纽约到圣迭戈都有单向的电话线。如果存在一条从数据中心  $a$  到  $b$  的电话线， $(a, b)$  就属于关系  $R$ 。我们怎样确定从一个中心是否有一条电话线或多条电话线(可能不直接)链接到另一个中心？由于所有的链接不一定是直接的，例如从波士顿可通过底特律到丹佛，因此不能直接使用  $R$  来回答这个问题。用关系的语言说， $R$  不是传递的，因而它不包含可能被链接的所有的对。正如我们将在本节证明的，我们可以通过构造包含  $R$  的最小的传递关系来找出每一对有着链接的数据中心。这个关系叫做  $R$  的传递闭包。

一般说来，设  $R$  是集合  $A$  上的关系。 $R$  可能具有或者不具有某些性质  $P$ ，例如自反性、对称性或传递性。如果存在包含  $R$  的具有性质  $P$  的关系  $S$ ，并且  $S$  是包含  $R$  且具有性质  $P$  的每一个关系的子集，那么  $S$  叫做  $R$  的关于  $P$  的闭包。(注意一个关系关于一个性质的闭包可能不存在，见节末的练习 8 和 17。)我们将说明怎样找关系的自反闭包、对称闭包和传递闭包。

### 5.4.2 闭包

集合  $A = \{1, 2, 3\}$  上的关系  $R = \{(1, 1), (1, 2), (2, 1), (3, 2)\}$  不是自反的。我们怎样才能产生一个包含  $R$  的尽可能小的自反关系呢？这可以通过把  $(2, 2)$  和  $(3, 3)$  加到  $R$  中来做到，因为只有它们是不在  $R$  中的形如  $(a, a)$  的对。很明显，这个新关系包含了  $R$ 。此外，任何包含  $R$  的自反关系必须包含  $(2, 2)$  和  $(3, 3)$ 。因为这个关系包含了  $R$ ，是自反的，并且被包含在每一个包含  $R$  的自反关系之中，因此它就是  $R$  的自反闭包。

正如这个例子所显示的，给定集合  $A$  上的关系  $R$ ，对于  $a \in A$ ，可以通过把形如  $(a, a)$  的所有对，除了已在  $R$  中的之外，都加到  $R$  中，就构成了  $R$  的自反闭包。加入这些对产生了一个新的自反的、包含  $R$  的关系，并且它被包含在任何包含  $R$  的自反关系之中。我们看到  $R$  的自反闭包等于  $R \cup \Delta$ ，其中  $\Delta = \{(a, a) \mid a \in A\}$  是  $A$  上的对角关系。(读者应能验证这个结果。)

**例 1** 整数集上的关系  $R = \{(a, b) \mid a < b\}$  的自反闭包是什么？

**解**  $R$  的自反闭包是

$$R \cup \Delta = \{(a, b) \mid a < b\} \cup \{(a, a) \mid a \in \mathbb{Z}\} = \{(a, b) \mid a \leq b\}$$

$\{1, 2, 3\}$  上的关系  $\{(1, 1), (1, 2), (2, 2), (2, 3), (3, 1), (3, 2)\}$  不是对称的。我们怎样产生一个包含  $R$  的尽可能小的对称关系？只需增加  $(2, 1)$  和  $(1, 3)$ ，因为只有它们是具有  $(a, b) \in R$  而  $(b, a)$  不在  $R$  中的那种  $(b, a)$  对。这个新关系是对称的，且包含了  $R$ 。此外，任何包含了  $R$  的对称关系一定包含这个新关系，因为一个包含  $R$  的对称关系必须包含  $(2, 1)$  和  $(1, 3)$ 。因此这个新关系叫做  $R$  的对称闭包。

正如这个例子所显示的，关系  $R$  的对称闭包可以通过增加所有形如  $(b, a)$  的有序对构成，其中  $(a, b)$  在关系中而  $(b, a)$  不在关系中。增加这些有序对产生一个关系，它是对称的，包含了  $R$ ，并且它被包含在任何包含  $R$  的对称关系之中。关系  $R$  的对称闭包可以通过取关系与它的逆的并来构造，即  $R \cup R^{-1}$  是  $R$  的对称闭包，其中  $R^{-1} = \{(b, a) \mid (a, b) \in R\}$ 。读者应能验证这个结果。

**例 2** 正整数集上的关系  $R = \{(a, b) \mid a > b\}$  的对称闭包是什么？

**解**  $R$  的对称闭包是关系

$$R \cup R^{-1} = \{(a, b) \mid a > b\} \cup \{(b, a) \mid a > b\} = \{(a, b) \mid a \neq b\}$$

**Example** 这最后一步相等是由于  $R$  包含了所有正整数构成的有序对，其中第一元素大于第二元素，并且  $R^{-1}$  包含了所有正整数构成的有序对，其中第一元素小于第二元素。

假设关系  $R$  不是传递的，我们怎样产生一个包含  $R$  的传递关系并使得这个新的关系被包含在

任何包含  $R$  的传递关系之中? 对于已经在  $R$  中的  $(a, b)$  和  $(b, c)$ , 可以通过增加所有形如  $(a, c)$  的有序对构成  $R$  的传递闭包吗? 考虑集合  $\{1, 2, 3, 4\}$  上的关系  $R = \{(1, 3), (1, 4), (2, 1), (3, 2)\}$ 。这个关系不是传递的, 因为对于在  $R$  中的  $(a, b)$  和  $(b, c)$ , 它不包含所有形如  $(a, c)$  的有序对。这种不在  $R$  中的有序对是  $(1, 2)$ ,  $(2, 3)$ ,  $(2, 4)$  和  $(3, 1)$ 。把这些有序对加到  $R$  中并不能产生一个传递关系, 因为所得的结果关系包含  $(3, 1)$  和  $(1, 4)$ , 但不包含  $(3, 4)$ 。这说明构造关系的传递闭包比起构造它们的自反闭包或对称闭包更复杂。下面我们将建立一个构造传递闭包的算法。正如将要显示的, 一个关系的传递闭包可以通过增加那些必须出现的有序对来得到, 要重复这个过程, 直到没有必须增加的有序对为止。

### 5.4.3 有向图的路径

我们将看到用有向图表示关系有助于构造关系的传递闭包。为此现在引入某些将要用到的术语。

通过沿有向图的边(按照这条边的箭头指示的相同方向)移经有向图就得到一条有向图中的路径。

**定义 1** 在有向图  $G$  中从  $a$  到  $b$  的一条路径是  $G$  中一条或多条边的序列  $(x_0, x_1), (x_1, x_2), (x_2, x_3), \dots, (x_{n-1}, x_n)$ , 其中  $x_0 = a, x_n = b$ 。即一个边的序列, 其中一条边的终点和路径中下一条边的始点相同。这条路径记为  $x_0, x_1, \dots, x_{n-1}, x_n$ , 长度为  $n$ 。在同一顶点开始和结束的路径叫做回路或圈。

有向图的一条路径可以多次通过一个顶点。此外, 有向图的一条边也可以多次出现在一条路径中。

**例 3** 下面哪些路径是图 5-9 的有向图中的路径:  $a, b, e, d$ ;  $a, e, c, d, b$ ;  $b, a, c, b, a, a, b$ ;  $d, c$ ;  $c, b, a$ ;  $e, b, a, b, a, b, e$ ? 这些路径的长度是多少? 这个列表中的哪些路径是回路?

**解** 因为  $(a, b)$ 、 $(b, e)$  和  $(e, d)$  都是边,  $a, b, e, d$  是长为 3 的路径。因为  $(c, d)$  不是边,  $a, e, c, d, b$  不是路径。因为  $(b, a)$ ,  $(a, c)$ ,  $(c, b)$ ,  $(b, a)$ ,  $(a, a)$  和  $(a, b)$  都是边,  $b, a, c, b, a, a, b$  是长为 6 的路径。我们也看到, 因为  $(d, c)$  是边,  $d, c$  是长为 1 的路径。还有由于  $(c, b)$  和  $(b, a)$  是边,  $c, b, a$  是长为 2 的路径。 $(e, b)$ ,  $(b, a)$ ,  $(a, b)$ ,  $(b, a)$ ,  $(a, b)$ ,  $(b, e)$  都是边, 因此  $e, b, a, b, a, b, e$  是长为 6 的路径。

两条路径  $b, a, c, b, a, a, b$  和  $e, b, a, b, a, b, e$  是回路, 因为它们在同一顶点开始和结束。路径  $a, b, e, d$ ;  $c, b, a$  和  $d, c$  不是回路。

术语路径也用于关系。把有向图的定义推广到关系可知, 如果存在一个元素的序列  $a, x_1, x_2, \dots, x_{n-1}, b$  具有  $(a, x_1) \in R, (x_1, x_2) \in R, \dots, (x_{n-1}, b) \in R$ , 那么在  $R$  中存在一条从  $a$  到  $b$  的路径。从关系中的路径定义可以得到定理 1。

**定理 1** 设  $R$  是集合  $A$  上的关系。从  $a$  到  $b$  存在一条长为  $n$  ( $n$  为正整数) 的路径, 当且仅当  $(a, b) \in R^n$ 。

**证** 我们将使用数学归纳法。根据定义, 从  $a$  到  $b$  存在一条长为 1 的路径, 当且仅当  $(a, b) \in R$ 。因此当  $n=1$  时定理为真。

假定对于正整数  $n$  定理为真, 这是归纳假设。从  $a$  到  $b$  存在一条长为  $n+1$  的路径, 当且仅当存在元素  $c \in A$  使得从  $a$  到  $c$  存在一条长为 1 的路径, 即  $(a, c) \in R$ , 以及一条从  $c$  到  $b$  的长为  $n$  的路径, 即  $(c, b) \in R^n$ 。因此, 由归纳假设, 从  $a$  到  $b$  存在一条长为  $n+1$  的路径, 当且仅当存在一个元素  $c$ , 使得  $(a, c) \in R$  和  $(c, b) \in R^n$ 。但是存在这样一个元素, 当且仅当  $(a, b) \in R^{n+1}$ 。因此, 从  $a$  到  $b$  存在一条长为  $n+1$  的路径, 当且仅当  $(a, b) \in R^{n+1}$ 。定理得证。  $\square$

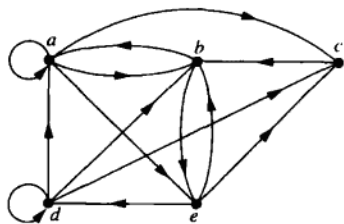


图 5-9 一个有向图

#### 5.4.4 传递闭包

我们现在证明找一个关系的传递闭包与在相关的有向图中确定哪些顶点对被路径连接是等价的。由此要定义一个新的关系。

**定义 2** 设  $R$  是集合  $A$  上的关系。连通性关系  $R^*$  由对  $(a, b)$  构成, 使得在  $R$  中从顶点  $a$  到  $b$  之间存在一条至少长为 1 的路径。

因为  $R^n$  由对  $(a, b)$  构成, 使得存在一条从  $a$  到  $b$  的长为  $n$  的路径, 从而  $R^*$  是所有集合  $R^n$  的并。换句话说,

$$R^* = \bigcup_{n=1}^{\infty} R^n$$

许多模型都用到连通性关系。

**例 4** 设  $R$  是世界上所有人的集合上的关系, 如果  $a$  认识  $b$ , 那么  $R$  包含  $(a, b)$ 。 $R^n$  是什么? 其中  $n$  是大于 2 的正整数。 $R^*$  是什么?

**解** 如果存在  $c$  使得  $(a, c) \in R$  且  $(c, b) \in R$ , 即存在  $c$  使得  $a$  认识  $c$ ,  $c$  认识  $b$ , 那么关系  $R^2$  包含  $(a, b)$ 。类似地, 如存在  $x_1, x_2, \dots, x_{n-1}$  使得  $a$  认识  $x_1$ ,  $x_1$  认识  $x_2$ ,  $\dots$ ,  $x_{n-1}$  认识  $b$ , 那么  $R^n$  包含对  $(a, b)$ 。

如果存在从  $a$  开始至  $b$  终止的序列, 使得序列中的每个人都认识序列中的下一个人, 那么  $R^*$  包含对  $(a, b)$ 。(关于  $R^*$  存在许多有趣的猜想。你认为这个连通关系包含以你作为第一元素, 蒙古的总统作为第二元素的一对元素吗? 在第 6 章我们将用图建立这个应用的模型。)

**例 5** 设  $R$  是纽约市所有地铁站集合的关系。如果可以从  $a$  站不换乘就旅行到  $b$  站, 那么  $R$  包含对  $(a, b)$ 。当  $n$  是正整数时,  $R^n$  是什么?  $R^*$  是什么?

**解** 如果换  $n-1$  次车就可以从  $a$  站旅行到  $b$  站, 关系  $R^n$  就包含  $(a, b)$ 。从  $a$  站旅行到  $b$  站, 如果需要可以换乘任意多次, 关系  $R^*$  就由这种有序对  $(a, b)$  组成。(读者应该能验证这些论断。)

**例 6** 设  $R$  是美国所有的州的集合上的关系, 如果  $a$  州和  $b$  州有公共的边界, 那么  $R$  包含  $(a, b)$ 。 $R^n$  是什么? 其中  $n$  是正整数。 $R^*$  又是什么?

**解** 关系  $R^n$  由对  $(a, b)$  构成, 可以从  $a$  州恰好跨越  $n$  次州界到达  $b$  州。 $R^*$  由对  $(a, b)$  构成, 可以从  $a$  州跨越任意多次边界到达  $b$  州(读者应该能够验证这些论断。)那些包含与美国大陆不相连的州(即含有阿拉斯加或夏威夷)的有序对是不在  $R^*$  中的。

定理 2 证明一个关系的传递闭包和相关的连通性关系是等同的。

**定理 2** 关系  $R$  的传递闭包等于连通性关系  $R^*$ 。

**证** 注意  $R^*$  包含  $R$ 。为证明  $R^*$  是  $R$  的传递闭包, 我们必须证明  $R^*$  是传递的且对一切包含  $R$  的传递关系  $S$ , 有  $R^* \subseteq S$ 。

首先, 我们证明  $R^*$  是传递的。如果  $(a, b) \in R^*$  和  $(b, c) \in R^*$ , 那么在  $R$  中存在从  $a$  到  $b$  和从  $b$  到  $c$  的路径。我们从  $a$  到  $b$  的路径开始, 并且沿着从  $b$  到  $c$  的路径就得到一条从  $a$  到  $c$  的路径。因此,  $(a, c) \in R^*$ 。这就得出  $R^*$  是传递的。

现在假设  $S$  是包含  $R$  的传递关系, 因为  $S$  是传递的,  $S^n$  也是传递的(读者应该能验证这一点), 并且  $S^n \subseteq S$ (由 5.1 节定理 1)。此外, 因为

$$S^* = \bigcup_{k=1}^{\infty} S^k$$

和  $S^k \subseteq S$ , 因此  $S^* \subseteq S$ 。现在注意到如果  $R \subseteq S$ , 那么  $R^* \subseteq S^*$ , 这是由于任何  $R$  中的路径也是  $S$  中的路径。从而  $R^* \subseteq S^* \subseteq S$ 。于是, 任何包含  $R$  的传递关系也一定包含  $R^*$ 。因此,  $R^*$  是  $R$  的传递闭包。□

既然我们知道传递闭包等于连通性关系, 我们考虑这个关系的计算问题。在一个有限的有向图中不需要检测任意长的路径来确定是否在两个顶点之间存在一条路径。正如下面的引理所证明的。检测包含不超过  $n$  条边的路径就足够了, 这里  $n$  是集合中的元素个数。

**引理 1** 设  $A$  是  $n$  元素集合,  $R$  是  $A$  上的关系。如果  $R$  中存在一条从  $a$  到  $b$  的长至少为 1 的

路径, 那么存在一条长度不超过  $n$  的这种路径。此外, 当  $a \neq b$  时, 如果在  $R$  中存在一条从  $a$  到  $b$  的路径, 那么存在一条长度不超过  $n-1$  的这种路径。

**证** 假设  $R$  中存在一条从  $a$  到  $b$  的路径。令  $m$  是这种路径的最短长度。假设  $x_0, x_1, x_2, \dots, x_{m-1}, x_m$  是一条这样的路径, 其中  $x_0 = a, x_m = b$ 。

假设  $a = b$  和  $m > n$ , 使得  $m \geq n+1$ 。由鸽巢原理, 因为  $A$  中有  $n$  个顶点, 在  $m$  个顶点  $x_0, x_1, x_2, \dots, x_{m-1}$  中至少有两个是相同的(见图 5-10)。

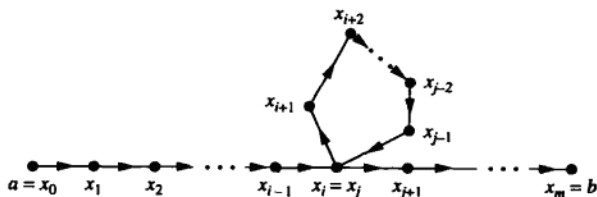


图 5-10 产生一条长度不超过  $n$  的路径

假设  $x_i = x_j$  满足  $0 \leq i < j \leq m-1$ 。那么这条路径包含一条从  $x_i$  到  $x_i$  自身的回路。可以把这条回路从由  $a$  到  $b$  的路径中删除, 剩下的路径, 即  $x_0, x_1, \dots, x_i, x_{j+1}, \dots, x_{m-1}, x_m$  是从  $a$  到  $b$  的更短的路径。因此, 最短长度的这种路径的长度一定小于等于  $n$ 。

$a \neq b$  的情况留给读者作为练习。 □

由引理 1, 我们看出  $R$  的传递闭包是  $R, R^2, R^3, \dots, R^n$  的并。这是由于在  $R^*$  的两个顶点之间存在一条路径, 当且仅当对某个正整数  $i (i \leq n)$  在  $R^i$  的这些顶点之间存在一条路径。因为

$$R^* = R \cup R^2 \cup R^3 \cup \dots \cup R^n$$

并且表示关系的并的 0-1 矩阵是这些关系的 0-1 矩阵的联合, 因此传递闭包的 0-1 矩阵是  $R$  的 0-1 矩阵的前  $n$  次幂的 0-1 矩阵的联合。

**定理 3** 设  $M_R$  是  $n$  元素集合上的关系  $R$  的 0-1 矩阵。那么传递闭包  $R^*$  的 0-1 矩阵是

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]} \vee \dots \vee M_R^{[n]}$$

**例 7** 求关系  $R$  的传递闭包的 0-1 矩阵, 其中

$$M_R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

**解** 由定理 3,  $R^*$  的 0-1 矩阵是

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]}$$

因为

$$M_R^{[2]} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad M_R^{[3]} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

所以

$$M_{R^*} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

定理 3 可以作为计算关系  $R^*$  的矩阵的算法基础。为求出这个矩阵, 要连续计算  $M_R$  的布尔幂, 直到第  $n$  次幂为止。当计算每次幂时就构成这个幂与所有较小的幂的联合。当做到第  $n$  次幂时, 就得到关于  $R^*$  的矩阵。这个过程见算法 1。

**算法 1** 计算传递闭包的过程

**procedure** *transitive closure* ( $M_R$ :  $n \times n$  的 0-1 矩阵)

$A := M_R$

$B := A$

**for**  $i := 2$  **to**  $n$

**begin**

$A := A \odot M_R$

$B := B \vee A$

**end** { $B$  是关于  $R'$  的 0-1 矩阵}

我们可以容易地求出用算法 1 确定关系的传递闭包所使用的位运算次数。计算布尔幂  $M_R$ ,  $M_R^{[2]}$ ,  $\dots$ ,  $M_R^{[n]}$  需要找到  $n-1$  个  $n \times n$  的 0-1 矩阵的布尔积。每个布尔积可以使用  $n^2(2n-1)$  次位运算求得。因此, 计算这些乘积使用  $n^2(2n-1)(n-1)$  次位运算。

为从  $n$  个  $M_R$  的布尔幂求  $M_{R'}$ , 需要求  $n-1$  个 0-1 矩阵的联合。计算每一个联合使用  $n^2$  次位运算。因此, 在这部分计算中使用  $(n-1)n^2$  次位运算。所以, 当使用算法 1 时, 用  $n^2(2n-1)(n-1) + (n-1)n^2 = 2n^3(n-1) = O(n^4)$  次位运算就可以求出  $n$  元素集合上关系的传递闭包的矩阵。下面我们将要描述一个更有效的求传递闭包的算法。

#### 5.4.5 沃舍尔算法

沃舍尔算法得名于史蒂芬·沃舍尔, 他在 1960 年给出该算法。这个算法能够有效计算关系的传递闭包。算法 1 求出  $n$  元素集合上关系的传递闭包使用  $2n^3(n-1)$  次位运算。而沃舍尔算法只使用  $2n^3$  次位运算就可以求出这个传递闭包。

**注意** 沃舍尔算法有时也叫做罗伊-沃舍尔算法, 因为伯纳德·罗伊(B·Roy)在 1959 年描述了这个算法。

假设  $R$  是  $n$  元素集合上的关系, 设  $v_1, v_2, \dots, v_n$  是这  $n$  个元素的任意排列。沃舍尔算法中用到一条路径的内点的概念。如果  $a, x_1, x_2, \dots, x_{m-1}, b$  是一条路径, 它的内点是  $x_1, x_2, \dots, x_{m-1}$ , 即除了第一和最后一个顶点之外出现在路径中的所有顶点。例如, 在有向图的一条路径  $a, c, d, f, g, h, b, j$  的内点是  $c, d, f, g, h, b$ 。 $a, c, d, a, f, b$  的内点是  $c, d, a, f$ 。(注意这条路径的起点不是内点, 除非这条路径再次访问它, 且不是作为终点来访问的。类似地, 这条路径的终点也不是内点, 除非它在这之前曾被这条路径访问过, 且不是作为起点来访问的。)

沃舍尔算法的基础是构造一系列 0-1 矩阵。这些矩阵是  $W_0, W_1, \dots, W_n$ , 其中  $W_0 = M_R$  是这个关系的 0-1 矩阵, 且  $W_k = [w_{ij}^{(k)}]$ 。如果存在一条从  $v_i$  到  $v_j$  的路径使得这条路径的所有内点都在集合  $\{v_1, v_2, \dots, v_k\}$  (表中的前  $k$  个顶点) 之中, 那么  $w_{ij}^{(k)} = 1$ , 否则为 0 (这条路径的起点和终点可能在表中的前  $k$  个顶点的集合之外)。注意  $W_n = M_{R'}$ , 因为  $M_{R'}$  的第  $(i, j)$  项是 1, 当且仅当存在一条从  $v_i$  到  $v_j$  的路径, 且全部内点都在集合  $\{v_1, v_2, \dots, v_n\}$  之中 (但这些就是有向图的所有顶点)。例 8 说明矩阵  $W_k$  表示什么。

**例 8** 设  $R$  是一个关系, 它的有向图如图 5-11 所示。设  $a, b, c, d$  是集合元素的排列。求矩阵  $W_0, W_1, W_2, W_3$  和  $W_4$ 。矩阵  $W_4$  是  $R$  的传递闭包。

**解** 令  $v_1 = a, v_2 = b, v_3 = c$ , 且  $v_4 = d$ 。 $W_0$  是这个关系的矩阵, 于是

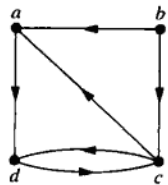


图 5-11 关系  $R$  的有向图

$$W_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

如果存在一条  $v_i$  到  $v_j$  的且只有  $v_1=a$  作为内点的路径,  $W_1$  的  $(i, j)$  项有 1。注意因为所有长为 1 的路径没有内点, 所以仍旧可以使用这些路径。此外存在一条从  $b$  到  $d$  的路径, 即  $b, a, d$ 。因此

$$W_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

如果存在一条  $v_i$  到  $v_j$  的且只有  $v_1=a$  和  $v_2=b$  作为内点的路径,  $W_2$  的  $(i, j)$  项有 1。因为没有边以  $b$  作为终点, 当我们允许  $b$  作为内点时不会得到新的路径。因此,  $W_2=W_1$ 。

若存在一条  $v_i$  到  $v_j$  的只用  $v_1=a, v_2=b$  和  $v_3=c$  作为内点的路径, 则  $W_3$  的  $(i, j)$  项有 1。我们现在有从  $d$  到  $a$  的路径, 即  $d, c, a$  和从  $d$  到  $d$  的路径  $d, c, d$ 。因此

$$W_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

最后, 如果存在一条从  $v_i$  到  $v_j$  的路径, 并且以  $v_1=a, v_2=b, v_3=c$  及  $v_4=d$  作为内点, 那么  $W_4$  的  $(i, j)$  项为 1。因为这是图的全部顶点, 此项为 1, 当且仅当存在一条从  $v_i$  到  $v_j$  的路径。因此

$$W_4 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

这个最后的矩阵就是传递闭包的矩阵。

沃舍尔算法通过有效地计算  $W_0=M_R, W_1, W_2, \dots, W_n=M_R$  来计算  $M_R$ 。不难看出可以直接从  $W_{k-1}$  计算  $W_k$ : 存在一条从  $v_i$  到  $v_j$  的只以  $v_1, v_2, \dots, v_k$  中的顶点作为内点的路径, 当且仅当要么存在一条从  $v_i$  到  $v_j$  的且内点是表中前  $k-1$  个顶点的路径, 要么存在从  $v_i$  到  $v_k$  的路径和从  $v_k$  到  $v_j$  的路径, 而这些路径的内点仅在表中的前  $k-1$  个顶点中。这就是说, 要么在  $v_k$  被允许作为内点之前从  $v_i$  到  $v_j$  已经存在一条路径, 要么允许  $v_k$  作为内点产生一条从  $v_i$  到  $v_k$  然后从  $v_k$  到  $v_j$  的路径。这两种情况如图 5-12 所示。

第一种类型的路径存在, 当且仅当  $w_{ij}^{[k-1]}=1$ , 且第二种类型的路径存在, 当且仅当  $w_{ik}^{[k-1]}=1$  和  $w_{kj}^{[k-1]}=1$ 。于是,  $w_{ij}^{[k]}$  是 1, 当且仅当或者  $w_{ij}^{[k-1]}=1$ , 或者  $w_{ik}^{[k-1]}=1$  和  $w_{kj}^{[k-1]}=1$ 。这就得到引理 2。

**引理 2** 设  $W_k=[w_{ij}^{[k]}]$  是 0-1 矩阵, 它在  $(i, j)$  位置有 1, 当且仅当存在一条从  $v_i$  到  $v_j$  的路径, 其内点取自集合  $\{v_1, v_2, \dots, v_k\}$ , 那么

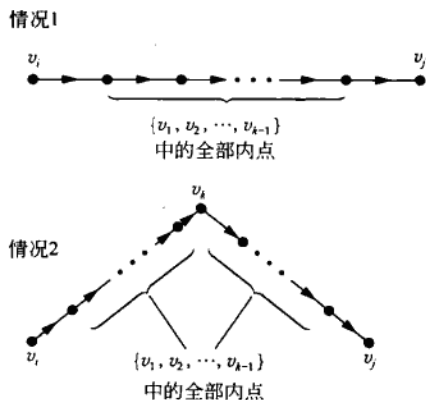


图 5-12 把  $v_k$  加到允许使用的内点集中



$$w_{ij}^{[k]} = w_{ij}^{[k-1]} \vee (w_{ik}^{[k-1]} \wedge w_{kj}^{[k-1]})$$

其中  $i, j$  和  $k$  是不超过  $n$  的正整数。

引理 2 提供了有效计算矩阵  $W_k (k=1, 2, \dots, n)$  的手段。我们使用引理 2 把沃舍尔算法的伪码在算法 2 中给出。

#### 算法 2 沃舍尔算法

**procedure** *Warshall* ( $M_R: n \times n$  的 0-1 矩阵)

$W := M_R$

**for**  $k := 1$  **to**  $n$

**begin**

**for**  $i := 1$  **to**  $n$

**begin**

**for**  $j := 1$  **to**  $n$

$w_{ij} := w_{ij} \vee (w_{ik} \wedge w_{kj})$

**end**

**end** {  $W = [w_{ij}]$  是  $M_R^*$  }

很容易以位运算次数计算沃舍尔算法的计算复杂度。使用引理 2, 从项  $w_{ij}^{[k-1]}$ ,  $w_{ik}^{[k-1]}$  和  $w_{kj}^{[k-1]}$  找到项  $w_{ij}^{[k]}$  需要 2 次位运算。为从  $W_{k-1}$  求出  $W_k$  的所有  $n^2$  个项需要  $2n^2$  次位运算。因为沃舍尔算法从  $W_0 = M_R$  开始, 计算  $n$  个 0-1 矩阵的序列  $W_1, W_2, \dots, W_n = M_R^*$ , 使用的位运算总次数是  $n \cdot 2n^2 = 2n^3$ 。

#### 练习

1. 设  $R$  是集合  $\{0, 1, 2, 3\}$  上的关系,  $R$  包含有序对  $(0, 1), (1, 1), (1, 2), (2, 0), (2, 2)$  和  $(3, 0)$ , 求
  - a)  $R$  的自反闭包
  - b)  $R$  的对称闭包
2. 设  $R$  是整数集上的关系  $\{(a, b) \mid a \text{ 整除 } b\}$ ,  $R$  的对称闭包是什么?

在练习 3~4 中画出给定有向图所表示关系的自反闭包的有向图。



5. 对于练习 3~4 的有向图表示的关系, 找出关系的对称闭包的有向图。
6. 对于练习 3~4 的每个有向图表示的关系, 求包含它的最小的自反对称关系的有向图。
7. 假设有穷集  $A$  上的关系  $R$  由矩阵  $M_R$  表示, 证明表示  $R$  的对称闭包的矩阵是  $M_R \vee M_R^T$ 。
8. 什么时候可能定义一个关系  $R$  的反自反闭包, 即一个包含  $R$  的关系是反自反的且被包含在每一个包含  $R$  的反自反关系之中?
9. 设  $R$  是集合  $\{1, 2, 3, 4, 5\}$  上的关系,  $R$  包含有序对  $(1, 3), (2, 4), (3, 1), (3, 5), (4, 3), (5, 1), (5, 2)$  和  $(5, 4)$ 。求
  - a)  $R^2$
  - b)  $R^3$
  - c)  $R^4$
  - d)  $R^5$
  - e)  $R^6$
  - f)  $R^*$
10. 设  $R$  是所有学生的集合上的关系, 如果  $a \neq b$  且  $a$  和  $b$  至少有一门公共课程, 则  $R$  包含了有序对  $(a, b)$ 。什么时候  $(a, b)$  在下面的关系中?
  - a)  $R^2$
  - b)  $R^3$
  - c)  $R^*$
11. 假设关系  $R$  是对称的, 证明  $R^*$  是对称的。

12. 使用算法 1 找出下面  $\{1, 2, 3, 4\}$  上的关系的传递闭包。
- a)  $\{(1, 2), (2, 1), (2, 3), (3, 4), (4, 1)\}$   
 b)  $\{(2, 1), (2, 3), (3, 1), (3, 4), (4, 1), (4, 3)\}$   
 c)  $\{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$   
 d)  $\{(1, 1), (1, 4), (2, 1), (2, 3), (3, 1), (3, 2), (3, 4), (4, 2)\}$
13. 使用沃舍尔算法找出练习 12 中关系的传递闭包。
14. 求出包含关系  $\{(1, 2), (1, 4), (3, 3), (4, 1)\}$  的最小的关系, 使得它是
- a) 自反的和传递的。      b) 对称的和传递的。      c) 自反的、对称的和传递的。
15. 已经设计出算法用  $O(n^{2.8})$  次位运算来计算两个  $n \times n$  的 0-1 矩阵的布尔积。假设可以使用这些算法, 给出用算法 1 和沃舍尔算法求  $n$  元素集合上关系的传递闭包所用位运算数的大  $O$  估计。
16. 修改算法 1 找出  $n$  元素集合上关系的传递闭包的自反闭包。
17. 证明: 集合  $\{0, 1, 2\}$  上的关系  $R = \{(0, 0), (0, 1), (1, 1), (2, 2)\}$  关于下述性质  $P$  的闭包不存在, 如果  $P$
- a) “不是自反的”。      b) “有奇数个元素”。

## 5.5 等价关系基础

### 5.5.1 引言

在一些程序设计语言中, 变量的命名可以包含无数字符。当编译器确定要检查两个变量是否相同时, 字符的数量就有限制。例如, 在传统 C 中, 编译器只检查变量名的前八个字符。(这些字符是大写或小写字母、数字或下划线。) 所以编译器对于比八个字符长的串就考虑它的前八个字符相同。设  $R$  是字符串集合上的关系,  $s$  和  $t$  是两个字符串, 如果  $s$  和  $t$  至少有八个字符长, 而  $s$  和  $t$  的前八个字符相符, 或者  $s=t$ , 则  $sRt$ 。容易得出  $R$  是自反的、对称的和传递的。而且  $R$  把所有字符串的集合分成一些类, 传统 C 的编译器认为在某一特定类中的所有字符串是相同的。

如果 4 整除  $a-b$ , 整数  $a$  和  $b$  有模 4 同余的关系。后面我们将证明这个关系是自反的、对称的和传递的。不难看出  $a$  和  $b$  相关, 当且仅当被 4 整除时,  $a$  和  $b$  有相同的余数。这个关系将整数集划分成不同的 4 类。当我们仅关心一个整数被 4 整除的余数时, 我们只需要知道它在哪个类而不必知道它的特定值。

这两个关系,  $R$  和模 4 同余关系是等价关系, 即是自反的、对称的和传递的。本节将证明这种关系把集合划分成由等价元素构成的不相交的类。当我们仅关心集合的一个元素是否在某个元素类, 而不介意它的特殊身份时, 就出现了等价关系。

### 5.5.2 等价关系



在这一节我们将研究具有一组特殊性质的关系, 可以用这组性质为在某一方面类似的相关个体之间建立联系。

**定义 1** 集合  $A$  上的关系叫做等价关系, 如果它是自反的、对称的和传递的。

等价关系在整个数学和计算机科学中都很重要。原因之一就是 在等价关系中, 当两个元素相联系就可以说它们是等价的。

**定义 2** 如果两个元素  $a$  和  $b$  由于等价关系而相联系, 则称它们是等价的。记法  $a \sim b$  通常用来表示对于某个特定的等价关系来说,  $a$  和  $b$  是等价的元素。

为了使等价元素的概念有意义, 每个元素都应该等价于它自身, 因为对于等价关系来说, 自反性是一定成立的。在等价关系中说  $a$  和  $b$  相互联系也是正确的 (而不仅是  $a$  联系于  $b$ ), 因为如果  $a$  联系于  $b$ , 由对称性,  $b$  也联系于  $a$ 。此外, 因为等价关系是传递的, 如果  $a$  和  $b$  等价且  $b$  和  $c$  等价, 则可得出  $a$  和  $c$  也是等价的。

例 1~例 5 说明了等价关系的概念。

**例 1** 设  $R$  是整数集上的关系, 满足  $aRb$ , 当且仅当  $a=b$  或  $a=-b$ 。在 5.1 节我们证明了  $R$  是自反的、对称的和传递的。因此  $R$  是等价关系。 ■

**例2** 设  $R$  是实数集上的关系, 满足  $aRb$ , 当且仅当  $a-b$  是整数.  $R$  是等价关系吗?

**解** 因为对所有的实数  $a$ ,  $a-a=0$  是整数, 即对所有的实数有  $aRa$ , 因此  $R$  是自反的. 现在假设  $aRb$ , 那么  $a-b$  是整数, 所以  $b-a$  也是整数. 因此有  $bRa$ .  $R$  是对称的. 如果  $aRb$  和  $bRc$ , 那么  $a-b$  和  $b-c$  是整数, 所以  $a-c=(a-b)+(b-c)$  也是整数. 因此  $aRc$ . 于是,  $R$  是传递的. 综上所述,  $R$  是等价关系. ■

最广泛使用的等价关系之一是模  $m$  同余关系, 其中  $m$  是大于 1 的正整数.

**例3** 模  $m$  同余. 设  $m$  是大于 1 的正整数. 证明以下关系是整数集上的等价关系.

$$R = \{(a, b) \mid a \equiv b \pmod{m}\}$$

**解**  $a \equiv b \pmod{m}$ , 当且仅当  $m$  整除  $a-b$ . 注意  $a-a=0$  被  $m$  整除, 因为  $0=0 \cdot m$ . 因此  $a \equiv a \pmod{m}$ , 从而模  $m$  同余关系是自反的. 现在假设  $a \equiv b \pmod{m}$ , 那么  $a-b$  被  $m$  整除, 即  $a-b=km$ , 其中  $k$  是整数. 从而  $b-a=(-k)m$ , 即  $b \equiv a \pmod{m}$ , 因此模  $m$  同余关系是对称的. 下面假设  $a \equiv b \pmod{m}$  和  $b \equiv c \pmod{m}$ , 那么  $m$  整除  $a-b$  和  $b-c$ . 因此, 存在整数  $k$  和  $l$ , 使得  $a-b=km$  和  $b-c=lm$ . 把这两个等式加起来得  $a-c=(a-b)+(b-c)=km+lm=(k+l)m$ . 于是,  $a \equiv c \pmod{m}$ , 从而模  $m$  同余关系是传递的. 综上所述, 模  $m$  同余关系是等价关系. ■

**例4** 设  $R$  是英语字母串的集合上的关系并且使得  $aRb$  当且仅当  $l(a)=l(b)$ , 其中  $l(x)$  是串  $x$  的长度.  $R$  是等价关系吗?

**解** 因为  $l(a)=l(a)$ , 从而只要  $a$  是一个串, 就有  $aRa$ , 故  $R$  是自反的. 其次, 假设  $aRb$ , 即  $l(a)=l(b)$ . 那么有  $bRa$ , 因为  $l(b)=l(a)$ , 因此  $R$  是对称的. 最后, 假设  $aRb$  和  $bRc$ , 那么有  $l(a)=l(b)$  和  $l(b)=l(c)$ . 因此  $l(a)=l(c)$ , 即  $aRc$ , 从而  $R$  是传递的. 由于  $R$  是自反的、对称的和传递的,  $R$  是等价关系. ■

**例5** 设  $n$  是正整数,  $S$  是字符串集. 假定  $R_n$  是  $S$  上的关系,  $sR_nt$  当且仅当  $s=t$  或者  $s$  和  $t$  都至少含有  $n$  个字符, 且  $s$  和  $t$  的前  $n$  个字符相同. 就是说, 少于  $n$  个字符的字符串只和它自身以关系  $R_n$  相联系; 至少含有  $n$  个字符的字符串  $s$  联系于  $t$  当且仅当  $t$  也含有至少  $n$  个字符且以  $s$  最前面的  $n$  个字符开始. 例如, 设  $n=3$ ,  $S$  是所有比特串的集合,  $sR_3t$  当  $s=t$  或者  $s$  和  $t$  的长度至少为 3, 且前 3 个比特相同时成立. 例如,  $01R_301$ ,  $00111R_300101$ , 但  $01R_3010$ ,  $01011R_301110$ .

证明: 对所有的字符串集  $S$  和所有的正整数  $n$ ,  $R_n$  是  $S$  上的等价关系.

**解** 如果  $s=s$ , 则  $sR_ns$  只要  $s$  是  $S$  中的一个字符串, 所以关系  $R_n$  是自反的. 如果  $sR_nt$ , 那么或者  $s=t$  或者  $s$  和  $t$  都至少含有  $n$  个字符, 且以相同的  $n$  个字符开始. 这就意味着  $tR_ns$  成立. 所以  $R_n$  是对称的.

现在假设  $sR_nt$  且  $tR_nu$ . 则有,  $s=t$  或者  $s$  和  $t$  都至少含有  $n$  个字符且以相同的  $n$  个字符开始; 还有  $t=u$  或者  $t$  和  $u$  都至少含有  $n$  个字符且以相同的  $n$  个字符开始; 由此可以推出  $s=u$  或者  $s$  和  $u$  都至少含有  $n$  个字符且以相同的  $n$  个字符开始 (因为在这种情形下, 我们知道  $s, t, u$  都至少有  $n$  个字符, 且  $s$  和  $u$  都与  $t$  一样以相同的  $n$  个字符开始). 所以  $R_n$  是传递的. 这就证明了  $R_n$  是一个等价关系. ■

在例 6 和例 7 中, 将看到两个非等价的关系.

**例6** 证明: 正整数上的“整除”关系不是等价关系.

**解** 由 5.1 节的例 9 和例 15 可知, “整除”关系是自反和传递的. 但是, 由 5.1 节的例 12, 知道此关系不是对称的 (例如,  $2 \mid 4$  但  $4 \nmid 2$ ). 所以得出, 正整数上的“整除”关系不是等价关系. ■

**例7**  $R$  是实数集上的关系,  $xRy$  当且仅当  $x$  和  $y$  是差小于 1 的实数, 即  $|x-y| < 1$ . 证明  $R$  不是等价关系.

**解**  $R$  是自反的, 因为只要  $x \in \mathbf{R}$ , 就有  $|x-x|=0 < 1$ .  $R$  是对称的, 因为如果  $xRy$ , 此处  $x$  和  $y$  是实数, 那么有  $|x-y| < 1$ , 这告诉我们  $|y-x|=|x-y| < 1$ , 因此  $yRx$ . 然而,  $R$  不是等价关系, 因为它不是传递的. 取  $x=2.8$ ,  $y=1.9$ , 和  $z=1.1$ , 这样  $|x-y|=|2.8-1.9|=0.9 < 1$ ,  $|y-z|=|1.9-1.1|=0.8 < 1$ , 但是  $|x-z|=|2.8-1.1|=1.7 > 1$ . 这

就是说,  $2.8 R 1.9$ ,  $1.9 R 1.1$  但  $2.8 R 1.1$ 。

### 5.5.3 等价类

设  $A$  是在你们学校所有的高中毕业生。考虑  $A$  上的关系  $R$ ,  $R$  由所有的对  $(x, y)$  构成, 其中  $x$  和  $y$  从同一高中毕业。给定学生  $x$ , 我们可以构造所有的关于  $R$  与  $x$  等价的学生的集合。这个集合由与  $x$  在同一高中毕业的所有学生构成。 $A$  的这个子集叫做这个关系的一个等价类。

**定义 3** 设  $R$  是集合  $A$  上的等价关系。与  $A$  中的一个元素  $a$  有关系的所有元素的集合叫做  $a$  的等价类。 $A$  的关于  $R$  的等价类记作  $[a]_R$ 。当只有一个关系被考虑时, 我们将省去下标  $R$  并把这个等价类写作  $[a]$ 。

换句话说, 如果  $R$  是集合  $A$  上的等价关系, 元素  $a$  的等价类是

$$[a]_R = \{s \mid (a, s) \in R\}$$

如果  $b \in [a]_R$ ,  $b$  叫做这个等价类的代表元。一个等价类的任何元素都可以作为这个类的代表元。也就是说, 对作为这个类的代表元所选择的特定元素没有特殊要求。

**例 8** 对于例 1 的等价关系, 一个整数的等价类是什么?

**解** 在这个等价关系中, 一个整数等价于它自身和它的相反数。从而  $[a] = \{-a, a\}$ 。这个集合包含两个不同的整数, 除非  $a=0$ 。例如,  $[7] = \{-7, 7\}$ ,  $[-5] = \{-5, 5\}$ ,  $[0] = \{0\}$ 。

**例 9** 对于模 4 同余关系, 0 和 1 的等价类是什么?

**解** 0 的等价类包含使得  $a \equiv 0 \pmod{4}$  的所有整数  $a$ 。这个类中的整数是被 4 整除的那些整数。因此, 对于这个关系 0 的等价类是

$$[0] = \{\dots, -8, -4, 0, 4, 8, \dots\}$$

1 的等价类包含使得  $a \equiv 1 \pmod{4}$  的所有整数  $a$ 。这个类中的整数是当被 4 除时余数为 1 的那些整数。因此, 对于这个关系 1 的等价类是

$$[1] = \{\dots, -7, -3, 1, 5, 9, \dots\}$$

在例 9 找到了 0 和 1 关于模 4 同余的等价类。用任何正整数  $m$  代替 4 很容易把例 9 加以推广。模  $m$  同余关系的等价类叫做模  $m$  同余类。整数  $a$  模  $m$  的同余类记作  $[a]_m$ , 满足  $[a]_m = \{\dots, a-2m, a-m, a, a+m, a+2m, \dots\}$ 。例如, 从例 9 得出  $[0]_4 = \{\dots, -8, -4, 0, 4, 8, \dots\}$  和  $[1]_4 = \{\dots, -7, -3, 1, 5, 9, \dots\}$ 。

**例 10** 对于例 5 中全体比特串集合上的等价关系  $R_s$  而言, 串 0111 的等价类是什么? (回忆  $s R_t$  当且仅当  $s, t$  是满足如下条件的比特串:  $s=t$  或者  $s$  和  $t$  都至少含有 3 个比特, 且  $s$  和  $t$  的前 3 个比特相同。)

**解** 等价于 0111 的是以 011 开始, 至少含有 3 个比特的比特串。它们是 011, 0110, 0111, 01100, 01101, 01110, 01111, 等等。所以

$$[011]_{R_s} = \{011, 0110, 0111, 01100, 01101, 01110, 01111, \dots\}$$

**例 11 C 程序设计语言中的标识符** 在 C 语言中, 标识符是变量、函数或者其他类型的实体的名字。每个标识符是一个非空字符串, 串中的每个字符可以是 大写或小写的英文字母、数字或下划线, 而第一个字符必须为 大写或小写的英文字母。标识符的长度是任意的, 这就使得开发者可以按照自己的意愿使用一定数量的字符来命名一个实体, 比如变量。然而, 对于某些版本的 C 编译器来说, 当比较两个名字看它们是否表示同一事物时, 实际检查的字符个数是有限制的。例如, 当两个标识符的前 31 个字符相同时, 标准 C 编译器就认为它们是相同的。所以, 开发者就必须小心, 不要使用前 31 个字符相同的标识符来表示不同的事物。我们可以看出, 如果两个标识符由例 5 中的关系  $R_{31}$  联系起来, 那么它们将被视做相同。由例 5 知道, 在标准 C 的标识符集上, 关系  $R_{31}$  是一个等价关系。

现有下标识符: Number\_of\_tropical\_storms, Number\_of\_named\_tropical\_storms, Number\_of\_named\_tropical\_storms\_in\_the\_Atlantic\_in\_2005, 它们的等价类各是什么?

**解** 注意, 当一个标识符的长度小于 31 的时候, 根据  $R_{31}$  的定义, 它的等价类只包含它自身。因为标识符 `Number_of_tropical_storms` 只含有 25 个字符, 所以它的等价类只含有一个元素, 即它自己。

标识符 `Number_of_named_tropical_storms` 的长度刚好为 31。一个标识符就与它等价意味着以这 31 个字符开始。所以, 每个长度至少为 31、以 `Number_of_named_tropical_storms` 开始的标识符都与这个标识符等价。所以得出, `Number_of_named_tropical_storms` 的等价类是全体以 `Number_of_named_tropical_storms` 这 31 个字符开始的标识符。

一个标识符与 `Number_of_named_tropical_storms_in_the_Atlantic_in_2005` 等价, 当且仅当它以 `Number_of_named_tropical_storms_in_the_Atlantic_in_2005` 的前 31 个字符开始。因为这 31 个字符是 `Number_of_named_tropical_storms`, 我们看到一个标识符与 `Number_of_named_tropical_storms_in_the_Atlantic_in_2005` 等价, 当且仅当它与 `Number_of_named_tropical_storms` 等价。就是说, 最后这两个标识符的等价类是相同的。 ■

#### 5.5.4 等价类与划分

设  $A$  是你们学校恰好主修一个专业的学生的集合,  $R$  是  $A$  上的关系, 如果  $x$  和  $y$  是主修同一专业的学生, 则  $(x, y)$  属于  $R$ 。那么正如读者可以验证的,  $R$  是等价关系。我们可以看出  $R$  将  $A$  中的所有学生分成不相交的子集, 其中每个子集包含某个特定专业的学生。例如, 一个子集包含所有(只主修)计算机专业的学生, 第二个子集包含所有历史专业的学生。而且这些子集是  $R$  的等价类。这个例子说明一个等价关系的等价类怎样把一个集合划分成不相交的非空子集。我们将在下面的讨论中把这些概念进一步精确化。

设  $R$  是集合  $A$  上的等价关系。下面的定理证明  $A$  的两个元素的等价类或是相等或是不相交。

**定理 1** 设  $R$  是集合  $A$  上的等价关系, 下面的命题是等价的。

- (i)  $aRb$                       (ii)  $[a] = [b]$                       (iii)  $[a] \cap [b] \neq \emptyset$

**证** 首先证明(i)推出(ii)。假设  $aRb$ , 我们将通过  $[a] \subseteq [b]$  和  $[b] \subseteq [a]$  来证明  $[a] = [b]$ 。假设  $c \in [a]$ , 那么  $aRc$ 。因为  $aRb$  和  $R$  的对称性, 有  $bRa$ 。又由于  $R$  是传递的以及  $bRa$  和  $aRc$ , 就得到  $bRc$ , 因而有  $c \in [b]$ 。这就证明了  $[a] \subseteq [b]$ 。类似地, 可证明  $[b] \subseteq [a]$ , 证明留给读者作为练习。

其次我们将证明(ii)推出(iii)。假设  $[a] = [b]$ 。这就证明了  $[a] \cap [b] \neq \emptyset$ , 因为  $[a]$  是非空的(由  $R$  的自反性  $a \in [a]$ )。

下面证明(iii)推出(i)。假设  $[a] \cap [b] \neq \emptyset$ , 那么存在元素  $c$  满足  $c \in [a]$  和  $c \in [b]$ 。换句话说,  $aRc$  和  $bRc$ 。由对称性有  $cRb$ 。再根据传递性, 由  $aRc$  和  $cRb$  就有  $aRb$ 。

因为(i)推出(ii), (ii)推出(iii), (iii)推出(i), 三个命题(i)、(ii)和(iii)是等价的。 □

我们现在将显示一个等价关系怎样划分一个集合。设  $R$  是集合  $A$  上的等价关系,  $R$  的所有等价类的并集就是  $A$  的全部, 因为  $A$  的每个元素  $a$  都在它自己的等价类即  $[a]_R$  中。换句话说,

$$\bigcup_{a \in A} [a]_R = A$$

此外, 由定理 1, 这些等价类或是相等的或是不相交的, 因此当  $[a]_R \neq [b]_R$  时,

$$[a]_R \cap [b]_R = \emptyset$$

这两个观察证明等价类构成  $A$  的划分, 因为它们将  $A$  分成不相交的子集。更确切地说, 集合  $S$  的划分是一族  $S$  的不相交的非空子集, 且  $S$  就是它们的并。换句话说, 一族子集  $A_i$ ,  $i \in I$ , (其中  $I$  是指标集)构成  $S$  的划分, 当且仅当

$$A_i \neq \emptyset, \quad i \in I$$

$$A_i \cap A_j = \emptyset, \quad i \neq j$$

和

$$\bigcup_{i \in I} A_i = S$$

(这里符号  $\bigcup_{i \in I} A_i$  表示对于所有的  $i \in I$  集合  $A_i$  的并集。)图 5-13 说明了集合的划分的概念。

**例 12** 假设  $S = \{1, 2, 3, 4, 5, 6\}$ , 一族集合  $A_1 = \{1, 2, 3\}$ ,  $A_2 = \{4, 5\}$  和  $A_3 = \{6\}$  构成  $S$  的一个划分, 因为这些集合是不相交的, 且它们的并是  $S$ 。■

我们已经看到集合上等价关系的等价类构成这个集合的划分。这个划分中的子集就是这些等价类。反过来, 可以用集合的每个划分来构成等价关系。两个元素关于这个关系是等价的, 当且仅当它们在这个划分的同一子集中。

为看到这一点, 假设  $\{A_i \mid i \in I\}$  是  $S$  的划分。设  $R$  是  $S$  上的由有序对  $(x, y)$  组成的等价关系, 其中  $x$  和  $y$  属于这个划分的同一子集  $A_i$ 。为证明  $R$  是等价关系, 我们必须证明  $R$  是自反的、对称的和传递的。

对于每个  $a \in S$  有  $(a, a) \in R$ , 因为  $a$  和它自己在同一子集中, 因此  $R$  是自反的。如果  $(a, b) \in R$ , 那么  $b$  和  $a$  在这个划分的同一子集中, 因此有  $(b, a) \in R$ 。从而  $R$  是对称的。如果  $(a, b) \in R$  和  $(b, c) \in R$ , 那么  $a$  和  $b$  在这个划分的同一子集中, 比如是  $X$ ,  $b$  和  $c$  也在这个划分的同一子集中, 比如是  $Y$ 。因为划分的子集是不相交的, 并且  $b$  属于  $X$  和  $Y$ , 必有  $X = Y$ 。因而  $a$  和  $c$  属于这个划分的同一子集, 即  $(a, c) \in R$ 。于是  $R$  是传递的。

这就证明了  $R$  是等价关系。 $R$  的等价类由  $S$  的子集构成, 这些子集包含了  $S$  的有关关系的元素, 根据  $R$  的定义, 它们就是划分的子集。定理 2 是对等价关系和划分之间建立的这种联系的总结。

**定理 2** 设  $R$  是集合  $S$  上的等价关系。那么  $R$  的等价类构成  $S$  的划分。相反, 给定集合  $S$  的划分  $\{A_i \mid i \in I\}$ , 存在着等价关系  $R$ , 它以集合  $A_i (i \in I)$  作为它的等价类。

例 13 说明了怎样从一个划分构造一个等价关系。

**例 13**  $A_1 = \{1, 2, 3\}$ ,  $A_2 = \{4, 5\}$ ,  $A_3 = \{6\}$  是例 12 给出的集合  $S = \{1, 2, 3, 4, 5, 6\}$  的划分, 列出这个划分所产生的等价关系  $R$  中的有序对。

**解** 划分中的子集是  $R$  的等价类。对  $(a, b) \in R$ , 当且仅当  $a$  和  $b$  在划分的同一个子集中。由于  $A_1 = \{1, 2, 3\}$  是一个等价类, 因此, 对  $(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)$  属于  $R$ ; 由于  $A_2 = \{4, 5\}$  是一个等价类, 因此, 对  $(4, 4), (4, 5), (5, 4), (5, 5)$  也属于  $R$ ; 最后, 由于  $\{6\}$  是一个等价类, 对  $(6, 6)$  属于  $R$ 。此外没有其他的对属于  $R$ 。■

模  $m$  同余类对定理 2 提供了一个有用的说明。存在  $m$  个不同的模  $m$  同余类, 对应于当一个整数除以  $m$  时可能得到的  $m$  个不同的余数。这  $m$  个同余类记作  $[0]_m, [1]_m, \dots, [m-1]_m$ 。它们构成了整数集合的划分。

**例 14** 模 4 同余产生的整数划分中的集合是什么?

**解** 存在 4 个同余类, 对应于  $[0]_4, [1]_4, [2]_4$  和  $[3]_4$ , 它们是集合

$$[0]_4 = \{\dots, -8, -4, 0, 4, 8, \dots\}$$

$$[1]_4 = \{\dots, -7, -3, 1, 5, 9, \dots\}$$

$$[2]_4 = \{\dots, -6, -2, 2, 6, 10, \dots\}$$

$$[3]_4 = \{\dots, -5, -1, 3, 7, 11, \dots\}$$

这些同余类是不相交的, 并且每个整数恰好在它们之中的一个。换句话说, 正如定理 2 所说, 这些同余类构成了一个划分。■

现在举一个例子: 从这个集合上的等价关系产生的全体串集合的划分。

**例 15** 设  $R_3$  为例 5 中的关系。在全体比特串集合上, 由  $R_3$  产生的对全体比特串集合的划分中是哪些集合? (回忆  $s, t$  是比特串,  $s R_3 t$ , 如果  $s = t$  或者  $s$  和  $t$  都至少含有 3 个比特, 且它们的前 3 个比特相同。)

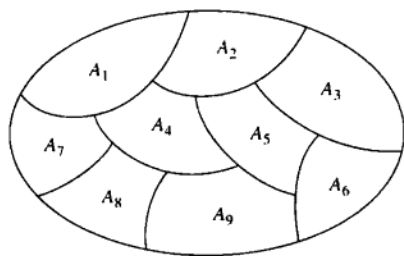


图 5-13 集合的划分

解 注意到每个长度小于3的比特串只和它自身等价, 有 $[\lambda]_{R_1} = \{\lambda\}$ ,  $[0]_{R_1} = \{0\}$ ,  $[1]_{R_1} = \{1\}$ ,  $[00]_{R_1} = \{00\}$ ,  $[01]_{R_1} = \{01\}$ ,  $[10]_{R_1} = \{10\}$ ,  $[11]_{R_1} = \{11\}$ . 再注意到每个长度大于或等于3的比特串必和000, 001, 010, 011, 100, 101, 110, 111这8个比特串之一等价, 我们有

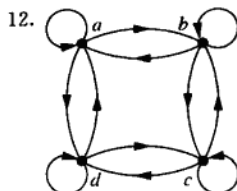
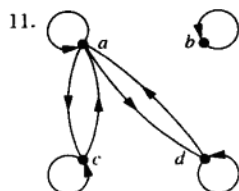
$$\begin{aligned} [000]_{R_1} &= \{000, 0000, 0001, 00000, 00001, 00010, 00011, \dots\} \\ [001]_{R_1} &= \{001, 0010, 0011, 00100, 00101, 00110, 00111, \dots\} \\ [010]_{R_1} &= \{010, 0100, 0101, 01000, 01001, 01010, 01011, \dots\} \\ [011]_{R_1} &= \{011, 0110, 0111, 01100, 01101, 01110, 01111, \dots\} \\ [100]_{R_1} &= \{100, 1000, 1001, 10000, 10001, 10010, 10011, \dots\} \\ [101]_{R_1} &= \{101, 1010, 1011, 10100, 10101, 10110, 10111, \dots\} \\ [110]_{R_1} &= \{110, 1100, 1101, 11000, 11001, 11010, 11011, \dots\} \\ [111]_{R_1} &= \{111, 1110, 1111, 11100, 11101, 11110, 11111, \dots\} \end{aligned}$$

这15个等价类是不相交的, 每个比特串都恰好属于它们之一。正如定理2告诉我们的, 这些等价类是全体比特串构成的集合的一个划分。 ■

### 练习

- 下面是 $\{0, 1, 2, 3\}$ 上的关系, 其中哪些是等价关系? 确定一个等价关系的性质, 这些性质是其他关系所欠缺的。
  - $\{(0, 0), (1, 1), (2, 2), (3, 3)\}$
  - $\{(0, 0), (0, 2), (2, 0), (2, 2), (2, 3), (3, 2), (3, 3)\}$
  - $\{(0, 0), (1, 1), (1, 2), (2, 1), (2, 2), (3, 3)\}$
  - $\{(0, 0), (1, 1), (1, 3), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}$
  - $\{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 2), (3, 3)\}$
- 下面是从 $\mathbf{Z}$ 到 $\mathbf{Z}$ 的所有函数集合上的关系, 其中哪些是等价关系? 确定一个等价关系的性质, 这些性质是其他关系所欠缺的。
  - $\{(f, g) \mid f(1)=g(1)\}$
  - $\{(f, g) \mid f(0)=g(0) \text{ 或 } f(1)=g(1)\}$
  - $\{(f, g) \mid \text{对所有的 } x \in \mathbf{Z}, f(x)-g(x)=1\}$
  - $\{(f, g) \mid \text{对某个 } C \in \mathbf{Z}, \text{对所有的 } x \in \mathbf{Z}, f(x)-g(x)=C\}$
  - $\{(f, g) \mid f(0)=g(1) \text{ 且 } f(1)=g(0)\}$
- 在大学校园里的建筑物集合上定义3个等价关系, 确定关于这些等价关系的等价类。
- 证明: 在所有复合命题集合上的逻辑等价关系是等价关系。这里T和F的等价类是什么?
- 假设A是非空集合,  $f$ 是以A作为定义域的函数, 设R是A上的关系, 若 $f(x)=f(y)$ , 则 $(x, y)$ 属于R。
  - 证明R是A上的等价关系。
  - R的等价类是什么?
- 设R是长度至少为3的所有二进位串的集合上的关系, R由对 $(x, y)$ 构成, 其中 $x$ 和 $y$ 在它们的前3位相同。证明R是等价关系。
- 设R是长度至少为3的所有二进位串的集合上的关系, R由序对 $(x, y)$ 构成, 其中 $x$ 和 $y$ 在它们的第1和第3位字符相同。证明R是等价关系。
- 设R是正整数的有序对集合上的关系,  $((a, b), (c, d)) \in R$ 当且仅当 $a+d=b+c$ 。证明R是等价关系。
- (需要微积分)
  - 设R是从 $\mathbf{R}$ 到 $\mathbf{R}$ 的所有可微分函数的集合上的关系, R由所有的对 $(f, g)$ 构成, 其中对所有实数 $x$ ,  $f'(x)=g'(x)$ 。证明R是等价关系。
  - 什么函数与函数 $f(x)=x^2$ 在同一个等价类中?
- 设R是所有URL(或Web地址)集合上的关系,  $xRy$ 当且仅当在 $x$ 的Web页与在 $y$ 的Web页相同, 证明R是等价关系。

在练习11~12中确定具有所有有向图的关系是否为等价关系。



13. 设  $R$  是所有二进位串的集合上的关系,  $sRt$  当且仅当  $s$  和  $t$  包含相同个数的 1, 证明  $R$  是等价关系。

14. 对于练习 13 中的等价关系, 二进位串 011 的等价类是什么?

15. 当  $n$  为下列各数时, 同余类  $[n]_5$  (即  $n$  关于模 5 同余的等价类) 是什么?

a) 2

b) 3

c) 6

d) -3

16. 描述每一个模 6 同余类。

17. a) 对于练习 8 中的等价关系 (1, 2) 的等价类是什么?

b) 对于练习 8 中的等价关系  $R$  解释等价类的含义。[提示: 差  $a-b$  对应  $(a, b)_0$ 。]

18. 下面哪些子集族是  $\{1, 2, 3, 4, 5, 6\}$  的划分?

a)  $\{1, 2\}, \{2, 3, 4\}, \{4, 5, 6\}$

b)  $\{1\}, \{2, 3, 6\}, \{4\}, \{5\}$

c)  $\{2, 4, 6\}, \{1, 3, 5\}$

d)  $\{1, 4, 5\}, \{2, 6\}$

19. 下面哪些子集族是 8 位二进位串的集合上的划分?

a) 以 1 开始的二进位串的集合, 以 00 开始的二进位串的集合, 以 01 开始的二进位串的集合。

b) 包含串 00 的二进位串的集合, 包含串 01 的二进位串的集合, 包含串 10 的二进位串的集合, 包含串 11 的二进位串的集合。

c) 以 00 结尾的二进位串的集合, 以 01 结尾的二进位串的集合, 以 10 结尾的二进位串的集合, 以 11 结尾的二进位串的集合。

d) 以 111 结尾的二进位串的集合, 以 011 结尾的二进位串的集合, 以 00 结尾的二进位串的集合。

e) 含  $3k+1$  个 1 的二进位串的集合, 其中  $k$  为非负整数; 含  $3k+1$  个 1 的二进位串的集合, 其中  $k$  为非负整数; 含  $3k+2$  个 1 的二进位串的集合, 其中  $k$  是正整数。

20. 下面哪些是整数的有序对的集合  $\mathbb{Z} \times \mathbb{Z}$  的划分?

a)  $x$  或  $y$  是奇数的  $(x, y)$  对的集合;  $x$  是偶数的  $(x, y)$  对的集合;  $y$  是偶数的  $(x, y)$  对的集合。

b)  $x$  和  $y$  都是奇数的  $(x, y)$  对的集合;  $x$  和  $y$  只有一个为奇数的  $(x, y)$  对的集合;  $x$  和  $y$  都是偶数的  $(x, y)$  对的集合。

c)  $x$  是正数的  $(x, y)$  对的集合;  $y$  是正数的  $(x, y)$  对的集合;  $x$  和  $y$  都是负数的  $(x, y)$  对的集合。

d)  $x$  和  $y$  都被 3 整除的  $(x, y)$  对的集合;  $x$  被 3 整除且  $y$  不被 3 整除的  $(x, y)$  对的集合;  $x$  不被 3 整除且  $y$  被 3 整除的  $(x, y)$  对的集合;  $x$  和  $y$  都不被 3 整除的  $(x, y)$  对的集合。

e)  $x > 0$  且  $y > 0$  的  $(x, y)$  对的集合;  $x > 0$  且  $y \leq 0$  的  $(x, y)$  对的集合;  $x \leq 0$  且  $y > 0$  的  $(x, y)$  对的集合;  $x \leq 0$  且  $y \leq 0$  的  $(x, y)$  对的集合。

f)  $x \neq 0$  且  $y \neq 0$  的  $(x, y)$  对的集合;  $x = 0$  且  $y \neq 0$  的  $(x, y)$  对的集合;  $x \neq 0$  且  $y = 0$  的  $(x, y)$  对的集合。

21. 列出由  $\{0, 1, 2, 3, 4, 5\}$  的划分产生的等价关系中的有序对。

a)  $\{0\}, \{1, 2\}, \{3, 4, 5\}$

b)  $\{0, 1\}, \{2, 3\}, \{4, 5\}$

c)  $\{0, 1, 2\}, \{3, 4, 5\}$

d)  $\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}$

如果在划分  $P_1$  中的每个集合都是划分  $P_2$  中每个集合的子集, 则  $P_1$  叫做  $P_2$  的加细。

22. 证明: 由模 6 同余类构成的划分是模 3 同余类构成的划分的加细。

23. 证明: 对于 16 位二进位串的集合, 最后 8 位相同的二进位串的等价类所构成的划分是由最后 4 位相同的二进位串的等价类所构成的划分的加细。

在练习 24 中,  $R_n$  表示例 5 中定义的等价关系族。回忆字符串  $s, t$  满足  $sR_n t$ , 如果  $s=t$  或者  $s$  和  $t$  都至少含有  $n$  个字符, 且它们的前 5 个字符相同。

24. 证明: 由等价关系  $R_3$  决定的标识符等价类构成的 C 语言中所有标识符的划分是由等价关系  $R_8$  决定的标识符等价类构成的划分的加细。(旧的 C 语言编译器只要多个标识符的前 8 个字符相同就将它们视为相同, 而标准 C 的编译器需要多个标识符的前 31 个字符相同才将它们视为相同。)



25. 求出在集合  $\{a, b, c, d, e\}$  上包含关系  $\{(a, b), (a, c), (d, e)\}$  的最小的等价关系。
26. 考虑例 3 的等价关系, 即  $R = \{(x, y) \mid x - y \text{ 是整数}\}$ 。
- a) 关于这个等价关系的 1 的等价类是什么?
- b) 关于这个等价关系的  $1/2$  的等价类是什么?
- \*27. 设  $R$  是  $2 \times 2$  棋盘的所有涂色集合上的关系, 其中 4 个方格中的每一个可以被涂成红色或蓝色。 $C_1$  和  $C_2$  是被这样涂色的  $2 \times 2$  棋盘,  $(C_1, C_2) \in R$ , 当且仅当  $C_2$  可以由旋转  $C_1$ , 或旋转  $C_1$  然后再翻转  $C_1$  得到。
- a) 证明  $R$  是等价关系。
- b)  $R$  的等价类是什么?
28. 通过列举确定 3 元集上的不同的等价关系个数。
- \*29. 当我们构造一个关系的自反闭包的对称闭包的传递闭包时, 一定得到一个等价关系吗?
30. 假设我们使用定理 2 从一个等价关系  $R$  构造一个划分  $P$ 。如果再次使用定理 2 从  $P$  构造一个等价关系, 那么结果的等价关系  $R$  是什么?
31. 设计一个算法找出包含一个给定关系的最小的等价关系。

## 5.6 偏序

### 5.6.1 引言

我们常常用关系对集合的某些元素或全体元素排序。例如, 使用包含着字对  $(x, y)$  的关系对字排序, 其中  $x$  按照字典顺序排在  $y$  的前面。使用包含着对  $(x, y)$  的关系安排课题, 其中  $x$  和  $y$  是课题中的任务并且  $x$  必须在  $y$  开始之前完成。使用包含着对  $(x, y)$  的关系排序整数集合, 其中  $x$  小于  $y$ 。当我们把所有形如  $(x, x)$  的对加到这些关系中时, 就得到了一个自反、反对称和传递的关系。这些都是刻画元素有序集合的关系特征的性质, 而这些关系使用它们相对的大小来排序集合的元素。

**定义 1** 集合  $S$  上的关系  $R$ , 如果它是自反的、反对称的和传递的, 就称为偏序。集合  $S$  与偏序  $R$  一起叫做偏序集, 记作  $(S, R)$ 。

**例 1** 证明: “大于或等于”关系  $(\geq)$  是整数集合上的偏序。

**解** 因为对所有整数  $a$  有  $a \geq a$ ,  $\geq$  是自反的。如果  $a \geq b$  且  $b \geq a$ , 那么  $a = b$ , 因此  $\geq$  是反对称的。最后, 因为  $a \geq b$  和  $b \geq c$  推出  $a \geq c$ , 所以  $\geq$  是传递的。从而  $\geq$  是整数集合上的偏序且  $(\mathbb{Z}, \geq)$  是偏序集。

**例 2** 整除关系 “ $|$ ” 是正整数集合上的偏序, 因为正如 5.1 节所述, 它是自反的、反对称的和传递的。我们看到  $(\mathbb{Z}^+, |)$  是偏序集 ( $\mathbb{Z}^+$  表示正整数集合)。

**例 3** 证明: 包含关系  $\subseteq$  是集合  $S$  的幂集上的偏序。

**解** 因为只要  $A$  是  $S$  的子集, 就有  $A \subseteq A$ ,  $\subseteq$  是自反的。因为  $A \subseteq B$  和  $B \subseteq A$  推出  $A = B$ , 因此它是反对称的。最后, 因为  $A \subseteq B$  和  $B \subseteq C$  推出  $A \subseteq C$ ,  $\subseteq$  是传递的。因此,  $\subseteq$  是  $P(S)$  上的偏序, 且  $(P(S), \subseteq)$  是偏序集。

**例 4** 举出了一个不是偏序的关系。

**例 4** 设  $R$  是由人构成的集合上的关系,  $x R y$  当且仅当  $x$  和  $y$  是人, 且  $x$  年纪大于  $y$ 。证明:  $R$  不是偏序。

**解** 注意  $R$  是反对称的, 因为如果有一个人  $x$  比另一个人  $y$  年长, 那么  $y$  就不会比  $x$  年长。也就是说, 如果  $x R y$ , 那么  $y \not R x$ 。关系  $R$  是传递的, 因为如果  $x$  比  $y$  年长, 而  $y$  又比  $z$  年长, 那么  $x$  肯定比  $z$  年长。就是说, 如果  $x R y$ ,  $y R z$ , 那么  $x R z$ 。但是,  $R$  不是自反的, 因为没有谁会比自己年长。即对于所有的人  $x$ ,  $x \not R x$ 。这就意味着  $R$  不是偏序。

在一个偏序集中记号  $a \leq b$  表示  $(a, b) \in R$ 。使用这个记号是由于“小于或等于”关系是偏序关系的范例。(注意符号  $\leq$  用来表示任意偏序集的关系, 并不仅仅是“小于或等于”关系。)记号  $a < b$  表示  $a \leq b$ , 但  $a \neq b$ 。如果  $a < b$ , 我们说“ $a$  小于  $b$ ”或“ $b$  大于  $a$ ”。

当  $a$  与  $b$  是偏序集  $(S, \leq)$  的元素时, 不一定有  $a \leq b$  或  $b \leq a$ 。例如, 在  $(P(\mathbb{Z}), \subseteq)$  中,

$\{1, 2\}$ 与 $\{1, 3\}$ 没有关系, 反之亦然, 因为没有集合被另一个集合包含。类似地, 在 $(\mathbb{Z}^+, |)$ 中, 2与3没关系, 3也与2没关系, 因为 $2 \nmid 3$ 且 $3 \nmid 2$ 。由此得到定义2。

**定义2** 偏序集 $(S, \leq)$ 的元素 $a$ 和 $b$ 叫做可比的, 如果 $a \leq b$ 或 $b \leq a$ 。当 $a$ 和 $b$ 是 $S$ 的元素并且既没有 $a \leq b$ , 也没有 $b \leq a$ , 则称 $a$ 与 $b$ 是不可比的。

**例5** 在偏序集 $(\mathbb{Z}^+, |)$ 中, 整数3和9是可比的吗? 5和7是可比的吗?

**解** 整数3和9是可比的, 因为 $3 \mid 9$ 。整数5和7是不可比的, 因为 $5 \nmid 7$ 且 $7 \nmid 5$ 。 ■

用形容词“部分的(偏的)”描述偏序是由于一些对元素可能是不可比的。当集合中的每对元素都可比时, 这个关系叫做全序。

**定义3** 如果 $(S, \leq)$ 是偏序集, 且 $S$ 的每对元素都是可比的, 则 $S$ 叫做全序集或线序集, 且 $\leq$ 叫做全序或线序。一个全序集也叫做链。

**例6** 偏序集 $(\mathbb{Z}, \leq)$ 是全序集, 因为只要 $a$ 和 $b$ 是整数, 就有 $a \leq b$ 或 $b \leq a$ 。 ■

**例7** 偏序集 $(\mathbb{Z}^+, |)$ 不是全序集, 因为它包含着不可比的元素, 例如5和7。 ■

我们注意到 $(\mathbb{Z}^+, \leq)$ 是良序的, 其中 $\leq$ 是通常的“小于或等于”关系。我们现在定义良序集。

**定义4** 对于偏序集 $(S, \leq)$ , 如果 $\leq$ 是全序, 并且 $S$ 的每个非空子集都有一个最小元素, 就称它为良序集。

**例8** 正整数的有序对的集合 $\mathbb{Z}^+ \times \mathbb{Z}^+$ 与 $\leq$ 构成良序集。对于 $(a_1, a_2)$ 和 $(b_1, b_2)$ , 如果 $a_1 < b_1$ , 或如果 $a_1 = b_1$ 且 $a_2 \leq b_2$ (字典顺序), 则 $(a_1, a_2) \leq (b_1, b_2)$ 。有关的验证留作节后的练习。集合 $\mathbb{Z}$ 与通常的 $\leq$ 序不是良序的, 因为负整数集合是 $\mathbb{Z}$ 的子集, 但没有最小元素。 ■

现在我们叙述使用良序归纳原理证明关于一个良序集的结果这个证明技术是有效的。

**定理1 良序归纳原理** 设 $S$ 是一个良序集, 如果下述条件成立:

归纳步骤: 对一切 $y \in S$ , 如果 $P(x)$ 对所有 $x \in S$ 且 $x < y$ 为真, 则 $P(y)$ 为真。

那么 $P(x)$ 对所有的 $x \in S$ 为真。

**证** 假若 $P(x)$ 不对所有的 $x \in S$ 为真。那么存在一个元素 $y \in S$ 使得 $P(y)$ 为假。于是集合 $A = \{x \in S \mid P(x) \text{ 为假}\}$ 是非空的。因为 $S$ 是良序的, 集合 $A$ 有最小元素 $a$ 。根据 $a$ 是选自 $A$ 的最小元素, 我们知道对所有的 $x \in S(x < a)$ 都有 $P(x)$ 为真。由归纳步这就可以推出 $P(a)$ 为真。这个矛盾就证明了 $P(x)$ 必须对所有 $x \in S$ 为真。 □

良序归纳原理对证明关于良序集的结果是一种通用的技术。即使可以使用关于正整数集合的数学归纳法证明一个定理时, 使用良序归纳原理甚至可能更简单。

### 5.6.2 字典顺序

字典中的字按照字母顺序或字典顺序排列, 字典顺序是以字母表中的字母顺序为基础的。这是从一个集合上的偏序构造一个集合上的串的序的特殊情况。我们将说明这种构造在任一个偏序集上是怎样做的。

首先, 我们将说明怎样在两个偏序集 $(A_1, \leq_1)$ 和 $(A_2, \leq_2)$ 的笛卡儿积上构造一个偏序。在 $A_1 \times A_2$ 上的字典顺序定义如下: 如果第一个对的第一个元素(在 $A_1$ 中)小于第二个对的第一个元素, 或者第一个元素相等, 但是第一个对的第二个元素(在 $A_2$ 中)小于第二个对的第二个元素, 那么第一个对小于第二个对。换句话说,  $(a_1, a_2)$ 小于 $(b_1, b_2)$ , 即

$$(a_1, a_2) < (b_1, b_2)$$

或者 $a_1 < b_1$ , 或者 $a_1 = b_1$ 且 $a_2 < b_2$ 。

把相等加到 $A_1 \times A_2$ 上的序 $<$ 就得到偏序 $\leq$ 。这个验证留作练习。

**例9** 确定在偏序集 $(\mathbb{Z} \times \mathbb{Z}, \leq)$ 中是否有 $(3, 5) < (4, 8)$ ,  $(3, 8) < (4, 5)$ 和 $(4, 9) < (4, 11)$ ? 这里的 $\leq$ 是从 $\mathbb{Z}$ 上通常的 $\leq$ 关系构造的字典顺序。

**解** 因为 $3 < 4$ , 故而 $(3, 5) < (4, 8)$ 且 $(3, 8) < (4, 5)$ 。因为 $(4, 9)$ 与 $(4, 11)$ 的第一元素相同, 但是 $9 < 11$ , 我们有 $(4, 9) < (4, 11)$ 。 ■

图5-14明显地显示了 $\mathbb{Z}^+ \times \mathbb{Z}^+$ 中比 $(3, 4)$ 小的有序对的集合。

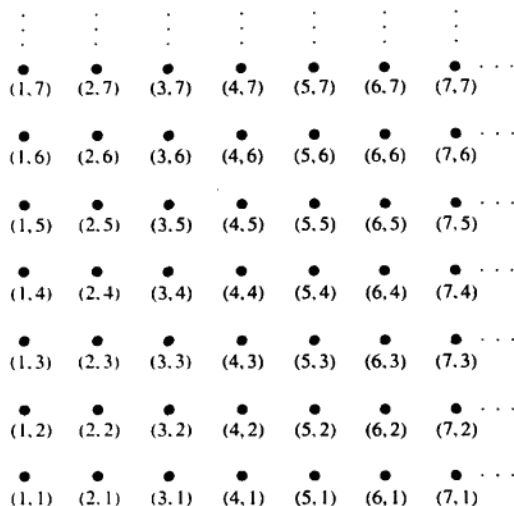


图 5-14 按照字典顺序比(3, 4)小的有序对

可以在  $n$  个偏序集  $(A_1, \leq_1), (A_2, \leq_2), \dots, (A_n, \leq_n)$  的笛卡儿积上定义字典顺序。如下定义  $A_1 \times A_2 \times \dots \times A_n$  上的偏序  $\leq$ : 如果  $a_1 <_1 b_1$ , 或者存在整数  $i > 0$  使得  $a_1 = b_1, \dots, a_i = b_i$ , 且  $a_{i+1} <_{i+1} b_{i+1}$ , 那么

$$(a_1, a_2, \dots, a_n) < (b_1, b_2, \dots, b_n)$$

如果  $a_1 <_1 b_1$ , 或者如果存在整数  $i > 0$ , 使得  $a_1 = b_1, \dots, a_i = b_i$ , 且  $a_{i+1} <_{i+1} b_{i+1}$ 。换句话说, 如果在两个  $n$  元组不同元素出现的第一位置上第一个  $n$  元组的元素小于第二个  $n$  元组的元素, 那么第一个  $n$  元组小于第二个  $n$  元组。

**例 10** 注意  $(1, 2, 3, 5) < (1, 2, 4, 3)$ , 因为这些 4 元组的前两位相同, 但是第一个 4 元组的第三位 3 小于第二个 4 元组的第三位 4 (这里的 4 元组上的字典顺序是整数集合上的通常的“小于或等于”关系导出的字典顺序)。

我们现在可以定义串上的字典顺序。考虑偏序集  $S$  上的串  $a_1 a_2 \dots a_m$  和  $b_1 b_2 \dots b_n$ , 假定这些串不相等。设  $t$  是  $m, n$  中较小的数。定义字典顺序如下:  $a_1 a_2 \dots a_m$  小于  $b_1 b_2 \dots b_n$ , 当且仅当

$$(a_1, a_2, \dots, a_t) < (b_1, b_2, \dots, b_t) \quad \text{或者}$$

$$(a_1, a_2, \dots, a_t) = (b_1, b_2, \dots, b_t) \quad \text{并且 } m < n$$

其中这个不等式中的  $<$  表示  $S'$  中的字典顺序。换句话说, 为确定两个不同串的序, 较长的串被切到较短的串的长度  $t$ , 即  $t = \min(m, n)$ 。然后使用  $S'$  上的字典顺序比较每个串的前  $t$  位组成的  $t$  元组。如果对应于第一个串的  $t$  元组小于第二个串的  $t$  元组, 或者这两个  $t$  元组相等, 但是第二个串更长, 那么第一个串小于第二个串。这是偏序的验证留给读者作为练习。

**例 11** 考虑小写英语字母串构成的集合。使用在字母表中的字母序, 可以构造在串的集合上的字典顺序。如果两个串第一次出现不同字母时, 第一个串的字母先于第二个串的字母, 或者如果第一个串和第二个串在所有的位都相同, 但是第二个串有更多的字母, 那么, 第一个串小于第二个串。这种排序和字典使用的排序相同。例如,

$$\text{discreet} < \text{discrete}$$

因为这两个串在第 7 位首次出现不同字母, 并且  $e < t$ 。

$$\text{discreet} < \text{discreetness}$$

因为这两个串前 8 个字母相同, 但是第二个串更长。此外,

$$\text{discrete} < \text{discretion}$$

因为

$discrete < discreti$

5.6.3 哈塞图

在有穷偏序集的有向图中有许多边可以不必显示出来，因为它们是必须存在的。例如，考虑在集合 $\{1, 2, 3, 4\}$ 上的偏序 $\{(a, b) \mid a \leq b\}$ 的有向图，参见图 5-15a。因为这个关系是偏序，它是自反的并且有向图在所有的顶点都有环。因此，我们不必显示这些环，因为它们是必须出现的。在图 5-15b 中没有显示这些环。由于一个偏序是传递的，我们不必显示那些由于传递性而必须出现的边。例如，在图 5-15c 中没有显示边 $(1, 3)$ ， $(1, 4)$ 和 $(2, 4)$ ，因为它们是必须出现的。如果假设所有边的方向是向上的(如图中所示)，我们不必显示边的方向；图 5-15c 没有显示方向。

一般说来，我们可以使用下面的过程表示一个有穷集上的偏序。从这个关系的有向图开始。由于偏序是自反的，每个顶点有一个环，移走这些环。移走所有由于传递性出现的边，因为偏序是传递的，这些边是必须出现的。例如，如果 $(a, b)$ 和 $(b, c)$ 在偏序中，移走边 $(a, c)$ ，因为它必须要出现。此外，如果 $(c, d)$ 也在偏序中，移走边 $(a, d)$ ，因为它也是必须出现的。最后，排列每条边使得它的起点在终点的下面(正如在纸上所画的)。移走所有的有向边上的箭头，因为所有的边向上指向它们的终点。(只有和偏序集的覆盖关系中的对对应的边留下来。)

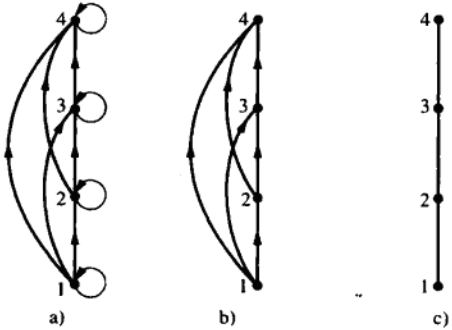


图 5-15 构造关于 $(\{1, 2, 3, 4\}, \leq)$ 的哈塞图

这些步骤是有着明确定义的，并且对于一个有穷偏序集只有有限步需要执行。当所有的步骤执行以后，就得到一个包含着足够的表示偏序信息的图。这个图叫做哈塞图，它是用 20 世纪德国数学家赫尔姆·哈塞的名字命名的。

**例 12** 画出表示 $\{1, 2, 3, 4, 6, 8, 12\}$ 上的偏序 $\{(a, b) \mid a \text{ 整除 } b\}$ 的哈塞图。

**解** 由这个偏序的有向图开始，如图 5-16a 所示。移走所有的环，如图 5-16b 所示。然后删除所有由传递性导出的边。这些边是 $(1, 4)$ ， $(1, 6)$ ， $(1, 8)$ ， $(1, 12)$ ， $(2, 8)$ ， $(2, 12)$ 和 $(3, 12)$ 。排列所有的边使得方向向上，并且删除所有的箭头得到哈塞图。结果如图 5-16c 所示。

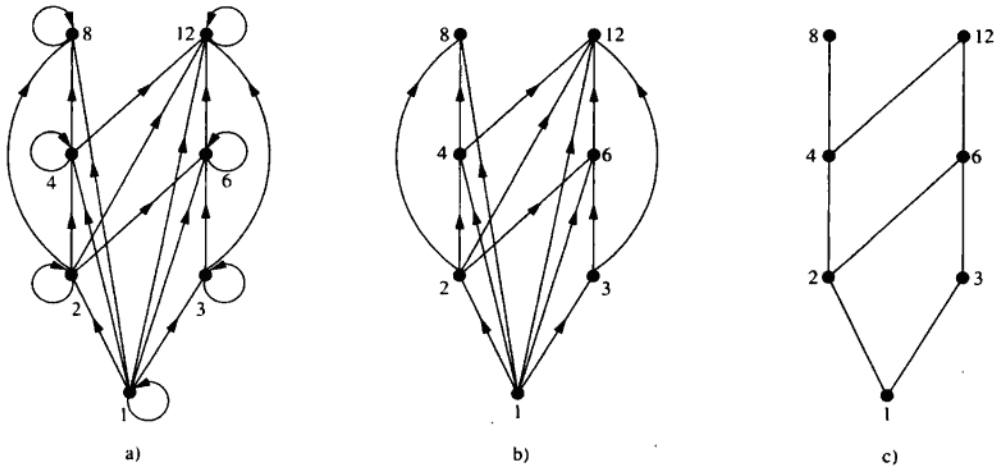


图 5-16 构造 $(\{1, 2, 3, 4, 6, 8, 12\}, \mid)$ 上的哈塞图

**例 13** 画出幂集  $P(S)$  上的偏序  $\{(A, B) \mid A \subseteq B\}$  的哈塞图, 其中  $S = \{a, b, c\}$ 。

**解** 关于这个偏序的哈塞图是由相关的有向图得到的, 先删除所有的环和所有由传递性产生的边, 即  $(\emptyset, \{a, b\})$ ,  $(\emptyset, \{a, c\})$ ,  $(\emptyset, \{b, c\})$ ,  $(\emptyset, \{a, b, c\})$ ,  $(\{a\}, \{a, b, c\})$ ,  $(\{b\}, \{a, b, c\})$  和  $(\{c\}, \{a, b, c\})$ 。最后, 使所有的边方向向上并删除箭头。得到的哈塞图如图 5-17 所示。

#### 5.6.4 极大元素与极小元素

具有极值性质的偏序集元素有许多重要应用。偏序集的一个元素叫做极大的, 当它不小于这个偏序集的任何其他元素。即  $a$  在偏序集  $(S, \leq)$  中是极大的, 当不存在  $b \in S$  使得  $a < b$ 。类似地, 偏序集的一个元素叫做极小的, 如果它不大于这个偏序集的任何其他元素。即  $a$  在偏序集  $(S, \leq)$  中是极小的, 如果不存在  $b \in S$  使得  $b < a$ 。使用哈塞图很容易识别极大元素与极小元素。它们是图中的“顶”元素与“底”元素。

**例 14** 偏序集  $(\{2, 4, 5, 10, 12, 20, 25\}, \mid)$  的哪些元素是极大的, 哪些是极小的?

**解** 图 5-18 关于这个偏序集的哈塞图显示了极大元素是 12, 20 和 25, 极小元素是 2 和 5。通过这个例子可以看出, 一个偏序集可以有多于一个的极大元素和多于一个的极小元素。

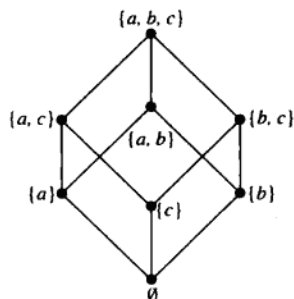


图 5-17  $(P(\{a, b, c\}), \subseteq)$  的哈塞图

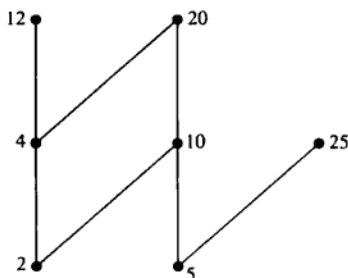


图 5-18 偏序集的哈塞图

有时在偏序集中存在一个元素大于每个其他的元素。这样的元素叫做最大元素。即  $a$  是偏序集  $(S, \leq)$  的最大元素, 当对所有的  $b \in S$  有  $b \leq a$ 。当最大元素存在时它是唯一的。类似地, 一个元素叫做最小元素, 当它小于偏序集的所有其他元素。即  $a$  是偏序集  $(S, \leq)$  的最小元素, 如果对所有的  $b \in S$  有  $a \leq b$ 。当最小元素存在时它也是唯一的。

**例 15** 确定图 5-19 的每个哈塞图表示的偏序集是否有最大元素和最小元素。

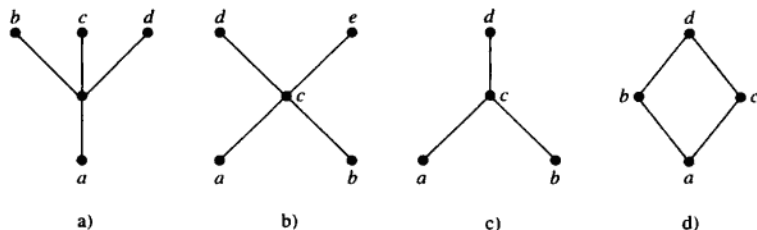


图 5-19 四个偏序集的哈塞图

**解** 哈塞图图 5-19a 的偏序集的最小元素是  $a$ 。这个偏序集没有最大元素。哈塞图图 5-19b 的偏序集既没有最小元素也没有最大元素。哈塞图图 5-19c 的偏序集没有最小元素, 它的最大元素是  $d$ 。哈塞图图 5-19d 的偏序集有最小元素  $a$  和最大元素  $d$ 。

**例 16** 设  $S$  是集合。确定偏序集  $(P(S), \subseteq)$  中是否存在最大元素与最小元素。

**解** 最小元素是空集, 因为对于  $S$  的任何子集  $T$ , 有  $\emptyset \subseteq T$ 。集合  $S$  是这个偏序集的最大元素, 因为只要  $T$  是  $S$  的子集, 就有  $T \subseteq S$ 。■

**例 17** 在偏序集  $(\mathbb{Z}^+, |)$  中是否存在最大元素和最小元素?

**解** 1 是最小元素, 因为只要  $n$  是正整数, 就有  $1 | n$ 。因为没有被所有正整数整除的整数, 所以不存在最大元素。■

有时候可以找到一个元素大于偏序集  $(S, \leq)$  的子集  $A$  中所有的元素。如果  $u$  是  $S$  的元素使得对所有的元素  $a \in A$ , 有  $a \leq u$ , 那么  $u$  叫做  $A$  的一个上界。类似地, 也可能存在一个元素小于  $A$  中的所有其他元素。如果  $l$  是  $S$  的一个元素使得对所有的元素  $a \in A$  有  $l \leq a$ , 那么  $l$  叫做  $A$  的一个下界。

**例 18** 找出在图 5-20 所示哈斯图的偏序集的子集  $\{a, b, c\}$ ,  $\{j, h\}$  和  $\{a, c, d, f\}$  的下界和上界。

**解**  $\{a, b, c\}$  的上界是  $e, f, j$  和  $h$ , 它的唯一的下界是  $a$ 。 $\{j, h\}$  没有上界, 它的下界是  $a, b, c, d, e$  和  $f$ 。 $\{a, c, d, f\}$  的上界是  $f, h$  和  $j$ , 它的下界是  $a$ 。■

元素  $x$  叫做子集  $A$  的最小上界, 如果  $x$  是一个上界并且它小于  $A$  的其他上界。因为如果这样的元素存在, 只存在一个, 称这个元素为最小上界是有意义的。即如果只要  $a \in A$  就有  $a \leq x$ , 并且只要  $z$  是  $A$  的上界, 就有  $x \leq z$ ,  $x$  就是  $A$  的最小上界。类似地, 如果  $y$  是  $A$  的下界, 并且只要  $z$  是  $A$  的下界, 就有  $z \leq y$ ,  $y$  就是  $A$  的最大下界。 $A$  的最大下界如果存在也是唯一的。一个子集  $A$  的最大下界和最小上界分别记作  $\text{glb}(A)$  和  $\text{lub}(A)$ 。■

**例 19** 在图 5-20 所示偏序集中如果  $\{b, d, g\}$  的最大下界和最小上界存在, 求出这个最大下界和最小上界。

**解**  $\{b, d, g\}$  的上界是  $g$  和  $h$ 。因为  $g < h$ ,  $g$  是最小上界。 $\{b, d, g\}$  的下界是  $a$  和  $b$ 。因为  $a < b$ ,  $b$  是最大下界。■

**例 20** 在偏序集  $(\mathbb{Z}^+, |)$  中, 如果集合  $\{3, 9, 12\}$  和  $\{1, 2, 4, 5, 10\}$  的最大下界和最小上界存在, 求出这些最大下界和最小上界。

**解** 如果  $3, 9, 12$  被一个整数整除, 那么这个整数就是  $\{3, 9, 12\}$  的下界。这样的整数只有 1 和 3。因为  $1 | 3$ , 3 是  $\{3, 9, 12\}$  的最大下界。集合  $\{1, 2, 4, 5, 10\}$  关系到 | 的下界只有 1, 因此 1 是  $\{1, 2, 4, 5, 10\}$  的最大下界。

一个整数是  $\{3, 9, 12\}$  的上界, 当且仅当它被 3, 9 和 12 整除。具有这种性质的整数就是那些被 3, 9 和 12 的最小公倍数 36 整除的整数。因此, 36 是  $\{3, 9, 12\}$  的最小上界。一个正整数是集合  $\{1, 2, 4, 5, 10\}$  的上界, 当且仅当它被 1, 2, 4, 5 和 10 整除。具有这种性质的整数就是被这些整数的最小公倍数 20 整除的整数。因此, 20 是  $\{1, 2, 4, 5, 10\}$  的最小上界。■

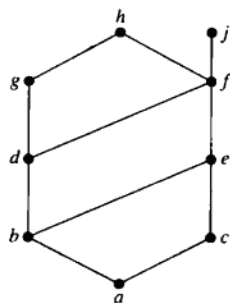


图 5-20 偏序集的哈塞图

### 5.6.5 格

如果一个偏序集的每对元素都有最小上界和最大下界, 就称这个偏序集为格。格有许多特殊的性质。此外, 格有许多不同的应用, 如用在信息流的模型。格在布尔代数中也起到了重要的作用。

**例 21** 确定图 5-21 的每个哈塞图表示的偏序集是否是格。

**解** 在图 5-21a 和图 5-21c 中的哈塞图表示的偏序集是格, 因为在每个偏序集中每对元素都有最小上界和最大下界, 读者应该能验证这一点。另一方面, 图 5-21b 所示的哈斯图的偏序集不是格, 因为元素  $b$  和  $c$  没有最小上界。为此只要注意到  $d, e$  和  $f$  中每一个都是上界, 但这 3 个元素的任何一个关于这个偏序集中的序都不大于其他 2 个。■

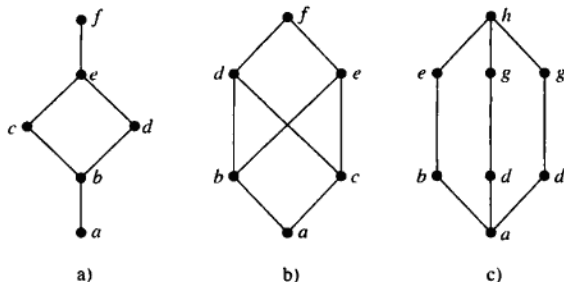


图 5-21 三个偏序集的哈塞图

**例 22** 偏序集 $(\mathbb{Z}^+, |)$ 是格吗?

**解** 设  $a$  和  $b$  是两个正整数。这两个整数的最小上界和最大下界分别是它们的最小公倍数和最大公约数, 读者应能验证这一点。因此这个偏序集是格。 ■

**例 23** 确定偏序集 $(\{1, 2, 3, 4, 5\}, |)$ 和 $(\{1, 2, 4, 8, 16\}, |)$ 是否为格。

**解** 因为 2 和 3 在 $(\{1, 2, 3, 4, 5\}, |)$ 中没有上界, 它们当然没有最小上界。因此第一个偏序集不是格。

第二个偏序集的每两个元素都有最小上界和最大下界。在这个偏序集中两个元素的最小上界是它们中间较大的元素, 而两个元素的最大下界是它们中间较小的元素。读者应能验证这一点。因此第二个偏序集是格。 ■

**例 24** 确定 $(P(S), \subseteq)$ 是否是格, 其中  $S$  是集合。

**解** 设  $A$  和  $B$  是  $S$  的两个子集。 $A$  和  $B$  的最小上界和最大下界分别是  $A \cup B$  和  $A \cap B$ , 读者可以证明这一点。因此 $(P(S), \subseteq)$ 是格。 ■

**例 25 信息流的格模型** 在许多设置中, 从一个人或计算机程序到另一个人或计算机程序的信息流要受到限制, 这可以通过安全权限来实现。我们可以使用格的模型来表示不同的信息流策略。例如, 一个通用的信息流策略是用于政府或军事系统中的多级安全策略。为每组信息分配一个安全级别, 并且每个安全级别用一个对 $(A, C)$ 表示, 其中  $A$  是权限级别,  $C$  是种类。然后允许人和计算机程序从一个被特别限制的安全类的集合中访问信息。

在美国政府中, 使用的典型的权限级别是不保密(0)、秘密(1)、机密(2)和绝密(3)。在安全级别中使用的种类是一个集合的子集, 这个集合含有与一个特定行业领域相关的所有的分部, 每个分部表示一个指定的对象域。例如, 如果分部的集合是{间谍, 鼯鼠, 双重间谍}, 那么存在 8 个不同的种类, 分部集合有 8 个子集, 对于每个子集有一类, 例如{间谍, 鼯鼠}。

我们可以对安全类排序, 规定 $(A_1, C_1) \leq (A_2, C_2)$ 当且仅当  $A_1 \leq A_2$  和  $C_1 \subseteq C_2$ 。信息允许从安全类 $(A_1, C_1)$ 流向安全类 $(A_2, C_2)$ 当且仅当 $(A_1, C_1) \leq (A_2, C_2)$ 。例如, 信息允许从安全类(机密, {间谍, 鼯鼠})流向安全类(绝密, {间谍, 鼯鼠, 双重间谍}), 反之, 信息不允许从安全类(绝密, {间谍, 鼯鼠})流向安全类(机密, {间谍, 鼯鼠, 双重间谍})或(绝密, {间谍})。 ■

### 5.6.6 拓扑排序

假设一个项目由 20 个任务构成。某些任务只能在其他任务结束之后完成。如何找到关于这些任务的顺序? 为了对这个问题构造模型, 我们建立一个任务集合上的偏序, 使得  $a < b$ , 当且仅当  $a$  和  $b$  是任务且直到  $a$  结束后  $b$  才能开始。为安排好这个项目, 需要得出与这个偏序相容的所有 20 个任务的顺序。我们将说明怎样做到这一点。

我们从定义开始。如果只要  $aRb$  就有  $a \leq b$ , 则称一个全序  $\leq$  与偏序  $R$  是相容的。从一个偏序构造一个相容的全序叫做拓扑排序。我们需要使用引理 1。

**引理 1** 每个有穷非空偏序集 $(S, \leq)$ 都有极小元素。

**证** 选择  $S$  的一个元素  $a_0$ 。如果  $a_0$  不是极小元素, 那么存在元素  $a_1$ , 满足  $a_1 < a_0$ 。如果  $a_1$  不





在项目的安排中常会用到拓扑排序。

**例 27** 一个计算机公司的开发项目需要完成 7 个任务。其中某些任务只能在其他任务结束后才能开始。考虑如下建立任务上的偏序，如果任务  $Y$  在  $X$  结束后才能开始，则任务  $X < \text{任务 } Y$ 。这 7 个任务关于这个偏序的哈塞图在图 5-23 中给出。求一个全序，使得可以按照这个全序执行这些任务以完成这个项目。

**解** 可以通过执行一个拓扑排序得到 7 个任务的排序。排序的步骤显示在图 5-24 中。这个排序的结果， $A < C < B < E < F < D < G$ ，给出了关于任务的一种可能的次序。

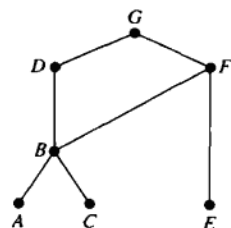


图 5-23 关于 7 个任务的哈塞图

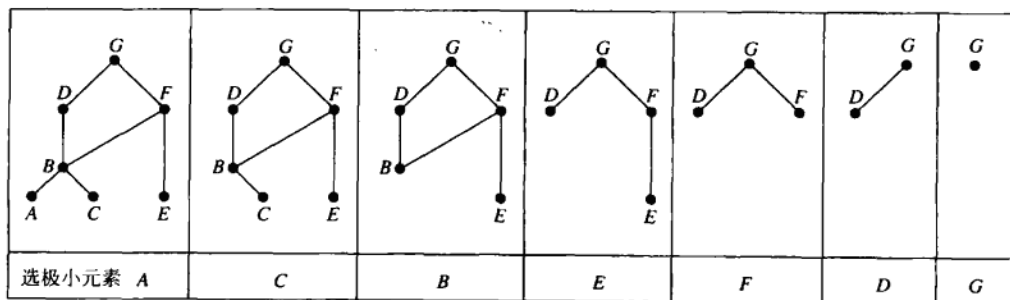


图 5-24 任务的拓扑排序

### 练习

1. 以下这些集合  $\{0, 1, 2, 3\}$  上的关系，哪些是偏序？如果不是偏序，请问它缺少偏序的哪些性质。

- $\{(0, 0), (1, 1), (2, 2), (3, 3)\}$
- $\{(0, 0), (1, 1), (2, 0), (2, 2), (2, 3), (3, 2), (3, 3)\}$
- $\{(0, 0), (1, 1), (1, 2), (2, 2), (3, 3)\}$
- $\{(0, 0), (1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}$
- $\{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 2), (3, 3)\}$

2. 设  $a$  和  $b$  是人， $S$  是全世界所有人构成的集合， $(a, b) \in R$ 。请问  $(S, R)$  是否为偏序，如果

- $a$  比  $b$  个子高？
- $a$  不比  $b$  高？
- $a=b$  或  $a$  是  $b$  的祖先？
- $a$  和  $b$  有共同的朋友？

3. 下面哪些集合是偏序集？

- $(\mathbb{Z}, =)$
- $(\mathbb{Z}, \neq)$
- $(\mathbb{Z}, \geq)$
- $(\mathbb{Z}, \neq)$

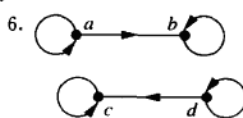
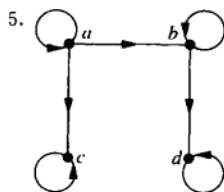
4. 确定以下 0-1 矩阵表示的关系是否为偏序。

a)  $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

b)  $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

c)  $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$

在练习 5~6 中确定有向图所表示的关系是否为偏序。



7. 求下面偏序集的对偶。

a)  $(\{0, 1, 2\}, \leq)$

b)  $(\mathbb{Z}, \geq)$

c)  $(P(\mathbb{Z}), \supseteq)$

d)  $(\mathbb{Z}^+, |)$

8. 在下面偏序集中找出两个不可比的元素。

a)  $(P(\{0, 1, 2\}), \subseteq)$

b)  $(\{1, 2, 4, 6, 8\}, |)$

9. 找出下面的  $n$  元组的字典顺序。

a)  $(1, 1, 2), (1, 2, 1)$

b)  $(0, 1, 2, 3), (0, 1, 3, 2)$

c)  $(1, 0, 1, 0, 1), (0, 1, 1, 1, 0)$

10. 找出二进制串  $0, 01, 11, 001, 010, 011, 0001$  和  $0101$  的基于  $0 < 1$  的字典顺序。

11. 画出  $\{0, 2, 5, 10, 11, 15\}$  上的“小于或等于”关系的哈塞图。

12. 画出下述集合上整除关系的哈塞图。

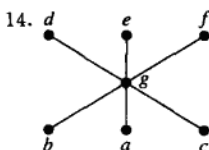
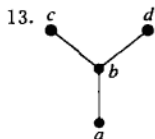
a)  $\{1, 2, 3, 4, 5, 6, 7, 8\}$

b)  $\{1, 2, 3, 5, 7, 11, 13\}$

c)  $\{1, 2, 3, 6, 12, 24, 36, 48\}$

d)  $\{1, 2, 4, 8, 16, 32, 64\}$

列出练习 13~14 所示哈塞图的偏序中的所有有序对。



设  $(S, \leq)$  是偏序集。我们说一个元素  $y \in S$  覆盖元素  $x \in S$ , 当  $x < y$  并且没有元素  $z \in S$  使得  $x < z < y$ 。使得  $y$  覆盖  $x$  的对  $(x, y)$  的集合叫做  $(S, \leq)$  的覆盖关系。

15.  $S$  的幂集上的偏序  $(A, B) \mid A \subseteq B$  的覆盖关系是什么? 其中  $S = \{a, b, c\}$ 。

16. 证明: 一个有穷偏序集可以从它的覆盖关系重新构造出来。[提示: 证明偏序集是它的覆盖关系的自反传递闭包。]

17. 对偏序集  $(\{3, 5, 9, 15, 24, 45\}, |)$ , 回答下述问题。

a) 求极大元素。

b) 求极小元素。

c) 存在最大元素吗?

d) 存在最小元素吗?

e) 找出  $\{3, 5\}$  的所有上界。

f) 如果存在的话, 求  $\{3, 5\}$  的最小上界。

g) 求  $\{15, 45\}$  的所有下界。

h) 如果存在的话, 求  $\{15, 45\}$  的最大下界。

18. 对偏序集  $(\{\{1\}, \{2\}, \{4\}, \{1, 2\}, \{1, 4\}, \{2, 4\}, \{3, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}, \subseteq)$ , 回答下述问题。

a) 求极大元素。

b) 求极小元素。

c) 存在最大元素吗?

d) 存在最小元素吗?

e) 求  $\{\{2\}, \{4\}\}$  的所有上界。

f) 如果存在的话, 求  $\{\{2\}, \{4\}\}$  的最小上界。

g) 求  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$  的所有下界。

h) 如果存在的话, 求  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$  的最大下界。

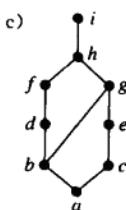
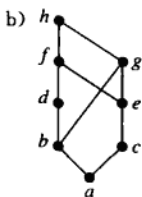
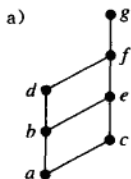
19. 证明: 字典顺序是两个偏序集的笛卡儿积上的偏序。

20. 假设  $(S, \leq_1)$  和  $(T, \leq_2)$  是偏序集。证明  $(S \times T, \leq)$  也是偏序集, 其中  $(s, t) \leq (u, v)$ , 当且仅当  $s \leq_1 u$  且  $t \leq_2 v$ 。

21. a) 证明: 在一个具有最大元素的偏序集中恰好存在一个极大元素。

b) 证明: 在一个具有最小元素的偏序集中恰好存在一个极小元素。

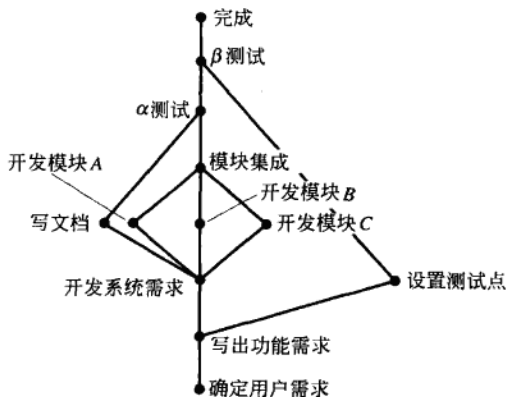
22. 确定具有下面哈塞图的偏序集是否为格。



23. 证明：一个格的每个有限非空子集有最小上界和最大下界。
24. 在一个公司里用信息流的格模型控制敏感信息，这些信息具有由有序对  $(A, C)$  表示的安全类别。这里  $A$  是权限级别，这种权限级别可以是非私有的(0)、私有的(1)、受限制的(2)或注册的(3)。种类  $C$  是所有项目集合{猎豹，黑斑羚，美洲狮}的子集(在公司里常常使用动物的名字作为项目的代码名字)。
- 信息是否允许从(私有的，{猎豹，美洲狮})流向(受限制的，{美洲狮})？
  - 信息是否允许从(受限制的，{猎豹})流向(注册的，{猎豹，黑斑羚})？
  - 允许信息从(私有的，{猎豹，美洲狮})流向哪个类？
  - 允许信息从哪个类流向安全类(受限制的，{黑斑羚，美洲狮})？
- \*25. 证明：一个集合上的所有划分的集合与关系  $S$  构成一个格，其中如果划分  $P_1$  是划分  $P_2$  的加细，则  $P_1 \leq P_2$  (见 5.5 节的练习 22 前面的说明)。
26. 证明：每个有限格都有一个最小元素和一个最大元素。
27. 验证  $(\mathbb{Z}^+ \times \mathbb{Z}^+, \leq)$  是一个良序集，其中  $\leq$  是如例 8 所声明的字典顺序。

如果偏序集中不存在元素的无限递减序列，即元素  $x_1, x_2, \dots, x_n \dots$  使得  $\dots < x_n < \dots < x_2 < x_1$ ，则这个偏序集是良基的。设偏序集  $(R, \leq)$ ，如果对所有的  $x \in S$  和  $y \in S(x < y)$  都存在元素  $z \in R$  使得  $x < z < y$ ，则称这个偏序集是稠密的。

28. 证明：偏序集  $(\mathbb{Z}, \leq)$  (其中  $x < y$  当且仅当  $|x| < |y|$ ) 是良基的，但不是全序集。
29. 证明：有理数和通常的“小于或等于”关系构成的偏序集  $(\mathbb{Q}, \leq)$  是稠密偏序集。
30. 证明：偏序集是良序的，当且仅当它是全序的并且是良基的。
31. 求一个与例 27 构造的序不同的完成开发项目中任务的序。
32. 如果关于一个软件项目的任务的哈塞图如下所示，对这个软件项目的任务进行排序。



## 关键术语和结果

### 术语

从  $A$  到  $B$  的二元关系： $A \times B$  的子集。

$A$  的关系：从  $A$  到自身的二元关系(即  $A \times A$  的子集)。

$S \circ R$ ： $R$  与  $S$  的合成。

$R^{-1}$ ： $R$  的逆关系。

$R^n$ ： $R$  的  $n$  次幂。

自反的： $A$  上的一个关系是自反的，如果对所有的  $a \in A$  有  $(a, a) \in R$ 。

对称的： $A$  上的一个关系是对称的，如果只要  $(a, b) \in R$  就有  $(b, a) \in R$ 。

反对称的： $A$  上的关系是反对称的，如果只要  $(a, b) \in R$  和  $(b, a) \in R$  就有  $a = b$ 。

传递的： $A$  上的关系  $R$  是传递的，如果从  $(a, b) \in R$  和  $(b, c) \in R$  推出  $(a, c) \in R$ 。

$A_1, A_2, \dots, A_n$  的  $n$  元关系： $A_1 \times A_2 \times \dots \times A_n$  的子集。

关系数据模型：一个使用  $n$  元关系表示数据库的模型。

主关键字： $n$  元关系的一个域，使得一个  $n$  元组被它在这个域的值唯一确定。

复合关键字：一个  $n$  元关系域的笛卡儿积，使得一个  $n$  元组被它在这些域的值唯一确定。

选择算子：一个函数，它在  $n$  元关系中选出满足特定条件的  $n$  元组。

投影：一个函数，它从一个  $n$  元关系通过删除域产生一个阶较小的关系。

连接：一个函数，它把具有某些相同的域的  $n$  元关系组合起来。

有向图：称为顶点的元素以及这些元素的有序对(也叫做边)的集合。

环：形如  $(a, a)$  的边。

关系  $R$  关于性质  $P$  的闭包：包含  $R$  的关系  $S$  (如果存在的话)具有性质  $P$ ，并且被任何包含  $R$  且具有性质  $P$  的关系所包含。

有向图中的路径：边的序列  $(a, x_1), (x_1, x_2), \dots, (x_{n-2}, x_{n-1}), (x_{n-1}, b)$  使得序列中每条边的终点是后面一条边的起点。

有向图的回路(或圈)：以同一顶点作为起点和终点的路径。

$R^*$  (连通性关系)：由有序对  $(a, b)$  构成的关系，条件是存在一条从  $a$  到  $b$  的路径。

等价关系：自反的、对称的和传递的关系。

等价：如果  $R$  是等价关系，若  $aRb$ ，那么  $a$  等价于  $b$ 。

$[a]_R$  ( $a$  关于  $R$  的等价类)： $A$  中所有等价于  $a$  的元素的集合。

$[a]_m$  (模  $m$  的同余类)：与  $a$  模  $m$  同余的整数的集合。

集合  $S$  的划分：一族两两不相交的非空子集，且这些子集的并就是  $S$ 。

偏序：自反的、反对称的和传递的关系。

偏序集  $(S, R)$ ：集合  $S$  与这个集合上的偏序  $R$ 。

可比的：偏序集  $(A, \leq)$  的元素  $a$  和  $b$  是可比的如果  $a \leq b$  或  $b \leq a$ 。

不可比的：一个偏序集的元素不是可比的。

全序(或线序)：一个偏序，并且它的每对元素都是可比的。

全序(或线序)集：具有一个全序(或线序)的偏序集。

良序集：偏序  $(S, \leq)$ ，其中  $\leq$  是全序， $S$  的每个非空子集至少有一个元素。

字典顺序：笛卡儿积或串的一个偏序(参见 5.6.2 节)。

哈塞图：偏序集的图表示，其中所有的环和由传递性导出的边不出现，并且顶点的位置指示了边的方向。

极大元素：偏序集的一个元素，它不小于这个偏序集的所有其他元素。

极小元素：偏序集的一个元素，它不大于这个偏序集的所有其他元素。

最大元素：偏序集的一个元素，它大于或等于这个集合的所有其他元素。

最小元素：偏序集的一个元素，它小于或等于这个集合的所有其他元素。

集合的上界：偏序集的一个元素，它大于这个集合的所有其他元素。

集合的下界：偏序集的一个元素，它小于这个集合的所有其他元素。

集合的最小上界：集合的一个上界，它小于所有其他的上界。

集合的最大下界：集合的一个下界，它大于所有其他的下界。

格：一个偏序集，其中每对元素都有一个最大下界和一个最小上界。

与一个偏序相容的全序：包含了给定偏序的一个全序。

拓扑排序：用给定偏序构造一个相容的全序。

## 结果

- 集合  $A$  的关系  $R$  的自反闭包等于  $R \cup \Delta$ ，其中  $\Delta = \{(a, a) \mid a \in A\}$ 。
- 集合  $A$  的关系  $R$  的对称闭包等于  $R \cup R^{-1}$ ，其中  $R^{-1} = \{(b, a) \mid (a, b) \in R\}$ 。
- 一个关系的传递闭包等于从这个关系构成的连通性关系。

- 为了求一个关系的传递闭包的沃舍尔算法(见 5.4.5 节)。
- 设  $R$  是等价关系, 那么下面三条语句是等价的:  
(1)  $aRb$  (2)  $[a]_R \cap [b]_R \neq \emptyset$  (3)  $[a]_R = [b]_R$
- 集合  $A$  的等价关系的等价类构成  $A$  的划分。相反, 从一个划分可以构造一个等价关系使得等价类就是划分中的子集。
- 良序归纳原理。
- 拓扑排序算法(见 5.6.5 节)。

## 复习题

- a) 什么是集合上的关系? b) 一个  $n$  元素集合上有多少个关系?
- a) 什么是自反关系? b) 什么是对称关系?  
c) 什么是反对称关系? d) 什么是传递关系?
- 给出一个集合  $\{1, 2, 3, 4\}$  上的关系的例子, 使得它是  
a) 自反的、对称的但不是传递的。 b) 不是自反的, 是对称的和传递的。  
c) 自反的、反对称的, 但不是传递的。 d) 自反的、对称的和传递的。  
e) 自反的、反对称的和传递的。
- a) 在一个  $n$  元素集合上有多少个自反的关系?  
b) 在一个  $n$  元素集合上有多少个对称的关系?  
c) 在一个  $n$  元素集合上有多少个反对称的关系?
- a) 解释在一个大学里怎样用一个  $n$  元关系表示有关学生的信息。  
b) 怎样用一个包含学生姓名、地址、电话号码、专业和平均学分绩点的 5 元关系构造包含学生姓名、专业和平均学分绩点的 3 元关系。  
c) 怎样将包含学生姓名、地址、电话号码和专业的 4 元关系和包含学生姓名、学号、专业和学分数的 4 元关系组合成一个单一的  $n$  元关系?
- a) 解释怎样使用一个 0-1 矩阵表示有穷集上的关系。  
b) 解释怎样使用表示关系的 0-1 矩阵来确定这个关系是否为自反的、对称的和反对称的。
- a) 解释怎样使用一个有向图来表示有穷集上的关系。  
b) 解释怎样使用表示关系的有向图来确定这个关系是否为自反的、对称的和反对称的。
- a) 定义一个关系的自反闭包和对称闭包。  
b) 可以怎样构造一个关系的自反闭包?  
c) 可以怎样构造一个关系的对称闭包?  
d) 求集合  $\{1, 2, 3, 4\}$  上的关系  $\{(1, 2), (2, 3), (2, 4), (3, 1)\}$  的自反闭包和对称闭包。
- a) 定义一个关系的传递闭包。  
b) 一个关系的传递闭包能够通过包含所有的对  $\langle a, c \rangle$  得到吗? 其中  $\langle a, c \rangle$  满足  $\langle a, b \rangle$  和  $\langle b, c \rangle$  属于  $R$ 。  
c) 描述求关系的传递闭包的两个算法。  
d) 求关系  $\{(1, 1), (1, 3), (2, 1), (2, 3), (2, 4), (3, 2), (3, 4), (4, 1)\}$  的传递闭包。
- a) 定义等价关系。  
b) 集合  $\{a, b, c, d\}$  上的哪些关系是包含了  $\langle a, b \rangle$  和  $\langle b, d \rangle$  的等价关系?
- a) 证明模  $m$  同余关系是等价关系, 其中  $m$  是正整数。  
b) 证明关系  $\{\langle a, b \rangle \mid a \equiv \pm b \pmod{7}\}$  是整数集上的等价关系。
- a) 什么是一个等价关系的等价类?  
b) 什么是模 5 同余关系的等价类?  
c) 什么是问题 11(b) 中等价关系的等价类?
- 解释在集合的等价关系与集合划分之间的联系。
- a) 定义偏序。 b) 证明正整数集合上的整除关系是偏序。
- 解释怎样用集合  $A_1$  和  $A_2$  上的偏序定义在集合  $A_1 \times A_2$  上的偏序。
- a) 解释怎样构造有穷集上的偏序的哈塞图。

- b) 画出集合  $\{2, 3, 5, 9, 12, 15, 18\}$  上的整除关系的哈塞图。
17. a) 定义一个偏序集的极大元素和最大元素。  
 b) 给出一个有 3 个极大元素的偏序集的例子。  
 c) 给出一个有 1 个最大元素的偏序集的例子。
18. a) 定义格。  
 b) 给出一个 5 元偏序集是格的例子和一个 5 元偏序集不是格的例子。
19. a) 证明一个格的每个有穷子集有一个最大下界和一个最小上界。  
 b) 证明每个具有有限元素的格有一个最小元素和一个最大元素。
20. a) 定义一个良序集。  
 b) 描述一个从偏序集产生良序集的算法。  
 c) 如果仅当某个或某些其他任务完成以后, 一个任务才可以开始, 解释怎样用 (b) 中的算法排序这个项目中的任务。

## 补充练习

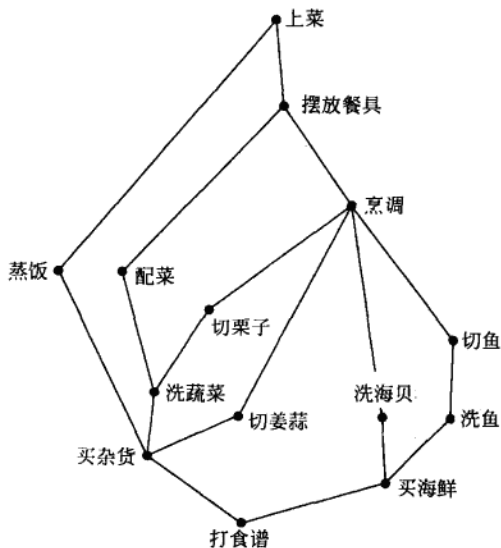
1. 设  $S$  是所有英语字母的串的集合。确定下面的关系是否是自反的、反自反的、对称的、反对称的和传递的。  
 a)  $R_1 = \{(a, b) \mid a \text{ 和 } b \text{ 没有公共字母}\}$   
 b)  $R_2 = \{(a, b) \mid a \text{ 和 } b \text{ 长度不相等}\}$   
 c)  $R_3 = \{(a, b) \mid a \text{ 比 } b \text{ 长}\}$
2. 构造集合  $\{a, b, c, d\}$  上的关系, 使得它是  
 a) 自反的、对称的, 但不是传递的。  
 b) 反自反的、对称的和传递的。  
 c) 反自反的、反对称的, 但不是传递的。  
 d) 自反的, 既不是对称的也不是反对称的, 是传递的。  
 e) 既不是自反的、反自反的、对称的和反对称的, 也不是传递的。
3.  $\mathbb{Z} \times \mathbb{Z}$  上的关系  $R$  定义如下:  $(a, b)R(c, d)$ , 当且仅当  $a + d = b + c$ 。证明:  $R$  是等价关系。
4. 证明: 一个反对称关系的子集也是一个反对称关系。
5. 设  $R$  是  $A$  上的自反关系, 证明:  $R \subseteq R^2$ 。
6. 假设  $R_1$  和  $R_2$  是集合  $A$  上的自反关系, 证明:  $R_1 \oplus R_2$  是反自反的。
7. 假设  $R_1$  和  $R_2$  是集合  $A$  上的自反关系,  $R_1 \cap R_2$  也是自反的吗?  $R_1 \cup R_2$  也是自反的吗?
8. 假设  $R$  是集合  $A$  上的对称关系,  $\bar{R}$  也是对称的吗?
9. 设  $R_1$  和  $R_2$  是集合  $A$  上的对称关系,  $R_1 \cap R_2$  也是对称的吗?  $R_1 \cup R_2$  也是对称的吗?
10. 一个关系  $R$  叫做循环的, 如果  $aRb$  和  $bRc$  推出  $cRa$ 。证明  $R$  是自反的和循环的, 当且仅当它是等价关系。
11. 证明: 一个  $n$  元关系的主关键字也是这个关系的任何投影的主关键字, 其中这个投影包含这个关键字作为它的一个域。
12. 一个  $n$  元关系的主关键字也就是由取这个关系与第二个关系的连接而得到的较大的关系的主关键字吗?
13. 证明: 一个关系的对称闭包的自反闭包和它的自反闭包的对称闭包是相同的。
14. 设  $R$  是所有数学家的集合上的关系,  $R$  包含有序对  $(a, b)$ , 当且仅当  $a$  与  $b$  合写了一篇文章。  
 a) 描述关系  $R^2$ 。  
 b) 描述关系  $R^*$ 。
15. a) 如果一个数学家与多产的匈牙利数学家保罗·埃德斯合写了一篇文章, 那么这个数学家的埃德斯数是 1。如果这个数学家没有与埃德斯合写一篇文章, 但是与某个与埃德斯合写过论文的人合写了一篇文章, 那么这个数学家的埃德斯数是 2, 依此下去 (埃德斯本人的埃德斯数是 0)。用  $R$  中路径的概念给出埃德斯数的定义。
16. a) 给出一个例子, 证明一个关系的对称闭包的传递闭包不一定与这个关系的传递闭包的对称闭包相等。  
 b) 证明: 一个关系的对称闭包的传递闭包一定包含这个关系的传递闭包的对称闭包。
17. a) 设  $S$  是一个计算机问题的子程序的集合。定义关系  $R$ , 如果在执行中子程序  $P$  调用子程序  $Q$ , 那么  $P R Q$ 。描述  $R$  的传递闭包。  
 b) 对于哪些子程序  $P$ ,  $(P, P)$  属于  $R$  的传递闭包?

c)描述  $R$  的传递闭包的自反闭包。

17. 假设  $R$  和  $S$  是集合  $A$  上的关系,  $R \subseteq S$ , 且  $R$  和  $S$  的关于性质  $P$  的闭包存在。证明  $R$  关于  $P$  的闭包是  $S$  关于  $P$  的闭包的子集。
18. 证明: 两个关系的并的对称闭包是它们的对称闭包的并。
- \*19. 设计一个基于内点概念的算法, 求有向图中两个顶点之间的最长路径的长度, 或确定在这些顶点之间存在任意长的路径。
20. 下面的哪些关系是所有人集合上的等价关系?
  - a)  $\{(x, y) \mid x \text{ 与 } y \text{ 有同样的星座}\}$
  - b)  $\{(x, y) \mid x \text{ 与 } y \text{ 出生在同一年}\}$
  - c)  $\{(x, y) \mid x \text{ 与 } y \text{ 曾待在同一城市}\}$
- \*21. 在 5 元素集合上有多少个不同的等价关系恰有 3 个不同的等价类?
22. 证明:  $\{(x, y) \mid x - y \in \mathbb{Q}\}$  是实数集上的等价关系, 其中  $\mathbb{Q}$  是有理数集合。 $[1]$ ,  $[1/2]$  和  $[\pi]$  是什么?
23. 设  $P_1 = \{A_1, A_2, \dots, A_m\}$  和  $P_2 = \{B_1, B_2, \dots, B_n\}$  都是集合  $S$  的划分。证明: 形如  $A_i \cap B_j$  的非空子集族是  $S$  的划分, 且是  $P_1$  和  $P_2$  的加细(见 5.5 节练习 22 前面的说明)。
- \*24. 证明: 关系  $R$  的自反闭包的对称闭包的传递闭包是包含  $R$  的最小的等价关系。
25. 设  $\mathbf{R}(S)$  是集合  $S$  上的所有关系的集合。如下定义  $\mathbf{R}(S)$  上的关系  $\leq$ , 如果  $R_1 \subseteq R_2$  则  $R_1 \leq R_2$ , 这里的  $R_1$  和  $R_2$  是  $S$  上的关系。证明:  $(\mathbf{R}(S), \leq)$  是偏序集。
26. 设  $\mathbf{P}(S)$  是  $S$  上的所有划分的集合。定义  $\mathbf{P}(S)$  上的关系如下: 如果  $P_1$  是  $P_2$  的加细, 则  $P_1 \leq P_2$  (见 5.5 节的练习 22)。证明:  $(\mathbf{P}(S), \leq)$  是偏序集。
27. 右下图为烹调中餐任务的哈塞图, 通过规定那些必需任务的顺序来安排烹调。

如果一个偏序集子集的每一对元素都是可比的, 就称这个子集为一条链。如果它的每一对元素都是不可比的, 就称这个子集为一条反链。

28. 找出 5.6 节练习 13~14 的哈塞图所表示的偏序集中所有的链。
29. 找出 5.6 节练习 13~14 的哈塞图所表示的偏序集中所有的反链。
30. 证明: 在一个有穷偏序集  $(S, \leq)$  中每个极大链包含  $S$  的一个极小元素。(一条极大链是一条链, 但不是一条更大的链的子链。)
- \*\*31. 证明: 一个偏序集可以被分成  $k$  条链, 其中  $k$  是在这个偏序集中一条反链的最多元素数。
- \*32. 证明: 在任意一组  $mn+1$  个人中要么存在  $m+1$  个人的一个列表, 其中每个人(除了表的第一个人以外)都是表中前一个人的后代; 要么存在  $n+1$  个人, 其中没有一个人是其他  $n$  个人中任何一个人的后代。[提示: 用练习 31.]



假设  $(S, \leq)$  是良基的偏序集。良基归纳原理说,

如果  $\forall x (\forall y (y < x \rightarrow P(y)) \rightarrow P(x))$ , 那么对所有的  $x \in S$ , 有  $P(x)$  为真。

33. 证明: 对于良基归纳原理不需要另外的基础情况, 就是说, 如果  $\forall x (\forall y (y < x \rightarrow P(y)) \rightarrow P(x))$ ,  $P(u)$  对于  $S$  中所有的极小元素  $u$  为真。
- \*34. 证明: 良基归纳原理是有效的。

集合  $A$  上的关系  $R$  叫做  $A$  上的近似序, 如果  $R$  是自反的和传递的。

35. 设  $R$  是从  $\mathbb{Z}^+$  到  $\mathbb{Z}^+$  的所有函数的集合上的关系, 使得  $(f, g)$  属于  $R$ , 当且仅当  $f$  是  $O(g)$ , 证明:  $R$  是近似序。
36. 设  $R$  是  $A$  上的近似序, 证明:  $R \cap R^{-1}$  是等价关系。
- \*37. 设  $R$  是近似序, 设  $S$  是  $R \cap R^{-1}$  的等价类的集合上的关系,  $C$  和  $D$  是  $R$  的等价类,  $(C, D)$  属于  $S$ , 当且

仅当存在  $C$  的元素  $c$ ,  $D$  的元素  $d$ , 使得  $(c, d)$  属于  $R$ 。证明  $S$  是偏序。

设  $L$  是一个格。由  $x \wedge y = \text{glb}(x, y)$  和  $x \vee y = \text{lub}(x, y)$  定义交 ( $\wedge$ ) 和并 ( $\vee$ ) 运算。

38. 证明下面的性质对格  $L$  的一切元素  $x, y, z$  成立:

a)  $x \wedge y = y \wedge x, x \vee y = y \vee x$  (交换律)

b)  $(x \wedge y) \wedge z = x \wedge (y \wedge z), (x \vee y) \vee z = x \vee (y \vee z)$  (结合律)

c)  $x \wedge (x \vee y) = x, x \vee (x \wedge y) = x$  (吸收律)

d)  $x \wedge x = x, x \vee x = x$  (幂等律)

39. 证明: 如果  $x$  和  $y$  是格的元素, 那么  $x \vee y = y$  当且仅当  $x \wedge y = x$ 。

一个格  $L$  是有界的, 如果它有一个上界, 记作  $1$ , 即对所有的  $x \in L$  有  $x \leq 1$ ; 并且有一个下界, 记作  $0$ , 即对所有的  $x \in L$  有  $0 \leq x$ 。

40. 证明: 如果  $L$  是具有上界  $1$  和下界  $0$  的有界格, 那么对所有的元素  $x \in L$ , 具有下面的性质:

a)  $x \vee 1 = 1$

b)  $x \wedge 1 = x$

c)  $x \vee 0 = x$

d)  $x \wedge 0 = 0$

41. 证明: 每个有限格是有界的。

一个格叫做分配的, 如果对  $L$  中所有的  $x, y, z$ , 有

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \text{ 和 } x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

\*42. 给出一个不是分配格的例子。

43. 证明: 格  $(P(S), \subseteq)$  是分配格, 其中  $P(S)$  是有穷集  $S$  的幂集。

44. 格  $(\mathbb{Z}^+, |)$  是分配格吗?

有界格  $L$  的元素  $a$  关于上界  $1$  和下界  $0$  的补元是元素  $b$ , 使得  $a \vee b = 1$  和  $a \wedge b = 0$ 。如果一个格的每个元素都有补元, 那么这个格叫做有补格。

45. 给出一个有限格的例子, 其中至少 1 个元素有多于 1 个的补元且至少 1 个元素没有补元。

46. 证明: 格  $(P(S), \subseteq)$  是有补格, 其中  $P(S)$  是有穷集  $S$  的幂集。

\*47. 证明: 如果  $L$  是有限分配格, 那么  $L$  的一个元素至多有 1 个补元。

1.7 节例 12 中介绍的 Chomp 游戏可以推广到在任意最小元素为  $a$  的有限偏序集  $(S, \leq)$  上玩。在这个游戏中, 一个移动由选择  $S$  中的元素  $x$ , 及移动  $x$  和所有  $S$  中比  $x$  大的元素构成。失败者是被迫选择最小元素  $a$  的那个玩家。

48. 证明: 1.7 节例 12 所描述的 Chomp 游戏(曲奇饼摆放在  $m \times n$  的矩形网格中)与偏序集  $(S, |)$  上的 Chomp 游戏相同, 其中  $S$  是所有除  $p^{m-1}q^{n-1}$  的正整数集合, 这里  $p$  和  $q$  是两个不同的素数。

49. 证明: 如果  $(S, \leq)$  有最大元素  $b$ , 那么在偏序集上存在一个获胜策略。[提示: 推广 1.7 节例 12 中的讨论。]

## 计算机题目

按下述给定的输入和输出写出程序。

1. 给定表示有穷集上关系的矩阵, 确定这个关系是否是自反或反自反的。

2. 给定表示有穷集上关系的矩阵, 确定这个关系是否是对称或反对称的。

3. 给定表示有穷集上关系的矩阵, 确定这个关系是否是传递的。

4. 给定正整数  $n$ , 显示一个  $n$  元素集合上所有的关系。

\*5. 给定一个正整数  $n$ , 确定  $n$  元素集合上的传递关系的个数。

\*6. 给定一个正整数  $n$ , 确定  $n$  元素集合上的等价关系的个数。

\*7. 给定一个正整数  $n$ , 显示  $n$  个最小的正整数集合上的所有等价关系。

8. 给定一个  $n$  元关系, 当某些特定的字段被删除以后求这个关系的投影。

9. 给定一个  $m$  元关系、一个  $n$  元关系和一个公共字段的集合, 找出这些关系关于这些公共字段的连接。

10. 给定表示一个有穷集上关系的矩阵, 求表示这个关系自反闭包的矩阵。

11. 给定表示一个有穷集上关系的矩阵, 求表示这个关系对称闭包的矩阵。

12. 给定表示一个有穷集上关系的矩阵, 通过计算表示这个关系的矩阵的幂的联合求表示这个关系传递闭包的矩阵。



13. 给定表示一个有穷集上关系的矩阵, 使用沃舍尔算法求表示这个关系的传递闭包的矩阵。
14. 给定表示一个有穷集上关系的矩阵, 求表示包含这个关系的最小的等价关系的矩阵。
15. 给定一个有穷集上的偏序, 使用拓扑排序找出一个与它相容的全序。

## 计算和研究

使用一个计算程序或你已完成的程序做下面的练习。

1. 显示一个 4 元素集合上的所有不同的关系。
2. 显示一个 6 元素集合上的所有不同的自反和对称的关系。
3. 显示一个 5 元素集合上的所有不同的自反和传递的关系。
- \*4. 对所有的正整数  $n$ ,  $n \leq 7$ , 确定在  $n$  元素集合上有多少个传递关系。
5. 使用对应于某个特定的运输或通信网络的有向链路的关系, 或者使用一个随机生成的关系, 在至少 20 个元素的集合上求所用关系的传递闭包。
6. 对于所有不超过 20 的正整数  $n$ , 计算  $n$  元素集合上的不同的等价关系个数。
7. 显示 7 元素集合上的所有的等价关系。
- \*8. 显示 5 元素集合上的所有的偏序。
- \*9. 显示 5 元素集合上的所有的格。

## 写作题目

用本书以外的资料写成短文回答下列问题。

1. 讨论模糊关系的概念。怎样使用模糊关系?
2. 不限于 5.2 节所介绍的内容, 描述关系数据库的基本原理。关系数据库与其他类型的数据库相比, 使用面有多广?
3. 查找沃舍尔和罗伊(Roy)的原始论文(法文), 在那篇论文中他们提出了求传递闭包的算法。讨论他们的方法。为什么可以认为沃舍尔算法是被多人独立发现的?
4. 描述怎样用等价类把有理数定义为整数对的类, 以及遵照这种方法怎样定义有理数的基本算术运算。
5. 解释赫尔姆·哈塞怎样使用我们现在称为哈塞图的图示。
6. 描述在计算机操作系统中用来执行信息流策略的某些机制。
7. 讨论计划评审技术(Program Evaluation and Review Technique, PERT)在安排一个大的复杂项目的任务中的应用。PERT 的使用面有多广?
8. 讨论关键路径方法(Critical Path Method, CPM)对找出完成项目的最短时间的应用。CPM 的使用面有多广?
9. 讨论格中的对偶性的概念。解释怎样用对偶性建立新的结果?
10. 解释模格的意义。描述模格的某些性质, 描述模格是怎样在投影几何的研究中产生的。

## 图

图是由顶点和连接顶点的边构成的离散结构。几乎可以想到的每个学科中的问题都可以运用图模型来求解。我们将给出实例说明如何在各种领域中运用图来建模。例如，如何用图表示生态环境里不同物种的竞争，如何用图表示组织中谁影响谁，如何用图表示循环锦标赛的输出。我们将描述如何用图对人们之间的相识关系、研究人员之间的合作关系、电话号码间的呼叫关系以及网站之间的链接关系进行建模。我们将说明如何用图对地图和一个组织内员工的工作指派进行建模。

运用图模型，可以确定能不能遍历一个城市的所有街道而不在任一条街道上走两遍，还能找出对地图上的区域着色所需要的颜色数；可以用图来确定某一个电路是否能够在平面电路板上实现；用图区分两种有着同样的分子式但结构不同的两种化合物；我们能够运用计算机网络的图模型确定两台计算机是否由通信链路连接；对其边赋予了权重的图可以求解诸如在运输网络中寻找两个城市间的最短路径这类问题。我们还可以用图来安排考试和指定电视台的频道。本章将介绍图论的基本内容，还将给出许多不同的图模型。为了求解能够用图研究的多种问题，我们将介绍许多不同的图的算法，还将研究这些算法的复杂度。

## 6.1 图和图模型

首先给出图的定义。

**定义 1** 一个图  $G$  由顶点(或结点)的非空集  $V$  和边集  $E$  构成，每条边有一个或两个顶点与它相连，这样的顶点称为边的端点。边连接它的端点。

**注意** 图  $G$  的顶点集  $V$  可能是无限的。顶点集为无限集的图称为无限图，与之相对，顶点集为有限集的图称为有限图。在本书中，通常只考虑有限图。

现在假设一个网络由数据中心和计算机之间的通信网络组成。可以把每个数据中心的位置用一个点来表示，把每个通信链接用一段线来表示，如图 6-1 所示。

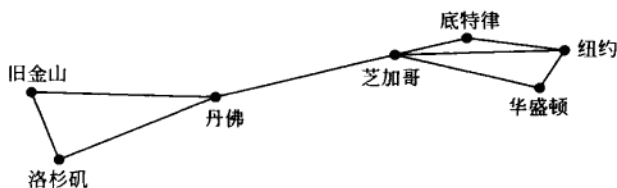


图 6-1 计算机网络

这个计算机网络可以用图来建模，图中的顶点表示数据中心，边表示通信链接。通常，用点表示顶点、用线段或者曲线表示边来可视化图。其中，表示边的线段的端点就是表示相应边的端点的点。当我们画图时，尽量不让它们的边相交。然而，并不是必须这样做，因为任意的用点表示顶点、任意形式的顶点间的连接表示边的描述方法均可使用。实际上，有些图不能够没有边相交地画在平面上(见 6.7 节)。关键一点是，只要正确地描述了顶点间的连接，画一个图的路线方式可以是任意的。

说明一下，表示这个计算机网络图的每条边都连接两个不同的顶点，即没有任何一个顶点连

接它自身,另外,也没有不同的两条边连接着相同的一对顶点。每条边都连接两个不同的顶点且没有不同的两条边连接着相同的一对顶点,这样的图称为简单图。注意,在简单图中,每条边与一对无序的顶点相关联,而且没有其他的边和这条边连接相同顶点。因此,当有一条简单图的边与 $\{u, v\}$ 相关联,也可以说 $\{u, v\}$ 是该图的一条边,这不会产生误解。

一个计算机网络可能在两个数据中心之间有多重链接,如图 6-2 所示。为这样的网络建模,需要有不只一条边连接同一对顶点的图。可能会有多重边连接同一对顶点的图称为多重图。当有 $m$ 条不同的边与相同的无序顶点对相关联,我们也说 $\{u, v\}$ 是一条多重度为 $m$ 的边。就是说,可以把这个边集考虑是边 $\{u, v\}$ 的 $m$ 个不同副本。

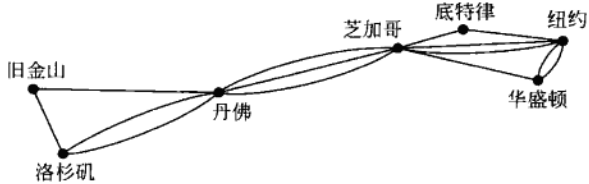


图 6-2 数据中心之间带有多重连接的计算机网络

有时候一个数据中心与它自己连接一条通信线,也许是一个为了诊断目的的反馈环。图 6-3 说明了这样的网络。为这个网络建模,需要包括把一个顶点连接到它自身的边的图。这样的边称为环。可能包含环、还有可能包含连接同一对顶点的多重边的图,称作伪图。

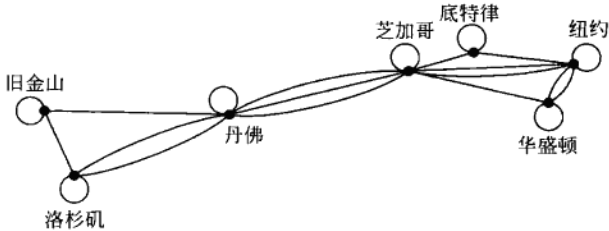


图 6-3 带诊断线的计算机网络

到目前为止,我们所介绍的图是无向图。它们的边也被认为是无向的。然而,要建立一个图模型,可能会发现有必要给这些边赋予方向。例如,在计算机网络中,一些连接可以只对一个方向操作(这种边称为单工线路)。这可能是这种情况,有大量的数据送到某些数据中心,但只有很少或者根本没有相反方向的数据传输。这样的网络如图 6-4 所示。

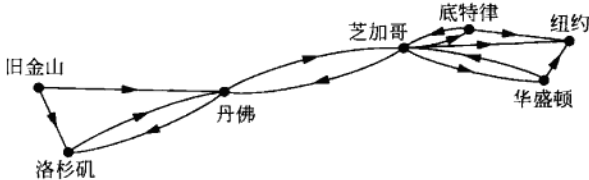


图 6-4 带有单向通信线的通信网络

我们使用有向图来为这样的计算机网络建模。有向图的每条边与一个有序对相关联。这里给有向图的定义比在第 5 章使用的更加广义,在第 5 章中使用有向图来表示关系。

**定义 2** 一个有向图 $(V, E)$ 由一个非空顶点集 $V$ 和一个有向边(或弧)集 $E$ 组成。每条有向边与一个顶点有序对相关联。认为与有序对 $(u, v)$ 相关联的有向边开始于 $u$ 、结束于 $v$ 。

当画线描述一个有向图时,我们用一个从 $u$ 指向 $v$ 的箭头来表示这条边的方向是开始于 $u$ 结束

于  $v$ 。一个有向图可能包含环，也有可能包含开始和结束于相同顶点的多重有向边。有向图也有可能包含连接  $u$  和  $v$  的两个方向上的有向边；就是说，当一个有向图含有从  $u$  到  $v$  的边，它也可能包含从  $v$  到  $u$  的一条或多条边。注意，当对一个无向图的每一条边都赋予方向后，就得到了一个有向图。当一个有向图不包含环和多重有向边时，就称为简单有向图。因为简单有向图在每一对有序顶点  $(u, v)$  最多连一条边。如果在图中它们只连一条边，则称为  $(u, v)$  边。

在某些计算机网络中，两个数据中心之间可能会有多重的通信链接，正如图 6-5 所示的那样。用可以包含从一个顶点指向第二个(也许是同一个)顶点的多重有向边的有向图来对这样的网络建模，我们称这样的图为有向多重图。当  $m$  条有向边中的每一条都与一对有序顶点对  $(u, v)$  相关联，我们就说  $(u, v)$  是一条多重度为  $m$  的边。

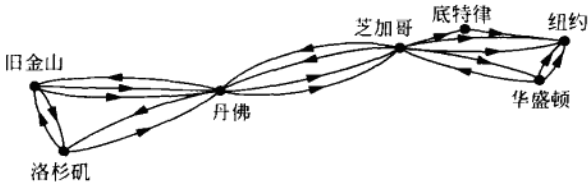


图 6-5 带多重单向线的计算机网络

对于某些模型，我们可能需要这样的图，其中有些边是无向的，而另一些边是有向的。有向边和无向边都有的图称为混合图。例如，可能会用一个混合图来给这样的计算机网络建模，该网络包含的链接有些可以双向运行，另外一些只能单向运行。

表 6-1 总结了各种图的专门术语。有时我们将用“图”作为一个通用的术语指代有向或无向的(或两者皆有)、有环或无环的，以及有多重边或无多重边的图。在其他情形下，当上下文清楚时，我们使用术语“图”只表示无向图。

表 6-1 图术语

| 类 型   | 边       | 允许多重边 | 允许环 |
|-------|---------|-------|-----|
| 简单图   | 无向      | 否     | 否   |
| 多重图   | 无向      | 是     | 否   |
| 伪图    | 无向      | 是     | 是   |
| 有向图   | 有向      | 否     | 是   |
| 有向多重图 | 有向      | 是     | 是   |
| 混合图   | 有向的和无向的 | 是     | 是   |

由于现代对图论的研究兴趣，还因为其在各个学科的广泛应用，图论中引入了许多不同的术语。不管什么时候遇到这些术语，读者应该注意它们的实际含义。数学家们用以描述图的术语已经逐步得到规范，但是讨论运用于其他学科的图术语仍然多种多样。

尽管描述图的术语可能区别很大，但是以下三个问题能够帮助我们理解图的结构：

- 图的边是有向的还是无向的(还是两者皆有)?
- 如果是无向图，是否存在连接相同顶点对的多重边? 如果是有向图，是否存在多重有向边?
- 是否存在环?

回答这些问题有助于我们理解图。而记住所使用的特定术语就不那么重要。

图模型

图可用在各种模型里。这里将介绍几种领域的图模型。本章的后续小节和后面几章将介绍其他模型。当建立一个图模型，需要确认已经正确回答了我们提出的关于图结构的三个关键问题。

**例1 生态学里栖息地重叠图** 图可用在涉及不同种类动物在一起活动的许多模型里。例如用栖息地重叠图来为生态系统里物种之间的竞争建模。用顶点表示每个物种。若两个物种竞争(即它们共享某些食物来源),则用无向边连接表示它们的顶点。栖息地重叠图是一个简单图,因为在此模型中不需要环和多重边。图6-6里的图表示森林生态系统。从这个图看出松鼠与浣熊竞争,但是乌鸦不与鼯鼠竞争。

**例2 熟人关系图** 可用图模型来表示人与人之间的各种关系。例如可用图来表示两个人是否互相认识,即他们是否是熟人。用顶点表示具体人群里的每个人。当两个人互相认识时,用无向边连接这两个人。不使用多重边,通常也不使用环。(如果我们想表达“自己认识自己”这层意思,就在图中包含环。)图6-7显示了一个小型熟人关系图。世界上所有人的熟人关系图有超过60亿个顶点和可能超过1万亿条边!在6.4节里将要进一步讨论这个图。

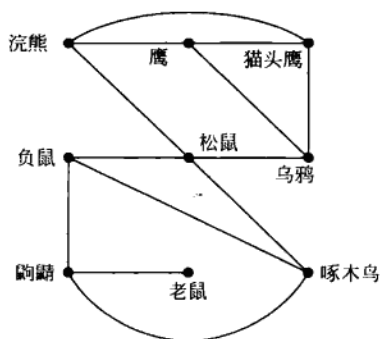


图 6-6 栖息地重叠图

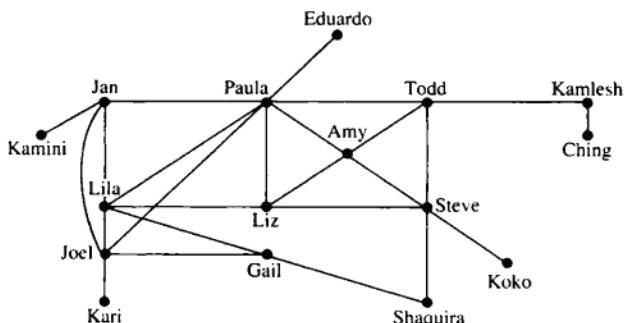


图 6-7 熟人关系图

**例3 影响图** 在对群体行为的研究里,可以观察到某些人能够影响其他人的思维。一种称为影响图的有向图可以用来为这样的行为建模。用顶点表示群体的每个人。当顶点 $a$ 所表示的人影响到顶点 $b$ 所表示的人时,就有从顶点 $a$ 到顶点 $b$ 的有向边。图6-8表示了群体成员的影响图的例子。在这个用影响图建模的群体里,Deborah影响Brain、Fred以及Linda,但是没有人影响Deborah。另外,Yvonne与Brain互相影响。

**例4 好莱坞图** 好莱坞图用顶点表示演员,并且当两个顶点所表示的演员共同出演一部电影时,就连接这两个顶点。这个图是简单图,因为它的边是无向边,不包含多重边和环。根据互联网电影数据库,在2006年1月,好莱坞图有637099个顶点和超过2000万条边,这些顶点所表示的演员出现在339896部电影中。稍后在6.4节里将要讨论好莱坞图。

**例5 循环赛** 每个队都与其他每队恰好比赛一次的联赛称为循环赛。用其中顶点表示每个队的有向图来为这样的比赛建模。注意若 $a$ 队击败 $b$ 队,则 $(a, b)$ 是边。该图是简单有向图,不包含环和多重有向边(因为没有任何两支队的比赛多于一次)。图6-9表示这样的有向图模型。注意在这个比赛里,队1无败绩而队3无胜绩。

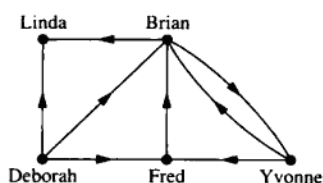


图 6-8 影响图

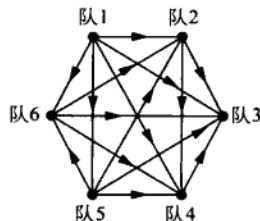


图 6-9 循环赛的图模型



**例 6 合作图** 合作图可用来为学术论文的合作者关系建模。在合作图中，顶点表示人（可能限制为某个学术圈子的成员），如果两个人合写论文，则用边连接这两个人。已经发现在数学研究论文上合作者的合作图有超过 400 000 个顶点和 675 000 条边。在 6.4 节里将对这个图补充说明。



**例 7 呼叫图** 可用来为网络(比如长途电话网)里的电话呼叫建模。具体地说，有向图用来为呼叫建模，其中用顶点表示每个电话号码，用有向边表示每次电话呼叫。表示呼叫的边是以发出呼叫的电话号码为起点，以接受呼叫的电话号码为终点。我们需要有向边，因为其表示的通话方向是有意义的。我们需要多重有向边，因为需要表示每一个从特定号码拨到第二个号码的通话。

一个小型的电话呼叫图如图 6-10a 所示，它表示 7 个电话号码。例如，这个图表示从 732-555-1234 到 732-555-9876 有 3 次呼叫和 2 次反向呼叫，但是从 732-555-4444 到其余 6 个号码(732-555-0011 除外)没有呼叫。当只关心是否有连接两个电话号码的呼叫时，就使用无向图，其中当两个号码之间有呼叫时，就用边连接这两个电话号码。这个版本的呼叫图如图 6-10b 所示。

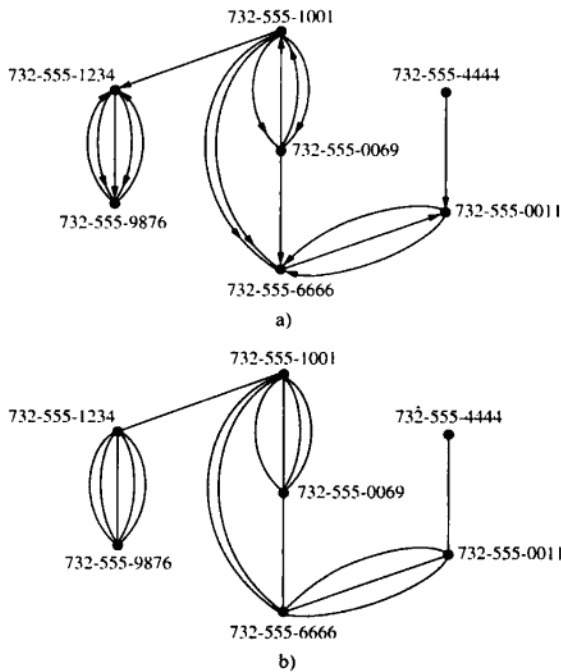


图 6-10 呼叫图

表示实际呼叫活动的呼叫图可以非常大。例如，在 AT&T 研究的一个呼叫图里，有大约 30 亿 9 千万个顶点和 40 亿条边，这个图表示在 20 天里进行的呼叫。在 6.4 节里将要进一步讨论呼叫图。



**例 8 网络图** 万维网可用有向图来建模，其中用顶点表示每个网页，并且若有从网页  $a$  指向网页  $b$  的链接，则有以  $a$  为起点以  $b$  为终点的边。因为几乎每秒钟都有新页面在网络上某处产生并且有其他页面被删除，所以网络图几乎是连续变化的。目前网络图有超过 30 亿个顶点和 200 亿条边。许多人正在研究网络图的性质，以便更好地理解网络的特性。6.4 节将继续讲解网络图，第 7 章将要解释网络爬虫(搜索引擎用它来产生网页的索引)是如何利用网络图的。

**例9 优先图与并发处理** 通过并发地执行某些语句, 计算机程序可以执行得更快。重要的是避免语句执行时还要用到尚未执行语句的结果。语句与前面语句的相关性可以表示成有向图。用顶点表示每个语句, 若在执行完第一个顶点所表示的语句之前不能执行第二个顶点所表示的语句, 则从第一个顶点到第二个顶点有一条边。这样的图称为优先图。图 6-11 显示计算机程序和优先图。例如, 该图说明在执行语句  $S_1$ 、 $S_2$  和  $S_4$  之前不能执行语句  $S_5$ 。

**例10 路线图** 可以用图对路线图建模。在这样的模型中, 定点表示交叉点, 而边表示路, 无向边表示双向道路, 有向边表示单行道; 多重无向边表示连接相同交叉点的多条双向道路, 多重有向边表示从一个交叉点开始、到第二个交叉点结束的多条单行道; 环表示环形路。因此, 如果一个路线图中只有双向道路, 没有环形路, 没有两条路连接相同的一对交叉点, 那么这个路线图可以用一个简单无向图来表示。一个路线图中只有单行道, 没有环形路, 没有两条路起始于相同的交叉点且终止于相同的交叉点, 那么它可以用一个简单有向图来表示。描述包含单行道和双向道路的路线图则需要混合图。

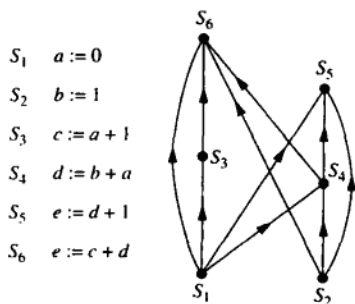
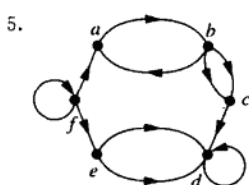
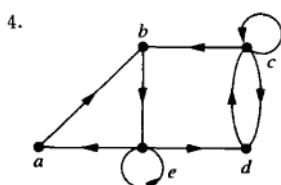
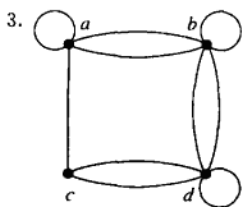
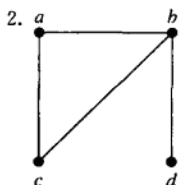


图 6-11 优先图

### 练习

- 画出表示航空公司航线的图模型, 并说出所用图的类型, 其中每天有 4 个航班从波士顿到纽华克, 2 个航班从纽华克到波士顿, 3 个航班从纽华克到迈阿密, 2 个航班从迈阿密到纽华克, 1 个航班从纽华克到底特律, 2 个航班从底特律到纽华克, 3 个航班从纽华克到华盛顿, 2 个航班从华盛顿到纽华克, 1 个航班从华盛顿到迈阿密, 其中
  - 若城市之间有航班(任何方向), 则在表示城市的顶点之间有边。
  - 对城市之间每个航班(任何方向)来说, 在表示城市的顶点之间有边。
  - 对城市之间每个航班(任何方向)来说, 在表示城市的顶点之间有边, 以及有表示在迈阿密起飞和降落特殊观光旅行的环。
  - 从表示航班出发城市的顶点到表示航班终止城市的顶点之间有边。
  - 对每个航班, 从表示出发城市的顶点到表示终止城市的顶点之间有边。

试确定练习 2~5 中所示的各个图是有向边还是无向边, 是否有多重边, 是否有一个或多个环。用你的答案指出该图属于表 6-1 中的哪种图。



- 设  $G$  是一个简单图。  $R$  是  $G$  的节点集上的关系,  $u R v$  当且仅当  $G$  中有与  $\{u, v\}$  相关联的边。证明: 关系  $R$  是对称的和非自反的关系。
- 集合  $A_1, A_2, \dots, A_n$  的交图对每个集合来说都有一个顶点, 若两个集合有非空交集, 则有一条边连接代表这两个集合的顶点。构造下列集合的交图。
  - $A_1 = \{0, 2, 4, 6, 8\}, A_2 = \{0, 1, 2, 3, 4\}$   
 $A_3 = \{1, 3, 5, 7, 9\}, A_4 = \{5, 6, 7, 8, 9\}$   
 $A_5 = \{0, 1, 8, 9\}$
  - $A_1 = \{\dots, -4, -3, -2, -1, 0\}$   
 $A_2 = \{\dots, -2, -1, 0, 1, 2, \dots\}$   
 $A_3 = \{\dots, -6, -4, -2, 0, 2, 4, 6, \dots\}$

$$A_4 = \{\dots, -5, -3, -1, 1, 3, 5, \dots\}$$

$$A_5 = \{\dots, -6, -3, 0, 3, 6, \dots\}$$

$$c) A_1 = \{x \mid x < 0\}, A_2 = \{x \mid -1 < x < 0\}$$

$$A_3 = \{x \mid 0 < x < 1\}$$

$$A_4 = \{x \mid -1 < x < 1\}$$

$$A_5 = \{x \mid x > -1\}$$

$$A_6 = \mathbf{R}$$

8. 构造6种鸟的栖息地重叠图, 其中隐士鸫与旅鸫以及蓝松鸦竞争, 旅鸫也与嘲鸫竞争, 嘲鸫也与蓝松鸦竞争, 以及鹤鸟与多毛啄木鸟竞争。
9. 可用图来表示两个人是否生活在同一时代。画出这样的图来表示本书前4章里有生平介绍的在1900年前去世的属于同一时代的数学家和计算机科学家。(如果在同一年里两个人都在世, 就假设他们生活在同一时代。)
10. 构造公司董事会成员的影响图, 主席影响技术总监、市场总监和营运总监; 技术总监影响营运总监; 市场总监影响营运总监; 无人影响首席财务官或受其影响。
11. 在循环赛里, 老虎队击败蓝松鸦队, 蓝松鸦队击败红衣主教队, 红衣主教队击败北美金莺队, 用有向图有这样的结果建模。
12. 解释如何用一月份和二月份的电话呼叫图来确定改变了电话号码的人的新电话号码。
13. 如何用表示网络里发送电子邮件的图来找出最近改变了原来电子邮件地址的人?
14. 描述一个图模型, 它表示在某个聚会上每个人是否知道另一个人的名字。图中的边应该是有向的还是无向的? 是否应该允许多重边? 是否应该允许环?
15. 描述一种表示传统婚姻的图模型。这个图有什么特殊性质?
16. 构造下列程序的优先图:
 
$$\begin{aligned} S_1: x &= 0 \\ S_2: x &= x+1 \\ S_3: y &= 2 \\ S_4: z &= y \\ S_5: x &= x+2 \\ S_6: y &= x+z \\ S_7: z &= 4 \end{aligned}$$
17. 描述一种用来为群体里成对个人之间关系建模的基于图的离散结构, 其中每个人可能喜欢或者不喜欢另一个人, 或者中立, 而反过来的关系可以是不同的。[提示: 给一个有向图添加结构。分别处理表示两个人的顶点之间的反向边。]

## 6.2 图的术语和几种特殊的图

### 6.2.1 引言



本节将介绍图论的一些基本词汇。当解决许多不同类型的问题时可以使用这些词汇。

其中一个这样的问题涉及判定能否把图画在平面里, 使得没有两条边是交叉的。另一个例子是判定两个图是否具有顶点之间的一一对应, 使得这样的对应能够产生边之间的一一对应。我们还将介绍在例子和模型里经常用到的几种重要的图族。在这些特殊类型的图出现的地方, 将会介绍几种重要的应用。

### 6.2.2 基本术语

首先给出描述无向图的顶点和边的一些术语。

**定义1** 若  $u$  和  $v$  是无向图  $G$  中的一条边的端点, 则称两个顶点  $u$  和  $v$  称为在  $G$  里邻接(或相邻)。若  $e$  和  $\{u, v\}$  相联系, 则边  $e$  称为关联顶点  $u$  和  $v$ 。也可以说边  $e$  连接  $u$  和  $v$ 。顶点  $u$  和  $v$  称为与  $\{u, v\}$  相联系的边的端点。

为了反映出有多少条边关联着一个顶点, 有下述的定义。

**定义2** 在无向图里顶点的度是与该顶点关联的边的数目, 例外的情形是, 顶点上的环为顶点的度做出双倍贡献。顶点  $v$  的度表示成  $\deg(v)$ 。



例1 如图6-12所示, 图 $G$ 和图 $H$ 的顶点的度是什么?

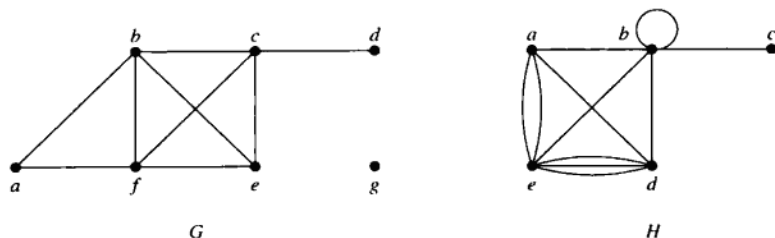


图6-12 无向图 $G$ 和 $H$

解 在 $G$ 里,  $\deg(a)=2$ ,  $\deg(b)=\deg(c)=\deg(f)=4$ ,  $\deg(d)=1$ ,  $\deg(e)=3$ ,  $\deg(g)=0$ .  
在 $H$ 里,  $\deg(a)=4$ ,  $\deg(b)=\deg(e)=6$ ,  $\deg(c)=1$ ,  $\deg(d)=5$ . ■

把度为0的顶点称为孤立的. 因此孤立点不与任何顶点相邻. 例1里图 $G$ 的顶点 $g$ 是孤立的. 顶点是悬挂的, 当且仅当它的度是1. 因此悬挂点恰与1个其他顶点相邻. 例1里图 $G$ 的顶点 $d$ 是悬挂的.

当对图 $G=(V, E)$ 的所有顶点的度求和时, 得出了什么? 每条边都为顶点的度之和贡献2, 因为一条边恰好关联2个(可能相同)顶点. 这意味着顶点的度之和是边数的2倍. 下面描述握手定理, 这是因为在一条边上两个端点可以类比为一次握手涉及两只手这种情形.

分析一个图模型中顶点的度, 能够提供关于该模型的有用信息, 如例2所示.

例2 栖息地重叠图(6.1节例1中介绍的)中一个顶点的度表示什么意义? 该图中的哪些顶点是悬挂的, 哪些是孤立的? 运用6.1节图6-6中所示的栖息地重叠图解释你的答案.

解 栖息地重叠图中的两个顶点之间有边, 当且仅当这两个顶点所代表的两个物种相互竞争. 因此, 栖息地重叠图中的一个顶点的度表示了该生态系统里与此顶点代表的物种竞争的物种数目. 如果一个物种恰好与另一种物种竞争, 则相应的顶点是悬挂的. 最后, 如果某一物种不与其他任何物种竞争, 那么代表该物种的顶点就是孤立的.

例如, 6.1节图6-6中代表松鼠的顶点的度是4, 因为松鼠于其他4种物种竞争: 乌鸦、负鼠、浣熊和啄木鸟. 图6-6所示的图中, 老鼠是唯一的一个由悬挂顶点表示的物种, 因为老鼠只与鼯鼠竞争, 而其余的所有物种都至少与两种以上的他物种竞争. 如果一个物种仅与另一种物种竞争, 则表示它的顶点是悬挂的. 图6-6所示的图中没有孤立的顶点, 因为每种物种都至少与生态系统中的其他一种物种竞争.

**定理1 握手定理** 设 $G=(V, E)$ 是有 $e$ 条边的无向图, 则

$$2e = \sum_{v \in V} \deg(v)$$

(注意即使出现多重边和环, 这个式子也仍然成立.)

例3 一个具有10个顶点而且每个顶点的度都为6的图, 有多少条边?

解 因为顶点的度之和是 $6 \cdot 10 = 60$ , 所以 $2e = 60$ . 因此 $e = 30$ . ■

定理1说明无向图顶点的度之和是偶数. 这可以推导出许多结论, 其中一个结论作为定理2给出.

**定理2** 无向图有偶数个奇数度顶点.

证 在无向图 $G=(V, E)$ 里, 设 $V_1$ 和 $V_2$ 分别是偶数度顶点和奇数度顶点的集合. 于是

$$2e = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v)$$

因为对 $v \in V_1$ 来说,  $\deg(v)$ 是偶数, 所以上面等式右端的第一项是偶数. 另外, 上面等式右端的两项之和是偶数, 因为和是 $2e$ . 因此, 和里的第二项也是偶数. 因为在第二项的和里, 所有的项都是奇数, 所以必然有偶数个这样的项. 因此, 有偶数个奇数度顶点. □

对带有有向边的图来说也有一些有用的术语。

**定义 3** 当 $(u, v)$ 是带有有向边的图 $G$ 的边时, 说 $u$ 邻接到 $v$ , 而且说 $v$ 从 $u$ 邻接。顶点 $u$ 称为 $(u, v)$ 的起点,  $v$ 称为 $(u, v)$ 的终点。环的起点和终点是相同的。

因为带有有向边的图的边是有序对, 所以这时顶点度的定义细化成反映这个顶点作为起点和作为终点的不同度数。

**定义 4** 在带有有向边的图里, 顶点 $v$ 的入度(表示成 $\deg^-(v)$ )是以 $v$ 作为终点的边数。顶点 $v$ 的出度(表示成 $\deg^+(v)$ )是以 $v$ 作为起点的边数(注意顶点上的环对这个顶点的入度和出度的贡献都是 1)。

**例 4** 求出图 6-13 所示带有有向边的图 $G$ 里每个顶点的入度和出度。

**解** 入度是:  $\deg^-(a)=2, \deg^-(b)=2, \deg^-(c)=3, \deg^-(d)=2, \deg^-(e)=3, \deg^-(f)=0$ 。出度是:  $\deg^+(a)=4, \deg^+(b)=1, \deg^+(c)=2, \deg^+(d)=2, \deg^+(e)=3, \deg^+(f)=0$ 。 ■

因为每条边都有一个起点和一个终点, 所以在带有有向边的图里, 所有顶点的入度之和等于出度之和。这两个和都等于图的边数。把这个结果表述成定理 3。

**定理 3** 设 $G=(V, E)$ 是带有有向边的图。于是

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|$$

带有有向边的图有许多性质是不依赖于边的方向的。因此忽略这些方向经常是有点用的。忽略边的方向后得出的无向图称为底无向图。带有有向边的图与它的底无向图有相同的边数。

6.2.3 一些特殊的简单图

下面要介绍几类简单图。这些图常常用作例子, 来自于许多应用。

**例 5 完全图**  $n$  个顶点的完全图(表示成 $K_n$ )是在每对不同顶点之间都恰有一条边的简单图。对 $n=1, 2, 3, 4, 5, 6$ 来说, 图 6-14 显示图 $K_n$ 。 ■

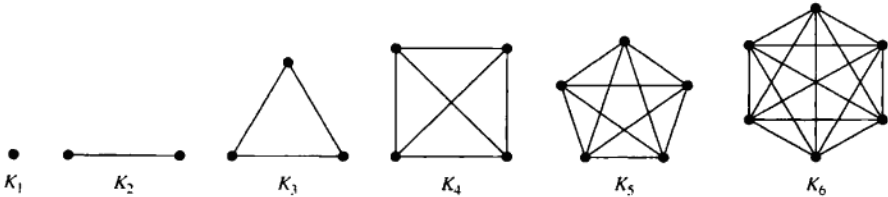


图 6-14 图 $K_n, n=1, 2, 3, 4, 5, 6$

**例 6 圈图** 圈图 $C_n (n \geq 3)$ 是由 $n$ 个顶点 $v_1, v_2, \dots, v_n$ 以及边 $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$ 组成的。图 6-15 显示圈图 $C_3, C_4, C_5, C_6$ 。 ■

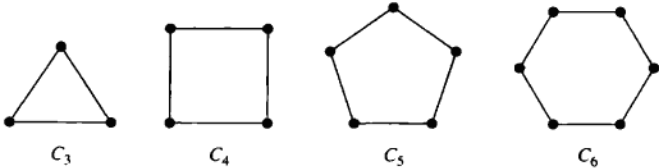


图 6-15 圈图 $C_3, C_4, C_5, C_6$

**例 7 轮图** 对 $n \geq 3$ 来说, 当给圈图 $C_n$ 添加另一个顶点, 而且把这个新顶点与 $C_n$ 里 $n$ 个顶

点逐个连接时,就得出轮图  $W_n$ 。图 6-16 显示轮图  $W_3, W_4, W_5, W_6$ 。

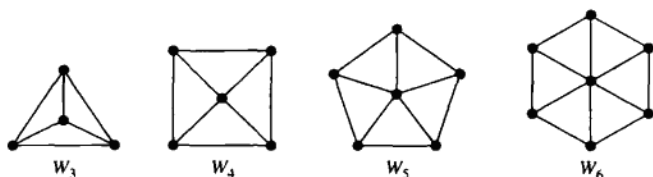


图 6-16 轮图  $W_3, W_4, W_5, W_6$

**例 8  $n$  立方体图**  $n$  立方体图(表示成  $Q_n$ )是用顶点表示  $2^n$  个长度为  $n$  的位串的图。两个顶点相邻,当且仅当它们所表示的位串恰相差一位。图 6-17 显示图  $Q_1, Q_2, Q_3$ 。注意可以从  $n$  立方体图  $Q_n$  来构造  $(n+1)$  立方体图  $Q_{n+1}$ ,方法是建立  $Q_n$  的两个副本,在  $Q_n$  的一个副本的顶点标记前加 0,在  $Q_n$  的另一个副本的顶点标记前加 1,并且加入连接那些标记只在第一位不同的两个顶点的边。在图 6-17 中,从  $Q_2$  来构造  $Q_3$ ,方法是画出  $Q_2$  的两个副本作为  $Q_3$  的顶面和底面,在底面每个顶点的标记前加 0,在顶面每个顶点的标记前加 1。(这里,“面”意为三维空间中立方体的一个面。试想在三维空间中画出以  $Q_2$  的两份拷贝作为立方体顶面和底面的  $Q_3$  的图形,然后在平面上画出结果描述方法(此  $Q_3$  图形——译者注)的投影。)

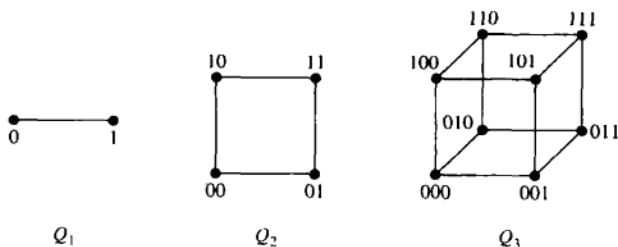


图 6-17 对于  $n=1, 2, 3$  的  $n$  立方体图  $Q_n$

#### 6.2.4 偶图

有时可以把图的顶点分成两个不相交的子集,使得每条边都连接一个子集里的顶点与另一个子集里的顶点。例如,考虑一下表示村民之间的婚姻关系的图,其中用顶点表示每个人,用边表示婚姻。在这个图里,每条边都连接表示男人的顶点子集里的顶点与表示女人的顶点子集里的顶点。这引出了定义 5。

**定义 5** 若把简单图  $G$  的顶点集分成两个不相交的非空集合  $V_1$  和  $V_2$ ,使得图里的每一条边都连接着  $V_1$  里的一个顶点与  $V_2$  里的一个顶点(因此  $G$  里没有边是连接着  $V_1$  里的两个顶点或  $V_2$  里的两个顶点),则  $G$  称为偶图<sup>⊖</sup>。当此条件成立,称  $(V_1, V_2)$  为  $G$  的顶点集的一个二部划分。

在例 9 里说明  $C_6$  是偶图,在例 10 里说明  $K_3$  不是偶图。

**例 9** 图 6-18 所示的  $C_6$  是偶图,因为把它的顶点集分成两个集合  $V_1 = \{v_1, v_3, v_5\}$  和  $V_2 = \{v_2, v_4, v_6\}$ ,  $C_6$  的每一条边都连接  $V_1$  里的一个顶点与  $V_2$  里的一个顶点。

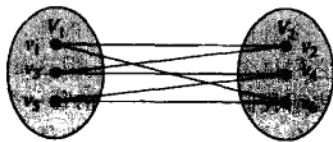


图 6-18  $C_6$  是偶图

**例 10**  $K_3$  不是偶图。为了看明白这一点,注意若把  $K_3$  的顶点集分成两个不相交的集合,则两个集合之一必然包含两个顶点。假如这个图是偶图,那么这两个顶点就不能用边连接,但是在  $K_3$  里每一个顶点都用边连接到其他每个顶点。

<sup>⊖</sup> 也称为二部图、二分图。——译者注

例 11 图 6-19 所示的图  $G$  和  $H$  是否为偶图?

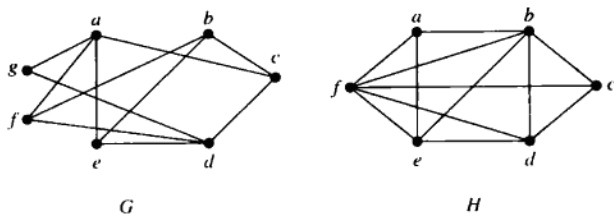


图 6-19 无向图  $G$  和  $H$

**解** 图  $G$  是偶图, 因为它的顶点集是两个不相交集  $\{a, b, d\}$  和  $\{c, e, f, g\}$  的并, 每条边都连接一个子集里的一个顶点与另一个子集里的一个顶点。(注意对偶图  $G$  来说, 不必让  $\{a, b, d\}$  里每一个顶点与  $\{c, e, f, g\}$  里每一个顶点都相邻。例如  $b$  与  $g$  就不相邻。)

图  $H$  不是偶图, 因为它的顶点集不能分成两个子集, 使得边都不连接同一个子集的两个顶点 (读者应当通过考虑顶点  $a, b, f$  来验证它)。

定理 4 给出了决定图是否为偶图的有用准则。

**定理 4** 一个简单图是偶图, 当且仅当能够对图中的每个顶点赋以两种不同的颜色, 而不让相邻接的顶点被赋以相同的颜色。

**证** 首先, 假设  $G=(V, E)$  是一个二分简单图。那么,  $V=V_1 \cup V_2$ , 其中  $V_1$  和  $V_2$  是不相交的顶点集且  $E$  中的每一条边都连接一个  $V_1$  中的顶点和一个  $V_2$  中的顶点。如果对  $V_1$  中的每个顶点赋予一种颜色而  $V_2$  中的顶点赋予第二种颜色, 那么相邻接的顶点就不会被赋予相同的颜色。

现在假设可以仅用两种颜色对图中的顶点着色, 以使得相邻接的顶点都被赋予不同的颜色。让  $V_1$  为一种颜色的顶点集,  $V_2$  为另一种颜色的顶点集, 则  $V_1$  和  $V_2$  不相交且  $V=V_1 \cup V_2$ 。此外, 每条边都连接一个  $V_1$  中的顶点和一个  $V_2$  中的顶点, 因为并无相邻接的顶点同在  $V_1$  中或同在  $V_2$  中。所以,  $G$  是偶图。

例 12 将说明如何用定理 4 判断一个图是不是偶图。

**例 12** 用定理 4 判断例 11 中的图是不是偶图。

**解** 首先考虑图  $G$ 。将图  $G$  中的每个顶点赋予两种颜色 (如红色和蓝色) 中的一种, 使得  $G$  中的每一条边都连接一个红色顶点和一个蓝色顶点。不失一般性, 我们先任意地赋予顶点  $a$  红色。然后, 必须对  $c, e, f$  和  $g$  顶点赋予蓝色, 因为这些顶点与顶点  $a$  相邻接。为了避免一条边有两个蓝色的端点, 所有与  $c, e, f$  或  $g$  顶点相邻接的顶点必须被染成红色。这就是说, 必须把  $b$  和  $d$  染成红色 (也意味着,  $a$  必须染成红色, 而  $a$  已经是红色的了)。现在, 已经将所有顶点都进行了染色,  $a, b$  和  $d$  为红色,  $c, e, f$  和  $g$  为蓝色。查看每一条边, 我们看见每条边都连接一个红色顶点和一个蓝色顶点。因此, 由定理 4, 图  $G$  是偶图。

接下来, 将对图  $H$  中的每个顶点赋予红色或蓝色, 以使  $H$  中的每一条边都连接一个红色顶点和一个蓝色顶点。不失一般性, 我们任意地对  $a$  赋以红色。然后, 必须对  $b, e$  和  $f$  赋以蓝色, 因为它们每个都与  $a$  邻接。但这是不可能的, 因为  $e$  和  $f$  相邻接, 因此不能两个都赋以蓝色。这一矛盾表明我们不能对  $H$  中的每一个顶点赋以两种颜色中的一种, 以使没有邻接的顶点被赋以相同的颜色。根据定理 4,  $H$  不是偶图。

定理 4 是图论中“图的着色”部分的一个结果示例。图的着色是图论中一个重要的部分, 有着许多重要的应用。我们将在 6.8 节进一步学习图的着色。

决定一个图是否是偶图的另一个标准是基于路径的概念, 将在 6.4 节学习这个概念。一个图是偶图, 当且仅当从一个顶点出发经过奇数条不同的边再回到它本身是不可能的。当我们在 6.4 节讨论图中的路径和环路时, 将会让这个概念变得更加精确 (参见那一节的练习 26)。

**例 13 完全偶图** 完全偶图  $K_{m,n}$  是顶点集分成分别含有  $m$  和  $n$  个顶点的两个子集的图。两个顶点之间有边当且仅当一个顶点属于第一个子集而另一个顶点属于第二个子集。图 6-20 显示完全偶图  $K_{2,3}$ ,  $K_{3,3}$ ,  $K_{3,5}$  以及  $K_{2,6}$ 。

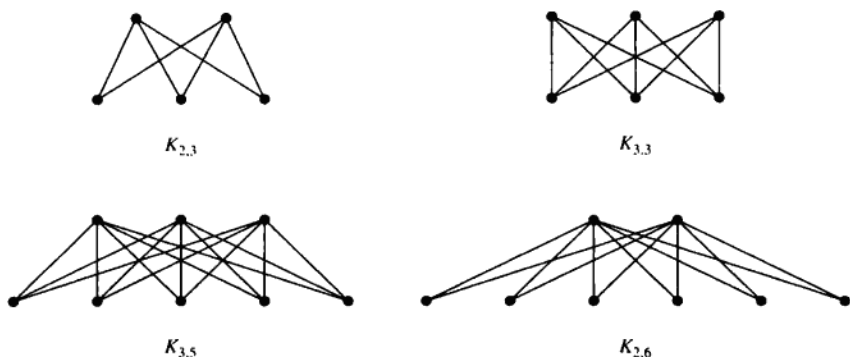


图 6-20 部分完全偶图

### 6.2.5 特殊类型的图的一些应用

以下例 14~16 将要说明如何把特殊类型的图用在数据通信和并行处理的模型里。

**例 14 任务分配** 设一个小组内有  $m$  名雇员，有  $j$  个不同的任务需要完成，其中  $m \leq j$ 。每个员工都受过相关训练，可以做这  $j$  个任务中的一个或多个。我们可以用一个图来为员工的能力建模。每个员工和每个任务都用一个顶点表示。对于每个员工，从表示该员工的顶点到表示该员工受训能做的所有任务，都作一条边。注意到这个图的顶点集可以被划分成两个不相交的集合，一个代表员工的顶点集，另一个代表任务的顶点集。而且每一条边都连接一个表示员工的顶点和一个表示任务的顶点。因此，该图是一个偶图。

例如，设一个小组由 4 名员工：Alvarez, Berkowitz, Chen 和 Davis；再设完成一个项目有 4 种工作要做：需求，架构，实现和测试。设 Alvarez 受训可以完成需求分析和测试；Berkowitz 接受的训练是完成架构、实现和测试；Chen 接受的训练是完成需求、架构和实现；而 Davis 只接受训练做需求。可以用图 6-21 所示的偶图对员工的这些能力建模。

为了完成项目，必须给员工指派任务，以满足每个任务都有一个员工接手，而且每个员工最多只分配一个任务。在这种情况下，可以让 Alvarez 做测试，Berkowitz 做实现，Chen 做架构，Davis 做需求。如图 6-21 所示（带颜色的线表示了任务的分配）。

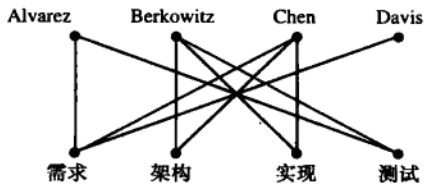


图 6-21 为受训员工分配工作建模

寻求一种对员工的任务分配方案可以看成是在图模型中寻找一个匹配。简单图的匹配是该图边集的一个子集，该子集中没有两条边与同一个顶点相关联。最大匹

配是所含边数最多的匹配。换句话说，匹配是边集的子集使得如果  $\{s, t\}$  和  $\{u, v\}$  是匹配中的边，那么  $s, t, u$  和  $v$  就是不同的顶点。在给职员分配任务的过程中，要给多数数的职员都分配，实际上就是寻找员工能力的图模型中的最大匹配。（感兴趣的读者可以在关于图论的书中找到更多有关匹配的内容，这些书中包括 [GrYe06]。）



**例 15 局域网** 在一座大楼里，像小型计算机和个人电脑这样的各种计算机，以及像打印机和绘图仪这样的外设，都可以用局域网来连接。一些这样的网络是基于星形拓扑，其中所有设备都连接到中央控制设备。局域网用图 6-22a 所示的完全偶图  $K_{1,n}$  来表示。通过中

央控制设备在设备间传输信息。

其他局域网是基于环形拓扑, 其中每个设备恰恰连接到两个其他设备。带环形拓扑的局域网用图 6-22b 所示的  $n$  圈图  $C_n$  来建模。消息围绕着圈从设备送到设备, 直到抵达消息目的地为止。

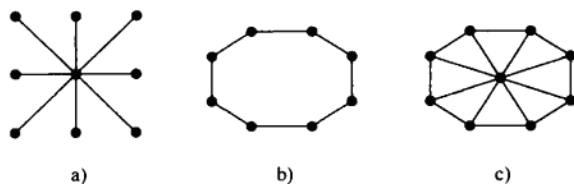


图 6-22 局域网的星形、环形以及混合拓扑

最后, 一些局域网采用这两种拓扑的混合形式。消息是围绕着环或通过中央设备来传送。这样的冗余使得网络更加可靠。带冗余的局域网用图 6-22c 所示的轮图  $W_n$  来建模。

**例 16 并行计算的互连网络** 直到最近, 计算机执行程序还是一次完成一个操作。因此, 为解决问题而写的算法都设计成一次执行一步; 这样的算法称为串行的(几乎所有本书描述的算法都是串行的)。不过, 像气象模拟、医学图像分析以及密码分析这样的许多高强度计算问题, 即使在超级计算机上, 也不能通过串行操作在合理的时间范围内解决。另外, 计算机执行基本操作的速度, 还存在物理的限制, 所以总是有问题不能用串行操作在合理的时间范围内解决。

并行处理利用包含许多自带内存的单处理器的计算机来帮助克服串行计算机的局限性。并行算法把问题分成可并发解决的若干子问题, 于是可以设计并行算法, 用带有多处理器的计算机来快速解决问题。在并行算法里, 单个指令流控制着算法的执行, 包括把子问题送到不同的处理器, 以及把子问题的输入和输出定向到适当的处理器。

采用并行处理时, 一个处理器需要另一个处理器产生的输出。因此处理器需要互联。可用适当类型的图来表示带有多重处理器的计算机里处理器的互连网络。在以下讨论里将要描述最常用类型的并行处理器互连网络。用来实现具体并行算法的互连网络的类型, 差异在于处理器之间交换数据的需求量和所需要的速度, 当然还有可用的硬件等。

最简单却又最昂贵的网络互联处理器, 包含每对处理器之间的双向连接。当有  $n$  个处理器时, 这样的网络表示成  $n$  个顶点上的完全图  $K_n$ 。不过, 这种类型的互连网络有严重的问题, 它所需要的连接数太大。实际上, 处理器的直接连接数目是有限的, 所以当处理器数很大时, 处理器不能直接连接到所有其他处理器。例如, 当有 64 个处理器时, 就需要  $C(64, 2) = 2016$  个连接, 每个处理器都得直接连接到其他 63 个处理器。

另一方面, 互联  $n$  个处理器的最简单方式或许是使用称为线性阵列的排列方式。除  $P_1$  和  $P_n$  外的每个处理器  $P_i$  都通过双向连接来连接到相邻处理器  $P_{i-1}$  和  $P_{i+1}$ 。  $P_1$  只连接  $P_2$ ,  $P_n$  只连接  $P_{n-1}$ 。图 6-23 显示 6 处理器线性阵列。这种方式的缺点是为了让处理器共享信息, 有时需要使用大量的称为 **hop** 的中间连接。

栅格网络(或二维阵列)是通用的互连网络。在这样的网络里, 处理器个数是一个完全平方数, 比方说  $n = m^2$ 。  $n$  个处理器标记成  $P(i, j)$ ,  $0 \leq i \leq m-1$ ,  $0 \leq j \leq m-1$ 。双向连接把处理器  $P(i, j)$  连接到它的 4 个相邻处理器  $P(i \pm 1, j)$  和  $P(i, j \pm 1)$ , 只要这些处理器是在栅格里。(注意, 栅格角上的 4 个处理器只有 2 个相邻处理器, 边界上其他处理器只有 3 个相邻处理器; 有时也用每个处理器恰有 4 个连接的变种的栅格网络。)栅格网络限制每个处理器的连接数。某些成对处理器之间的通信需要  $O(\sqrt{n}) = O(m)$  个中间连接(见本节末的练习 33)。图 6-24 表示 16 个处理器的栅格网络的图。



图 6-23 6 个处理器的线性阵列

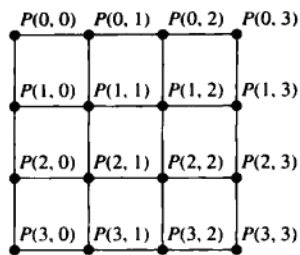


图 6-24 16 个处理器的栅格网络

超立方体是互连网络的一个重要类型。对这样的网络来说，处理器数是 2 的幂， $n=2^m$ 。 $n$  个处理器标记成  $P_0, P_1, \dots, P_{n-1}$ 。每个处理器都有到  $m$  个其他处理器的双向连接。连接到处理器  $P_i$  上的处理器，其下标的二进制表示与  $i$  的二进制表示恰恰相差 1 位。超立方体网络在每个处理器的直接连接数与保证处理器通信的中间连接数之间取得了平衡。已经用超立方体网络建造了许多计算机，而且用超立方体网络设计了许多算法。 $n$  立方体图  $Q_n$  表示带  $n$  个处理器的超立方体网络。图 6-25 显示 8 个处理器的超立方体网络(图 6-25 中画  $Q_3$  的方式与图 6-17 所示的方式不同)。

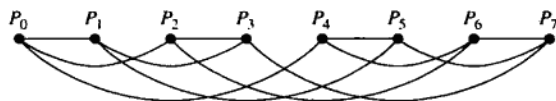


图 6-25 8 个处理器的超立方体网络

### 6.2.6 从旧图到新图

有时解决问题只需要图的一部分。例如，只关心大型计算机网络中涉及纽约、丹佛、底特律以及亚特兰大的计算机中心的那一部分。其他的计算机中心以及不连接到这 4 个具体计算机中心里任何 2 个的所有电话线可以忽略。在大网络的图模型里，可以删除除这 4 处外的计算机中心所对应的顶点，可以删除所有与所删除顶点关联的边。删除后所剩下的图称为原图的子图。

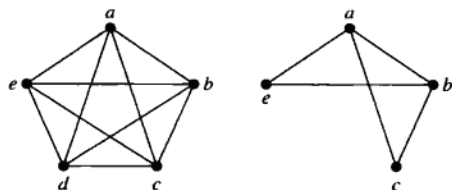
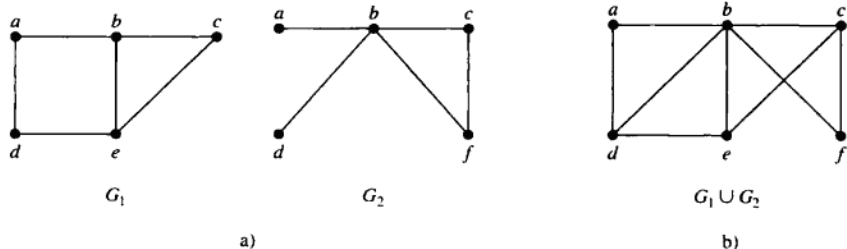
**定义 6** 图  $G=(V, E)$  的子图是图  $H=(W, F)$ ，其中  $W \subseteq V$  而且  $F \subseteq E$ 。

**例 17** 图 6-26 所示的图  $G$  是  $K_5$  的子图。

可以用各种方式组合两个或更多的图。把这些图的所有顶点和边所组成的图称为这些图的并图。两个简单图的并图的定义如下。

**定义 7** 两个简单图  $G_1=(V_1, E_1)$  和  $G_2=(V_2, E_2)$  的并图是带有顶点集  $V_1 \cup V_2$  和边集  $E_1 \cup E_2$  的简单图。 $G_1$  和  $G_2$  的并图表示成  $G_1 \cup G_2$ 。

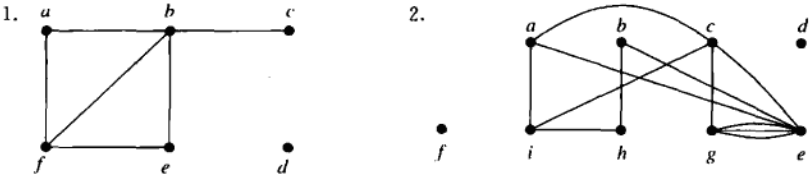
**例 18** 求图 6-27a 所示的图  $G_1$  和  $G_2$  的并图。

图 6-26  $K_5$  的子图图 6-27 a) 简单图  $G_1$  和  $G_2$ ；b) 它们的并  $G_1 \cup G_2$

解 并图  $G_1 \cup G_2$  的顶点集是两个顶点集的并, 即  $\{a, b, c, d, e, f\}$ 。并图的边集是两个边集的并。并图显示在图 6-27b 中。

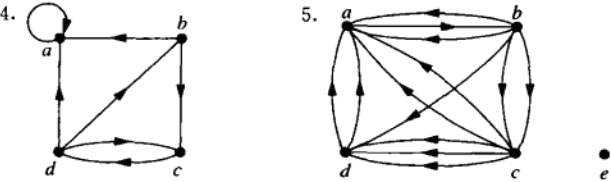
练习

在练习 1~2 里, 求所给无向图的顶点数、边数以及每个顶点的度。指出所有孤立点和悬挂点。



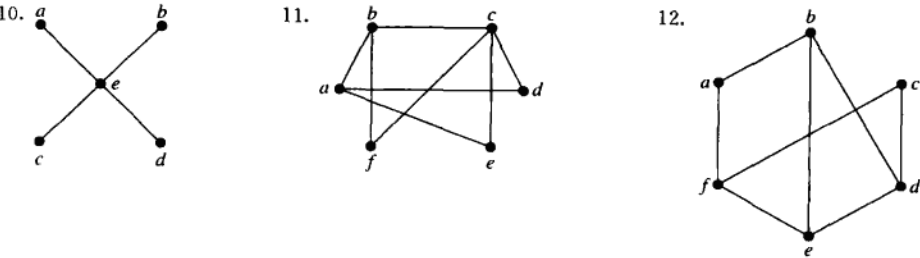
3. 带有 15 个顶点而且每个顶点的度都为 5 的简单图能否存在?

在练习 4~5 里, 对给定的有向多重图来说, 确定顶点数和边数, 并求出每个顶点的入度和出度。



6. 构造出图 6-13 里带有向边的图的底图。
7. 在合作图里, 顶点的度表示什么? 孤立点和悬挂点表示什么?
8. 在 6.1 节例 7 所描述的电话呼叫图里, 顶点的入度和出度表示什么? 在这个图的无向版本里, 顶点的度表示什么?
9. 在为循环赛建模的有向图里, 顶点的入度和出度表示什么?

在练习 10~12 里, 确定图是否为偶图。



13. 设一个新公司有 5 名员工: Zamora、Agraharam、Smith、Chou 和 Macintyre。每个员工承担下列 6 种责任之一: 策划, 宣传, 销售, 市场, 开发, 以及工业关系。每个员工能够做这些工作中的一种或多种: Zamora 能做策划、销售、市场或工业关系; Agraharam 能做策划或开发; Smith 能做宣传、销售或工业关系; Chou 能做策划、销售或工业关系; Macintyre 能做策划、宣传、销售或工业关系。
- a) 用一个偶图对这些员工的能力建模。
- b) 找出一个能力的指派, 使得每个员工被指定一种能力。
14. 下列图有多少个顶点和多少条边?
- a)  $K_n$                       b)  $C_n$                       c)  $W_n$
- d)  $K_{m,n}$                       e)  $Q_n$

一个图的度序列是由该图的各节点的度按非递增顺序排列的序列。例如, 本节例 1 中图  $G$  的度序列就是 4, 4, 4, 3, 2, 1, 0。

15. 求下列各个图的度序列。



- a)  $K_4$                       b)  $C_4$                       c)  $W_4$   
d)  $K_{2,3}$                     e)  $Q_3$

16. 图  $K_n$  的度序列是什么(其中  $n$  是正整数)? 并解释你的答案。  
17. 若图的顶点有度 5, 2, 2, 2, 2, 1, 则它有多少条边? 画出这样的图。

如果一个序列是一个简单图的度序列, 那么该序列是成图的。

18. 判断下列序列是否是成图的。如果是, 请画出一个简单图使其具有给定的度序列。

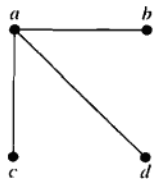
- a) 3, 3, 3, 3, 2                      b) 5, 4, 3, 2, 1  
c) 4, 4, 3, 2, 1                      d) 4, 4, 3, 3, 3  
e) 3, 2, 2, 1, 0                      f) 1, 1, 1, 1, 1

19. 证明: 一个由非负整数非递增排列的序列  $d_1, d_2, \dots, d_n$  是成图序列当且仅当把序列  $d_2 - 1, \dots, d_{d_1 - 1} - 1, d_{d_1 - 2}, \dots, d_n$  中的元素重新排序为非递增而得到的序列是成图序列。

20. 证明: 每个元素之和为偶数的非递增非负整数序列都是某个伪图的度序列。伪图是允许有环的无向图。  
[提示: 首先通过给每个顶点添加尽可能多的环来构造一个图, 然后添加一些边连接度为奇数的顶点。解释为什么这种构造方法能够证明此问题。]

21. 至少带有 1 个顶点的  $K_3$  的子图有多少个?

22. 画出右面这个图的所有子图。



若简单图每个顶点的度都相等, 则这个图称为正则的。若正则图每个顶点的度都为  $n$ , 则这个图称为  $n$  正则的。

23. 对哪些  $n$  值来说, 下列图是正则图?

- a)  $K_n$                       b)  $C_n$                       c)  $W_n$                       d)  $Q_n$

24. 度都为 4 而且带有 10 条边的正则图有多少个顶点?

25. 求右面的简单图对的并图(假设带有相同端点的边是相同的)。

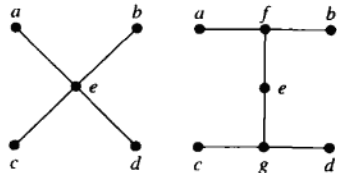
26. 简单图  $G$  的补图  $\bar{G}$  与  $G$  有相同的顶点。两个顶点在  $\bar{G}$  里相邻, 当且仅当它们在  $G$  里不相邻。求下列图。

- a)  $\bar{K_n}$                       b)  $\bar{K_{m,n}}$                       c)  $\bar{C_n}$                       d)  $\bar{Q_n}$

27. 若简单图  $G$  有  $v$  个顶点和  $e$  条边, 则  $\bar{G}$  有多少条边?

28. 简单图  $G$  的度序列为  $d_1, d_2, \dots, d_n$ 。求  $\bar{G}$  的度序列。

29. 证明: 若  $G$  是有  $n$  个顶点的简单图, 则  $G$  和  $\bar{G}$  的并图是  $K_n$ 。



表示成  $G$  的有向图  $G=(V, E)$  的逆图是图  $(V, F)$ , 其中  $(u, v) \in F$ , 当且仅当  $(v, u) \in E$ 。

30. 描述 6.1 节练习 4~5 里每个图的逆图。

31. 证明: 图  $G$  是它自身的逆图, 当且仅当  $G$  所关联的关系(参见 8.3 节)是对称的。

32. 画出 9 个并行处理器互联的栅格网络。

33. 证明: 在  $n=m^2$  个处理器的栅格网络里, 用  $O(\sqrt{n})=O(m)$  个 hop 就能让每一对处理器互相通信。

## 6.3 图的表示和图的同构

### 6.3.1 引言

图的表示方式有很多种。在本章里将看到选择最方便的表示会有助于对图的处理。本节将要说明如何用多种不同的方式来表示图。

有时两个图在这样一种意义下恰好具有相同的形式, 就是在两个图的顶点之间存在着一一对应, 这个对应保持边的对应关系。在这种情形下就说两个图是同构的。确定两个图是否同构, 这是本节里将要研究的重要图论问题。

### 6.3.2 图的表示

表示不带多重边的图的一种方式列出这个图的所有边。另一种表示不带多重边的图的方式是用邻接表, 它规定与图的每个顶点相邻的顶点。

例 1 用邻接表描述图 6-28 所示的简单图。

解 表 6-2 列出与图的每个顶点相邻的顶点。

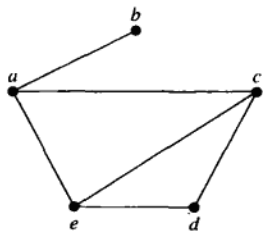


图 6-28 简单图

表 6-2 简单图的边表

| 顶 点 | 相 邻 顶 点 |
|-----|---------|
| a   | b, c, e |
| b   | a       |
| c   | a, d, e |
| d   | c, e    |
| e   | a, c, d |

例 2 通过列举在图的每个顶点上发出的边的终点，表示图 6-29 所示的有向图。

解 表 6-3 表示图 6-29 所示的有向图。

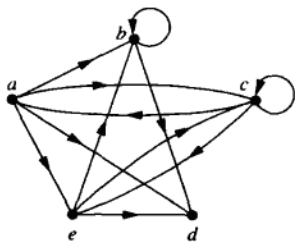


图 6-29 一个有向图

表 6-3 有向图的边表

| 起 点 | 终 点        |
|-----|------------|
| a   | b, c, d, e |
| b   | b, d       |
| c   | a, c, e    |
| d   | a, c, e    |
| e   | b, c, d    |

6.3.3 邻接矩阵

若图里有许多边，则把图表示成边表或邻接表，会不便于执行图的算法。为了简化计算，可用矩阵表示图。在此将给出两种类型的常用的表示图的矩阵。一种类型是基于顶点的相邻关系，另一种类型是基于顶点与边的关联关系。

假设  $G=(V, E)$  是简单图，其中  $|V|=n$ 。假设把  $G$  的顶点任意地排列成  $v_1, v_2, \dots, v_n$ 。对这个顶点表来说， $G$  的邻接矩阵  $A$  (或  $A_G$ ) 是一个  $n \times n$  的 0-1 矩阵，它满足这样的性质：当  $v_i$  和  $v_j$  相邻时第  $(i, j)$  项是 1，当  $v_i$  和  $v_j$  不相邻时第  $(i, j)$  项是 0。换句话说，若邻接矩阵是  $A=[a_{ij}]$ ，则

$$a_{ij} = \begin{cases} 1 & \text{若 } \{v_i, v_j\} \text{ 是 } G \text{ 的一条边} \\ 0 & \text{否则} \end{cases}$$

例 3 用邻接矩阵表示图 6-30 所示的图。

解 把顶点排列成  $a, b, c, d$ 。表示这个图的矩阵是

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

例 4 画出带有相对于顶点顺序  $a, b, c, d$  的邻接矩阵的图。

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

解 图 6-31 显示带有这个邻接矩阵的图。

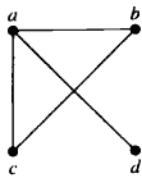


图 6-30 简单图

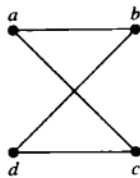


图 6-31 带有给定的邻接矩阵的图

注意，图的邻接矩阵依赖于所选择的顶点的顺序。因此带  $n$  个顶点的图有  $n!$  个不同的邻接矩阵，因为  $n$  个顶点有  $n!$  个不同的顺序。

简单图的邻接矩阵是对称的，即  $a_{ij} = a_{ji}$ ，因为当  $v_i$  和  $v_j$  相邻时，这两个项都是 1，否则都是 0。另外，因为简单图无环，所以每一项  $a_{ii}$  都是 0， $i = 1, 2, 3, \dots, n$ 。

邻接矩阵也可用来表示带环和多重边的无向图。把顶点  $a_i$  上的环表示成邻接矩阵第  $(i, i)$  位置上的 1。当出现多重边时，邻接矩阵不再是 0-1 矩阵，这是因为邻接矩阵的第  $(i, j)$  项等于与  $\{a_i, a_j\}$  关联的边数。包括多重图与伪图在内的所有无向图都具有对称的邻接矩阵。

**例 5** 用邻接矩阵表示图 6-32 所示的伪图。

**解** 顶点顺序为  $a, b, c, d$  的邻接矩阵是

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

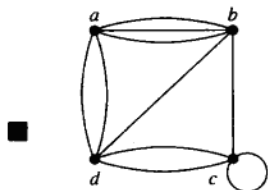


图 6-32 伪图

我们曾在第 5 章里用 0-1 矩阵表示有向图。若有向图  $G = (V, E)$  从  $v_i$  到  $v_j$  有边，则它的矩阵在  $(i, j)$  位置上有 1，其中  $v_1, v_2, \dots, v_n$  是有向图任意的顶点序列。换句话说，若  $A = [a_{ij}]$  是相对于这个顶点表的邻接矩阵，则

$$a_{ij} = \begin{cases} 1 & \text{若 } \{v_i, v_j\} \text{ 是 } G \text{ 的一条边} \\ 0 & \text{否则} \end{cases}$$

有向图的邻接矩阵不必是对称的，因为当从  $a_i$  到  $a_j$  有边时，从  $a_j$  到  $a_i$  可以没有边。

邻接矩阵也可用来表示有向多重图。同样，当有连接两个顶点的同向多重边时，这样的矩阵不是 0-1 矩阵。在有向多重图的邻接矩阵里， $a_{ij}$  等于关联到  $(v_i, v_j)$  的边数。

**在邻接表和邻接矩阵之间取舍** 当一个简单图包含的边相对较少，即该图是一个稀疏图的时候，通常邻接表比邻接矩阵更适合表示它。例如，当每个顶点的度都不超过  $c$ ， $c$  是一个比  $n$  小很多的常数，则邻接表中的每一条链表都最多只含有  $c$  个节点。所以整个邻接表中的元素不超过  $cn$  个。另一方面， $n$  个顶点的图的邻接矩阵含有  $n^2$  个元素。但是，需要注意的是，稀疏图的邻接矩阵是一个稀疏矩阵，即矩阵中只有少量元素不为 0。有特别的技术专门表示和处理稀疏矩阵。

现在设想一个稠密的简单图，它含有很多条边，例如，它含有的边数超过总的所有可能有的边的一半。在这种情形下，用邻接矩阵来表示图就比用邻接表好。为了知道原因，我们来比较判断某条边  $\{v_i, v_j\}$  是否存在的复杂度。在邻接矩阵中，可以通过查看第  $(i, j)$  个元素来决定这条边是否存在。如果该元素是 1 边就存在，如果是 0 边就不存在。所以，只需要一次比较，即将第  $(i, j)$  个元素和 0 比较，就可以判断这条边是否存在。而另一方面，如果使用邻接表表示这个图，就需要搜索  $v_i$  或  $v_j$  的链表中的顶点才能判断这条边是否存在。当图含有的边很多时，这需要  $O(|V|)$  次的比较。

6.3.4 关联矩阵

表示图的另一种常用方式是用关联矩阵。设  $G=(V, E)$  是无向图。设  $v_1, v_2, \cdots, v_n$  是顶点而  $e_1, e_2, \cdots, e_m$  是边。则相对于  $V$  和  $E$  的这个顺序的关联矩阵是  $n \times m$  矩阵  $M=[m_{ij}]$ ，其中

$$m_{ij} = \begin{cases} 1 & \text{当边 } e_j \text{ 关联 } v_i \text{ 时} \\ 0 & \text{否则} \end{cases}$$

例 6 用关联矩阵表示图 6-33 所示的图。

解 关联矩阵是

|       | $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ |
|-------|-------|-------|-------|-------|-------|-------|
| $v_1$ | 1     | 1     | 0     | 0     | 0     | 0     |
| $v_2$ | 0     | 0     | 1     | 1     | 0     | 1     |
| $v_3$ | 0     | 0     | 0     | 0     | 1     | 1     |
| $v_4$ | 1     | 0     | 1     | 0     | 0     | 0     |
| $v_5$ | 0     | 1     | 0     | 1     | 1     | 0     |

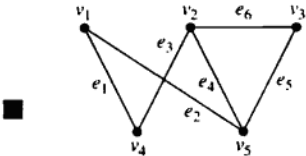


图 6-33 无向图

关联矩阵也可用来表示多重图边和环。在关联矩阵里用各项相等的列来表示多重边，因为这些边关联同样一对顶点。用恰有一项等于 1 的列来表示环，它对应于环所关联的顶点。

例 7 用关联矩阵表示图 6-34 所示的伪图。

解 这个图的关联矩阵是

|       | $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $e_8$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $v_1$ | 1     | 1     | 1     | 0     | 0     | 0     | 0     | 0     |
| $v_2$ | 0     | 1     | 1     | 1     | 0     | 1     | 1     | 0     |
| $v_3$ | 0     | 0     | 0     | 1     | 1     | 0     | 0     | 0     |
| $v_4$ | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 1     |
| $v_5$ | 0     | 0     | 0     | 0     | 1     | 1     | 0     | 0     |

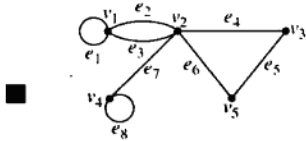


图 6-34 伪图

6.3.5 图的同构

我们经常需要知道是否有可能以同样的方式来画出两个图。例如，在化学里用图为化合物建模。不同的化合物可能分子式相同但结构不同。这样的化合物不能用以同样方式来画的图表示。表示过去已知化合物的图可以用来判定想象中的新化合物是否已经研究过了。

对具有同样结构的图来说，存在着一些有用的术语。

**定义 1** 设  $G_1=(V_1, E_1)$  和  $G_2=(V_2, E_2)$  是简单图，若存在一对一的和映上的从  $V_1$  到  $V_2$  的函数  $f$ ，且  $f$  具有这样的性质：对  $V_1$  里所有的  $a$  和  $b$  来说， $a$  和  $b$  在  $G_1$  里相邻当且仅当  $f(a)$  和  $f(b)$  在  $G_2$  里相邻，就说  $G_1$  与  $G_2$  是同构的。这样的函数  $f$  称为同构<sup>⊖</sup>。

换句话说，当两个简单图同构时，两个图的顶点之间具有保持相邻关系的一一对应。

例 8 证明：图 6-35 所示图  $G=(V, E)$  和  $H=(W, F)$  同构。

解 函数  $f$  满足  $f(u_1)=v_1, f(u_2)=v_4, f(u_3)=v_3, f(u_4)=v_2$ ，它是  $V$  和  $W$  之间的一一对应。为了看出这个对应保持相邻关系，注意  $G$  里相邻的顶点是  $u_1$  和  $u_2, u_1$  和  $u_3, u_2$  和  $u_4$ ，以及  $u_3$  和  $u_4$ ，由  $f(u_1)=v_1$  和  $f(u_2)=v_4, f(u_1)=v_1$  和  $f(u_3)=v_3, f(u_2)=v_4$  和  $f(u_4)=v_2$ ，以及  $f(u_3)=v_3$  和  $f(u_4)=v_2$  所组成的每一对顶点都是在  $H$  里相邻的。 ■

判定两个简单图是否同构常常是一件困难的事情。在两个带有  $n$  个顶点的简单图顶点集之间有  $n!$  种可能的一一对应。若  $n$  太大，则通过检验每一种对应来看它是否保持相邻关系是不可行的。

⊖ 同构(isomorphism)这个词来自两个希腊语字根：表示“相等”的 isos 和表示“形式”的 morphe。

有时候要说明两个图不同构并不困难。特别地,如果能找到某个属性,两个图中只有一个图具有这个属性,且该属性只在讨论同构时使用,就可以说这两个图不同构。这种保留在图的同构问题中使用的属性被称为图的非变体性。例如,同构的简单图必然具有相同的顶点数,因为在这些图的顶点集之间有着——对应。

同构的简单图必然有相同的边数,因为在顶点之间的一一对应建立了边之间的一一对应。另外,同构的简单图的对应顶点的度必然相同。即图  $G$  中的  $d$  度顶点  $v$  必然对应图  $H$  中的  $d$  度顶点  $f(v)$ ,这是因为在图  $G$  中顶点  $w$  与  $v$  相邻,当且仅当在图  $H$  中  $f(v)$  与  $f(w)$  相邻。

**例 9** 说明图 6-36 所示的图不同构。

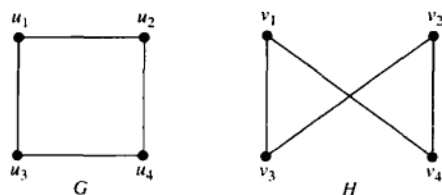


图 6-35 图  $G$  和  $H$

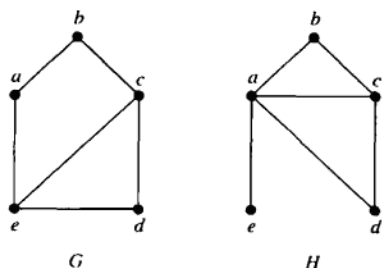


图 6-36 图  $G$  和  $H$

**解**  $G$  和  $H$  都具有 5 个顶点 6 条边。不过,  $G$  有 1 度顶点  $e$  而  $H$  没有 1 度顶点。所以  $G$  与  $H$  是不同构的。 ■

顶点数、边数以及顶点度都是在同构下的不变量。若两个简单图的这些量有任何不同,则这两个图就不是同构的。不过,当这些量都相同时,也不一定意味着两个图同构。目前还没有已知的用来判定简单图是否同构的不变量集。

**例 10** 判定图 6-37 所示的图是否同构。

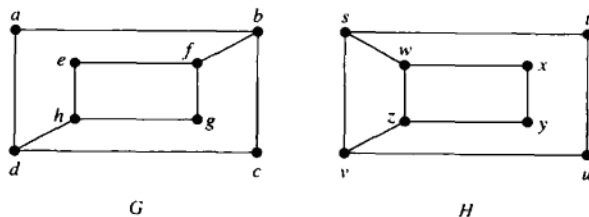


图 6-37 图  $G$  和  $H$

**解** 图  $G$  和  $H$  都具有 8 个顶点 10 条边。它们都具有 4 个 2 度顶点和 4 个 3 度顶点。因为这些不变量都相同,所以它们可能会是同构的。

然而  $G$  和  $H$  不是同构的。为了看明白这一点,注意到因为在  $G$  里  $\deg(a)=2$ , 所以  $a$  必然对应于  $H$  里的  $t$ ,  $u$ ,  $x$  或  $y$ , 这是因为这些顶点是  $H$  里的 2 度顶点。不过,  $H$  里的这 4 个顶点中的每一个都与  $H$  里另一个 2 度顶点相邻,但是在  $G$  里  $a$  却不是这样的。

看出  $G$  与  $H$  不同构的另一种方式是,注意到若这两个图同构,则由 3 度顶点和连接它们的边所组成的子图同构(读者应当验证它)。然而图 6-38 所示的这些子图却

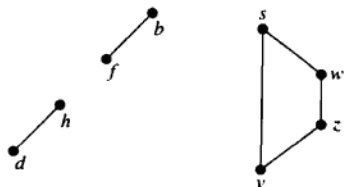


图 6-38 由 3 度顶点和连接它们的边所组成的  $G$  和  $H$  的子图

不是同构的。

为了说明从图  $G$  的顶点集到图  $H$  的顶点集的函数  $f$  是一个同构，需要说明  $f$  保持边的关系。一种有助于这样做的方式是利用邻接矩阵。具体地说，为了说明  $f$  是一个同构，就说明  $G$  的邻接矩阵与  $H$  的邻接矩阵相同，其中  $G$  的邻接矩阵的行和列的标记都是  $G$  里的顶点， $H$  的邻接矩阵的行和列的标记都是  $G$  里的对应顶点在  $f$  下的像。例 11 解释如何这样做。

例 11 判定图 6-39 所示的图  $G$  和  $H$  是否同构。

解  $G$  和  $H$  都具有 6 个顶点 7 条边，都具有 4 个 2 度顶点和 2 个 3 度顶点。还容易看出由 2 度顶点和连接它们的边所组成的  $G$  和  $H$  的子图是同构的(读者应当验证它)。因为  $G$  和  $H$  对这些不变量来说是相同的，这就有理由试着找出一个同构  $f$ 。

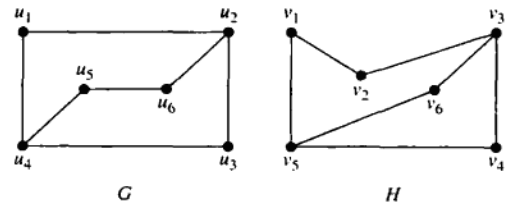


图 6-39 图  $G$  和  $H$

现在定义函数  $f$ ，然后判定它是否同构。因为  $\deg(u_1)=2$  而且  $u_1$  不与任何其他 2 度顶点相邻，所以  $u_1$  的像必然是  $v_1$  或  $v_6$ ，它们是  $H$  里仅有的不与 2 度顶点相邻的顶点。任取  $f(u_1)=v_6$ 。(如果发现这个选择得不出同构，就接着试验  $f(u_1)=v_1$ 。)因为  $u_2$  与  $u_1$  相邻，所以  $u_2$  可能的像是  $v_3$  和  $v_5$ 。任取  $f(u_2)=v_3$ 。照这样继续下去，用顶点的相邻关系和度作为指引，令  $f(u_3)=v_1$ ， $f(u_4)=v_5$ ， $f(u_5)=v_1$ ，以及  $f(u_6)=v_2$ 。现在已经有了在  $G$  的顶点集与  $H$  的顶点集之间的一一对应，即  $f(u_1)=v_6$ ， $f(u_2)=v_3$ ， $f(u_3)=v_1$ ， $f(u_4)=v_5$ ， $f(u_5)=v_1$ ，以及  $f(u_6)=v_2$ 。为了查看  $f$  是否保持边，就检查  $G$  的邻接矩阵：

$$A_G = \begin{matrix} & \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{matrix} \\ \begin{matrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

以及用  $G$  里的对应顶点的像来标记行和列的  $H$  的邻接矩阵：

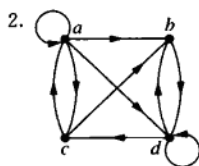
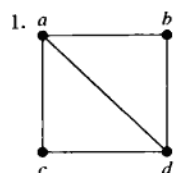
$$A_H = \begin{matrix} & \begin{matrix} v_6 & v_3 & v_4 & v_5 & v_1 & v_2 \end{matrix} \\ \begin{matrix} v_6 \\ v_3 \\ v_4 \\ v_5 \\ v_1 \\ v_2 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

因为  $A_G=A_H$ ，所以  $f$  是保持边的。由此得出  $f$  是一个同构，所以  $G$  与  $H$  是同构的。注意，若事实证明  $f$  不是一个同构，是无法得出  $G$  与  $H$  不是同构的，因为  $G$  和  $H$  里的顶点的另一个对应仍然可以是同构。

判定两个图是否同构，已知的最好算法具有指数的最坏情形时间复杂性(对图的顶点数来说)。不过，解决这个问题的线性平均情形时间复杂性的算法已经找到，而且有希望找到判定两个图是否同构的多项式最坏情形时间复杂性的算法。一种名为 NAUTY 的最佳实用算法，在现代个人电脑上可在 1 秒钟之内判定带有 100 个顶点的两个图是否同构。可以在因特网上下载 NAUTY 软件并用它做实验。

## 练习

在练习 1~2 中, 用邻接表表示给定的图。



3. 用邻接矩阵表示练习 1 里的图。

4. 用邻接矩阵表示练习 2 里的图。

5. 用邻接矩阵表示下列每一个图。

a)  $K_4$

b)  $K_{1,4}$

c)  $K_{2,3}$

d)  $C_4$

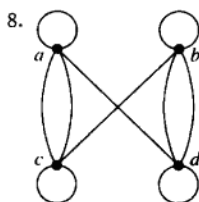
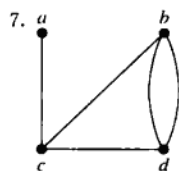
e)  $W_4$

f)  $Q_3$

6. 画出用下面的邻接矩阵表示的图。

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

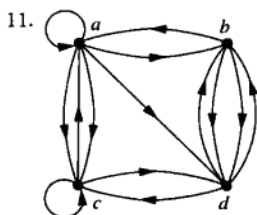
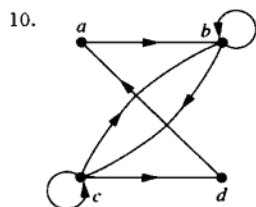
在练习 7~8 中, 用邻接矩阵表示给定的图。



9. 画出用下面的邻接矩阵所表示的无向图。

$$\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

在练习 10~11 中, 求给定的有向图的邻接矩阵。



12. 画出用下面的邻接矩阵表示的图。

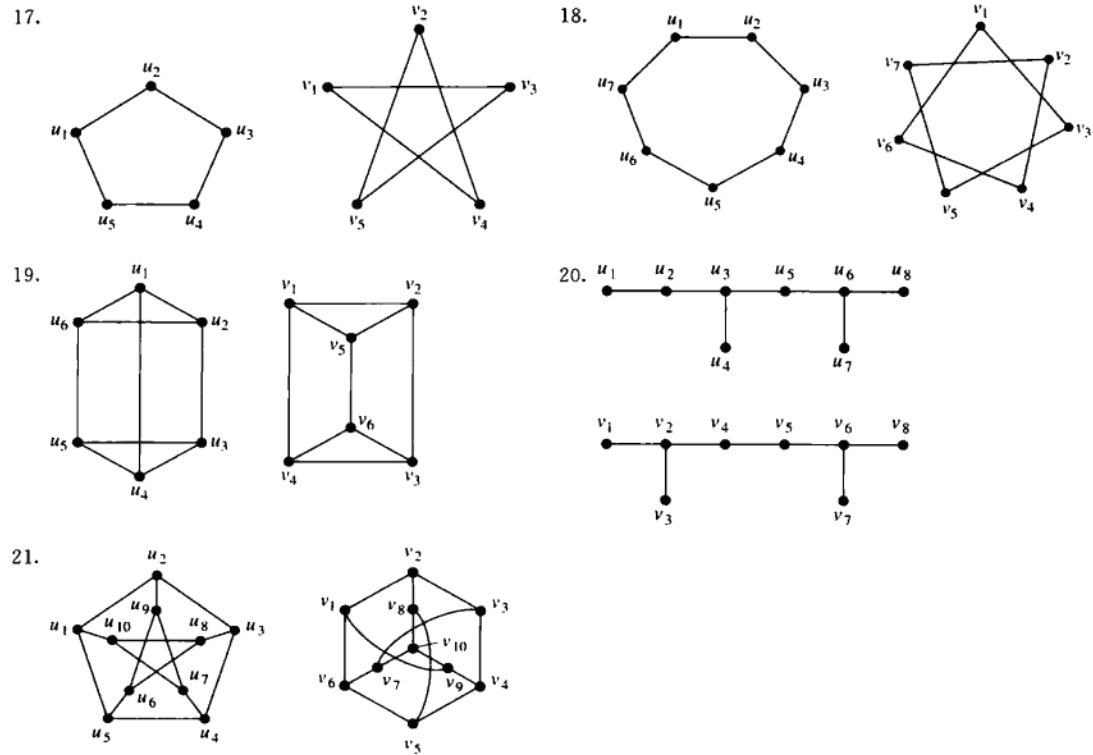
$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix}$$

13. 每一个对称的和对角线全为 0 的 0-1 方阵是否都是简单图的邻接矩阵?

14. 用关联矩阵表示练习 7~8 中的图。

- \*15. 什么是无向图的邻接矩阵的一列里的各项之和？对有向图来说呢？
- 16. 什么是无向图的关联矩阵的一列里的各项之和？对有向图来说呢？

在练习 17~21 中，判定所给定的一对图是否同构。给出一个同构或给出不存在同构的严格证明。



- 22. 证明：简单图的同构关系是等价关系。
- 23. 描述一下图的邻接矩阵里对应于孤立点的行和列。
- 24. 证明：带有 2 个以上顶点的偶图的顶点可以排序，使得邻接矩阵形如

$$\begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix}$$

其中所示的四项都是矩形块。

若简单图  $G$  和  $\bar{G}$  是同构的，则  $G$  称为自补图。

- 25. 求带有 5 个顶点的自补简单图。
- 26. 对哪些整数  $n$  来说  $C_n$  是自补图？
- 27. 带有 5 个顶点和 3 条边的非同构的简单图有多少个？
- 28. 带有下列邻接矩阵的简单图是否同构？

a)  $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

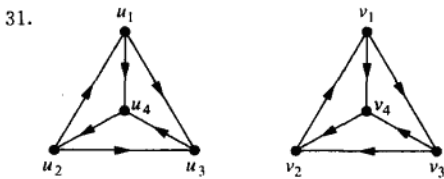
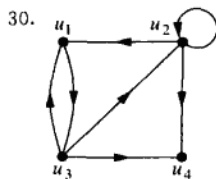
b)  $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

c)  $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$



29. 把简单图的同构定义推广到包含环和多重边的无向图。

在练习 30~31 里, 判定所给定的一对图是否同构。



32. 证明: 若  $G$  和  $H$  是同构的有向图, 则  $G$  和  $H$  的逆(在 9.2 节练习 61 之前定义)也是同构的。

33. 找出一对非同构的图, 它们具有相同的度序列, 但一个是偶图而另一个不是。

\*34. 无向图的关联矩阵与它的转置之积是什么?

## 6.4 连通性

### 6.4.1 引言

许多问题可以用沿图的边旅行所形成的通路来建模。例如, 判定在两个计算机之间用中间连接能否传递消息的问题, 就可以用图模型来研究。利用图里的通路所组成的模型, 可以解决投递邮件、收取垃圾以及计算机网络诊断等有效规划路线的问题。

### 6.4.2 通路

非形式化地说, 通路是边的序列, 它从图的一个顶点上开始并且行经图的边, 总是连接相邻的顶点对。定义 1 给出通路的形式化定义和相关术语。

**定义 1** 设  $n$  是非负整数且  $G$  是无向图。在  $G$  中从  $u$  到  $v$  的长度为  $n$  的通路是  $G$  的  $n$  条边  $e_1, \dots, e_n$  的序列, 使得  $f(e_1) = \{x_0, x_1\}$ ,  $f(e_2) = \{x_1, x_2\}$ ,  $\dots$ ,  $f(e_n) = \{x_{n-1}, x_n\}$ , 其中  $x_0 = u$  而  $x_n = v$ 。当这个图是简单图时, 就用顶点序列  $x_0, x_1, \dots, x_n$  表示这条通路(这是因为列出这些顶点就唯一地确定了通路)。若一条通路在相同的顶点上开始和结束, 即  $u = v$  且长度大于 0, 则它是一条回路。把这条通路或回路说成是经过顶点  $x_1, x_2, \dots, x_{n-1}$  或遍历边  $e_1, e_2, \dots, e_n$ 。若通路或回路不重复地包含相同的边, 则它是简单的。

当没有必要区分多重边时, 就用顶点序列  $x_0, x_1, \dots, x_n$  表示通路  $e_1, e_2, \dots, e_n$ , 其中对  $i = 1, 2, \dots, n$  来说,  $f(e_i) = \{x_{i-1}, x_i\}$ 。这种记法仅仅指出通路所经过的顶点。经过这个顶点序列可以有多条通路。注意长度为 0 的通路由单个顶点组成。

**注意** 关于定义 1 中的概念, 有很多不同的术语。例如在有些书里, 使用路径(walk)而不是通路(path), 这时路径被定义为图的顶点和边相互交替的序列,  $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$ , 其中  $v_{i-1}$  和  $v_i$  是  $e_i$  的端点,  $i = 1, 2, 3, \dots, n$ 。当使用“路径”这个术语时, 就会使用“闭合路径”(closed walk)而不是回路(circuit)来表示起始和终止于相同顶点的路径; 使用路线(trail)来表示没有重复边的路径(代替“简单通路”)。当使用路线这一术语时, 术语通路(path)通常就会用来表示没有重复顶点的路线, 这与定义 1 中的术语相冲突。由于这些术语上的各种变体, 当你在特定的书或者文章中阅读有关遍历图的边的内容时, 需要弄清楚使用的是哪一组定义。文章[GrYe06]是关于本注释中提到的各替代术语的一本好的参考文献。

**例 1** 在图 6-40 所示的图里,  $a, d, c, f, e$  是长度为 4 的简单通路, 因为  $\{a, d\}, \{d, c\}, \{c, f\}$  和  $\{f, e\}$  都是边。但是  $d, e, c, a$  不是通路, 因为  $\{e, c\}$  不是边。注意  $b, c, f, e, b$  是长度为 4 的回路, 这是因为  $\{b, c\}, \{c, f\}, \{f, e\}$  和  $\{e, b\}$  都是边, 这条通路在  $b$  上开始和结束。长度为 5 的通路  $a, b, e, d, a, b$  不是简单的, 因为它两次包含边  $\{a, b\}$ 。■

有向图里的通路和回路在第 5 章里介绍过。现在对有向多重图来定义这样的通路。

**定义 2** 设  $n$  是非负整数且  $G$  是有向多重图。在  $G$  中从  $u$  到  $v$  的长度为  $n$  的通路是  $G$  的边的序列  $e_1, e_2, \dots, e_n$ , 使得  $f(e_1) = (x_0, x_1), f(e_2) = (x_1, x_2), \dots, f(e_n) = (x_{n-1}, x_n)$ , 其中  $x_0 = u$  而  $x_n = v$ 。当有向图里没有多重边时, 就用顶点序列  $x_0, x_1, \dots, x_n$  表示这条通路。把在相同的顶点上开始和结束的长度大于 0 的通路称为回路或圈。若一条通路或回路不重复地包含相同的边, 则把它称为简单的。

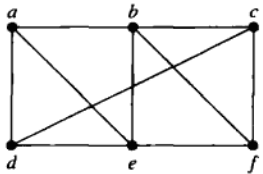


图 6-40 简单图

**注意** 由定义 2 给出的术语并非经常用来表示其中描述的概念。特别地, 使用路径(walk)、闭合路径(closed walk)、路线(trail)、通路(path)的术语(在定义 1 之后的注解中介绍过)可能用来描述有向图。详见[GrYe06]。

注意, 通路上一条边的终点是这条通路上下一条边的起点。当没有必要区分多重边时, 就用顶点序列  $x_0, x_1, \dots, x_n$  表示通路  $e_1, e_2, \dots, e_n$ , 其中对  $i = 1, 2, \dots, n$  来说,  $f(e_i) = (x_{i-1}, x_i)$ 。这种记法仅仅指出通路所经过的顶点。可以有多条通路经过这个顶点序列。

通路表示许多图模型里的有用信息, 如例 2~4 所示。

**例 2 熟人关系图中的通路** 在熟人关系图中, 如果相邻两个人彼此认识则将此二人连接起来, 形成一条人链。例如在 6.1 节的图 6-7 中, 有一条连接 Kamini 和 Ching 的 6 个人的链。许多社会学家猜想, 是否可以用只包含 5 个或更少的人的短链来连接世界上几乎每一对人。这意味着世界上所有人的熟人关系图中几乎每对顶点都通过长度不超过 4 的通路来连接。约翰·奎尔(John Guare)的戏剧《六度分离》(*Six Degrees of Separation*)就是基于这个概念。

**例 3 合作图中的通路** 在合作图中, 如果表示作者的两个顶点  $a$  和  $b$  之间有从  $a$  开始到  $b$  结束的一系列作者, 使得每条边端点所表示的两个作者写过一篇联名论文, 则  $a$  和  $b$  就通过一条通路而连接。在所有数学家的合作图中, 数学家  $m$  的埃德斯数(在第 5 章补充练习 14 中用关系术语定义过)就是在  $m$  和代表成果极其丰富的数学家保罗·埃德斯(1996 年去世)的顶点之间的最短通路的长度。换句话说, 一个数学家的埃德斯数就是从保罗·埃德斯开始到这个数学家结束的最短的数学家链的长度, 其中每一对相邻的数学家都联名写过论文。根据“埃德斯数项目”, 2006 年具有各个埃德斯数的数学家的数目如表 6-4 所示。

表 6-4 具有给定埃德斯数的数学家的数目

| 埃德斯数 | 人 数    | 埃德斯数 | 人 数    |
|------|--------|------|--------|
| 0    | 1      | 7    | 11 591 |
| 1    | 504    | 8    | 3 146  |
| 2    | 6 593  | 9    | 819    |
| 3    | 33 605 | 10   | 244    |
| 4    | 83 642 | 11   | 68     |
| 5    | 87 760 | 12   | 23     |
| 6    | 40 014 | 13   | 5      |

**例 4 好莱坞图中的通路** 在好莱坞图(参见 6.1 节例 4)中, 当存在连接两个顶点  $a$  和  $b$  的演员链, 其中在这个链上每两个相邻的演员都出演过同一部电影时,  $a$  和  $b$  就被连接。在好莱坞图中演员  $c$  的培根数定义为连接  $c$  和著名演员凯文·培根的最短通路的长度。随着新电影(包括凯文·培根的新电影)的不断产生, 演员们的培根数也在不断地发生变化。表 6-5 中显示的是从培根网站得来的 2006 年具有各个培根数的演员的数目。

表 6-5 具有给定培根数的演员数

| 培根数 | 人 数     | 培根数 | 人 数   |
|-----|---------|-----|-------|
| 0   | 1       | 5   | 8 168 |
| 1   | 1 902   | 6   | 810   |
| 2   | 160 463 | 7   | 81    |
| 3   | 457 231 | 8   | 14    |
| 4   | 111 310 |     |       |

### 6.4.3 无向图的连通性

若消息通过一个或多个中间计算机来传递,则计算机网络何时具有每对计算机都可共享信息的性质?当利用图来表示这个计算机网络,用顶点表示计算机而用边表示通信连接时,这个问题就变成:何时在图里任何两个顶点之间都存在通路?

**定义 3** 若无向图每一对不同的顶点之间都有通路,则该图称为连通的。

因此,在网络里任何两个计算机都可以通信,当且仅当这个网络图是连通的。

**例 5** 图 6-41 里的图  $G_1$  是连通的,这是因为在每一对不同的顶点之间都有通路(读者应当验证它)。但是图 6-41 里的图  $G_2$  不是连通的。例如,在顶点  $a$  和  $d$  之间没有通路。

在第 7 章里将用到下述定理。

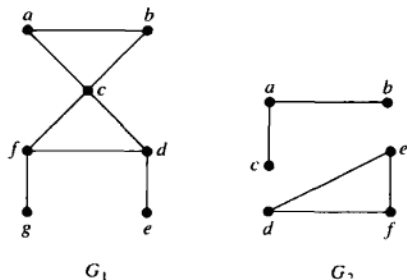
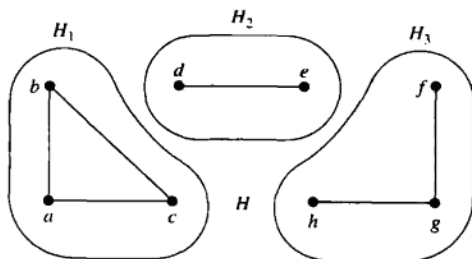
**定理 1** 在连通无向图的每一对不同顶点之间都存在简单通路。

**证** 设  $u$  和  $v$  是连通无向图  $G=(V, E)$  的两个不同的顶点。因为  $G$  是连通的,所以  $u$  和  $v$  之间至少有 1 条通路。设  $x_0, x_1, \dots, x_n$  是长度最短的通路的顶点序列,其中  $x_0=u$  而  $x_n=v$ 。这条长度最短的通路是简单的。为了看明白这一点,假设它不是简单的。则对满足  $0 \leq i < j$  的某个  $i$  和  $j$  来说,有  $x_i = x_j$ 。这意味着通过删除顶点序列  $x_i, \dots, x_{j-1}$  所对应的边,就获得了带有顶点序列  $x_0, x_1, \dots, x_{i-1}, x_j, \dots, x_n$  的从  $u$  到  $v$  的更短的通路。□

不连通的图是 2 个或 2 个以上连通子图之并,每一对子图都没有公共的顶点。这些不相交的连通子图称为图的连通分支。

**例 6** 图 6-42 所示的图  $H$  的连通分支是什么?

**解** 图  $H$  是图 6-42 所示的 3 个不相交连通子图  $H_1$ 、 $H_2$  和  $H_3$  之并。这 3 个子图是  $H$  的连通分支。

图 6-41 图  $G_1$  和  $G_2$ 图 6-42 图  $H$  和它的连通分支  $H_1$ 、 $H_2$  和  $H_3$ 

**例 7 呼叫图的连通分支** 当电话呼叫图(参见 6.1 节例 7)中存在一系列从  $x$  开始到  $y$  结束的电话呼叫时,两个顶点  $x$  和  $y$  就属于同一个连通分支。当分析 AT&T 网络中特定一天内发生的电话呼叫的呼叫图时,发现这个图具有 53 767 087 个顶点、超过 1 亿 7 千万条边和超过 370 万个连通分支。这些连通分支大多数都很小:大约四分之三是由表示只在彼此之间呼叫的一对电话号码的两个顶点所组成。这个图具有一个包含 44 989 297 个顶点(占总数的 80%)的巨

大的连通分支。另外，这个连通分支中每个顶点都通过一条不超过 20 个顶点的链连接到任何其他顶点。

有时删除一个顶点和它所关联的边，就产生带有比原图更多的连通分支的子图。把这样的顶点称为割点(或节点)。从连通图里删除割点，就产生不连通的子图。同理，把一旦删除就产生带有比原图更多的连通分支的子图的边称为割边或桥。

**例 8** 求出图 6-43 所示的图  $G$  的割点和割边。

**解** 图  $G$  的割点是  $b$ 、 $c$  和  $e$ 。删除这些顶点中的一个(和它的邻边)，就使得这个图不再是连通的。割边是  $\{a, b\}$  和  $\{c, e\}$ 。删除这些边中的一条，就使得  $G$  不再是连通的。

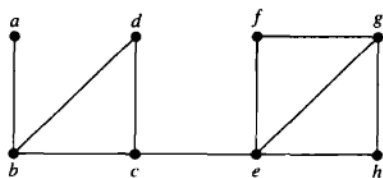


图 6-43 图  $G$

#### 6.4.4 有向图的连通性

根据是否考虑边的方向，在有向图里有两种连通性概念。

**定义 4** 若每当  $a$  和  $b$  都是一个有向图里的顶点时，就有从  $a$  到  $b$  和从  $b$  到  $a$  的通路，则该图是强连通的。

为了让一个有向图是强连通的，从这个图里任何一个顶点到任何另外一个顶点必须存在有向边的序列。有向图可以不是强连通的但还是“一整块”。定义 5 准确地说明了这个概念。

**定义 5** 若在有向图的底图里，任何两个顶点之间都有通路，则该有向图是弱连通的。

有向图是弱连通的，当且仅当在忽略边的方向时，任何两个顶点之间总是存在通路。显然，任何强连通有向图也是弱连通的。

**例 9** 图 6-44 所示的有向图  $G$  和  $H$  是否为强连通的？是否为弱连通的？

**解**  $G$  是强连通的，因为在这个有向图里，任何两个顶点之间都存在通路(读者应当验证它)。因此  $G$  也是弱连通的。图  $H$  不是强连通的，在这个图里，从  $a$  到  $b$  没有有向通路。但是  $H$  是弱连通的，因为在  $H$  的底图里，任何两个顶点之间都有通路(读者应当验证它)。

有向图  $G$  的子图是强连通的而不包含在更大的强连通子图中，即极大强连通子图，可称之为  $G$  的强连通分支或强分支。

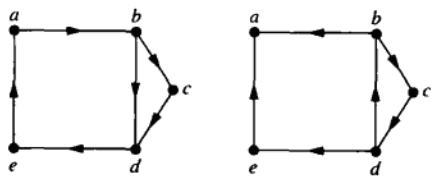


图 6-44 有向图  $G$  和  $H$

**例 10** 图 6-44 中的图  $H$  有 3 个强连通分支，包括：顶点  $a$ ；顶点  $e$ ；由顶点  $b$ 、 $c$  和  $d$  以及边  $(b, c)$ 、 $(c, d)$  和  $(d, b)$  所组成的图。

**例 11 网络图的强连通分支** 6.1 节例 8 介绍的网络图用顶点表示网页并且用有向边表示链接。网络在 1999 年的快照产生了具有 2 亿个顶点和 15 亿条边的网络图(参见 [Br00] 了解详情)。网络图的底无向图不是连通的并且有一个连通分支包含了这个图中大约 90% 的顶点。底无向图的这个连通分支所对应的原来有向图的子图(即具有相同的顶点以及连接这个图中顶点的所有有向边)有一个非常大的强连通分支和许多小的强连通分支。前者称为这个有向图的巨型强连通分支。可以遵循从这个分支里任何其他网页开始的链接来到达这个分支里的某一个网页。已经发现这项研究产生的网络图里的巨型强连通分支有超过 5300 万个顶点。这个无向图的大型连通分支里的其余顶点表示 3 种不同类型的网页：可以从巨型强连通分支里的网页来到达，但是不能通过一系列链接来返回前面这些网页的网页；可以通过一系列链接来返回巨型强连通分支里的网页，但是不能通过巨型强连通分支里网页上的链接来到达的网页；既不能到达巨型强连通分支里的网页，也不能通过一系列链接从巨型强连通分支里网页来到达的网页。在这项研究里，发现其余这三个集合中每个都具有大约 4400 万个顶点(这三个集合都接近同样的规模，这是相当令人惊讶的)。

### 6.4.5 通路与同构

有多种方式可以利用通路和回路来帮助判定两个图是否同构。例如, 具有特定长度的简单回路的存在性, 就是一种可以用来证明两个图是不同构的有用的不变量。另外, 可以利用通路来构造可能的同构映射。

前面指出过, 简单图的一个有用的同构不变量是长度为  $k$  的简单回路的存在性, 其中  $k$  是大于 2 的正整数。例 12 说明如何用这个不变量来证明两个图不同构。

**例 12** 判定图 6-45 所示的图  $G$  和  $H$  是否同构。

**解**  $G$  和  $H$  都具有 6 个顶点和 8 条边。各自具有 4 个 3 度顶点和 2 个 2 度顶点。所以对两个图来说有 3 个不变量(顶点数、边数以及顶点度)都是相同的。但是  $H$  有长度为 3 的简单回路, 即  $v_1, v_2, v_6, v_1$ , 而  $G$  没有长度为 3 的简单回路( $G$  里的所有简单回路的长度至少为 4)。因为长度为 3 的简单回路的存在性是一个同构不变量, 所以  $G$  和  $H$  不同构。■

我们已经说明了如何用某种类型的通路, 即具有特定长度的简单回路, 来证明两个图是不同构的。还可以用通路来求出潜在的同构映射。

**例 13** 判定图 6-46 所示的图  $G$  和  $H$  是否同构。

**解**  $G$  和  $H$  都具有 5 个顶点和 6 条边, 都具有 2 个 3 度顶点和 3 个 2 度顶点, 都具有 1 个长度为 3 的简单回路, 1 个长度为 4 的简单回路, 以及 1 个长度为 5 的简单回路。因为所有这些同构不变量都是相同的, 所以  $G$  和  $H$  可能是同构的。为了求出可能的同构, 沿着经过所有顶点并且使得两图里的对应顶点的度都相同的通路前进。例如,  $G$  里的通路  $u_1, u_4, u_3, u_2, u_5$  和  $H$  里的通路  $v_3, v_2, v_1, v_5, v_4$  都经过图里的每一个顶点, 都在 3 度顶点上开始, 都分别经过 2 度顶点和 2 度顶点并且在 2 度顶点上结束。通过在图里沿着这些通路前进, 定义映射  $f$  满足  $f(u_1) = v_3, f(u_4) = v_2, f(u_3) = v_1, f(u_2) = v_5$  和  $f(u_5) = v_4$ 。通过说明  $f$  保持边, 或者通过说明在顶点的适当顺序下  $G$  和  $H$  的邻接矩阵是相同的, 读者就可以说明  $f$  是一个同构, 所以  $G$  与  $H$  是同构的。■

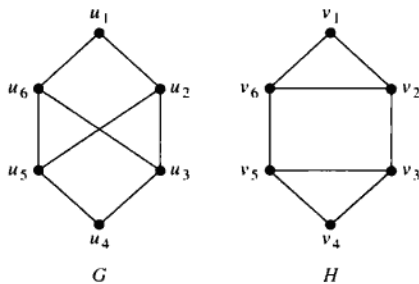


图 6-45 图  $G$  和  $H$

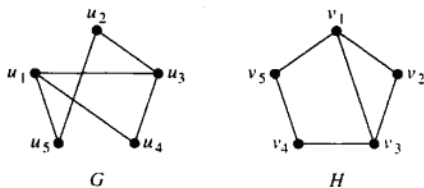


图 6-46 图  $G$  和  $H$

### 6.4.6 计算顶点之间的通路数

在一个图里两个顶点之间的通路的数目, 可以用这个图的邻接矩阵来确定。

**定理 2** 设  $G$  是带有相对于顶点顺序  $v_1, v_2, \dots, v_n$  的邻接矩阵  $A$  的图(允许带有无向或有向边、带有多重边和环)。从  $v_i$  到  $v_j$  的长度为  $r$  的不同通路的数目等于  $A^r$  的第  $(i, j)$  项, 其中  $r$  是正整数。

**证** 用数学归纳法证明。设  $G$  是带有邻接矩阵  $A$  的图(假设  $G$  的顶点具有顺序  $v_1, v_2, \dots, v_n$ )。从  $v_i$  到  $v_j$  长度为 1 的通路数是  $A$  的第  $(i, j)$  项, 这是因为该项是从  $v_i$  到  $v_j$  的边数。

假设  $A^r$  的第  $(i, j)$  项是从  $v_i$  到  $v_j$  的长度为  $r$  的不同通路的个数。这是归纳假设。因为  $A^{r+1} = A^r A$ , 所以  $A^{r+1}$  的第  $(i, j)$  项等于

$$b_{i1}a_{1j} + b_{i2}a_{2j} + \dots + b_{in}a_{nj}$$

其中  $b_{ik}$  是  $A^r$  的第  $(i, k)$  项。根据归纳假设,  $b_{ik}$  是从  $v_i$  到  $v_k$  的长度为  $r$  的通路数。

从  $v_i$  到  $v_j$  的长度为  $r+1$  的通路, 包括从  $v_i$  到某个中间顶点  $v_k$  的长度为  $r$  的通路以及从  $v_k$  到  $v_j$  的边。根据计数的乘法规则, 这样的通路的个数是从  $v_i$  到  $v_k$  的长度为  $r$  的通路数(即  $b_{ik}$ )与从  $v_k$  到  $v_j$  的边数(即  $a_{kj}$ )之积。当对所有可能的中间顶点  $v_k$  求这些乘积之和时, 根据计数的加法规则,

就得出所需要的结果。

例 14 在图 6-47 所示的简单图  $G$  里, 从  $a$  到  $d$  的长度为 4 的通路有多少条?

解  $G$  的邻接矩阵(顶点顺序为  $a, b, c, d$ )是

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

因此从  $a$  到  $d$  的长度为 4 的通路数是  $A^4$  的第(1, 4)项。因为

$$A^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

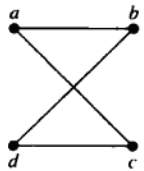


图 6-47 图  $G$

所以恰好有 8 条从  $a$  到  $d$  的、长度为 4 的通路。通过检查这个图, 我们看出  $a, b, a, b, d$ ;  $a, b, a, c, d$ ;  $a, b, d, b, d$ ;  $a, b, d, c, d$ ;  $a, c, a, b, d$ ;  $a, c, a, c, d$ ;  $a, c, d, b, d$  和  $a, c, d, c, d$  是 8 条从  $a$  到  $d$  的通路。

定理 2 可以用来求出在图的两个顶点之间的最短通路的长度, 还可以用来判定图是否连通(见练习 25)。

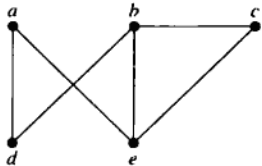
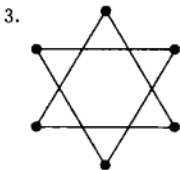
练习

1. 下述的每个顶点列表是否可以构成右图里的通路? 哪些通路是简单的?

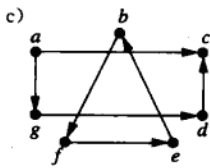
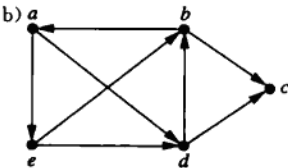
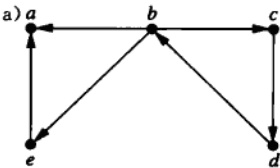
哪些是回路? 那些通路的长度是多少?

- a)  $a, e, b, c, b$
- b)  $a, e, a, d, b, c, a$
- c)  $e, b, a, d, b, e$
- d)  $c, b, d, a, e, c$

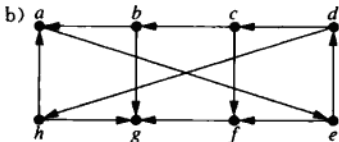
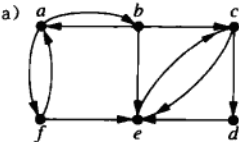
在练习 2~3 里, 判定所给的图是否连通。

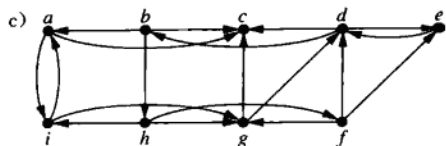


- 4. 熟人关系图的连通分支表示什么?
- 5. 解释一下为什么在数学家的合作图里, 表示一个数学家的顶点与表示保罗·埃德斯的顶点是在同一个连通分支里, 当且仅当这个数学家具有有穷的埃德斯数。
- 6. 判断下列各图是否是强连通的, 如果不是, 再判断是否是弱连通的。



- 7. 电话呼叫图的强连通分支表示什么?
- 8. 求下列每个图的强连通分支。

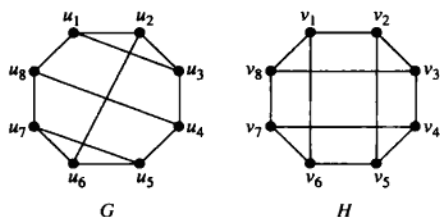




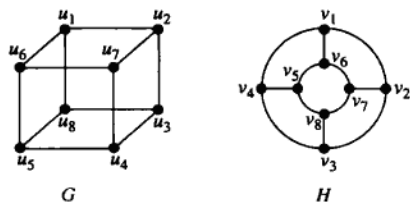
9. 求出  $K_4$  里两个不同顶点之间长度为  $n$  的通路数目, 若  $n$  是

- a) 2      b) 3      c) 4      d) 5

10. 运用通路要么证明这两个图不是同构的, 要么找出这两图之间的一个同构。



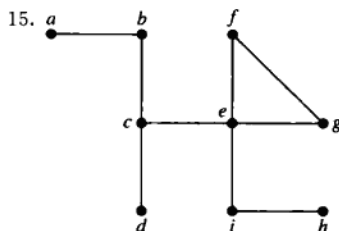
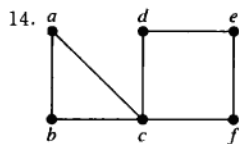
11. 运用通路要么证明这两个图不是同构的, 要么找出这两图之间的一个同构。



12. 对练习 9 里的  $n$  值来说, 求出  $K_{3,3}$  里任意两个不相邻顶点之间长度为  $n$  的通路数目。

13. 设  $G=(V, E)$  是简单图。设  $R$  是  $V$  上的关系, 它是由这样的顶点对  $(u, v)$  所组成的, 使得存在从  $u$  到  $v$  的通路或使得  $u=v$ 。证明:  $R$  是一个等价关系。

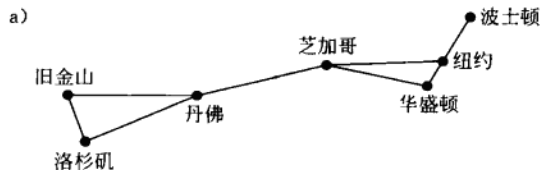
在练习 14~15 里, 求所给图的所有割点。

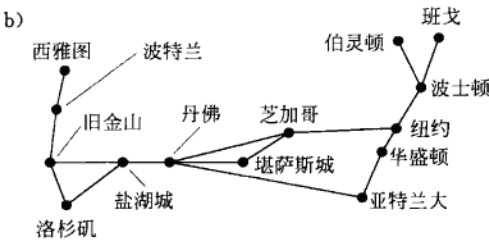


\*16. 假设  $v$  是一条割边的端点。证明:  $v$  是割点当且仅当它不是悬挂点。

\*17. 证明: 在至少有 2 个顶点的简单图中, 至少有 2 个顶点不是割点。

18. 若网络里的通信连接故障会导致不能传送某些消息, 则应当提供备份连接。对下面(a)和(b)里所示的通信网络来说, 确定出那些应当有备份的连接。





一个有向图里的顶点基是顶点的一个集合，使得该有向图里该集合之外的任何顶点，都有从该集合里某个顶点开始的一条通路，并且该集合里任何两个顶点之间都没有通路。

19. 在影响图(在 6.1 节例 3 里描述)里顶点基的重要性是什么？找出在该例里的影响图的顶点基。

\*20. 证明：若简单图  $G$  有  $k$  个连通分支，而且这些分支分别具有  $n_1, n_2, \dots, n_k$  个顶点，则  $G$  的边数不超过

$$\sum_{i=1}^k C(n_i, 2)$$

\*21. 证明：若带有  $n$  个顶点的简单图  $G$  具有超过  $(n-1)(n-2)/2$  条边，则它是连通的。

22. 当  $n$  取如下值时，存在多少个不同构的、带有  $n$  个顶点的连通简单图？

a) 2

b) 3

c) 4

d) 5

23. 用定理 2 求出图 6-40 里的多重图从  $a$  到  $f$  的最短通路的长度。

24. 设  $P_1$  和  $P_2$  是简单图  $G$  里顶点  $u$  和  $v$  之间的没有相同边的两条简单通路。证明：在  $G$  里存在简单回路。

25. 解释一下如何用定理 2 判定图是否连通。

26. 证明：简单图  $G$  是偶图，当且仅当  $G$  没有包含奇数条边的回路。

## 6.5 欧拉通路与哈密顿通路

### 6.5.1 引言

能否从一个顶点出发沿着图的边前进，恰好经过图的每条边一次并且回到这个顶点？同样，能否从一个顶点出发沿着图的边前进，恰好经过图的每个顶点一次并且回到这个顶点？虽然这两个问题有相似之处，但是对于所有的图来说，可以轻而易举地回答第一个关于是否具有欧拉回路的问题，却非常难以解决第二个关于是否具有哈密顿回路的问题。本节将研究这些问题并且讨论这些问题的难点。虽然两个问题在许多不同领域里都有实际应用，但是都来源于古老的智力题。下面将介绍这些古老的智力题以及现代的实际应用。

### 6.5.2 欧拉通路与欧拉回路

普鲁士的哥尼斯堡镇(现名加里宁格勒，属于俄罗斯共和国)被普雷格尔河支流分成四部分。这四部分包括河两岸、河中心岛以及两条支流之间的部分。在 18 世纪这四部分用七座桥连接起来，如图 6-48 所示。

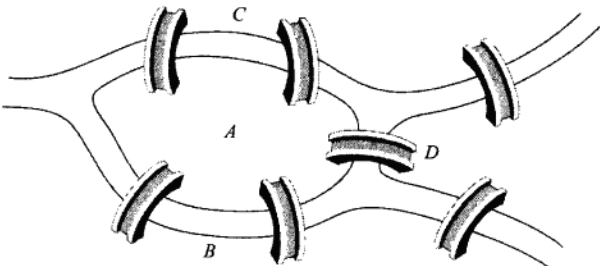


图 6-48 哥尼斯堡七桥



镇上的人们在周日里穿过镇子进行长距离的散步。他们想弄明白是否可能从镇里某个位置出发不重复地经过所有桥并且返回出发点。

瑞士数学家列昂哈德·欧拉解决了这个问题。他的解答发表在1736年，这也许是人们第一次使用图论。欧拉利用多重图来研究这个问题，其中用顶点表示四个部分，用边表示桥，如图6-49所示。

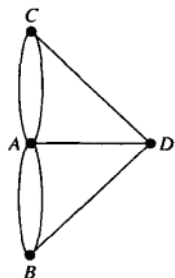


图 6-49 哥尼斯堡镇的多重图模型

不重复地经过每一座桥来旅行的问题可以利用这个模型来重新叙述。问题变成：在这个多重图里是否存在包含每一条边的简单回路？

**定义 1** 图  $G$  里的欧拉回路是包含着  $G$  的每一条边的简单回路。图  $G$  里的欧拉通路是包含着  $G$  的每一条边的简单通路。

例 1 和例 2 解释了欧拉回路和欧拉通路的概念。

**例 1** 在图 6-50 里，哪些无向图具有欧拉回路？在没有欧拉回路的那些图里，哪些具有欧拉通路？

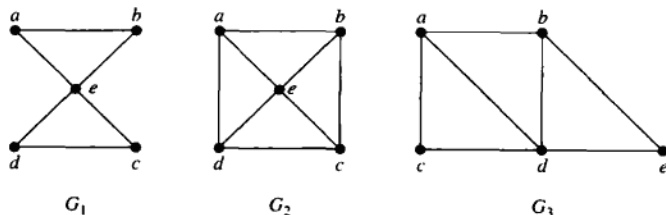


图 6-50 无向图  $G_1$ ,  $G_2$  和  $G_3$

**解** 图  $G_1$  具有欧拉回路，例如  $a, e, c, d, e, b, a$ 。图  $G_2$  和  $G_3$  都没有欧拉回路（读者应当验证它）。但是  $G_3$  具有欧拉通路，即  $a, c, d, e, b, d, a, b$ 。图  $G_2$  没有欧拉通路（读者应当验证它）。 ■

**例 2** 在图 6-51 里，哪些有向图具有欧拉回路？在没有欧拉回路的那些图里，哪些具有欧拉通路？

**解** 图  $H_2$  具有欧拉回路，例如  $a, g, c, b, g, e, d, f, a$ 。图  $H_1$  和  $H_3$  都没有欧拉回路（读者应当验证它）。图  $H_3$  具有欧拉通路，即  $c, a, b, c, d, b$ ，但是  $H_1$  没有欧拉通路（读者应当验证它）。 ■

**欧拉回路和欧拉通路的充要条件** 对判断多重图是否具有欧拉回路和欧拉通路来说，存在着简单的标准。欧拉在解决著名的哥尼斯堡七桥问题时发现了它们。假设在本节里讨论的所有图都具有无穷多个顶点和边。

若一个连通多重图具有欧拉回路，则它有什么性质呢？可以说明的是：每一个顶点都必有偶数条边。为此，首先注意一条欧拉回路从顶点  $a$  开始，接着是  $a$  关联的一条边，比方说  $\{a, b\}$ 。边  $\{a, b\}$  为  $\deg(a)$  贡献 1 度。这条回路每次经过一个顶点就为该顶点的度贡献 2 度，这是因为这条回路经过关联该顶点的边进入，又经过另一条这样的边离开。最后，这条回路在它开始的地方结束，为  $\deg(a)$  贡献 1 度。因此  $\deg(a)$  必为偶数，

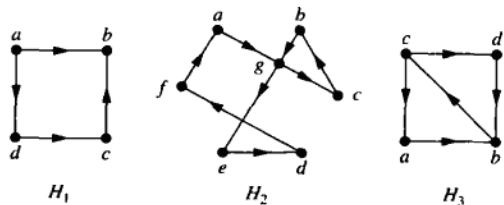


图 6-51 有向图  $H_1$ ,  $H_2$  和  $H_3$

这是因为当回路开始时它贡献 1 度，当回路结束时它贡献 1 度，每次经过  $a$  都贡献 2 度（如果它又经过了  $a$ ）。除  $a$  外的其余顶点都有偶数度，这是因为每次回路经过一个顶点就为该顶点的度贡献 2 度。由此得出结论若连通图有欧拉回路，则每一个顶点必有偶数度。

欧拉回路的存在性的这个必要条件是否也是充分的？即若在连通多重图里所有顶点都有偶数

度, 则是否必有欧拉回路? 这个问题可以通过构造来解决。

假设  $G$  是连通多重图而且  $G$  的每一个顶点都有偶数度。构造从  $G$  的任意顶点  $a$  开始的简单回路。设  $x_0 = a$ 。首先任意地选择一条关联  $a$  的边  $\{x_0, x_1\}$ 。通过建立尽量长的简单通路  $\{x_0, x_1\}$ ,  $\{x_1, x_2\}$ ,  $\dots$ ,  $\{x_{n-1}, x_n\}$  来继续构造。例如, 在图 6-52 的图  $G$  里, 从  $a$  开始而且连续地选择边  $\{a, f\}$ ,  $\{f, c\}$ ,  $\{c, b\}$  和  $\{b, a\}$ 。

这样的通路必然结束, 这是因为图的边数是有穷的。它在  $a$  上以形如  $\{a, x\}$  的边开始, 而且在  $a$  上以形如  $\{y, a\}$  的边结束。这是因为通路每次经过一个偶数度顶点时, 它只用 1 条边进入这个顶点, 所以至少还剩下 1 条边让通路离开这个顶点。这条通路可能用完了所有的边, 也可能没用完。

若所有的边都已经用完, 则欧拉回路已经构造好了。否则, 考虑通过从  $G$  里删除已经用过的边和不关联任何剩余的边的顶点这样所得到的子图  $H$ 。当从图 6-52 里的图删除回路  $a, f, c, b, a$  时, 就得到标记为  $H$  的子图。

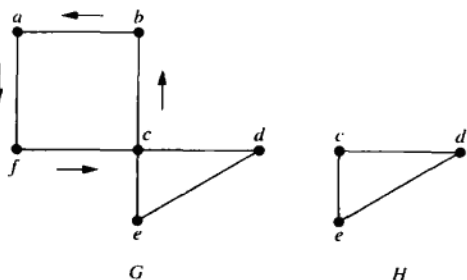


图 6-52 构造  $G$  里的欧拉回路

因为  $G$  是连通的, 所以  $H$  与已经删除的回路至少具有 1 个公共顶点。设  $w$  是这样的顶点(此例中  $c$  是这个顶点)。

$H$  里的每一个顶点都有偶数度。(因为  $G$  里的所有顶点都有偶数度, 对每个顶点来说, 把与这个顶点关联的边成对地删除, 以便形成  $H$ 。)注意  $H$  可能是不连通的。像在  $G$  里做过的那样, 在  $w$  上开始, 通过尽可能地选择边来构造  $H$  里的简单回路。这条回路必然在  $w$  上结束。例如, 在图 6-52 里,  $c, d, e, c$  是  $H$  里的回路。下一步通过把  $H$  里的回路与  $G$  里原来的回路拼接起来形成  $G$  里的回路(这是可行的, 因为  $w$  是这个回路的顶点之一)。当在图 6-52 的图里这样做时, 就得到回路  $a, f, c, d, e, c, b, a$ 。

继续进行这个过程, 直到已经用完了所有的边为止(这个过程必然结束, 这是因为图里只有有穷的边数)。这样就产生出欧拉回路。这样的构造说明, 若连通多重图的顶点都有偶数度, 则该图具有欧拉回路。

把这些结果总结成定理 1。

**定理 1** 连通多重图具有欧拉回路当且仅当它的每个顶点都有偶数度。

现在可以解决哥尼斯堡七桥问题了。因为图 6-49 所示的表示这些桥的多重图具有 4 个奇数度顶点, 所以它没有欧拉回路。从给定点开始、恰好经过每座桥一次并返回开始点是无法实现的。

算法 1 给出了在定理 1 之前的讨论里所给的求欧拉回路的构造过程。(因为这个过程里的回路是任意地选择的, 所以存在一些不确定性。这里不再通过更精确地说明过程的步骤来消除这些不确定性。)

#### 算法 1 构造欧拉回路

**procedure** Euler( $G$ : 所有顶点有偶数度的连通多重图)

$circuit$  := 在  $G$  里任选的顶点开始, 连续地加入边所形成的回到该顶点的回路

$H$  := 删除这条回路的边之后的  $G$

**while**  $H$  还有边

**begin**

$subcircuit$  := 在既是  $H$  里的顶点也是  $circuit$  的边的端点处开始的  $H$  里的一条回路

$H$  := 删除  $subcircuit$  的边和所有孤立点之后的  $H$

$circuit$  := 在适当顶点上插入  $subcircuit$  之后的  $circuit$

**end**{ $circuit$  是欧拉回路}

下一个例子说明如何利用欧拉通路和欧拉回路来解决一种类型的智力题。

**例3** 有许多智力题要求用铅笔连续移动, 不离开纸面并且不重复地画出图形。利用欧拉回路和欧拉通路来解决这样的智力题。例如, 能否用这样的方法画出图 6-53 所示的穆罕默德短弯刀? 其中图形是在同一个顶点上开始和结束。

**解** 可以解决这个问题, 因为图 6-53 所示的图  $G$  具有欧拉回路。它具有这样的回路是因为它的所有顶点都有偶数度。用算法 1 来构造欧拉回路。首先形成回路  $a, b, d, c, b, e, i, f, e, a$ 。通过删除这条回路的边并且删除因此产生的孤立点, 就得到子图  $H$ 。然后形成  $H$  里的回路  $d, g, h, j, i, h, k, g, f, d$ 。形成这条回路之后就用完了  $G$  里的所有边。在适当的地方拼接这条回路和第一条回路, 就产生出欧拉回路  $a, b, d, g, h, j, i, h, k, g, f, d, c, b, e, i, f, e, a$ 。这条回路给出了铅笔不离开纸面并且不重复地画出弯刀的方法。

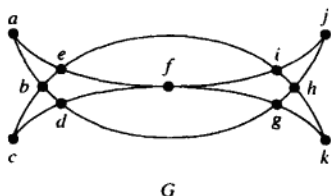


图 6-53 穆罕默德短弯刀

构造欧拉回路的另一个算法称为弗勒里算法, 在本节末尾的练习里描述它。

现在说明, 连通多重图具有欧拉通路(不是欧拉回路)当且仅当它恰有 2 个奇数度顶点。首先假设连通多重图有从  $a$  到  $b$  的欧拉通路, 但不是欧拉回路。通路的第一条边为  $a$  的度贡献 1 度。通路每次经过  $a$  就为  $a$  的度贡献 2 度。通路的最后一条边为  $b$  的度贡献 1 度。通路每次经过  $b$  就为  $b$  的度贡献 2 度。所以  $a$  和  $b$  的度都是奇数。每一个其他顶点都具有偶数度, 这是因为每当通路经过一个顶点时, 就为这个顶点的度贡献 2 度。

现在反过来考虑。假设这个图恰有 2 个奇数度顶点, 比方说  $a$  和  $b$ 。考虑由原来的图和边  $\{a, b\}$  所组成的更大的图。这个更大的图的每一个顶点都有偶数度, 所以具有欧拉回路。删除新边就产生原图的欧拉通路。定理 2 总结了这些结果。

**定理 2** 连通多重图具有欧拉通路但无欧拉回路当且仅当它恰有 2 个奇数度顶点。

**例 4** 图 6-54 所示的哪些图具有欧拉通路?

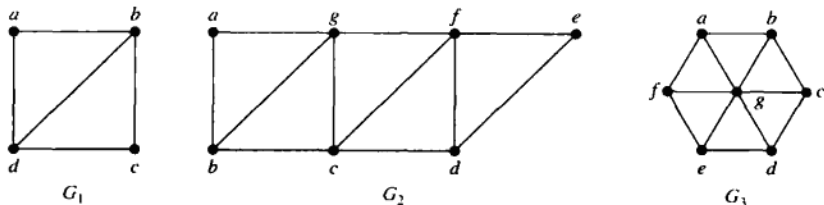


图 6-54 三个无向图

**解**  $G_1$  恰有 2 个奇数度顶点, 即  $b$  和  $d$ 。因此它具有必须用  $b$  和  $d$  作为端点的欧拉通路。一条这样的欧拉通路是  $d, a, b, c, d, b$ 。同理,  $G_2$  恰有 2 个奇数度顶点, 即  $b$  和  $d$ 。因此它具有必须用  $b$  和  $d$  作为端点的欧拉通路。一条这样的欧拉通路是  $b, a, g, f, e, d, c, g, b, c, f, d$ 。 $G_3$  没有欧拉通路, 因为它具有 6 个奇数度顶点。

回到 18 世纪的哥尼斯堡, 是否有可能在镇里某点开始, 旅行经过所有的桥, 在镇里其他某点结束? 通过判定表示哥尼斯堡七桥的多重图是否具有欧拉通路, 就可以回答这个问题。因为这个多重图有四个奇数度顶点, 没有欧拉通路, 所以这样的旅行是不可能的。

有向图里欧拉通路和欧拉回路的充要条件, 在本节末尾的练习里讨论。

可以用欧拉通路和欧拉回路解决许多实际问题。例如, 许多应用都要求一条通路或回路, 它要恰好一次地经过一个街区里的每条街道、一个交通网里的每条道路、一个高压输电网里的每个连接或者一个通信网络里的每个链接。求出适当的图模型里的欧拉通路或欧拉回路就可以解决这样的问题。例如, 如果一个邮递员可以求出表示他所负责投递的街道的图中的欧拉通路, 则这条通路就产生恰好一次经过每条街道的投递路线。这个问题称为中国邮递员问题。

以纪念在1962年提出这个问题的中国科学家管梅谷。参看[MiRo91]了解关于不存在欧拉通路时中国邮递员问题的解的更多信息。

应用欧拉通路和欧拉回路的其他领域有：电路布线、网络组播和分子生物学，在分子生物学中欧拉通路用于DNA测序。

### 6.5.3 哈密顿通路与哈密顿回路

包含多重图每一条边恰好一次的通路和回路的存在性的充要条件已经得出。那么包含图的每一个顶点恰好一次的简单通路和回路的存在性的充要条件是否也能得出呢？

**定义2** 在图  $G=(V, E)$  里，若  $V=\{x_0, x_1, \dots, x_{n-1}, x_n\}$  并且对  $0 \leq i < j \leq n$  来说有  $x_i \neq x_j$ ，则通路  $x_0, x_1, \dots, x_{n-1}, x_n$  称为哈密顿通路。在图  $G=(V, E)$  里，若  $x_0, x_1, \dots, x_{n-1}, x_n$  是哈密顿通路，则  $x_0, x_1, \dots, x_{n-1}, x_n, x_0$  (其中  $n > 1$ ) 称为哈密顿回路。

这个术语来自爱尔兰数学家威廉·罗万·哈密顿爵士在1857年发明的智力题。哈密顿的智力题用到了木质十二面体(如图6-55a所示，十二面体有12个正五边形表面)，十二面体每个顶点上的钉子，以及细线。十二面体的20个顶点用世界上的不同城市作标记。智力题要求从一个城市开始，沿十二面体的边旅行，访问其他19个城市每个恰好一次，回到第一个城市结束。旅行经过的回路用钉子和细线来标记。

因为作者不可能向每位读者提供带钉子和细线的木质十二面体，所以考虑一个等价的问题：图6-55b中的图是否具有恰好经过每个顶点一次的回路？它就是对原题的解，因为该图同构于包含十二面体顶点和边的图。图6-56是哈密顿智力题的解。

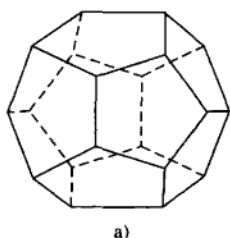


图 6-55 哈密顿的“周游世界”智力题

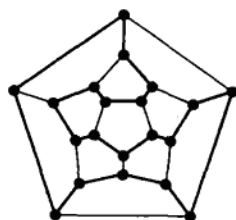
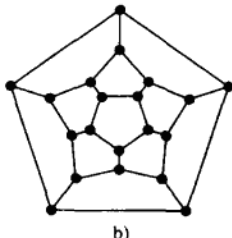


图 6-56 “周游世界”智力题的解

**例5** 在图6-57里，哪些简单图具有哈密顿回路？或者没有哈密顿回路但是有哈密顿通路？

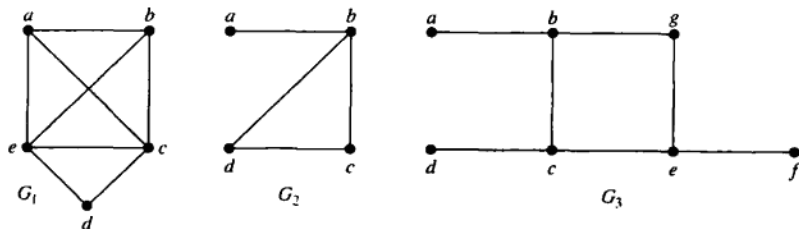


图 6-57 三个简单图

**解**  $G_1$  有哈密顿回路： $a, b, c, d, e, a$ 。 $G_2$  没有哈密顿回路(从包含每一个顶点的任何回路必然两次包含边  $\{a, b\}$  可以看出这一点)，但是  $G_2$  确实有哈密顿通路，即  $a, b, c, d$ 。 $G_3$  既无哈密顿回路也无哈密顿通路，这是因为包含所有顶点的任何通路都必然多次包含边  $\{a, b\}$ ， $\{e, f\}$  和  $\{c, d\}$  其中之一。

是否存在简单方式来判定一个图有无哈密顿回路或哈密顿通路？初看起来，似乎应当有判定这一点的简单方式，因为存在简单方式来回答一个图有无欧拉回路这样的相似问题。令人吃惊的

是, 没有已知的简单的充要条件来判定哈密顿回路的存在性。不过, 已经有许多定理对哈密顿回路的存在性给出了充分条件。另外, 某些性质可以用来证明一个图没有哈密顿回路。例如, 带有 1 度顶点的图没有哈密顿回路, 因为在哈密顿回路里每个顶点都关联着回路里的两条边。另外, 若图中有 2 度顶点, 则关联这个顶点的两条边属于任意一条哈密顿回路。此外注意, 当构造哈密顿回路而且该回路经过某一个顶点时, 除了回路所用到的两条边以外, 这个顶点所关联的其他所有边不用再考虑。而且, 哈密顿回路里不能包含更小的回路。

**例 6** 证明图 6-58 里的图都没有哈密顿回路。

**解**  $G$  里没有哈密顿回路, 因为  $G$  有 1 度顶点, 即  $e$ 。现在考虑  $H$ 。因为顶点  $a, b, d$  和  $e$  的度都为 2, 所以这些顶点关联的每一条边都必然属于任意一条哈密顿回路。现在容易看出  $H$  里不存在哈密顿回路, 因为任何这样的哈密顿回路都不得不包含 4 条关联  $c$  的边, 这是不可能的。

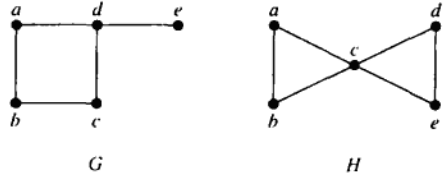


图 6-58 两个没有哈密顿回路的图

**例 7** 证明: 每当  $n \geq 3$  时,  $K_n$  就有哈密顿回路。

**解** 从  $K_n$  里的任意一个顶点开始来形成哈密顿回路。以所选择的任意顺序来访问顶点, 只要求通路在同一个顶点开始和结束, 而且对其他每个顶点恰好访问一次。这样做是可能的, 因为在  $K_n$  里任意两个顶点之间都有边。

虽然还未发现任何有用的关于哈密顿回路存在性的充要条件, 但很多充分条件已经找到了。注意到一个图的边越多, 这个图就越可能有哈密顿回路。另外, 加入边(而不是顶点)到已经有哈密顿回路的图中就产生有相同哈密顿回路的图。因此, 当加入边到图中时, 特别是当确保给每个顶点都加入边时, 这个图存在哈密顿回路的可能性就更大了。这里叙述充分条件中最重要的两个。这些条件是加布里尔·A·狄拉克(Gabriel A. Dirac)在 1952 年和奥伊斯坦·奥尔(Oystein Ore)在 1960 年发现的。

**定理 3 狄拉克定理** 如果  $G$  是带  $n$  个顶点的连通简单图, 其中  $n \geq 3$ , 并且  $G$  中每个顶点的度都至少为  $n/2$ , 则  $G$  有哈密顿回路。

**定理 4 奥尔定理** 如果  $G$  是带  $n$  个顶点的连通简单图, 其中  $n \geq 3$ , 并且对于  $G$  中每一对不相邻的顶点  $u$  和  $v$  来说, 都有  $\deg(u) + \deg(v) \geq n$ , 则  $G$  有哈密顿回路。

本节末的练习 27 粗略介绍了奥尔定理的证明。狄拉克定理可以作为奥尔定理的推论来证明, 因为狄拉克定理的条件蕴含了奥尔定理的条件。

奥尔定理和狄拉克定理都给出了连通简单图有哈密顿回路的充分条件。但是, 这些定理没有给出哈密顿回路存在性的必要条件。例如, 图  $C_5$  有哈密顿回路, 但既不满足奥尔定理的假设也不满足狄拉克定理的假设, 读者可以验证这一点。

已知的最好的求一个图里的哈密顿回路或判定这样的回路不存在的算法具有指数的最坏情形时间复杂性(相对于图的顶点数来说)。找到具有多项式最坏情形时间复杂性的解法算法是 NP 复杂的, 它的发现将具有重大意义, 因为这意味着其他许多难题都会迎刃而解。

可以用哈密顿通路和哈密顿回路来解决实际问题。例如, 许多应用都要求一条通路或回路, 它要恰好一次地访问一个城市里的每个路口、一个设备网格里的每个管道交汇处或者一个通信网络里的每个节点。求出适当的图模型里的哈密顿通路或哈密顿回路就可以解决这样的问题。著名的旅行商问题要求一个旅行商为了访问一组城市所应当选取的最短路线。这个问题可归结为求完全图的哈密顿回路, 使得这个回路的边的权的总和尽可能小。

现在给出哈密顿回路在编码上的一种不太明显的应用。

**例 8 格雷码** 旋转的指针的位置可以表示成数字的形式。一种方式是把圆周等分成  $2^n$  段弧并且用长度为  $n$  的位串给每段弧赋值。图 6-59 里显示出了用长度为 3 的位串来这样做的两种方式。

用  $n$  个接触点来确定指针位置的数字表示。每个接触点用来读出位置的数字表示中的一位。图 6-60 针对图 6-59 里的两种赋值对此做了解释。

当指针靠近两段弧的边界时，在读出指针位置时可能发生错误。这可能引起读出的位串里有一个大的错误。例如，在图 6-59a 的编码方案里，若在确定指针位置的过程里发生了一个小的错误，则读出的位串是 100 而不是 011。所有三位都是错的！为了把在确定指针位置的过程里的错误的影响降到最低，用位串对  $2^n$  段弧的赋值，应当使相邻的弧所表示的位串只相差一位。在图 6-59b 里的编码方案的情况恰好就是这样的。在确定指针位置的过程里的一个错误误给出的位串为 010 而不为 011。只有一位是错的。

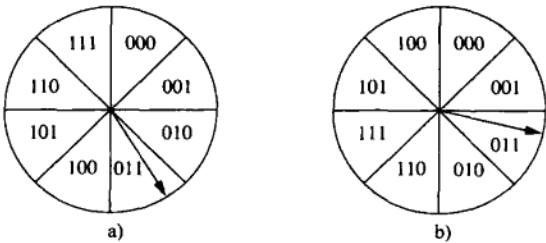


图 6-59 把指针位置转换成数字形式

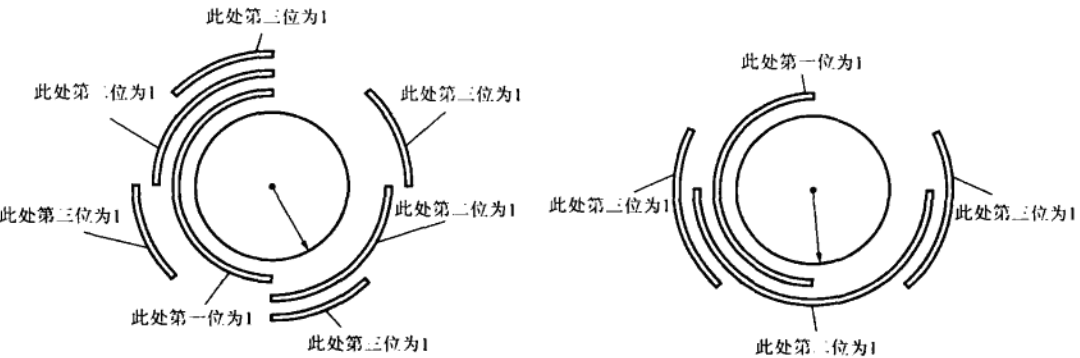


图 6-60 指针位置的数字表示

格雷码是圆周的弧的一种标记，使得相邻的弧具有恰好相差一位的位串标记。在图 6-59b 里的赋值是一个格雷码。可以这样找出格雷码：以下述的方式列出所有长度为  $n$  的位串，使得每一个串与前一个位串恰好相差一位，而且最后一个串与第一个位串恰好相差一位。可以用  $n$  立方体  $Q_n$  来为这个问题建模。解决这个问题所需要的是  $Q_n$  里的一条哈密顿回路。这样的哈密顿回路容易求出。例如， $Q_3$  的一条哈密顿回路显示在图 6-61 里。这条哈密顿回路所产生的前后恰好相差一位的位串序列是 000, 001, 011, 010, 110, 111, 101, 100。

格雷码是以弗兰克·格雷的名字来命名的，20 世纪 40 年代，格雷在贝尔实验室研究如何把数字信号传送过程中错误的影响降到最低时发明了它们。

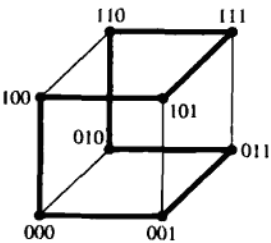
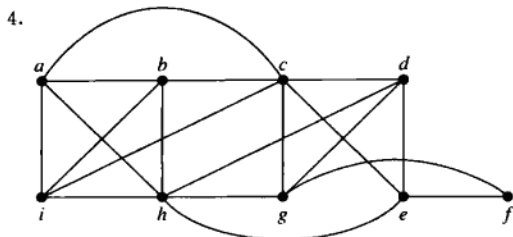
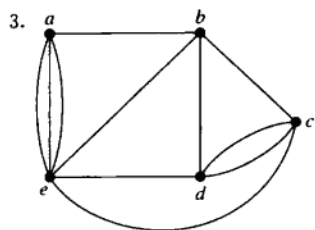


图 6-61  $Q_3$  的哈密顿回路

练习

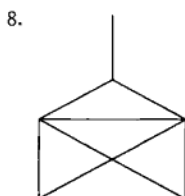
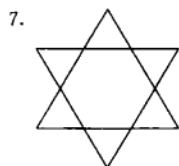
在练习 1~4 里，判定给定的图是否具有欧拉回路。当存在欧拉回路时构造出这样的回路。如果不存在欧拉回路，就确定这个图是否具有欧拉通路，若存在欧拉通路则构造出这样的通路。





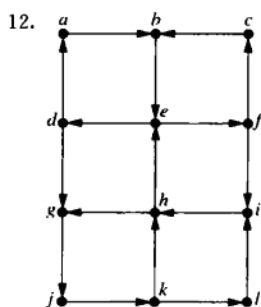
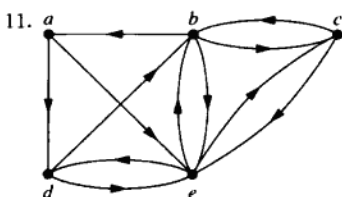
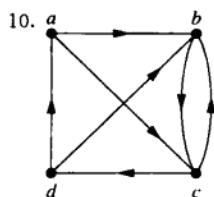
5. 在加里宁格勒(哥尼斯堡现名), 除了18世纪的七座桥之外, 还有另外两座桥。这些新桥分别连接区域B和C以及区域B和D。是否有人能够经过加里宁格勒的九座桥恰好一次并且回到出发点?
6. 何时可以画出一个城市里街道的中心线而不重复经过街道(假设所有街道都是双向街道)?

在练习7~8里, 判定是否可以用一支铅笔连续移动, 不离开纸面并且不重复地画出所示的图形。



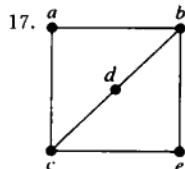
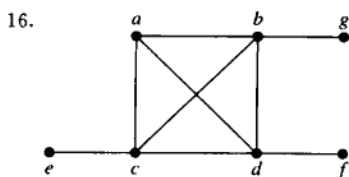
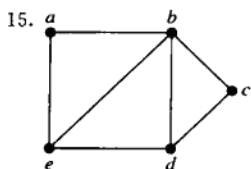
- \*9. 证明: 不带有孤立点的有向多重图具有欧拉通路而没有欧拉回路, 当且仅当该图是弱连通的并且存在两个顶点, 一个顶点的入度比出度大1而另外一个顶点的出度比入度大1, 其余每个顶点的入度与出度都相等。

在练习10~12里, 确定所示的有向图是否具有欧拉回路。若存在欧拉回路, 则构造出一条欧拉回路。如果不存在欧拉回路, 就确定这个有向图是否具有欧拉通路。若存在欧拉通路, 则构造出一条欧拉通路。



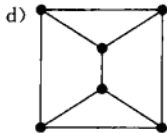
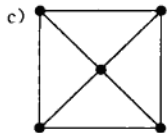
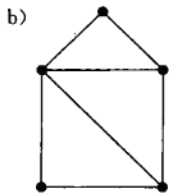
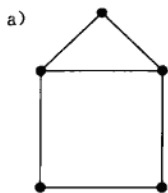
13. 设计一个算法, 构造有向图里的欧拉通路。
14. 当不重复任何部分地画出练习1~4里的每个图时, 求出铅笔必须离开纸面的最少次数。

在练习15~17里, 确定所给的图是否具有哈密顿回路。若有哈密顿回路, 则求出这样一条回路。若没有哈密顿回路, 则论证为什么这样的回路不存在。



18. 对哪些  $m$  和  $n$  值来说, 完全偶图  $K_{m,n}$  具有哈密顿回路?

19. 对于下列每个图确定：(i)能否用狄拉克定理来证明这个图有哈密顿回路；(ii)能否用奥尔定理来证明这个图有哈密顿回路；(iii)这个图是否具有哈密顿回路。



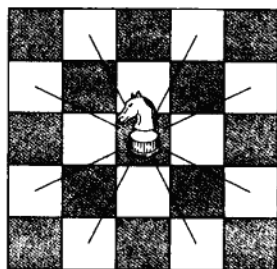
- \*20. 证明：每当  $n$  是正整数时，就存在  $n$  阶格雷码，或者等价地证明： $n > 1$  的  $n$  立方体  $Q_n$  总是具有哈密顿回路。[提示：用数学归纳法。证明如何从  $n-1$  阶格雷码产生  $n$  阶格雷码。]

**例 6.10** 构造欧拉回路的弗留利算法是从连通多重图的任意一个顶点开始，连续地选择边来形成一条回路。一旦选择了一条边，就删除这条边。连续地选择边，使得每条边从上一条边结束的地方开始，而且使得这条边不是一条割边，除非别无选择。

- \*21. 用伪代码表达弗留利算法。  
\*22. 给出变化的弗留利算法来产生欧拉通路。  
23. 证明：带有奇数个顶点的偶图没有哈密顿回路。

马在国际象棋中是这样一种棋子，它的移动可以是水平两格加垂直一格，或者是水平一格加垂直两格。即在  $(x, y)$  格子的马可以移动到 8 个格子  $(x \pm 2, y \pm 1)$ ,  $(x \pm 1, y \pm 2)$  中的任何一个，只要这些格子是在棋盘上，如右图所示。

**例 6.11** 马的周游是马的合法移动的序列，马在某个格子上开始而且访问每个格子恰好一次。若存在一种合法移动，它让马从周游的最后一个格子回到周游开始的地方，则马的周游称为重迭的。可以用图来马的周游建模，棋盘上每个格子都用一个顶点来表示，若马可以在两个顶点所表示的格子之间合法地移动，则用一条边连接这两个顶点。



24. 画出表示马在  $3 \times 4$  棋盘上的合法移动的图。  
\*25. 证明：存在马在  $3 \times 4$  棋盘上的周游。  
\*26. 证明：不存在马在  $4 \times 4$  棋盘上的周游。  
27. 本练习粗略介绍奥尔定理的证明。假设  $G$  是带有  $n$  个顶点的简单图， $n \geq 3$ ，并且每当  $x$  和  $y$  是  $G$  中不相邻的顶点时， $\deg(x) + \deg(y) \geq n$ 。奥尔定理称在这些条件下  $G$  有哈密顿回路。  
a) 证明：如果  $G$  没有哈密顿回路，则存在另一个带有与  $G$  相同顶点的图  $H$ ，可以这样来构造  $H$ ：加入边到  $G$  使得再加入一条边就产生  $H$  中的哈密顿回路。[提示：依次在  $G$  的每个顶点处加入不产生哈密顿回路的尽可能多的边。]  
b) 证明：在  $H$  中存在哈密顿通路。  
c) 设  $v_1, v_2, \dots, v_n$  是  $H$  中的哈密顿通路。证明： $\deg(v_1) + \deg(v_n) \geq n$  并且至多存在  $\deg(v_1)$  个顶点不与  $v_n$  相邻(包括  $v_n$  在内)。  
d) 设  $S$  是与哈密顿通路上  $v_1$  相邻的每个顶点前面的顶点的集合。证明  $S$  包含  $\deg(v_1)$  个顶点并且  $v_n \notin S$ 。  
e) 证明： $S$  包含与  $v_n$  相邻的顶点  $v_k$ 。这蕴含着存在连接  $v_1$  与  $v_{k+1}$  和  $v_k$  与  $v_n$  的边。  
f) 证明：(e) 蕴含着  $v_1, v_2, \dots, v_{k-1}, v_k, v_n, v_{n-1}, \dots, v_{k+1}, v_1$  是  $G$  中的哈密顿回路。从这个矛盾得出奥尔定理成立。

## 6.6 最短通路问题

### 6.6.1 引言

许多问题可以用边上赋权的图来建模。作为说明，考虑一下航线系统如何建模。如果用顶点



表示城市，用边表示航班。给边赋上城市之间的距离，就可以为涉及距离的问题建模；给边赋上飞行时间，就可以为涉及飞行时间的问题建模；给边赋上票价，就可以为涉及票价的问题建模。图 6-62 显示了给一个图的边赋权的三种不同方式，分别表示距离、飞行时间和票价。

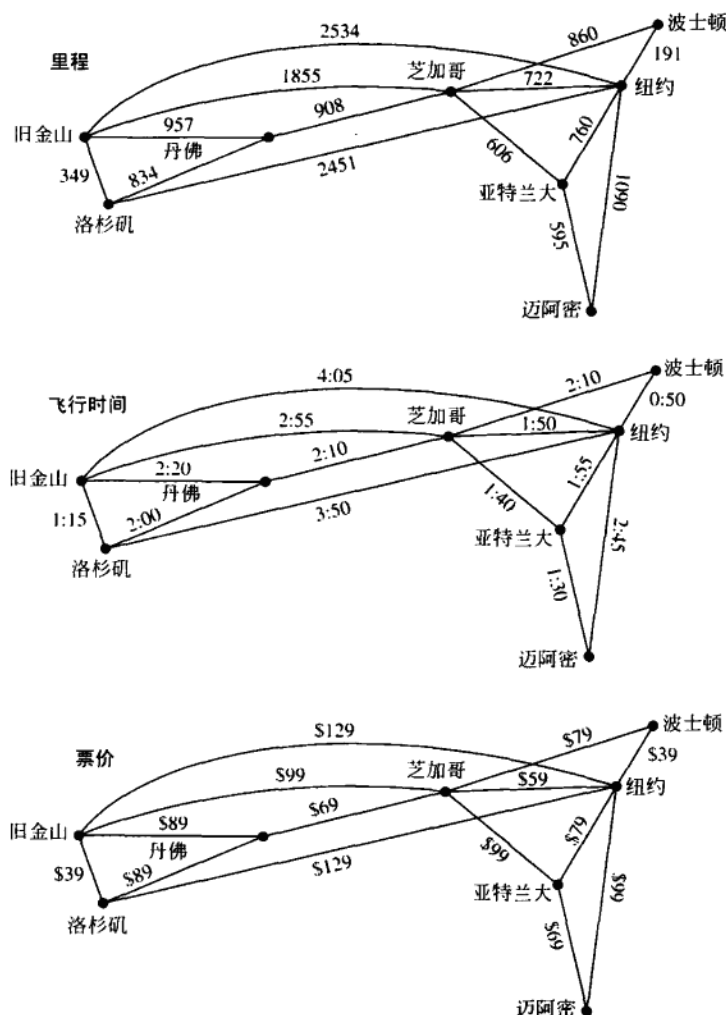


图 6-62 为航线系统建模的带权图

给每条边赋上一个数的图称为带权图。带权图用来为计算机网络建模。通信成本(比如租用电话线的月租费)、计算机在这些线路上的响应时间或计算机之间的距离等都可以用带权图来研究。图 6-63 显示三个带权图，它们表示给计算机网络的图的边赋权的三种方式，分别对应于距离、响应时间和成本。

与带权图有关的几种类型的问题出现得很多。确定网络里两个顶点之间长度最短的通路就是一个这样的问题。说得更具体些，设带权图里一条通路的长度是这条通路上各边的权的总和。(读者应当注意，对术语长度的这种用法，与表示不带权的图的通路里边数的的长度的用法，这两者是不同的。)问题是：什么是最短通路，即什么是在两个给定顶点之间长度最短的通路？例如，在图

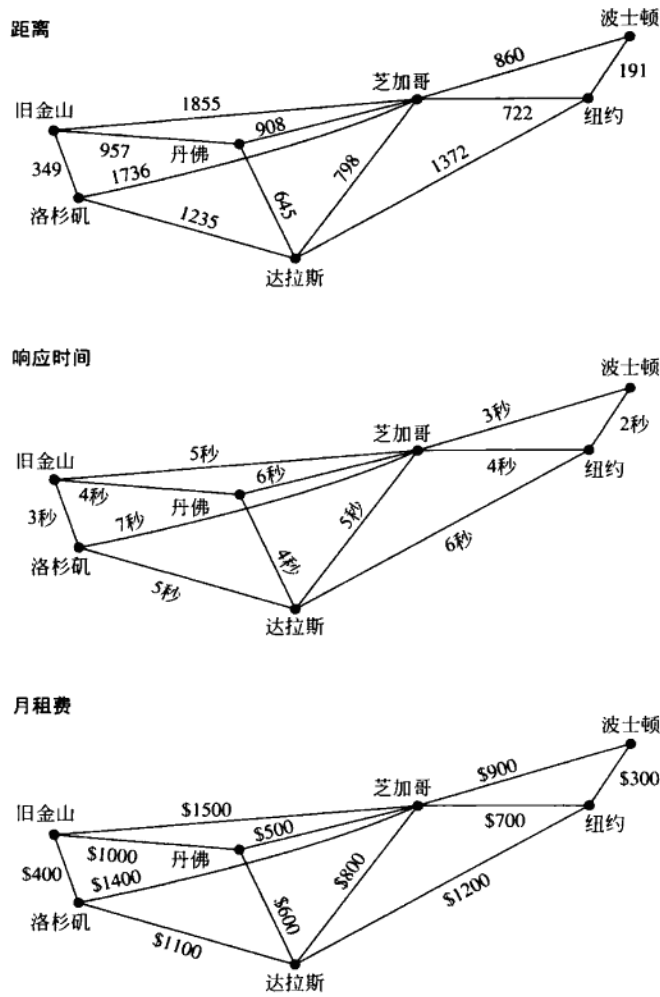


图 6-63 为计算机网络建模的带权图

6-62 所示带权图表示的航线系统里，在波士顿与洛杉矶之间以空中距离计算的最短通路是什么？在波士顿与洛杉矶之间什么样的航班组合的总飞行时间（即在空中的总时间，不包括航班之间的时间）最短？在这两个城市之间的最低费用是多少？在图 6-63 所示的计算机网络里，连接旧金山的计算机与纽约的计算机所需要的最便宜的一组电话线是什么？哪一组电话线给出旧金山与纽约之间的通信的最短响应时间？哪一组电话线有最短的总距离？

与带权图有关的另外一个重要问题是：求访问完全图每个顶点恰好一次的、总长度最短的回路。这就是著名的旅行商问题，它求一位推销员应当以什么样的顺序来访问其路程上的每个城市恰好一次，使得他旅行的总距离最短。本节后面将讨论旅行商问题。

6.6.2 最短通路算法

求带权图里两个顶点之间的最短通路有几个不同的算法。下面将给出荷兰数学家 E·迪克斯特拉在 1959 年所发现的一个解决无向带权图中最短通路问题的算法，其中所有的

权都是正数。可以很容易地将它修改后来解决有向图里的最短通路问题。

在给出这个算法的形式化表示之前将给出一个启发性的例子。

**例 1** 在图 6-64 所示的带权图里,  $a$  和  $z$  之间的最短通路的长度是多少?

**解** 虽然通过观察就容易求出最短通路, 但是需要想出一些有助于理解迪克斯特拉算法的办法。要解决的问题就是: 求从  $a$  到各个相继的顶点的最短通路, 直到到达  $z$  为止。

从  $a$  开始, (直到到达终点为止) 不包含除  $a$  以外的顶点的唯一通路是  $a, b$  和  $a, d$ 。因为  $a, b$  和  $a, d$  的长度分别为 4 和 2, 所以  $d$  是与  $a$  最靠近的顶点。

可以通过查看 (直到到达终点为止) 只经过  $a$  和  $d$  的所有通路, 来求出下一个最靠近  $a$  的顶点。到  $b$  的最短通路仍然是  $a, b$ , 长度为 4, 而到  $e$  的最短通路是  $a, d, e$ , 长度为 5。所以, 下一个与  $a$  最靠近的顶点是  $b$ 。

为了找出第三个与  $a$  最靠近的顶点, 只需要检查 (直到到达终点为止) 只经过了  $a, d$  和  $b$  的那些通路。存在长度为 7 到  $c$  的通路, 即  $a, b, c$ , 以及长度为 6 到  $z$  的通路, 即  $a, d, e, z$ 。所以,  $z$  是下一个与  $a$  最靠近的顶点, 而且到  $z$  的最短通路的长度为 6。

例 1 说明了在迪克斯特拉算法里使用的一般原理。注意通过检查就可能求出从  $a$  到  $z$  的最短通路。不过, 无论对人还是对计算机来说, 检查边数很多的图都是不切实际的。

现在将考虑一般问题: 在无向连通简单带权图里, 求出  $a$  与  $z$  之间的最短通路的长度。迪克斯特拉算法如下进行: 求出从  $a$  到第一个顶点的最短通路的长度, 从  $a$  到第二个顶点的最短通路的长度, 依此类推, 直到求出从  $a$  到  $z$  的最短通路的长度为止。

这个算法依赖于一系列的迭代。通过在每次迭代里添加一个顶点来构造出特殊顶点的集合。在每次迭代里完成一个标记过程。在这个标记过程里, 用只包含特殊顶点的从  $a$  到  $w$  的最短通路的长度来标记  $w$ 。添加到特殊顶点集里的顶点是尚在集合之外的那些顶点中带有最小标记的顶点。

现在给出迪克斯特拉算法的细节。它首先用 0 标记  $a$  而用  $\infty$  标记其余的顶点。用记号  $L_0(a) = 0$  和  $L_0(v) = \infty$  表示在没有发生任何迭代之前的这些标记 (下标 0 表示“第 0 次”迭代)。这些标记是从  $a$  到这些顶点的最短通路的长度, 其中这些通路只包含顶点  $a$ 。(因为不存在从  $a$  到其他顶点的这种通路, 所以  $\infty$  是  $a$  与这样的顶点之间的最短通路的长度。)

迪克斯特拉算法是通过形成特殊顶点的集合来进行的。设  $S_k$  表示在标记过程  $k$  次迭代之后的特殊顶点集。首先让  $S_0 = \emptyset$ 。集合  $S_k$  是通过把不属于  $S_{k-1}$  的带最小标记的顶点  $u$  添加到  $S_{k-1}$  里形成的。一旦把  $u$  添加到  $S_k$  里, 就更新所有不属于  $S_k$  的顶点的标记, 使得顶点  $v$  在第  $k$  个阶段的标记  $L_k(v)$  是只包含  $S_k$  里顶点 (即已有的特殊顶点再加上  $u$ ) 的、从  $a$  到  $v$  的最短通路的长度。

设  $v$  是不属于  $S_k$  的一个顶点。为了更新  $v$  的标记, 注意  $L_k(v)$  是只包含  $S_k$  里顶点的从  $a$  到  $v$  的最短通路的长度。当利用下面的观察结果时, 就可以有效地完成这个更新: 只包含  $S_k$  里顶点的从  $a$  到  $v$  的最短通路, 要么是只包含  $S_{k-1}$  里顶点 (即不包括  $u$  在内的特殊顶点) 的从  $a$  到  $v$  的最短通路, 要么是在第  $k-1$  阶段加上边  $(u, v)$  的从  $a$  到  $u$  的最短通路。换句话说,

$$L_k(a, v) = \min\{L_{k-1}(a, v), L_{k-1}(a, u) + w(u, v)\}$$

这个过程这样迭代: 相继添加顶点到特殊顶点集里, 直到添加  $z$  为止。当把  $z$  添加到特殊顶点集里时, 它的标记就是从  $a$  到  $z$  的最短通路的长度。在算法 1 里给出迪克斯特拉算法。随后将证明这个算法的正确性。

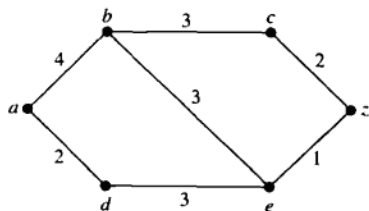


图 6-64 一个带权的简单图

**算法1 迪克斯特拉算法****procedure** *Dijkstra*( $G$ : 所有权都为正数的带权连通简单图){ $G$ 带有顶点 $a=v_0, v_1, \dots, v_n=z$ 和权 $w(v_i, v_j)$ , 其中若 $\{v_i, v_j\}$ 不是 $G$ 里的边, 则 $w(v_i, v_j)=\infty$ }**for**  $i := 1$  **to**  $n$  $L(v_i) := \infty$  $L(a) := 0$  $S := \emptyset$ {现在初始化标记, 使得 $a$ 的标记为0而所有其余标记为 $\infty$ ,  $S$ 是空集合}**while**  $z \notin S$ **begin** $u :=$  不属于 $S$ 的 $L(u)$ 最小的一个顶点 $S := S \cup \{u\}$ **for** 所有不属于 $S$ 的顶点 $v$ **if**  $L(u) + w(u, v) < L(v)$  **then**  $L(v) := L(u) + w(u, v)$ {这样就给 $S$ 里添加带最小标记的顶点, 并且更新不在 $S$ 里的顶点的标记}**end**{ $L(z)$ 是从 $a$ 到 $z$ 的最短通路的长度}

下面的例子说明迪克斯特拉算法是如何工作的。随后我们将证明这个算法总是产生带权图里两个顶点之间最短通路的长度。

**例2** 用迪克斯特拉算法求图6-65a所示的带权图里顶点 $a$ 与 $z$ 之间最短通路的长度。

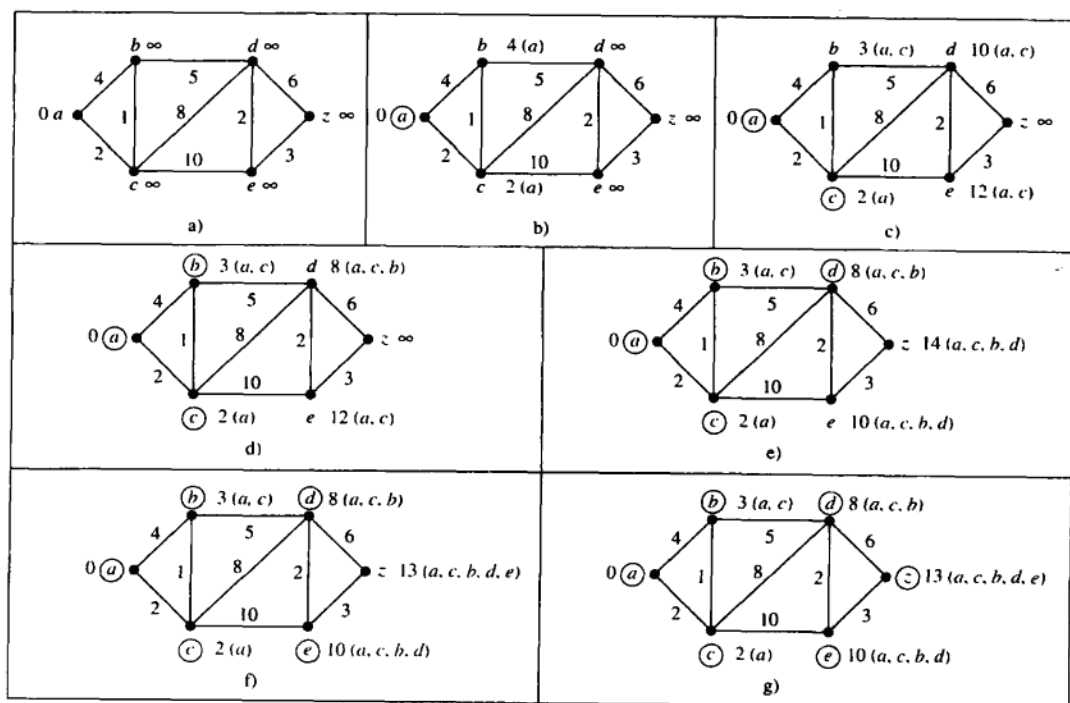


图 6-65 用迪克斯特拉算法求从  $a$  到  $z$  的最短通路

**解** 图 6-65 里显示了迪克斯特拉算法求  $a$  与  $z$  之间最短通路所用的步骤。在算法的每次迭代里, 用圆圈圈起集合  $S_i$  里的顶点。对每次迭代都标明了只包含  $S_i$  里顶点的从  $a$  到每个顶点的最短

通路。当圆圈圈到  $z$  时, 算法终止。找到从  $a$  到  $z$  的最短通路是  $a, c, b, d, e, z$ , 长度为 13。■

**注意** 在执行迪克斯特拉算法时, 为了更便于在每步里跟踪顶点的标记, 有时可以用一个表来代替, 而不再对每步都重新画出这个图。

下一步, 用归纳论证来证明迪克斯特拉算法产生无向连通带权图里两个顶点  $a$  与  $z$  之间最短通路的长度。用下列断言作为归纳假设: 在第  $k$  次迭代里

(i)  $S$  里的顶点  $v(v \neq 0)$  的标记是从  $a$  到这个顶点的最短通路的长度。

(ii) 不在  $S$  里的顶点的标记是(这个顶点自身除外)只包含  $S$  里顶点的、从  $a$  到这个顶点的最短通路的长度。

当  $k=0$  时, 在没有执行任何迭代之前,  $S=\{a\}$ , 所以从  $a$  到除  $a$  外的顶点的最短通路的长度是  $\infty$ 。设  $v$  是在第  $k+1$  次迭代里添加到  $S$  里的顶点, 使得  $v$  是在第  $k$  次迭代结束时带最小标记的不在  $S$  里的顶点(在该顶点不唯一的情形里, 可以采用带最小标记的任意顶点)。

根据归纳假设, 可以看出在第  $k+1$  次迭代之前,  $S$  里的顶点都用从  $a$  出发的最短通路的长度来标记。另外, 必须用从  $a$  到  $v$  的最短通路的长度来标记  $v$ 。假如情况不是这样, 那么在第  $k$  次迭代结束时, 就可能存在包含不在  $S$  里的顶点的、长度小于  $L_k(v)$  的通路(因为  $L_k(v)$  是在第  $k$  次迭代之后、只包含  $S$  里顶点的、从  $a$  到  $v$  的最短通路的长度)。设  $u$  是在这样的通路里不属于  $S$  的第一个顶点。则存在一条只包含  $S$  里顶点的、从  $a$  到  $u$  的长度小于  $L_k(v)$  的通路。这与对  $v$  的选择相矛盾。因此, 在第  $k+1$  次迭代结束时(i)成立。

设  $u$  是在第  $k+1$  次迭代之后不属于  $S$  的一个顶点。只包含  $S$  里顶点的从  $a$  到  $u$  的最短通路要么包含  $v$ 、要么不包含  $v$ 。若它不包含  $v$ , 则根据归纳假设, 它的长度是  $L_k(u)$ 。若它确实包含  $v$ , 则它必然是这样组成的: 一条只包含  $S$  里除  $v$  之外的顶点的、从  $a$  到  $v$  的、具有最短可能长度的通路, 后面接着从  $v$  到  $u$  的边。在这种情形里它的长度是  $L_k(v) + w(v, u)$ 。这样就证明了(ii)为真, 因为  $L_{k+1}(u) = \min\{L_k(u), L_k(v) + w(v, u)\}$ 。

已经证明了定理 1。

**定理 1** 迪克斯特拉算法求出连通简单无向带权图里两个顶点之间最短通路的长度。

现在可以估计迪克斯特拉算法的计算复杂性(就加法和比较而言)。这个算法使用的迭代次数不超过  $n-1$  次, 因为在每次迭代里添加一个顶点到特殊顶点集里。若可以估计每次迭代所使用的运算次数, 则大功告成了。可以用不超过  $n-1$  次比较来找出不在  $S_k$  里的带最小标记的顶点。于是我们使用一次加法和一次比较来更新不在  $S_k$  中的每一个顶点的标记, 所以在每次迭代里的运算不超过  $2(n-1)$  次, 因为在每次迭代里要更新的标记不超过  $n-1$  个。因为迭代次数不超过  $n-1$  次, 每次迭代的运算次数不超过  $2(n-1)$  次, 所以有下面的定理。

**定理 2** 迪克斯特拉算法使用  $O(n^2)$  次运算(加法和比较)来求出连通简单无向带权图里两个顶点之间最短通路的长度。

### 6.6.3 旅行商问题

 现在讨论与带权图有关的一个重要问题。考虑下面的问题: 一位旅行商想要访问  $n$  个城市中每个城市恰好一次, 并且返回到出发点。例如, 假定这个旅行商想要访问底特律、托莱多、萨吉诺、格兰特·拉皮滋以及卡拉玛祖(见图 6-66)。他应当以什么顺序访问这些城市以便旅行总距离最短? 为了解决这个问题, 可以假定旅行商从底特律出发(因为这个城市必须是回路的一部分), 并且检查他访问其余四个城市然后返回底特律的所有可能方式(从别处出发将产生相同的回路)。这样的回路总共有 24 条, 但是因为往返路程距离相同, 所以在求最短总距离时, 只需要考虑 12 条不同的回路即可。列出这 12 条不同回路和每条回路旅行的最短总距离。从表 6-6 可以看出, 回路底特律-托莱多-卡拉玛祖-格兰特·拉皮滋-萨吉诺, 对应 458 英里的最短总距离。

上面描述了旅行商问题的一个实例。旅行商问题求带权完全无向图里访问每个顶点恰好一次并且返回出发点的总权数最小的回路。这等价于求完全图里总权数最小的哈密顿回路, 因为在问

路里访问每个顶点恰好一次。

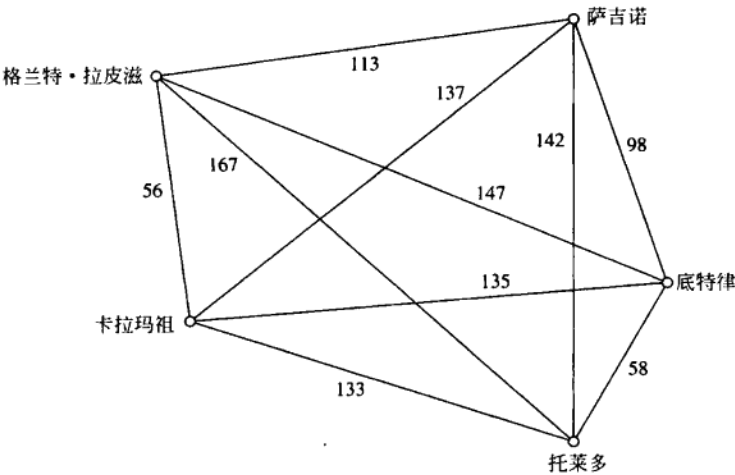


图 6-66 说明五个城市之间距离的图

表 6-6

| 路 线                          | 总距离(英里) |
|------------------------------|---------|
| 底特律-托莱多-格兰特·拉皮滋-萨吉诺-卡拉玛祖-底特律 | 610     |
| 底特律-托莱多-格兰特·拉皮滋-卡拉玛祖-萨吉诺-底特律 | 516     |
| 底特律-托莱多-卡拉玛祖-萨吉诺-格兰特·拉皮滋-底特律 | 588     |
| 底特律-托莱多-卡拉玛祖-格兰特·拉皮滋-萨吉诺-底特律 | 458     |
| 底特律-托莱多-萨吉诺-卡拉玛祖-格兰特·拉皮滋-底特律 | 540     |
| 底特律-托莱多-萨吉诺-格兰特·拉皮滋-卡拉玛祖-底特律 | 504     |
| 底特律-萨吉诺-托莱多-格兰特·拉皮滋-卡拉玛祖-底特律 | 598     |
| 底特律-萨吉诺-托莱多-卡拉玛祖-格兰特·拉皮滋-底特律 | 576     |
| 底特律-萨吉诺-卡拉玛祖-托莱多-格兰特·拉皮滋-底特律 | 682     |
| 底特律-萨吉诺-格兰特·拉皮滋-托莱多-卡拉玛祖-底特律 | 646     |
| 底特律-格兰特·拉皮滋-萨吉诺-托莱多-卡拉玛祖-底特律 | 670     |
| 底特律-格兰特·拉皮滋-托莱多-萨吉诺-卡拉玛祖-底特律 | 728     |

最直截了当的求解旅行商问题实例的方式是检查所有可能的哈密顿回路，并且挑出总权数最小的一条回路。若在图中有  $n$  个城市，则为了求解这个问题，得检查多少条回路？一旦选定了出发点，需要检查的不同的哈密顿回路就有  $(n-1)!$  条，因为第二个顶点有  $n-1$  种选择，第三个顶点有  $n-2$  种选择，依此类推。因为可以用相反顺序来经过一条哈密顿回路，所以只需要检查  $(n-1)!/2$  条回路来求出答案。注意  $(n-1)!/2$  增长得极快。当只有几十个顶点时，试图用这种方式来解决旅行商问题就是不切实际的。例如，假如有 25 个顶点，那么就不得不考虑总共  $24!/2$  (约为  $3.1 \times 10^{23}$ ) 条不同的哈密顿回路。假定检查每条哈密顿回路只花费 1 纳秒( $10^{-9}$  秒)，那么就需要大约一千万年才能求出这个图里长度最短的一条哈密顿回路。

因为旅行商问题在实践和理论上都具有重要意义，所以已经投入了巨大的努力来设计解决它的有效算法。不过，还没有已知的解决这个问题的多项式最坏情形时间复杂性的算法。进一步地，假如这种算法找到了，那么许多其他难题(比如在第 1 章里讨论过的确定  $n$  个变元的命题公式是否重言式)也可以迎刃而解。这个结果是从 NP 完全性理论得出的(关于这个理论的更多信息请参考 [GaJo79])。

当有许多需要访问的顶点时,解决旅行商问题的实际方法是使用近似算法。近似算法是这样的算法,它们不必产生问题的精确解,取而代之的是保证产生接近精确解的解。即它们可能产生带总权数  $W'$  的哈密顿回路,使得  $W \leq W' \leq cW$ , 其中  $W$  是精确解的总长度,而  $c$  是一个常数。例如,存在多项式最坏情形时间复杂性算法使得  $c=3/2$ 。对于一般带权图和每个正实数  $k$  来说,总是产生至多  $k$  倍于最优解的解的算法还是未知。假如这样的算法存在,那就可能证明 P 类等于 NP 类,这是关于算法复杂性的最著名的未解决问题。

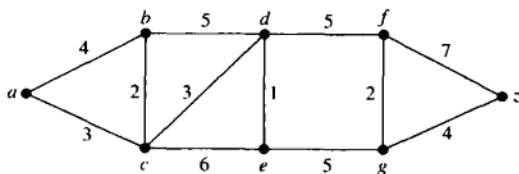
在实际中,已经研究出这样的算法,它们可以只用几分钟的机时,就解决多达 1000 个顶点的旅行商问题,误差在精确解的 2% 之内。关于旅行商问题的更多信息,包括历史、应用和算法等,见《离散数学的应用》(*Applications of Discrete Mathematics*) [MiRo91] 中关于这个主题的那一章。

### 练习

1. 对下列关于地铁系统的每个问题,描述一个可以用来解决这个问题的带权图模型。

- 在两站之间旅行所需要的最短时间是什么?
- 从一站到达另外一站所经过的最短距离是什么?
- 若把各站之间的票价求和就得出总票价,则在两站之间的最低票价是什么?

2. 求下面的带权图里  $a$  与  $z$  之间的最短通路的长度。



3. 在练习 2 的带权图中,  $a$  与  $z$  之间的最短通路是什么?

4. 在图 6-63 所示的通信网络里,求下列每对城市的计算机中心之间的(距离)最短路线。

- 波士顿与洛杉矶
- 纽约与旧金山
- 达拉斯与旧金山
- 丹佛与纽约

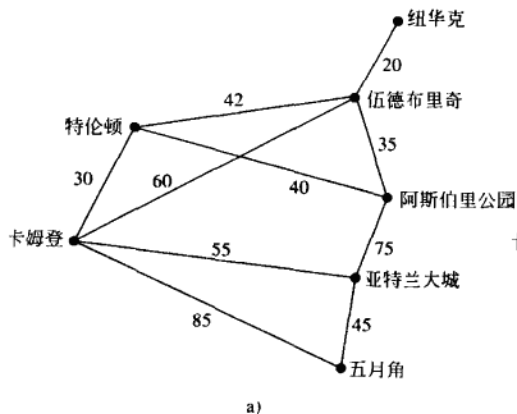
5. 利用图 6-63 里给出的租费,求在练习 4 中成对的计算机中心之间月租费最便宜的路线。

6. 推广求带权简单有向连通图里两个顶点之间最短通路的迪克斯特拉算法,以便求出在顶点  $a$  与图中其余每个顶点之间的最短通路的长度。

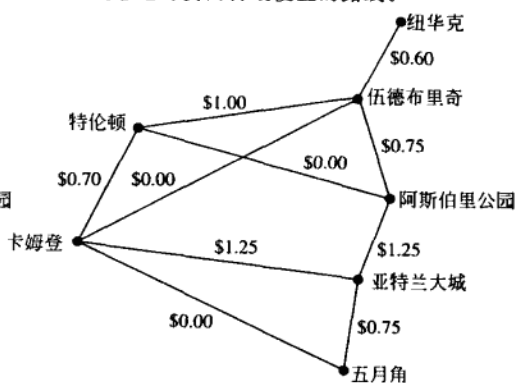
7. 在下图里的带权图说明新泽西的一些主要道路。图 a 说明这些道路上的城市之间的距离;图 b 说明通行费。

a) 利用这些道路,求在纽华克与卡姆登之间,以及在纽华克与五角角之间的距离最短的路线。

b) 利用给出的道路图,求在本题部分(a)里成对城市之间就总通行费而言最便宜的路线。



a)



b)

8. 哪些应用必须求出带权图里两个顶点之间的最长简单通路的长度?

可以用弗洛伊德算法来求出带权连通简单图里所有顶点对之间最短通路的长度。不过,不能用这个算法来构造出最短通路(在下面,把无穷权赋给任何一对不被图中的边所连接的顶点上)。

**算法 2 弗洛伊德算法**

**procedure** *Floyd*(*G*: 带权简单图)

{*G* 有顶点  $v_1, v_2, \dots, v_n$  和权  $w(v_i, v_j)$ , 其中若  $(v_i, v_j)$  不是边, 则  $w(v_i, v_j) = \infty$ }

**for**  $i := 1$  **to**  $n$

**for**  $j := 1$  **to**  $n$

$d(v_i, v_j) := w(v_i, v_j)$

**for**  $i := 1$  **to**  $n$

**for**  $j := 1$  **to**  $n$

**for**  $k := 1$  **to**  $n$

**if**  $d(v_j, v_i) + d(v_i, v_k) < d(v_j, v_k)$

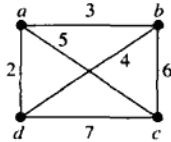
**then**  $d(v_j, v_k) := d(v_j, v_i) + d(v_i, v_k)$

{ $d(v_i, v_j)$  是在  $v_i$  与  $v_j$  之间的最短通路的长度}

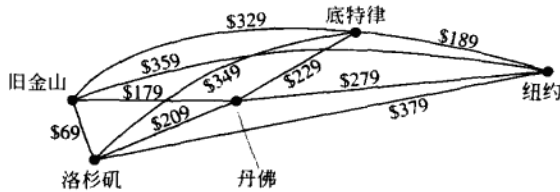
9. 用弗洛伊德算法求图 6-65a 中带权图里所有顶点对之间的距离。

\*10. 给出弗洛伊德算法为了确定在带有  $n$  个顶点的带权简单图里、所有顶点对之间的最短距离而使用的运算(比较和加法)次数的大  $O$  估计。

11. 通过求出所有哈密顿回路的总权数并且确定出总权数最小的回路, 来解决下图的旅行商问题。



12. 求访问下图里每个城市的机票总价最低的路线, 其中边上的权是在这两个城市之间的航班所提供的最低票价。



13. 构造一个带权无向图, 使得对于访问某些顶点超过一次的回路来说, 访问每个顶点至少一次的回路的总权数是最小的。[提示: 存在带三个顶点的例子。]

## 6.7 可平面图

### 6.7.1 引言

考虑把三座房屋与三种设施每种都连接起来的问题, 如图 6-67 所示。是否有可能这样来连接这些房屋与设施, 使得在这样的连接里不发生交叉? 这个问题可以用完全偶图  $K_{3,3}$  来建模。原来的问题可以重新叙述为: 能否在平面里画出  $K_{3,3}$ , 使得没有两条边发生交叉?

本节将研究能否在平面里让边不交叉地画出一个图的问题。特别是, 将回答这个房屋与设施的问题。

图的表示方式有许多种。何时有可能至少求出一种方式来在平面里表示这个图而让边没有任何交叉?

**定义 1** 若可以在平面里画出一个图而边没有任何交叉(其中边的交叉是表示边的直线或弧线在它们的公共端点以外的地方相交), 则这个图是可平面的。这种画法称为这个图的平面表示。

即使通常带交叉地画出了一个图, 这个图也仍然可能是可平面的, 这是因为有可能以不同的方式来不带交叉地画出这个图。



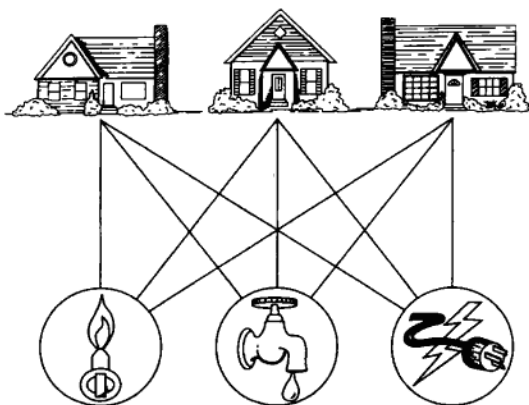
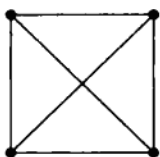
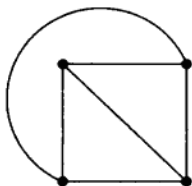


图 6-67 三座房屋与三种设施

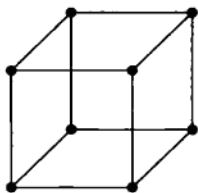
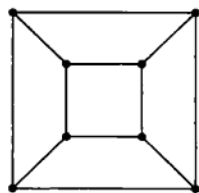
例 1  $K_4$  (如图 6-68 所示, 有两条边交叉) 是可平面吗?

解  $K_4$  是可平面的, 因为可以不带交叉地画出它, 如图 6-69 所示。

图 6-68 图  $K_4$ 图 6-69 不带交叉的图  $K_4$ 

例 2 图 6-70 所示的  $Q_3$  是可平面吗?

解  $Q_3$  是可平面的, 因为可以画出它而没有任何边交叉, 如图 6-71 所示。

图 6-70 图  $Q_3$ 图 6-71  $Q_3$  的一种平面表示

可以通过显示一种平面表示来证明一个图是可平面的。更难的是证明一个图是非可平面的。下面给出一个例子说明如何以一种特别的方法来做到这一点。

例 3 图 6-72 所示的  $K_{3,3}$  是可平面吗?

解 在平面里画出  $K_{3,3}$  而没有边交叉, 任何这样的尝试都注定是失败的。现在说明这是为什么。在  $K_{3,3}$  的任何平面表示里, 顶点  $v_1$  和  $v_2$  都必须同时与  $v_4$  和  $v_5$  连接。这四条边所形成的封闭曲线把平面分割成两个区域  $R_1$  和  $R_2$ , 如图 6-73a 所示。顶点  $v_3$  属于  $R_1$  或  $R_2$ 。当  $v_3$  属于闭曲线的内部  $R_2$  时, 在  $v_3$  和  $v_4$  之间以及在  $v_3$  和  $v_5$  之间的边, 把  $R_2$  分割成两个区域  $R_{21}$  和  $R_{22}$ , 如图 6-73b 所示。

下一步, 已经没有办法来不交叉地放置最后一个顶点  $v_6$ 。因为若  $v_6$  属于  $R_1$ , 则不能不带交叉地画出  $v_6$  和  $v_3$  之间的边。若  $v_6$  属于  $R_{21}$ , 则不能不带交叉地画出  $v_2$  和  $v_6$  之间的边。若  $v_6$  属于

$R_{22}$ , 则不能不带交叉地画出  $v_1$  和  $v_6$  之间的边。

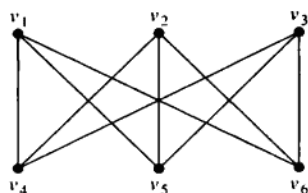


图 6-72 图  $K_{3,3}$

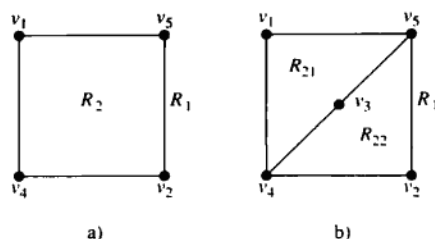


图 6-73 证明  $K_{3,3}$  是非可平面的

当  $v_3$  属于  $R_1$  时, 可以使用类似的论证。请读者来完成这个论证。所以  $K_{3,3}$  是非可平面的。■

例 3 解决了在本节开头所描述的设施与房屋的问题。不能在平面里连接这三座房屋与三种设施而不发生交叉。可以用类似的论证来证明  $K_5$  是非可平面的(见本节末的练习 6)。

图的平面性在电子电路的设计中有重要作用。可以用图来为电路建立模型, 用顶点表示电路的器件并且用边表示器件之间的连接。如果表示一个电路的图是可平面的, 则可以把这个电路无连接交叉地印刷在单个电路板上。当这个图不是可平面的时, 就必须求助于成本更高的选项。例如, 可以把表示电路的图的顶点划分成可平面子图。然后使用多层来构造这个电路。每当连接交叉时就可以用绝缘线来构造电路。在这种情况下, 重要的是以尽可能少的交叉来画出这个图。

## 6.7.2 欧拉公式

一个图的可平面表示把平面分割成一些面, 包括一个无界的面。例如, 图 6-74 所示的图的可平面表示把平面分割成 6 个面并加以标记。欧拉证明过一个图的所有可平面表示都把平面分割成相同数目的面。他是这样证明的: 求出可平面图的面数、顶点数以及边数之间的关系。

**定理 1 欧拉公式** 设  $G$  是带  $e$  条边和  $v$  个顶点的连通可平面简单图。设  $r$  是  $G$  的可平面表示里的面数。则  $r = e - v + 2$ 。

**证** 首先规定  $G$  的可平面表示。将要这样证明定理: 构造一系列子图  $G_1, G_2, \dots, G_r = G$ , 相继地在每个阶段添加一条边。用下面的归纳定义来这样做。任意地选择一条  $G$  的边来获得  $G_1$ 。这样通过  $G_{n-1}$  获得  $G_n$ : 任意地添加一条与  $G_{n-1}$  里顶点相关联的边, 若与这条边关联的另一个顶点还不在于  $G_{n-1}$  里, 则添加这个顶点。这样的构造是可能的, 因为  $G$  是连通的。在添加  $e$  条边之后就获得  $G$ 。设  $r_n, e_n$  和  $v_n$  分别表示  $G$  的可平面表示所引导出的  $G_n$  的可平面表示的面数、边数和顶点数。

现在通过归纳来进行证明。对  $G_1$  来说, 关系  $r_1 = e_1 - v_1 + 2$  为真, 因为  $e_1 = 1, v_1 = 2$ , 而  $r_1 = 1$ 。这种情形如图 6-75 所示。

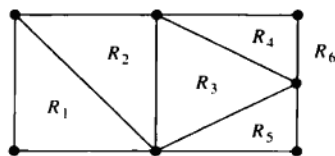


图 6-74 图的可平面表示的面

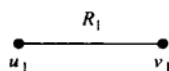


图 6-75 欧拉公式的证明中的基础情形

现在假定  $r_n = e_n - v_n + 2$ 。设  $\{a_{n+1}, b_{n+1}\}$  是为了获得  $G_{n+1}$  而添加到  $G_n$  上的边。有两种情形需要考虑。在第一种情形里,  $a_{n+1}$  和  $b_{n+1}$  都已经在  $G_n$  里了。这两个顶点必然是在一个公共面  $R$  的边界上, 否则就不可能把边  $\{a_{n+1}, b_{n+1}\}$  添加到  $G_n$  里而没有两条边交叉(并且  $G_{n+1}$  是可平面的)。这条新边的添加把  $R$  分割成两个面。所以, 在这种情形里,  $r_{n+1} = r_n + 1, e_{n+1} = e_n + 1$ , 而且  $v_{n+1} = v_n$ 。因此, 关系到面数、边数、顶点数的公式两边都恰好增加 1, 所以这个公式仍然为真。换句话

说,  $r_{n+1} = e_{n+1} - v_{n+1} + 2$ . 在图 6-76a 里说明这种情形。

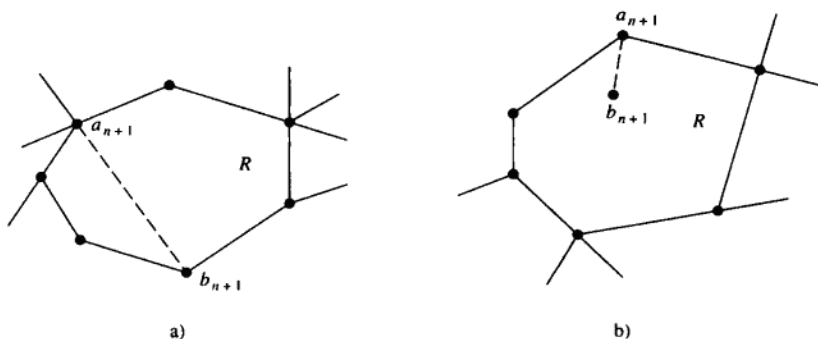


图 6-76 添加一条边到  $G_n$  来产生  $G_{n+1}$

在第二种情形里, 新边的两个顶点之一还不在于  $G_n$  里。假定  $a_{n+1}$  在  $G_n$  里但是  $b_{n+1}$  不在  $G_n$  里。添加这条新边不产生任何新的面, 因为  $b_{n+1}$  必然是在边界上有  $a_{n+1}$  的一个面里。所以,  $r_{n+1} = r_n$ 。另外,  $e_{n+1} = e_n + 1$  且  $v_{n+1} = v_n + 1$ 。关系到面数、边数、顶点数的公式两边都保持相等, 所以这个公式仍然为真。换句话说,  $r_{n+1} = e_{n+1} - v_{n+1} + 2$ 。在图 6-76b 里说明这种情形。

已经完成了归纳论证。因此对所有  $n$  来说, 都有  $r_n = e_n - v_n + 2$ 。因为原图是在添加了  $e$  条边之后所获得的图  $G_r$ , 所以这个定理为真。□

例 4 解释欧拉公式。

例 4 假定连通可平面简单图有 20 个顶点, 每个顶点度都为 3。这个可平面图表示把平面分割成多少个面?

解 这个图有 20 个顶点, 每个顶点的度都为 3, 所以  $v=20$ 。因为这些顶点的度之和  $3v = 3 \cdot 20 = 60$  等于边数的两倍  $2e$ , 所以有  $2e=60$  或  $e=30$ 。所以, 根据欧拉公式, 面数是

$$r = e - v + 2 = 30 - 20 + 2 = 12$$

可以用欧拉公式来建立可平面图所必须满足的一些不等式。在下面的推论里给出一个这样的不等式。

**推论 1** 若  $G$  是带  $e$  条边和  $v$  个顶点的连通可平面简单图, 其中  $v \geq 3$ , 则  $e \leq 3v - 6$ 。

在证明推论 1 之前先用它证明下面这个有用的结论。

**推论 2** 若  $G$  是连通可平面简单图, 则  $G$  有度数不超过 5 的顶点。

证 如果  $G$  有 1 个或 2 个顶点, 则结果为真。如果  $G$  至少有 3 个顶点, 则根据推论 1 知道  $e \leq 3v - 6$ , 所以  $2e \leq 6v - 12$ 。假如每个顶点的度数至少是 6, 则由于  $2e = \sum_{v \in V} \deg(v)$  (根据握手定理), 所以就有  $2e \geq 6v$ 。但是这与  $2e \leq 6v - 12$  相矛盾。所以必定存在度数不超过 5 的顶点。□

推论 1 的证明是基于面的次数的概念, 它定义为这个面的边界上的边数。当一条边在边界上出现两次(所以当描画边界时就描画它两次)时, 它贡献的次数是 2。图 6-77 里显示出面的次数。

现在可以给出推论 1 的证明了。

证 画在平面里的连通可平面简单图把平面分割成面, 比如说  $r$  个面。每个面的次数至少为 3。(因为这里所讨论的图都是简单图, 不允许带有可能产生 2 次面的多重边, 或者可能产生 1 次面的环。)特别地, 注意无界的面次数

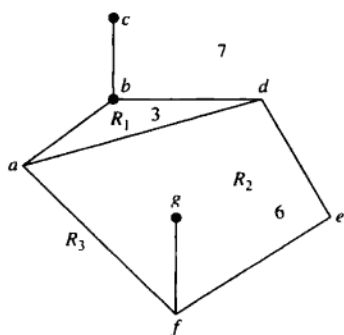


图 6-77 面的次数

至少为3,因为在图里至少有三个顶点。

注意各面的次数之和恰好是图中边数的两倍,因为每条边都在面的边界上出现两次(或者在两个不同面里,或者两次在相同面里)。因为每个面都有大于或等于3的次数,所以

$$2e = \sum_{\text{所有区域} R} \deg(R) \geq 3r$$

因此,

$$(2/3)e \geq r$$

利用  $r = e - v + 2$  (欧拉公式), 就得到

$$e - v + 2 \leq (2/3)e$$

所以  $e/3 \leq v - 2$ 。这样就证明了  $e \leq 3v - 6$ 。

可以用这个推论来证明  $K_5$  是非可平面的。

**例5** 用推论1证明:  $K_5$  是非可平面的。

**解** 图  $K_5$  有5个顶点和10条边。不过,对这个图来说,不满足不等式  $e \leq 3v - 6$ , 因为  $e = 10$  和  $3v - 6 = 9$ 。因此,  $K_5$  不是可平面的。

前面已经证明了  $K_{3,3}$  不是可平面的。不过,注意这个图有6个顶点和9条边。这意味着满足不等式  $e = 9 \leq 12 = 3 \cdot 6 - 6$ 。所以,满足不等式  $e \leq 3v - 6$  的事实并不意味着一个图是可平面的。不过,可以利用下面定理1的推论来证明  $K_{3,3}$  不是可平面的。

**推论3** 若连通可平面简单图有  $e$  条边和  $v$  个顶点,  $v \geq 3$  并且没有长度为3的回路, 则  $e \leq 2v - 4$ 。

推论3的证明类似于推论1的证明,不同之处在于,在这种情形里,没有长度为3的回路的事实蕴含着面的次数必然至少为4。把这个证明的细节留给读者(见本节末的练习8)。

**例6** 用推论3证明:  $K_{3,3}$  是非可平面的。

**解** 因为  $K_{3,3}$  没有长度为3的回路(容易看出这一点,因为它是偶图),所以可以使用推论3。 $K_{3,3}$  有6个顶点和9条边。因为  $e = 9$  和  $2v - 4 = 8$ , 所以推论3证明  $K_{3,3}$  是非可平面的。

### 6.7.3 库拉图斯基定理

我们已经看到  $K_{3,3}$  和  $K_5$  都不是可平面的。显然,若一个图包含这两个图作为子图,则它不是可平面的。另外,所有非可平面图必然包含一个子图,它是可以利用某些允许的操作从  $K_{3,3}$  或  $K_5$  来获得的。

若一个图是可平面的,则通过删除一条边  $\{u, v\}$  并且添加一个新顶点  $w$  和两条边  $\{u, w\}$  与  $\{w, v\}$ , 所获得的任何图也是可平面的。这样的操作称为初等细分。若可以从相同的图通过一系列初等细分来获得图  $G_1 = (V_1, E_1)$  和图  $G_2 = (V_2, E_2)$ , 则它们称为是同胚的。

**例7** 证明: 图6-78所示的图  $G_1, G_2, G_3$  是同胚的。

**解** 因为这三个图都可以从图  $G_1$  通过初等细分得到,所以它们是同胚的。 $G_1$  可以从它自身出发,通过一个空的初等细分序列而得到。要从  $G_1$  得到  $G_2$ , 采用如下初等细分序列:

- 1) 删掉边  $\{a, c\}$ , 增加顶点  $f$ , 然后添加边  $\{a, f\}$  和  $\{f, c\}$ 。
- 2) 删掉边  $\{b, c\}$ , 增加顶点  $g$ , 然后添加边  $\{b, g\}$  和  $\{g, c\}$ 。
- 3) 删掉边  $\{b, g\}$ , 增加顶点  $h$ , 然后添加边  $\{g, h\}$  和  $\{h, b\}$ 。

把找出由  $G_1$  到  $G_2$  的初等细分序列的任务留给读者。

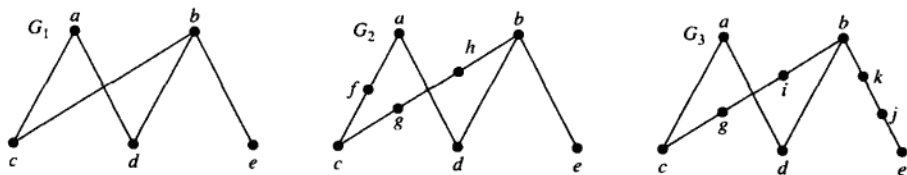


图 6-78 同胚图

波兰数学家库拉图斯基在1930年证明了下面的定理，这个定理用图同胚的概念刻画了可平面图。

**定理2** 一个图是非可平面的当且仅当它包含一个同胚于  $K_{3,3}$  或  $K_5$  的子图。

显然，一个包含着同胚于  $K_{3,3}$  或  $K_5$  的子图的图是非可平面的。不过，相反方向的命题——即每个非可平面图都包含一个同胚于  $K_{3,3}$  或  $K_5$  的子图——证明起来是复杂的，因而不在这里给出。下面的例子说明如何使用库拉图斯基定理。

**例8** 确定图6-79所示的图  $G$  是否是可平面的。

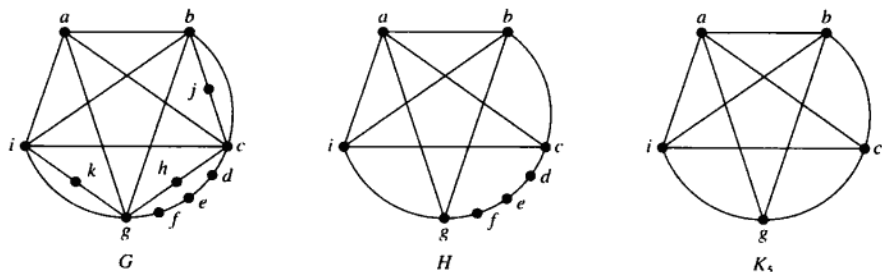


图 6-79 无向图  $G$ 、同胚于  $K_5$  的子图  $H$  以及  $K_5$

**解**  $G$  有同胚于  $K_5$  的子图  $H$ 。 $H$  是这样获得的：删除  $h, j$  和  $k$  以及所有与这些顶点关联的边。 $H$  是同胚于  $K_5$  的，因为从  $K_5$  (带有顶点  $a, b, c, g$  和  $i$ ) 通过一系列初等细分，添加顶点  $d, e$  和  $f$  就可以获得  $H$  (读者应当构造出这样的一系列初等细分)。因此， $G$  是非可平面的。

**例9** 在图6-80a中所示的彼得森图是可平面吗？(丹麦数学家朱利乌斯·彼得森在1891年介绍过这个图；它常用来说明关于图的各种性质的理论。)

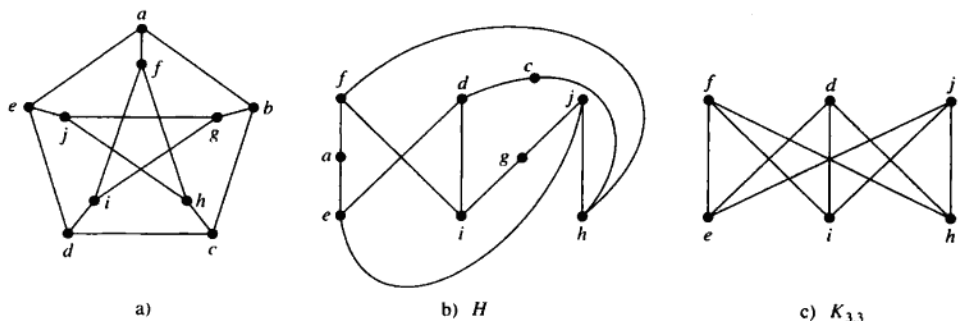
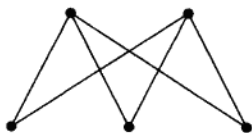


图 6-80 a) 彼得森图；b) 同胚于  $K_{3,3}$  的子图  $H$ ；c)  $K_{3,3}$

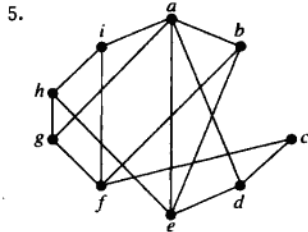
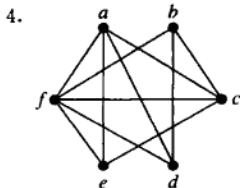
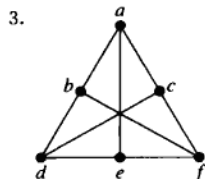
**解** 彼得森图的子图  $H$  是这样获得的：删除  $b$  和以  $b$  作为端点的三条边，如图6-78b所示，它是同胚于带有顶点集合  $\{f, d, j\}$  和  $\{e, i, h\}$  的  $K_{3,3}$  的，这是因为可以通过一系列初等细分来获得它，删除  $\{d, h\}$  并且添加  $\{c, h\}$  和  $\{c, d\}$ ，删除  $\{e, f\}$  并且添加  $\{a, e\}$  和  $\{a, f\}$ ，删除  $\{i, j\}$  并且添加  $\{g, i\}$  和  $\{g, j\}$ 。因此，彼得森图不是可平面的。

## 练习

1. 五座房屋能否不带连接交叉地与两种设施相连接？
2. 不带任何交叉地画出下面的可平面图。



在练习 3~5 里, 确定所给的图是否可平面。若是可平面的, 则画出它使得没有边交叉。



6. 用类似于在例 3 里给出的论证来证明:  $K_5$  是非可平面的。

7. 假定一个连通可平面图有 6 个顶点, 每个顶点的度都为 4。这个图的可平面表示把平面分割成多少个面?

8. 证明推论 3。

\*9. 假定一个带有  $e$  条边和  $v$  个顶点的连通可平面简单图不包含长度为 4 或更短的回路。证明: 若  $v \geq 4$  则  $e \leq (5/3)v - (10/3)$ 。

10. 下面的哪些非可平面图具有这样的性质: 删除任何一个顶点以及与这个顶点关联的所有边就产生一个可平面图?

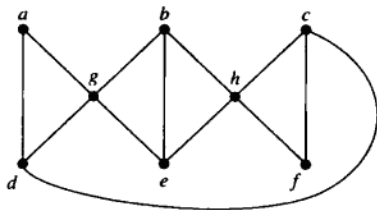
a)  $K_5$

b)  $K_6$

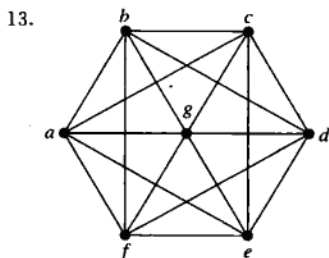
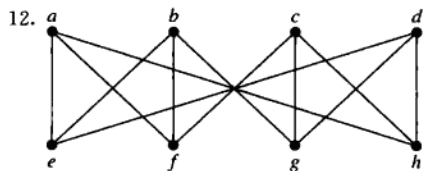
c)  $K_{3,3}$

d)  $K_{3,4}$

11. 确定下面的图是否同胚于  $K_{3,3}$ 。



在练习 12~13 里, 用库拉图斯基定理来确定所给的图是不是可平面的。



一个简单图的交叉数, 是指当在平面上画出这个图(其中不允许任何三条表示边的弧线在同一个点交叉)时交叉的最少次数。

\*\*14. 求下面每个非可平面图交叉数。

a)  $K_5$

b)  $K_6$

c)  $K_7$

d)  $K_{3,4}$

e)  $K_{4,4}$

f)  $K_{5,5}$

\*\*15. 证明: 若  $m$  和  $n$  都是偶正整数, 则  $K_{m,n}$  的交叉数小于或等于  $mn(m-2)(n-2)/16$ 。[提示: 沿着  $x$  轴放置  $m$  个顶点, 使得它们间距相等并且关于原点对称, 再沿着  $y$  轴放置  $n$  个顶点, 使得它们间距相等并且关于原点对称。现在连接  $x$  轴上  $m$  个顶点中的每一个与  $y$  轴上  $n$  个顶点中的每一个, 并且计算交叉数。]

简单图  $G$  的厚度是指, 以  $G$  作为它们的并图的  $G$  的可平面子图的最小个数。

\*16. 求练习 14 里的图的厚度。

\*17. 在一个环面上画出  $K_{3,3}$ , 使得没有边交叉。

## 6.8 图着色

### 6.8.1 引言

在图论中，有许多与地图区域(比如世界各部分的地图)着色有关的理论成果。当为一幅地图着色时<sup>①</sup>，具有公共边界的两个区域在传统上指定为不同的颜色。一种确保两个相邻的区域永远没有相同的颜色的方法是对每个区域都使用不同的颜色。不过，这种方法效率不高，而且在具有许多区域的地图上，可能难以区分相似的颜色。另一种方法是，应当尽可能地使用少数几种颜色。考虑这样的问题：确定可以用来给一幅地图着色的颜色的最小数目，使得相邻的区域永远没有相同的颜色。例如，对图 6-81 左侧地图来说，四色是足够的，但是三色就不够(读者应当验证这一点)。对图 6-81 右侧地图来说，三色是足够的(但是二色就不够)。

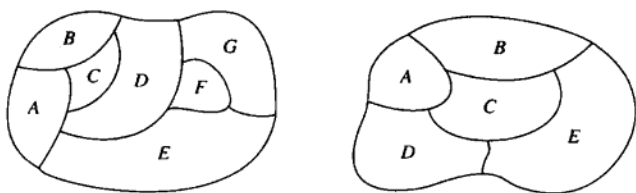


图 6-81 两幅地图

平面里的每幅地图都可以表示成一个图。为了建立这样的对应关系，地图的每个区域都表示成一个顶点。若两个顶点所表示的区域具有公共边界，则用边连接这两个顶点。只相交于一个点的两个区域不算是相邻的。这样所得到的图称为这个地图的对偶图。根据地图的对偶图的构造方式，显然在平面里的任何地图都具有可平面的对偶图。图 6-82 显示对应于图 6-81 所示地图的对偶图。

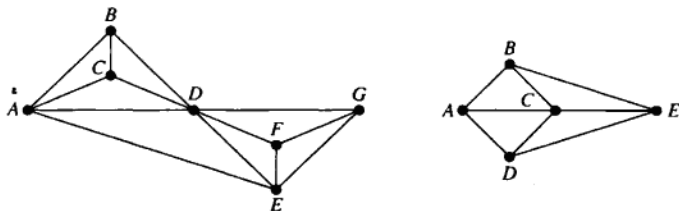


图 6-82 图 6-81 地图的对偶图

给地图的区域着色的问题等价于这样的问题：给对偶图的顶点着色，使得在对偶图里没有两个相邻的顶点具有相同的颜色。给出下面的定义。

**定义 1** 简单图的着色是对该图的每个顶点都指定一种颜色，使得没有两个相邻的顶点颜色相同。

通过对每个顶点都指定一种不同的颜色，就可以着色一个图。不过，对大多数的图来说，可以找到所用颜色数少于图中顶点数的着色。什么是所需要的最少颜色数？

**定义 2** 图的色数是着色这个图所需要的最少颜色数。

注意，求平面图的颜色数等于是求平面地图着色所需要的最少颜色数，使得没有两个相邻的区域指定为相同的颜色。这个问题已经研究了 100 多年。数学里最著名的定理之一给出了它的答案。

**定理 1 四色定理** 平面图的颜色数不超过四。

① 假定地图里所有区域都是连通的。这样就消除了像密歇根这样的地理状况所引起的问题，因为密歇根州是由两个分离的半岛组成的。

 四色定理最早是作为猜想在19世纪50年代提出的。美国数学家肯尼思·阿佩尔和沃尔夫冈·黑肯最终在1976年证明了它。在1976年之前，发表过许多不正确的证明，其中的错误常常难以发现。另外，还尝试过画出需要超过四色的地图来构造反例，而这样做是无效的。(证明五色定理就没有这样困难。)

也许迄今为止在数学里最有名的错误证明就是伦敦的律师和业余的数学家阿尔弗雷德·肯普在1879年所发表的四色定理证明。数学家们一直认为他的证明是正确的，直到1890年珀西·希伍德发现了一处错误，才发现肯普的论证是不完全的。不过，事实证明，肯普的推理路线是阿佩尔和黑肯所给出的成功证明的基础。他们的证明依赖于计算机所完成的对各种情形的仔细分析。他们证明，若四色定理为假，则存在一个反例，它是大约2000种不同类型中的一种，然后他们证明这些类型都没有导致反例。在他们的证明中使用了1000多个小时的机时。计算机在证明过程中起到如此重要的作用，由此引发了广泛的争论。例如，在计算机程序里有没有导致不正确结果的错误？假如论证依赖于或许不可靠的计算机得出，那么它是不是真正的证明？

注意，四色定理只适用于平面图。在例2里将证明非可平面图可以有任意大的色数。

证明一个图的色数为 $n$ 需要做两件事。首先必须证明：用 $n$ 种颜色可以着色这个图。构造出这样的着色就可以完成这件事。其次证明：用少于 $n$ 种颜色不能着色这个图。例1至例4说明如何求出色数。

**例1** 图6-83所示的图 $G$ 和 $H$ 的色数是什么？

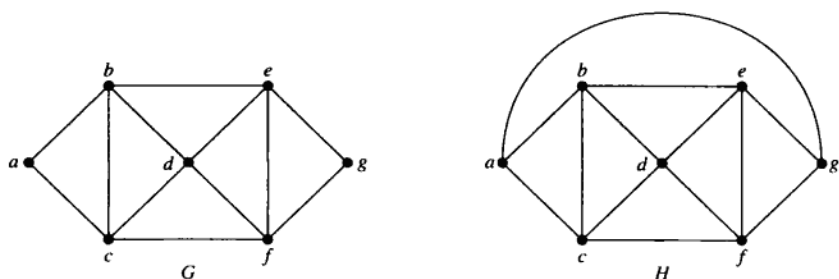


图6-83 简单图 $G$ 和 $H$

 **解** 图 $G$ 的色数至少为3，因为顶点 $a$ 、 $b$ 和 $c$ 必须为不同的颜色。为了看出是否可以三种颜色来对 $G$ 着色，指定 $a$ 为红， $b$ 为蓝， $c$ 为绿。于是，可以(而且必须)令 $d$ 为红，因为它与 $b$ 和 $c$ 相邻。另外，可以(而且必须)令 $e$ 为绿，因为它只与红色和蓝色顶点相邻，可以(而且必须)令 $f$ 为蓝，因为它只与红色和绿色顶点相邻。最后，可以(而且必须)令 $g$ 为红，因为它只与蓝色和绿色的顶点相邻。这样就产生出恰好使用三种颜色的 $G$ 的着色，如图6-84所示。

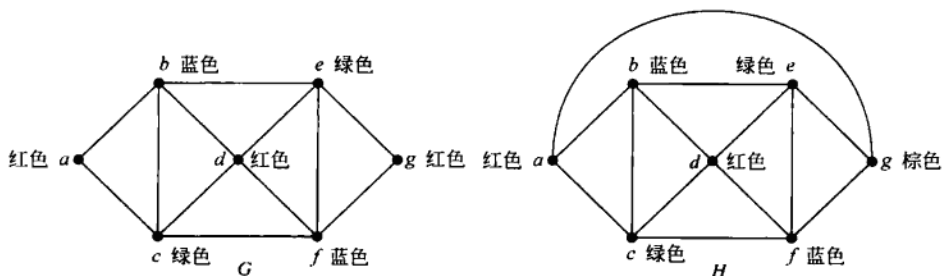


图6-84 图 $G$ 和 $H$ 的着色

图 $H$ 是由图 $G$ 和连接 $a$ 与 $g$ 的一条边所组成的。用三种颜色来着色 $H$ 的任何尝试都必须遵循与着色 $G$ 所用的同样的推理，不同之处是在最后阶段当除了 $g$ 以外的所有顶点都已经着色时，因



为  $g$  与红色、蓝色和绿色顶点(在  $H$  里)相邻, 所以需要使用第四种颜色, 比如棕色。因此,  $H$  的色数为四。图 6-84 右图显示  $H$  的一种着色。

**例 2**  $K_n$  的色数是什么?

**解** 通过给每个顶点指定一种不同的颜色, 用  $n$  种颜色可以构造  $K_n$  的着色。使用的颜色能否更少一些? 答案是不能。没有两个顶点可以指定相同的颜色, 因为这个图的每两个顶点都是相邻的。因此,  $K_n$  的色数  $= n$ 。(回忆一下, 当  $n \geq 5$  时  $K_n$  不是可平面图, 所以这个结果与四色定理并不矛盾。)在图 6-85 里显示使用五种颜色的  $K_5$  的着色。

**例 3** 完全偶图  $K_{m,n}$  的色数是什么? 其中  $m$  和  $n$  都是正整数。

**解** 需要的颜色数似乎依赖于  $m$  和  $n$ 。不过, 仅仅需要两种颜色。 $m$  个顶点为一种颜色,  $n$  个顶点为另外一种颜色。因为边都只能连接  $m$  个顶点中的一个顶点与  $n$  个顶点中的一个顶点, 所以没有相邻的顶点具有相同的颜色。在图 6-86 里显示带有两种颜色的  $K_{3,4}$  的着色。

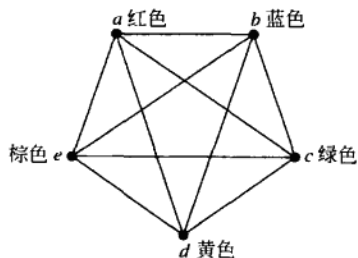


图 6-85  $K_5$  的着色

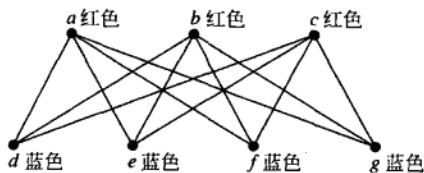


图 6-86  $K_{3,4}$  的着色

**例 4** 图  $C_n$  的色数是什么( $n \geq 3$ )? (回忆一下,  $C_n$  是带有  $n$  个顶点的圈图。)

**解** 首先考虑一些个别情形。首先, 设  $n=6$ 。挑选一个顶点并且把它着色成红色。在图 6-87 所示的  $C_6$  的平面画法里顺时针前进。必须给到达的下一个顶点指定第二种颜色, 比如蓝色。以顺时针方向继续下去, 可以令第三个顶点为红色, 第四个顶点为蓝色, 第五个顶点为红色。最后, 令第六个顶点为蓝色, 它与第一个顶点是相邻的。因此,  $C_6$  的色数为 2。图 6-87 显示这里构造的着色。

其次, 设  $n=5$  并且考虑  $C_5$ 。挑选一个顶点并且令它为红色。顺时针前进, 必须给到达的下一个顶点指定第二种颜色, 比如蓝色。以顺时针方向继续下去, 可以令第三个顶点为红色, 第四个顶点为蓝色。第五个顶点既不能为红色也不能为蓝色, 因为它与第四个顶点和第一个顶点都相邻。所以, 对这个顶点就需要第三种颜色。注意, 假如以逆时针方向对顶点着色, 同样需要三种颜色。因此,  $C_5$  的色数是 3。用 3 种颜色对  $C_5$  着色见图 6-87。

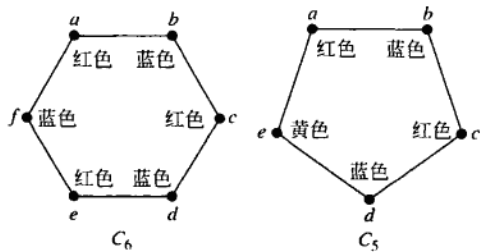


图 6-87  $C_5$  和  $C_6$  的着色

在一般情形里, 当  $n$  是偶数时, 对  $C_n$  着色需要

两种颜色。为了构造这样的着色, 简单地挑选一个顶点并且令它为红色。然后(利用图的平面表示)以顺时针方向绕图前进, 令第二个顶点为蓝色, 第三个顶点为红色, 依此类推。可以令第  $n$  个顶点为蓝色, 因为它与它相邻的两个顶点(即第  $n-1$  个顶点和第一个顶点)都是红色。

当  $n$  是奇数而且  $n > 1$  时,  $C_n$  的色数为 3。为了看出这一点, 挑选一个初始顶点。为了只用两种颜色, 当以顺时针方向遍历这个图时, 必须交替使用颜色。不过, 所到达的第  $n$  个顶点与带不同颜色的两个顶点(第一个顶点和第  $n-1$  个顶点)相邻。因此, 必须使用第三种颜色。



已知最好的求图的色数的算法(对图的顶点数来说)具有指数的最坏情形时间复杂性。即使求图的色数的近似值也是很难的。已经证明, 假如存在具有多项式最坏情形时间

复杂性的、可以达到 2 倍地近似图的色数的算法(即构造出一个不超过图的色数的两倍的界限),那么也存在具有多项式最坏情形时间复杂性的求图的色数的算法。

6.8.2 图着色的应用

图着色在与调度和分配有关的问题中具有多种应用。(注意,由于不知道图着色的有效算法,所以这并不能得出调度和分配的有效算法。)在这里将给出这样的应用的例子。第一个应用是用来安排期末考试。

**例 5 安排期末考试** 如何安排一所大学里的期末考试,使得没有学生同时要考两门试?

**解** 这样的安排问题可以用图模型来解决,用顶点表示科目,若有学生要考两门试,则在表示考试科目的两个顶点之间有边。用不同颜色来表示期末考试的每个时间段。考试的安排就对应于所关联的图的着色。

例如,假定要安排七门期末考试。假定科目从 1 到 7 编号。假定下列各对科目的考试有学生要都参加: 1 和 2, 1 和 3, 1 和 4, 1 和 7, 2 和 3, 2 和 4, 2 和 5, 2 和 7, 3 和 4, 3 和 6, 3 和 7, 4 和 5, 4 和 6, 5 和 6, 5 和 7, 以及 6 和 7。在图 6-88 里显示这组科目所关联的图。一种安排就是由这个图的一种着色来组成的。

因为这个图的色数为 4(读者应当验证这一点),所以需要四个时间段。图 6-89 显示使用了四种颜色的这个图的着色以及所关联的调度。

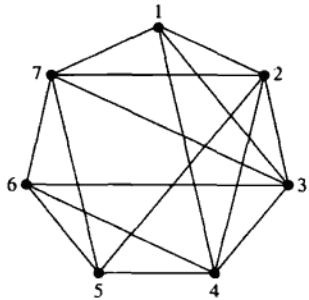


图 6-88 表示期末考试安排的图

现在考虑对电视频道的分配。

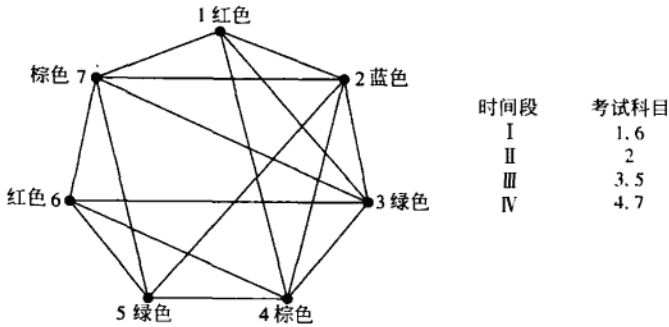


图 6-89 用着色来安排期末考试

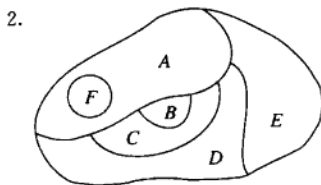
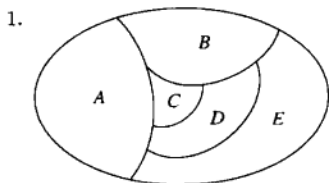
**例 6 频率分配** 把频道 2 到 13 分配给在北美洲的电视台,要避免 150 英里之内的两家电视台在相同频道上播出。如何用图为频道分配建模?

**解** 这样构造一个图:给每个电视台指定一个顶点。若两个电视台彼此位于 150 英里以内,则用边连接这两个顶点。频道指派就对应于这个图的着色,其中每种颜色表示一个不同的频道。

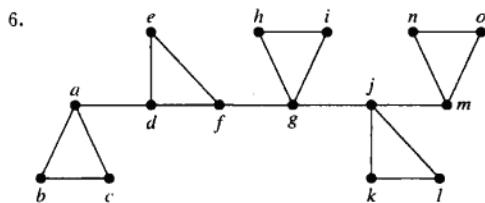
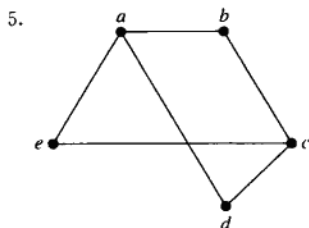
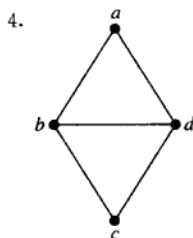
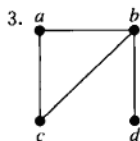
**例 7 变址寄存器** 在有效的编译器里,当把频繁地使用的变量暂时保存在中央处理单元的变址寄存器里,而不是保存在常规内存里时,可以加速循环的执行。对于给定的循环来说,需要多少个变址寄存器?可以用图着色模型来讨论这个问题。为了建立这个模型,设图的每个顶点表示循环里的一个变量。若在循环执行期间两个顶点所表示的变量必须同时保存在变址寄存器里,则在这两个顶点之间有边。所以,这个图的色数就给出所需要的变址寄存器数,因为当表示变量的顶点在图中相邻时,就必须给这些变量分配不同的寄存器。

## 练习

在练习 1~2 里, 构造所示地图的对偶图。然后求给这个地图着色使得相邻的两个区域都没有相同的颜色所需要的颜色数。



在练习 3~6 里, 求给定的图的色数。



7. 哪些图具有色数 1?

8.  $W_n$  的色数是什么?

9. 假定除了科目 Math 115 与 CS 473, Math 116 与 CS 473, Math 195 与 CS 101, Math 195 与 CS 102, Math 115 与 Math 116, Math 115 与 Math 185, Math 185 与 Math 195 以外, 其他每种组合里都有学生要参加组合里所有科目的考试, 请使用最少个数的不同时间段, 来为 Math 115, Math 116, Math 185, Math 195, CS 101, CS 102, CS 273 和 CS 473 安排期末考试日程表。

10. 数学系有六个委员会, 都是每月开一次会。假定委员会是  $C_1 = \{\text{阿林豪斯, 布兰德, 沙斯拉夫斯基}\}$ ,  $C_2 = \{\text{布兰德, 李, 罗森}\}$ ,  $C_3 = \{\text{阿林豪斯, 罗森, 沙斯拉夫斯基}\}$ ,  $C_4 = \{\text{李, 罗森, 沙斯拉夫斯基}\}$ ,  $C_5 = \{\text{阿林豪斯, 布兰德}\}$  和  $C_6 = \{\text{布兰德, 罗森, 沙斯拉夫斯基}\}$ , 那么怎样安排才能确保没有人同时参加两个会议。

11. 图的边着色是指对各边指定颜色, 使得关联到相同顶点的边指定不同的颜色。图的边色数是在该图的边着色里可以使用的最少颜色数。

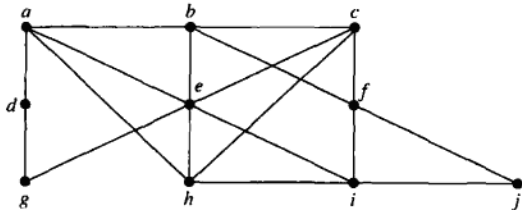
12. 求练习 3~6 里每个图的边色数。

13. 七个变量出现在计算机程序的循环里。这些变量以及必须保存它们的计算步骤是:  $t$ : 步骤 1 到 6;  $u$ : 步骤 2;  $v$ : 步骤 2 到 4;  $w$ : 步骤 1, 3 和 5;  $x$ : 步骤 1 和 6;  $y$ : 步骤 3 到 6; 以及  $z$ : 步骤 4 和 5。在执行期间需要多少个不同的变址寄存器来保存这些变量?

下面的算法可以用来为简单图着色。首先, 以度递减的顺序列出顶点  $v_1, v_2, \dots, v_n$  使得  $\deg(v_1) \geq \deg(v_2) \geq \dots \geq \deg(v_n)$ 。把颜色 1 指定给  $v_1$  和在表中不与  $v_1$  相邻的下一个顶点(若存在一个这样的顶点), 并且继续指定给每一个在表中不与已经指定了颜色 1 的顶点相邻的顶点。然后把颜色 2 指定给表中还没有着色的第一个顶点。继续把颜色 2 指定给那些在表中还没有着色, 而且不与指定了颜色 2 的顶点相邻的顶点。若

还有未着色的顶点, 则指定颜色 3 给表中还没有着色的第一个顶点, 并且用颜色 3 继续对还没有着色, 而且不与指定了颜色 3 的顶点相邻的那些顶点着色。继续这个过程直到所有顶点都着色为止。

13. 用这个算法构造下图的着色。



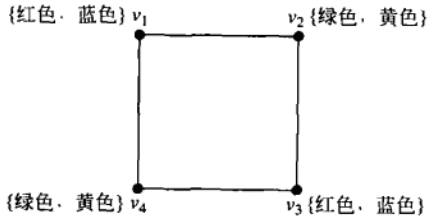
\*14. 证明: 这个算法所产生的着色数可能比着色一个图所需的颜色数更多。

如果一个连通图  $G$  的色数为  $k$ , 但是对于  $G$  的任意一条边  $e$ , 从  $G$  中删掉边  $e$  后得到的新图的色数都是  $k-1$ , 则称  $G$  为着色  $k$  关键的。

15. 证明: 只要  $n$  是正的奇数且  $n \geq 3$ , 那么  $W_n$  就是着色 4 关键的。

16. 证明: 如果图  $G$  为着色  $k$  关键的, 那么  $G$  中的各个顶点的度至少是  $k-1$ 。

图  $G$  的  $k$  重着色是对  $G$  的顶点指定含有  $k$  种不同颜色的集合, 使得相邻的顶点不具有相同的颜色。用  $\chi_k(G)$  来表示使得  $G$  有用  $n$  种颜色的  $k$  重着色的最小正整数  $n$ 。例如,  $\chi_2(C_4) = 4$ 。为了看出这一点, 注意如图所示, 只用四种颜色, 就可以对  $C_4$  的每个顶点指定两种颜色, 使得两个相邻顶点不具有相同的颜色。另外, 少于四种颜色是不够的, 因为顶点  $v_1$  和  $v_2$  每个都必须指定两种颜色, 而且不能对  $v_1$  和  $v_2$  指定相同颜色。(关于  $k$  重着色的更多信息, 见[MiRo91]。)



\*17. 设  $G$  和  $H$  是图 6-82 所示的图。求

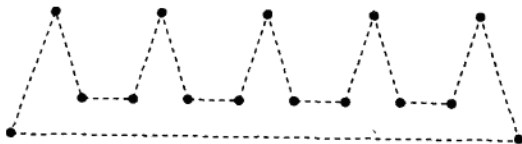
- a)  $\chi_2(G)$
- b)  $\chi_2(H)$
- c)  $\chi_3(G)$
- d)  $\chi_3(H)$

18. 移动广播(或蜂窝)电话的频率是按地段分配的。每个地段分配一组该地段里的车辆所使用的频率。在产生干扰问题的地段里不能使用相同频率。解释一下如何用  $k$  重着色来对一个地区里的每个移动广播地段分配  $k$  种频率。

著名的艺术馆问题求需要多少名保安才能看全艺术馆的所有部分, 这里艺术馆是一个  $n$  边形的边界及它所围的内部。为了更精确地描述这个问题, 需要一些术语。简单多边形  $P$  边界上或  $P$  内部的点  $x$  覆盖或看见  $P$  边界上或  $P$  内部的点  $y$ , 如果线段  $xy$  上所有的点都在  $P$  边界上或  $P$  内部。如果对于  $P$  边界上或  $P$  内部的每一个点  $y$ , 都能够在一个点的集合中找到一个看见  $y$  的点  $x$ , 就说这个点的集合是简单多边形的看守集。把看守简单多边形  $P$  所需的最少点数的看守集记为  $G(P)$ 。艺术馆问题求的就是一个函数  $g(n)$ , 它是所有  $n$  顶点的简单多边形  $P$  的看守集  $G(P)$  (的元素个数) 的最大值(考虑  $n$  顶点的简单多边形  $P$  的各种形态——译者注)。也就是说,  $g(n)$  是一个最小的正整数, 使得一个  $n$  顶点的简单多边形  $P$  保证可以被  $g(n)$  个或更少的保安看守。

\*19. 证明:  $g(5) = 1$ , 即所有的五边形都能够被一个点守卫。[提示: 先分出有 0 个、1 个、2 个顶点的内角大于 180 度, 然后再说明在各种情况下一个保安都足够了。]

\*20. 证明:  $g(n) \geq \lfloor n/3 \rfloor$ 。[提示: 考虑具有  $3k$  个顶点和  $k$  个齿尖的梳子状的多边形, 就像如图所示的 15 边形一样。]



## 关键术语和结果

### 术语

无向边：与集合 $\{u, v\}$ 关联的边，其中 $u$ 和 $v$ 都是顶点。

有向边：与有序对 $(u, v)$ 关联的边，其中 $u$ 和 $v$ 都是顶点。

多重边：连接同样一对顶点的不同的边。

环：连接一个顶点与它自身的边。

无向图：一组顶点以及连接这些顶点的一组无向边。

简单图：没有多重边和环的无向图。

多重图：可能包含多重边但不包含环的无向图。

伪图：可能包含多重边和环的无向图。

有向图：一组顶点以及连接这些顶点的一组有向边。

有向多重图：可能包含多重有向边的带有向边的图。

相邻：若两个顶点之间有边则它们是相邻的。

关联：若一个顶点是一条边的端点则那条边关联那个顶点。

$\deg(v)$  (无向图里顶点 $v$ 的度)：与 $v$ 关联的边的数目。

$\deg^-(v)$  (带有向边的图里顶点 $v$ 的入度)：以 $v$ 作为终点的边的数目。

$\deg^+(v)$  (带有向边的图里顶点 $v$ 的出度)：以 $v$ 作为起点的边的数目。

带有向边的图的底图：通过忽略边的方向所获得的无向图。

$K_n$  ( $n$ 个顶点的完全图)：带 $n$ 个顶点的无向图，其中每对顶点都用一条边连接。

偶图：顶点集划分成两个子集合 $V_1$ 和 $V_2$ 的图，使得每条边都连接 $V_1$ 里的顶点和 $V_2$ 里的顶点。

$K_{m,n}$  (完全偶图)：顶点集划分成 $m$ 个元素的子集和 $n$ 个元素的子集，使得两个顶点被一条边所连接，当且仅当一个顶点属于第一个子集而另外一个顶点属于第二个子集。

$C_n$  (大小为 $n$ 的圈图)， $n \geq 3$ ：带有 $n$ 个顶点 $v_1, v_2, \dots, v_n$ 和边 $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$ 的图。

$W_n$  (大小为 $n$ 的轮图)， $n \geq 3$ ：通过向 $C_n$ 添加一个顶点以及从这个顶点到 $C_n$ 里原来每个顶点的一条边所获得的图。

$Q_n$  ( $n$ 立方体图)， $n \geq 1$ ：用 $2^n$ 个长度为 $n$ 的位串作为顶点，边连接恰好相差一位的每对位串的图。

孤立点：0度顶点。

悬挂点：1度顶点。

正则图：所有顶点都有相同的度的图。

图 $G=(V, E)$ 的子图：图 $(W, F)$ ，其中 $W$ 是 $V$ 的子图而 $F$ 是 $E$ 的子图。

$G_1 \cup G_2$  ( $G_1$ 与 $G_2$ 的并图)：图 $(V_1 \cup V_2, E_1 \cup E_2)$ ，其中 $G_1=(V_1, E_1)$ 而 $G_2=(V_2, E_2)$ 。

邻接矩阵：利用顶点的相邻关系来表示图的矩阵。

关联矩阵：利用边与顶点的关联关系来表示图的矩阵。

同构的简单图：对简单图 $G_1=(V_1, E_1)$ 和简单图 $G_2=(V_2, E_2)$ 来说，若存在从 $V_1$ 到 $V_2$ 的一一对应 $f$ ，使得对所有属于 $V_1$ 的 $v_1$ 和 $v_2$ 来说， $\{f(v_1), f(v_2)\} \in E_2$ 当且仅当 $\{v_1, v_2\} \in E_1$ ，则 $G_1$ 与 $G_2$ 是同构的。

同构不变量：同构的图都有或都没有的性质。

无向图里从 $u$ 到 $v$ 的通路：一条或多条边的序列 $e_1, e_2, \dots, e_n$ ，其中对 $i=0, 1, \dots, n$ 来说， $e_i$ 关联着 $\{x_i, x_{i+1}\}$ ，其中 $x_0=u$ 而 $x_{n+1}=v$ 。

有向图里从 $u$ 到 $v$ 的通路：一条或多条边的序列 $e_1, e_2, \dots, e_n$ ，其中对 $i=0, 1, \dots, n$ 来说， $e_i$ 关联着 $(x_i, x_{i+1})$ ，其中 $x_0=u$ 而 $x_{n+1}=v$ 。

简单通路：不多次包含一条边的通路。

回路：在相同顶点处开始与结束的通路，通路长度 $n \geq 1$ 。

连通图：在图中每对顶点之间都有通路的无向图。

图  $G$  的连通分支：图  $G$  的最大连通子图。

强连通有向图：从每个顶点到每个顶点都存在有向通路的有向图。

有向图  $G$  的强连通分支： $G$  的最大强连通子图。

欧拉回路：恰好包含图的每条边一次的回路。

欧拉通路：恰好包含图的每条边一次的通路。

哈密顿通路：简单图  $G=(V, E)$  里的通路  $x_0, x_1, \dots, x_n$ ，使得  $\{x_0, x_1, \dots, x_n\}=V$  而且对  $0 \leq i < j \leq n$  来说有  $x_i \neq x_j$ 。

哈密顿回路：简单图里的回路  $x_0, x_1, \dots, x_n, x_0$ ，使得  $x_0, x_1, \dots, x_n$  是哈密顿通路。

带权图：为各边指定数字的图。

最短通路问题：确定带权图里的通路以使得这条通路里的边的权之和在所规定的顶点之间的所有通路上是最小值这样的问题。

旅行商问题：求访问图的每个顶点恰好一次的、总长度最短的回路的问题。

可平面图：可以画在平面上而没有边交叉的图。

可平面图的平面表示面：该图的平面表示把平面所分割成的区域。

初等细分：删除无向图的边  $\{u, v\}$  而且添加新顶点  $w$  以及边  $\{u, w\}$  和边  $\{w, v\}$ 。

同胚：若两个无向图是从同一个无向图通过一系列初等细分来获得的，则它们同胚。

图着色：给图的顶点指定颜色，使得相邻的两个顶点没有相同的颜色。

色数：在图的着色里所需要的最少颜色数。

## 结果

在连通多重图里存在欧拉回路当且仅当每个顶点都有偶数度。

在连通多重图里存在欧拉通路当且仅当至多两个顶点有奇数度。

迪克斯特拉算法：在带权图里求出两个顶点之间最短通路的过程(见 6.6 节)。

欧拉公式： $r=e-v+2$ ，其中  $r$ 、 $e$  和  $v$  分别是可平面图的平面表示的面数、边数和顶点数。

库拉图斯基定理：图是不可平面的当且仅当它包含同胚于  $K_{3,3}$  或  $K_5$  的子图。(其证明超出本书范围。)

四色定理：每个平面图都可以用不超过四种颜色来着色。(其证明远远超出本书范围!)

## 复习题

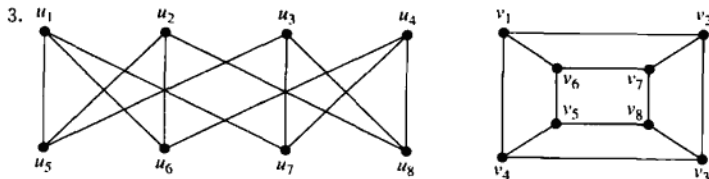
1. a) 定义：简单图，多重图，伪图，有向图，有向多重图。  
b) 用例子说明：如何用 a) 里每种类型的图来建模。例如，解释一下如何为计算机网络或飞行航线的不同方面来建模。
2. 给出如何用图建模的至少四个例子。
3. 在无向图里，顶点度数之和与该图里边数之间的关系是什么？解释这个关系为什么成立。
4. 为什么在无向图里奇数度顶点的个数必然是偶数？
5. 在有向图里顶点的入度之和与出度之和之间的关系是什么？解释这个关系为什么成立。
6. 描述下列图族。
  - a)  $K_n$ ，在  $n$  个顶点上的完全图
  - b)  $K_{m,n}$ ，在  $m$  和  $n$  个顶点上的完全偶图
  - c)  $C_n$ ，带  $n$  个顶点的圈图
  - d)  $W_n$ ，大小为  $n$  的轮图
  - e)  $Q_n$ ， $n$  立方体
7. 在练习 6 的图族里，每个图有多少个顶点和多少条边？
8. a) 什么是偶图？  
b) 图  $K_n$ 、 $C_n$  和  $W_n$  中哪些是偶图？

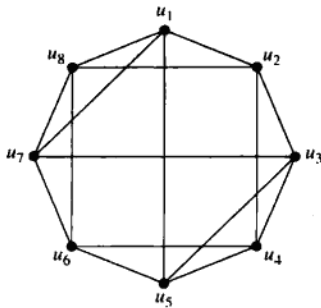
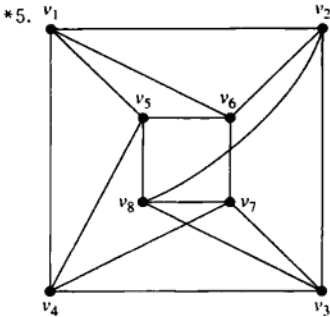
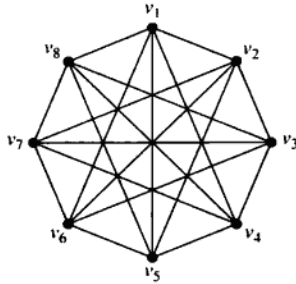
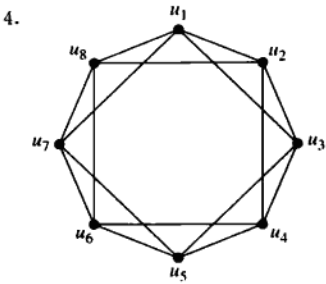
- c) 你如何确定无向图是否为偶图?
9. a) 描述用来表示图的三种不同方法。  
b) 画出至少带 5 个顶点和 8 条边的简单图。说明如何用你在 a) 里所描述的方法来表示它。
10. a) 两个简单图是同构的是什么意思?  
b) 对于简单图的同构来说, 不变量是什么意思? 给出至少五个这样的不变量的例子。  
c) 给出带有相同的顶点数、边数和顶点度但不同构的两个简单图的例子。  
d) 是否有一组已知的不变量可以用来有效地确定两个简单图是否同构?
11. a) 图是连通的是什么意思?  
b) 什么是图的连通分支?
12. a) 解释一下如何用邻接矩阵来表示图。  
b) 如何用邻接矩阵来确定从图  $G$  的顶点集到图  $H$  的顶点集的函数是否是同构的?  
c) 如何用图的邻接矩阵来确定在图的两个顶点之间长度为  $r$  的通路数? 其中  $r$  是正整数。
13. a) 定义无向图里的欧拉回路和欧拉通路。  
b) 描述著名的哥尼斯堡七桥问题, 并且解释一下如何利用欧拉回路来重新叙述它。  
c) 如何确定无向图是否具有欧拉通路?  
d) 如何确定无向图是否具有欧拉回路?
14. a) 定义简单图里的哈密顿回路。  
b) 给出一些性质, 这些性质蕴含着简单图没有哈密顿回路。
15. 给出至少两个可以通过求出带权图里最短通路来解决的问题的例子。
16. a) 描述求在带权图两个顶点之间的最短通路的迪克斯特拉算法。  
b) 画出至少带 10 个顶点和 20 条边的带权图。用迪克斯特拉算法求出在图中你所选择的两个顶点之间的最短通路。
17. a) 图是可平面的是什么意思?  
b) 给出不可平面的简单图的例子。
18. a) 平面图的欧拉公式是什么?  
b) 如何用平面图的欧拉公式来证明简单图是不可平面的?
19. 叙述关于图的可平面性的库拉图斯基定理, 并且解释一下它如何刻画了哪些图是可平面的。
20. a) 定义图的色数。  
b) 当  $n$  是正整数时, 图  $K_n$  的色数是什么?  
c) 当  $n$  是大于 2 的正整数时, 图  $C_n$  的色数是什么?  
d) 当  $m$  和  $n$  都是正整数时, 图  $K_{m,n}$  的色数是什么?
21. 叙述四色定理。有没有不能用四种颜色来着色的图?
22. 解释一下在建模里可以如何使用图的着色。至少举两个不同的例子。

## 补充练习

1. 一个带 100 个顶点的 50 正则图有多少条边?  
2.  $K_3$  有多少种非同构的子图?

在练习 3~5 里, 确定所给的成对的图是否同构。





完全  $m$  部图  $K_{n_1, n_2, \dots, n_m}$  的顶点划分成  $m$  个子集合, 各有  $n_1, n_2, \dots, n_m$  个元素, 而且顶点相邻当且仅当它们属于这个划分的不同子集合。

6. 画出下列图。

a)  $K_{1,2,3}$

b)  $K_{2,2,2}$

c)  $K_{1,2,2,3}$

\*7. 完全  $m$  部图  $K_{n_1, n_2, \dots, n_m}$  有多少个顶点和多少条边?

\*8. a) 证明或反驳: 在至少有两个顶点的有穷简单图里, 总是存在两个度数相同的顶点。

b) 对有穷多重图做与(a)里同样的事。

设  $G=(V, E)$  是简单图。顶点集合  $V$  的子集合  $W$  的导出子图是图  $(W, F)$ , 其中边集合  $F$  包含着  $E$  里的一条边当且仅当这条边的两个端点都属于  $W$ 。

9. 考虑 6.4 节图 6-42 所示的图。求下列顶点的导出子图

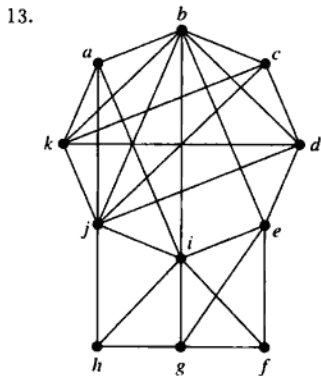
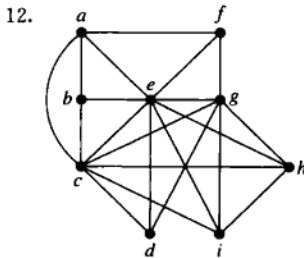
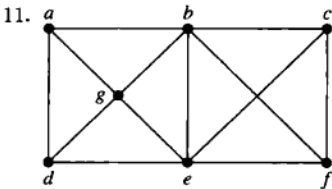
a)  $\{a, b, c\}$

b)  $\{a, e, g\}$

c)  $\{b, c, f, g, h\}$

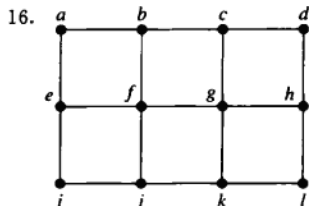
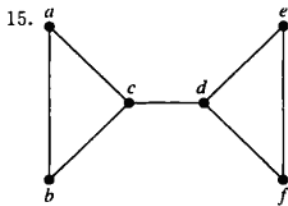
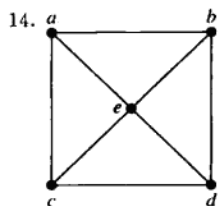
10. 设  $n$  是正整数。证明:  $K_n$  的顶点集的非空子集的导出子图是完全图。

~~定义~~ 简单无向图里的团是一个完全子图, 它不包含在任何更大的完全子图里。在练习 11~13 里, 求所给的图的所有团。





简单图里顶点的支配集是顶点的一个集合,使得其他每个顶点都与这个集合里至少一个顶点是相邻的。带最少顶点数的支配集称为最小支配集。在练习 14~16 里,求所给的图的最小支配集。



简单图可用来确定在棋盘上控制整个棋盘的最少皇后数。一个  $n \times n$  的棋盘具有  $n^2$  个格子。如下图所示,在所给位置里的皇后控制着同行、同列以及包含这个格子的两条斜线上的所有格子。与此对应的简单图具有  $n^2$  个顶点,每个顶点表示一个格子,而且若一个顶点所表示的格子里的皇后控制着另外一个顶点所表示的格子,则这两个顶点是相邻的。

17. 构造表示  $n \times n$  棋盘的简单图,用边表示皇后对格子的控制,其中

a)  $n=3$                       b)  $n=4$

18. 解释一下最小支配集的概念如何应用到确定控制  $n \times n$  棋盘的最少皇后数的问题。

- \*\*19. 求控制  $n \times n$  棋盘的最少皇后数,其中

a)  $n=3$                       b)  $n=4$                       c)  $n=5$

20. 假定  $G_1$  和  $H_1$  是同构的而且  $G_2$  和  $H_2$  是同构的。证明或反证:  $G_1 \cup G_2$  和  $H_1 \cup H_2$  是同构的。

21. 证明:下列性质是同构的简单图都有或都没有的不变量。

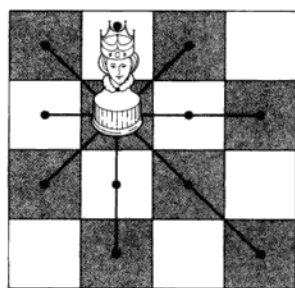
a) 连通性                      b) 哈密顿回路的存在性  
c) 欧拉回路的存在性                      d) 有交叉数  $C$   
e) 有  $n$  个孤立顶点                      f) 是偶图

22. 如何从  $G$  的邻接矩阵求  $\bar{G}$  的邻接矩阵? 其中  $G$  是简单图。

23. 有多少种非同构的、带有 4 个顶点的连通简单偶图?

- \*24. 有多少种非同构的简单连通图带有 5 个顶点并且

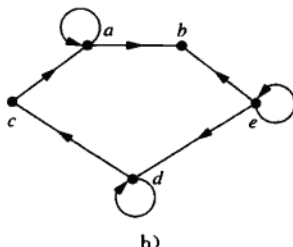
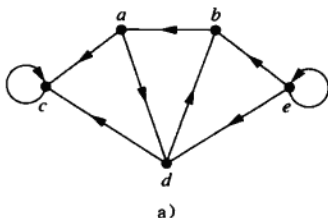
a) 没有任何顶点的度超过 2?  
b) 色数等于 4?  
c) 非可平面的?



皇后所控制的格子

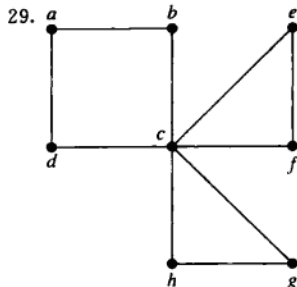
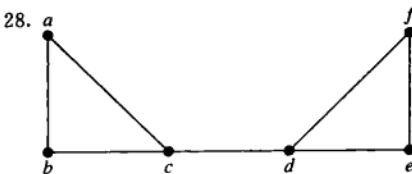
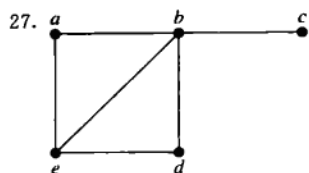
若有向图与它的逆同构,则它是自逆的。

25. 确定下列图是否为自逆的。



26. 证明:若有向图  $G$  是自逆的而且  $H$  是同构于  $G$  的有向图,则  $H$  也是自逆的。

无向简单图的定向就是指定它的各边的方向,使得所得到的有向图是强连通的。当无向图有定向时,这个图称为可定向的。在练习 27~29 里,确定给定的图是否可定向。



30. 因为在城市中心区交通流量正在增长, 所以交通工程师正在计划把目前所有双向街道都变成单行街道。解释一下如何为这个问题建模。

\*31. 证明: 若一个图具有割边, 则它不是可定向的。

竞赛图是简单有向图, 使得若  $u$  和  $v$  是图里不同的顶点, 则  $(u, v)$  和  $(v, u)$  中恰好有一个是图中的边。

32. 有多少种不同的带  $n$  个顶点的竞赛图?

33. 在竞赛图里一个顶点的入度与出度之和是什么?

\*34. 证明: 每个竞赛图都有哈密顿通路。

35. 给定鸡群里的两只鸡, 其中一只占优势。这样就定义了这个鸡群的啄食次序。如何用竞赛图来为啄食次序建模?

36. 假定  $G$  是带  $2k$  个奇数度顶点的连通多重图。证明: 存在  $k$  个子图, 它们的并图是  $G$ , 其中每个子图都有欧拉通路并且任何两个子图都没有公共边。[提示: 向该图添加  $k$  条边, 连接成对的奇数度顶点, 并且利用这个变大了的图的欧拉回路。]

\*37. 设  $G$  是带有  $n$  个顶点的简单图。 $G$  的带宽表示成  $B(G)$ , 它是  $\max\{|i-j| \mid a_i \text{ 与 } a_j \text{ 是相邻的}\}$  在  $G$  的顶点  $a_1, a_2, \dots, a_n$  的所有排列上所取的最小值。即带宽是赋给相邻顶点的下标的最大差值在顶点的所有列表上所取的最小值。求下列图的带宽。

a)  $K_5$       b)  $K_{1,3}$       c)  $K_{2,3}$       d)  $K_{3,3}$       e)  $Q_3$       f)  $C_5$

\*38. 连通简单图的两个不同顶点  $v_1$  和  $v_2$  之间的距离是在  $v_1$  和  $v_2$  之间的最短通路的长度(边数)。图的半径是从顶点  $v$  到其他顶点的最大距离在所有顶点  $v$  上所取的最小值。图的直径是在两个不同顶点之间的最大距离。求下列图的半径和直径。

a)  $K_6$       b)  $K_{4,5}$       c)  $Q_3$       d)  $C_5$

\*39. a) 证明: 若简单图  $G$  的直径至少为 4, 则它的补图  $\bar{G}$  的直径不超过 2。

b) 证明: 若简单图  $G$  的直径至少为 3, 则它的补图  $\bar{G}$  的直径不超过 3。

\*40. 假定一个多重图有  $2m$  个奇数度顶点。证明: 任何包含该图中每条边的回路, 必然至少包含  $m$  条边超过一次。

41. 求 6.6 节图 6-64 中在顶点  $a$  与  $z$  之间的次最短通路。

42. 设计一个算法, 它在简单连通带权图里, 求在两个顶点之间的最短通路。

43. 求 6.6 节图 6-65 中在顶点  $a$  与  $z$  之间经过顶点  $e$  的最短通路。

44. 设计一个算法, 它在简单连通带权图里, 求在两个顶点之间经过第三个指定顶点的最短通路。

\*45. 证明: 若  $G$  是至少带 11 个顶点的简单图, 则或者  $G$  不是可平面的, 或者  $\bar{G}$  不是可平面的,  $\bar{G}$  是  $G$  的补图。

若在图中一组顶点的集合里任何两个顶点都不相邻, 则这个顶点集合称为独立的。图的独立数是图中顶点独立集里的最大顶点数。

\*46. 下列图的独立数是什么?

a)  $K_n$       b)  $C_n$       c)  $Q_n$       d)  $K_{m,n}$

47. 证明: 一个简单图里的顶点数小于或等于这个图的独立数与色数之积。

48. 证明: 一个图的色数小于或等于  $v-i+1$ , 其中  $v$  是这个图的顶点数,  $i$  是这个图的独立数。

49. 假定为了生成带有  $n$  个顶点的随机简单图, 首先选择满足  $0 \leq p \leq 1$  的实数  $p$ 。对  $C(n, 2)$  对不同顶点中的每一对, 都生成一个在 0 与 1 之间的随机数  $x$ 。若  $0 \leq x \leq p$ , 则用一条边连接这两个顶点; 否则就不连接

这两个顶点。

- a) 生成带有  $m$  条边的图的概率是什么? 其中  $0 \leq m \leq C(n, 2)$ 。  
 b) 若以概率  $p$  来包含每一条边, 则在随机生成的带有  $n$  个顶点的图中, 边数的期望值是什么?  
 c) 证明: 若  $p=1/2$ , 则以相等的概率生成每一个带  $n$  个顶点的简单图。

每当向简单图添加更多的边(不添加顶点)时, 都还保持的性质称为单调增的, 每当从简单图删除边(不删除顶点)时, 都还保持的性质称为单调减的。

50. 对下列每个性质来说, 确定它是否为单调增的并且确定它是否为单调减的。

- a) 图  $G$  是连通的      b) 图  $G$  不是连通的      c) 图  $G$  有欧拉回路      d) 图  $G$  有哈密顿回路  
 e) 图  $G$  是可平面的      f) 图  $G$  的色数为 4      g) 图  $G$  的半径为 3      h) 图  $G$  的直径为 3

51. 证明: 图的性质  $P$  是单调增的当且仅当图的性质  $Q$  是单调减的, 其中  $Q$  是不具有性质  $P$  的性质。

\*\*52. 假定  $P$  是简单图的单调增的性质。证明: 带  $n$  个顶点的随机图有性质  $P$  的概率是挑选一条边属于图的概率  $p$  的单调非减函数。

## 计算机题目

按下述给定的输入与输出写出程序。

1. 给定无向图的各边所关联的顶点对, 确定每个顶点的度。
2. 给定有向图的各边所关联的有序顶点对, 确定每个顶点的入度和出度。
3. 给定简单图的边列表, 确定这个图是否为偶图。
4. 给定图的各边所关联的顶点对, 构造这个图的邻接矩阵。
5. 给定图的邻接矩阵, 列出这个图的各边, 并且给出每条边出现的次数。
6. 给定无向图各边所关联的顶点对, 以及每条边出现的次数, 构造这个图的关联矩阵。
7. 给定无向图的关联矩阵, 列出它的各边, 并且给出每条边出现的次数。
8. 给定正整数  $n$ , 通过产生图的邻接矩阵来生成无向图, 使得以相等的概率来生成所有的简单图。
9. 给定正整数  $n$ , 通过产生图的邻接矩阵来生成有向图, 使得以相等的概率来生成所有的有向图。
10. 给定两个都带不超过 6 个顶点的简单图的边列表, 确定这两个图是否同构。
11. 给定图的邻接矩阵和正整数  $n$ , 求顶点两两之间长度为  $n$  的通路数(产生对有向图和无向图来说都能工作的程序)。
- \*12. 给定简单图的边列表, 确定它是否连通, 若它不连通, 则求连通分支数。
13. 给定多重图的各边所关联的顶点对, 确定它是否有欧拉回路, 若没有欧拉回路, 则确定它是否有欧拉通路。若存在欧拉通路或欧拉回路, 则构造这样的通路或回路。
- \*14. 给定有向多重图的各边所关联的有序顶点对, 若存在欧拉通路或欧拉回路, 则构造这样的通路或回路。
- \*\*15. 给定简单图的边列表, 产生一条哈密顿回路, 或者确定该图没有这样的回路。
- \*\*16. 给定简单图的边列表, 产生一条哈密顿通路, 或者确定该图没有这样的通路。
17. 给定带权连通简单图的边及其权的列表, 以及该图中的两个顶点, 用迪克斯特拉算法求这两点间最短通路的长度。另外, 求出这条通路。
18. 给定无向图的边的表, 用 6.8 节练习里所给的算法求这个图的着色。
19. 给定学生及其注册课程的表, 构造期末考试日程表。
20. 给定各对电视台之间的距离, 为这些台分配频率。

## 计算和研究

利用计算程序或你所编写的程序来做下面的练习。

1. 显示带 4 个顶点的所有简单图。
2. 显示全套的带 6 个顶点的所有非同构的简单图。
3. 显示全套的带 4 个顶点的所有非同构的有向图。
4. 随机地生成 10 个不同的简单图, 每个带 20 个顶点, 使得每个这样的图都是以相等的概率来生成的。
5. 构造一种格雷码, 其中码字都是长度为 6 的位串。
6. 构造马在不同大小的棋盘上的周游路线。

7. 确定你在本组练习的练习 4 里生成的每个图是否为平面图。若你可以做到, 则确定每个非平面图的厚度。
8. 确定你在本组练习的练习 4 里生成的每个图是否连通。若有一个图不连通, 则确定这个图的连通分支数。
9. 随机地生成带 10 个顶点的简单图。当你生成了一个带欧拉回路的图时停止。显示这个图里的一个欧拉回路。
10. 随机地生成带 10 个顶点的简单图。当你生成了一个带哈密顿回路的图时停止。显示这个图里的一个哈密顿回路。
11. 求你在本组练习的练习 4 里所生成的每个图的色数。
- \*\*12. 求旅行推销员访问美国 50 个州的每个首府所能采取的最短路线, 飞机在各城市之间直线通行。
- \*13. 对每个不超过 10 的正整数  $n$  来说, 估计随机生成的带  $n$  个顶点的简单图连通的概率, 方法是生成一组随机简单图并且确定每个图是否连通。
- \*\*14. 研究这个问题: 确定  $K(7, 7)$  的交叉数是否为 77, 79 或 81。已知它等于这三个数中的一个。

## 写作题目

利用本书以外的资料, 就下列问题写作短文。

1. 描述一下在 1900 年以前图论的起源和发展。
2. 讨论一下图论在生态系统研究中的应用。
3. 讨论一下图论在社会学和心理学中的应用。
4. 讨论一下通过研究网络图的性质可以了解到什么?
5. 描述一下给定一个图的顶点和边, 在纸面或屏幕上画出这个图的算法。
6. 一个输入、显示和操纵各种图的软件工具应当具有什么功能? 现有的工具都具有这些功能中的哪些?
7. 描述一下确定两个图是否同构的一些可用算法和这些算法的计算复杂性。目前已知最有效的算法是什么?
8. 描述一下如何用欧拉通路来帮助确定 DNA 序列。
9. 定义一下德布鲁因序列并且讨论它们如何出现在应用里。解释如何用欧拉回路来构造德布鲁因序列。
10. 描述一下中国邮递员问题并且解释如何解决这个问题。
11. 描述一下蕴含着图具有哈密顿回路的一些不同条件。
12. 描述一下用来解决旅行推销员问题的几个不同策略和算法。
13. 描述一下确定一个图是否可平面的几个不同算法。每个算法的计算复杂性是什么?
14. 在建模中, 有时把大规模集成电路(VLSI)图嵌入到一本书中, 让顶点都在书脊上而边都在不同的书页上(在同一页之内边与边互不相交)。定义一下图的书页数(book number, 即上述做法所需要的最少页数——译者注)并且对  $n=3, 4, 5$  和 6 求包括  $K_n$  在内的各种图的书页数。
15. 描述一下四色定理的历史。
16. 描述一下在四色定理的证明中计算机所扮演的角色。如何肯定一个依赖计算机的证明是正确的?
17. 就是否产生最少颜色的着色以及复杂性而言, 描述并比较一下给图着色的几个不同算法。
18. 解释一下在各种不同模型里如何使用图的多重着色。
19. 解释一下在带特定性质的图的非构造性存在性证明中如何使用随机图理论。

# 树

不包含简单回路的连通图称为树。早在1857年英国数学家亚瑟·凯莱就用树去计数某些类型的化合物。本章中的例子将说明从那时起，树已经被用来解决各种学科分支里的问题。

树在计算机科学里特别有用，尤其是在算法中。例如，用树来构造求出项在表中位置的有效算法；可以用树构造有效编码以节省数据存储和传输成本，比如赫夫曼编码；可以用树来研究诸如跳棋和象棋这样的博弈，并且可以帮助确定进行这些博弈的取胜策略；可以用树来为通过一系列决策而完成的过程建立模型。构造这些模型可以帮助确定比如排序算法这样的基于一系列决策的算法的计算复杂性。

通过深度优先搜索或宽度优先搜索，可以系统地遍历图的顶点，构造出一棵包括每个顶点的树来。通过深度优先搜索来探索图的顶点，也称为回溯，允许系统地搜索五花八门的问题的解，比如确定在棋盘上如何放置8个王后使得这些王后不能互相攻击。

可以给树的边赋权来为许多问题建立模型。例如，用加权树可以开发出构造网络的算法，这些网络含有最便宜的连接不同网络节点的电话线集合。

## 7.1 概述

第6章说明了如何用图来建立模型和解决许多问题。本章将集中讨论称为树的一种特殊类型的图，之所以这样命名是因为这样的图就像是树。例如，家族树是表示族谱图的图。家族树用顶点表示家族成员并且用边表示亲子关系。图7-1显示瑞士数学世家伯努利家族的家族树。表示家族树的无向图是树的一个例子。

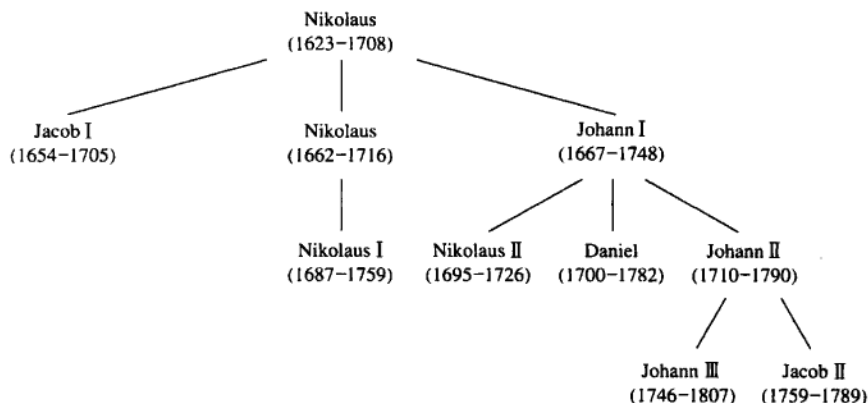


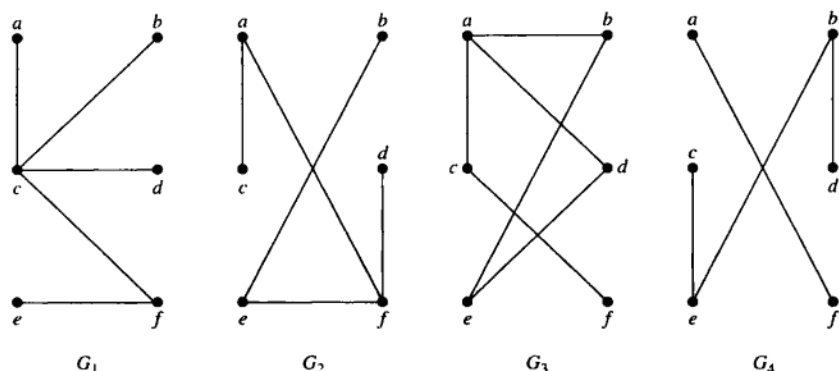
图 7-1 伯努利数学世家

**定义 1** 树是没有简单回路的连通无向图。

因为树没有简单回路，所以树不含多重边或环。因此任何树都必然是简单图。

**例 1** 在图 7-2 所示的图中，哪些图是树？

**解**  $G_1$  和  $G_2$  是树，因为都是没有简单回路的连通图。 $G_3$  不是树，因为  $e, b, a, d, e$  是这个图中的简单回路。最后， $G_4$  不是树，因为它不连通。

图 7-2  $G_1$  和  $G_2$  是树,  $G_3$  和  $G_4$  不是树

任何一个不包含简单回路的连通图都是树。不含简单回路但不一定连通的图是什么? 这些图称为森林, 而且具有这样的性质: 它们的每个连通分支都是树。图 7-3 显示一个森林。

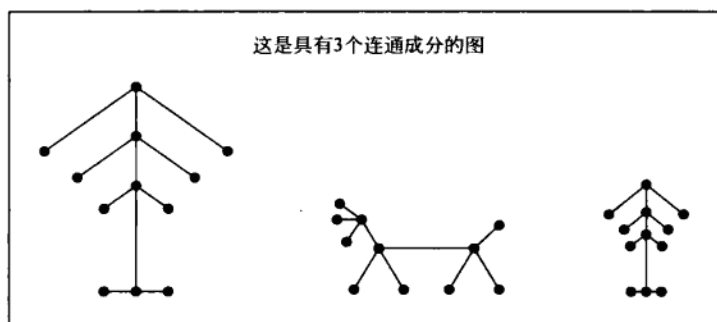


图 7-3 一个森林的例子

通常把树定义成具有在每对顶点之间存在唯一简单通路的性质的无向图。下面的定理说明这个定义与原来的定义等价。

**定理 1** 一个无向图是树当且仅当在它的每对顶点之间存在唯一简单通路。

**证** 首先假定  $T$  是树。则  $T$  是没有简单回路的连通图。设  $x$  和  $y$  是  $T$  的两个顶点。因为  $T$  是连通的, 根据 6.4 节定理 1, 在  $x$  和  $y$  之间存在一条简单通路。另外, 这条通路必然是唯一的, 因为假如存在第二条这样的通路, 那么从  $x$  到  $y$  的第一条通路以及将第二条通路逆转后所得到的从  $y$  到  $x$  的通路, 将组合起来形成回路。利用 6.4 节练习 24, 这蕴含着在  $T$  中存在简单回路。因此, 在树的任何两个顶点之间存在唯一简单通路。

现在假定在图  $T$  的任何两个顶点之间存在唯一简单通路。则  $T$  是连通的, 因为在它的任何两个顶点之间存在通路。另外,  $T$  没有简单回路。为了看出这是个真命题, 假定  $T$  有包含顶点  $x$  和  $y$  的简单回路。则在  $x$  和  $y$  之间就有两条简单通路, 因为这条简单回路包含一条从  $x$  到  $y$  的简单通路和一条从  $y$  到  $x$  的简单通路。因此, 在任何两个顶点之间存在唯一简单通路的图是树。□

在树的许多应用里, 指定树的一个特殊顶点作为根。一旦规定了根, 就可以给每条边指定方向如下。因为从根到图的每个顶点存在唯一通路(根据定理 1), 所以指定每条边是离开根的方向。因此, 树与它的根一起产生一个有向图, 称为根树。

**定义 2** 根树是指定一个顶点作为根并且每条边的方向都离开根树。

也可以递归地定义根树。通过选择任何一个顶点来作为根, 就可以把非根树变成根树。注意

对根的不同选择会导致产生不同的根树。例如，图 7-4 显示通过在树  $T$  里分别指定  $a$  和  $c$  作为根所形成的根树。通常在画根树时把根画在图的顶端。指示根树中边的方向的箭头可以省略，因为对根的选择确定了边的方向。

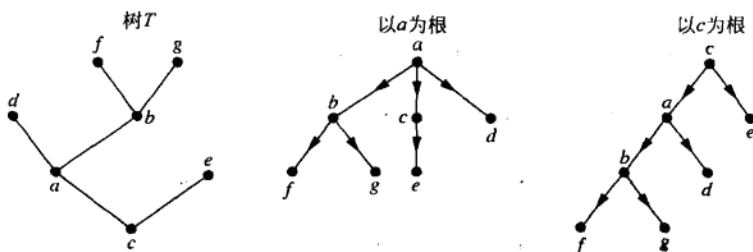


图 7-4 一棵树以及指定两个根所形成的根树

树这个术语起源于植物学和族谱学。假定  $T$  是根树。若  $v$  是  $T$  里非根的顶点，则  $v$  的父母是使得从  $u$  到  $v$  存在有向边的唯一的顶点  $u$  (读者应当证明这样的顶点  $u$  是唯一的)。当  $u$  是  $v$  的父母时， $v$  称为  $u$  的子女。具有相同父母的顶点称为兄弟。非根顶点的祖先是根到该顶点的通路上的顶点，不包括该顶点自身但包括根 (即该顶点的父母，该顶点的父母的父母，等等，一直到根为止)。顶点  $v$  的后代是以  $v$  作为祖先的顶点。树的顶点若没有子女则称为树叶。有子女的顶点称为内点。根是内点，除非它是图中唯一的顶点，在这种情况下它是树叶。

若  $a$  是树里的顶点，则以  $a$  为根的子树是由  $a$  和  $a$  的后代以及这些顶点所关联的边所组成的该树的子图。

**例 2** 在图 7-5 所示的根树里 (有根  $a$ )，求  $c$  的父母， $g$  的子女， $h$  的兄弟， $e$  的所有祖先， $b$  的所有后代，所有内点以及所有树叶。什么是根在  $g$  处的子树？

**解**  $c$  的父母是  $b$ 。 $g$  的子女是  $h$ ， $i$  和  $j$ 。 $h$  的兄弟是  $i$  和  $j$ 。 $e$  的祖先是  $c$ ， $b$  和  $a$ 。 $b$  的后代是  $c$ ， $d$  和  $e$ 。内点是  $a$ ， $b$ ， $c$ ， $g$ ， $h$  和  $j$ 。树叶是  $d$ ， $e$ ， $f$ ， $i$ ， $k$ ， $l$  和  $m$ 。根在  $g$  处的子树如图 7-6 所示。

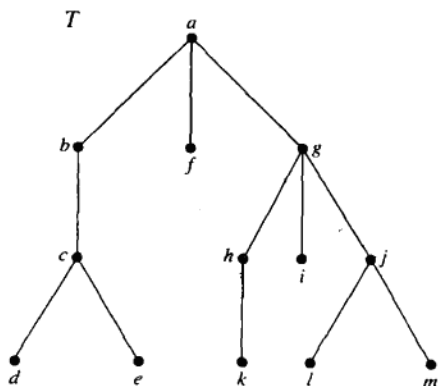


图 7-5 根树  $T$

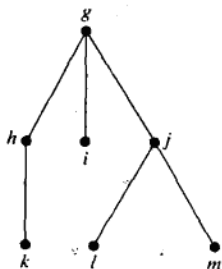


图 7-6 根在  $g$  处的子树

在许多不同的应用里都使用具有下面性质的根树：它们的所有内点都有同样个数的子女。在本章后面将用这样的树去研究涉及搜索、排序和编码的问题。

**定义 3** 若根树的每个内点都有不超过  $m$  个子女，则称它为  $m$  元树。若该树的每个内点都恰好有  $m$  个子女，则称它为正则  $m$  元树。把  $m=2$  的正则  $m$  元树称为二叉树。

**例 3** 图 7-7 里的根树，对某个正整数  $m$  来说是否为正则  $m$  元树？

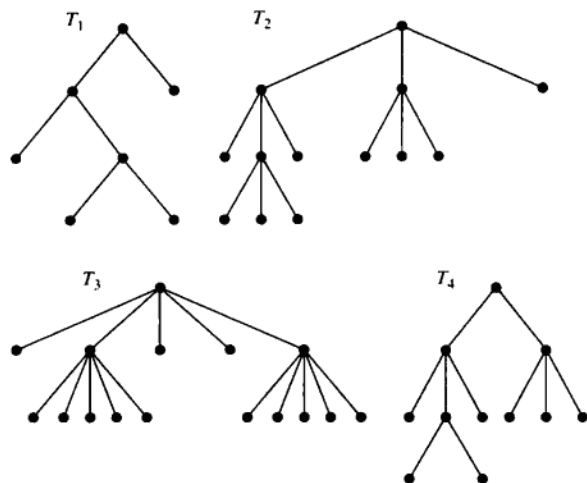


图 7-7 四个根树

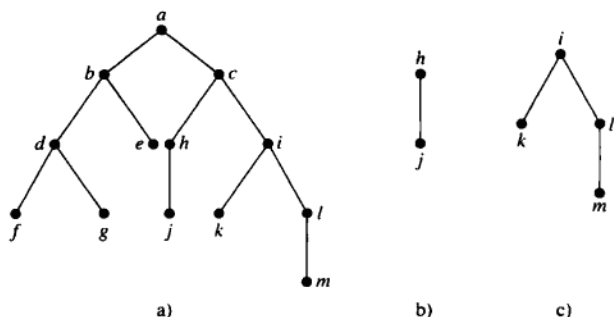
**解**  $T_1$  是正则二叉树, 因为它的每个内点都有两个子女。 $T_2$  是正则三元树, 因为它的每个内点都有三个子女。在  $T_3$  里每个内点都有五个子女, 所以它是正则五元树。对任何  $m$  来说,  $T_4$  都不是正则  $m$  元树, 因为它的某些内点有两个子女而其他内点有三个子女。 ■

有序根树是把每个内点的子女们都排序的根树。画有序根树时, 以从左向右的顺序来显示每个内点的子女们。注意在常规方式下根树的表示将确定它的边的一种顺序。我们将在画图时使用边的这种顺序, 但不明确地指出认为根树是有序的。

在有序二叉树(通常只称为二叉树)中, 若一个内点有两个子女, 则第一个子女称为左子女而第二个子女称为右子女。根处在一个顶点的左子女处的树称为该顶点的左子树, 而根处在一个顶点的右子女处的树称为该顶点的右子树。读者应当注意, 对某些应用来说, 二叉树的每个非根顶点都指定为其父母的右子女或左子女。即使当某些顶点仅有一个子女时也这样做。具体指定方式视需要而定。

可以递归地定义有序根树。

**例 4** 在图 7-8a 所示二叉树  $T$  里, 什么是  $d$  的左子女和右子女(其中顺序是画法所蕴含的)? 什么是  $c$  的左子树和右子树?

图 7-8 二叉树  $T$  和顶点  $c$  的左子树和右子树

**解**  $d$  的左子女是  $f$  而右子女是  $g$ 。在图 7-8b 和图 7-8c 中分别显示  $c$  的左子树和右子树。 ■  
与图的情形恰好一样, 不存在用来描述树、根树、有序根树和二叉树等的标准术语。出现这



种非标准的术语是因为在计算机科学里大量地使用树，而计算机科学还是相对年轻的领域。每当碰到关于树的术语时，读者就应当仔细地核对这些术语所表示的意思。

### 7.1.1 树作为模型

以树为模型的应用领域非常广泛，比如计算机科学、化学、地理学、植物学和心理学等。下面将描述基于树的各式各样的模型。

**例5 饱和碳氢化合物与树** 图可以用来表示分子，其中用顶点表示原子，用边表示原子之间的化学键。英国数学家亚瑟·凯莱在1857年发现了树，当时他正在试图列举形如  $C_nH_{2n+2}$  的化合物的同分异构体，它们都称为饱和碳氢化合物。

在饱和碳氢化合物的图模型里，用4度顶点表示每个碳原子，用1度顶点表示每个氢原子。在形如  $C_nH_{2n+2}$  的化合物的表示图里有  $3n+2$  个顶点。在这个图中，边数是顶点度数之和的一半。因此，在这个图中有  $(4n+2n+2)/2=3n+1$  条边。因为这个图是连通的，而且边数比顶点数少1，所以它必然是树。

带有  $n$  个4度顶点和  $2n+2$  个1度顶点的非同构的树，就表示  $C_nH_{2n+2}$  的不同的同分异构体。例如，当  $n=4$  时，恰好存在两个这种类型的不同的同分异构体。它们的结构如图7-9所示。这两种同分异构体称为丁烷和异丁烷。

**例6 表示组织机构** 大的组织机构的结构可以用根树来建模。在这个树里每个顶点表示机构里的一个职务。从一个顶点到另外一个顶点的边的起点所表示的人是终点所表示的人的(直接)上司。图7-10所示就是这样的树。在这个树所表示的组织机构里，硬件开发主任直接为研发经理工作。

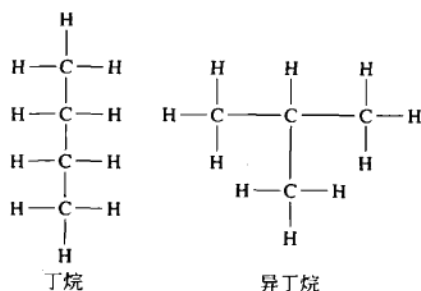


图7-9 丁烷的两种同分异构体

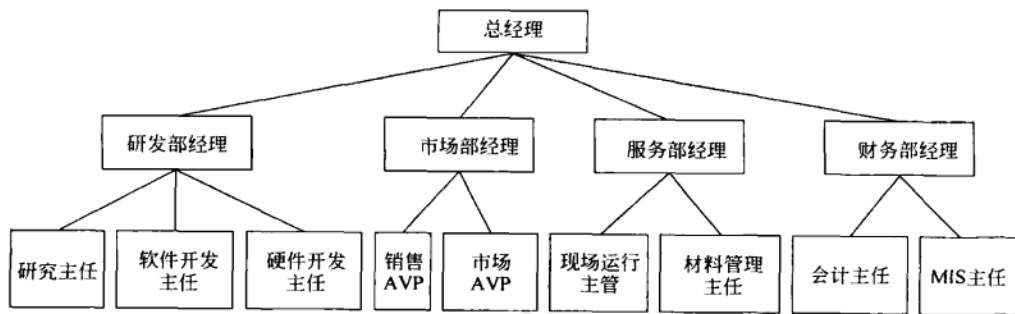


图7-10 一家计算机公司的组织机构图

**例7 计算机文件系统** 计算机存储器中的文件可以组织成目录。目录可以包含文件和子目录。根目录包含整个文件系统。因此，文件系统可以表示成根树，其中根表示根目录，内点表示子目录，树叶表示文件或空目录。在图7-11中显示一个这样的文件系统。在该系统中，文件 *kh* 属于目录 *rje*。(注意文件的链接，同一个文件会有一个以上的路径名，导致计算机文件系统中有回路。)

**例8 树形连接并行处理系统** 在6.2节例16里描述过多种并行处理的互连网络。树形连接网络是把处理器互相连接的另外一种重要方式。这样的网络的表示图是正则二叉树。这样的网络把  $n=2^k-1$  个处理器互连起来，其中  $k$  是正整数。一个非根也非树叶的顶点  $v$  所表示的处理器具有三个双向连接，一个连接通向  $v$  的父母所表示的处理器，两个连接通向  $v$  的两个子女所表示的处理器。根所表示的处理器具有两个双向连接，分别通向  $v$  的两个子女所表示的处理器。树叶所表

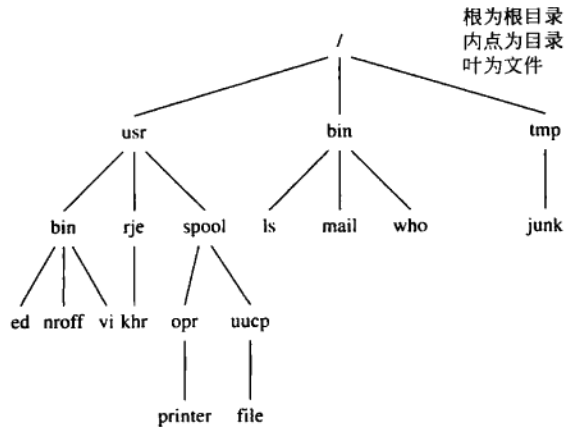


图 7-11 一个计算机文件系统

示的处理器具有一个双向连接，通向  $v$  的父母。在图 7-12 里显示带 7 个处理器的树形连接网络。

将要说明并行计算是如何使用树形连接网络的。具体地说，将说明图 7-12 里的处理器是如何用三步来完成 8 个数的相加的。在第一步用  $P_4$  相加  $x_1$  和  $x_2$ 。用  $P_5$  相加  $x_3$  和  $x_4$ 。用  $P_6$  相加  $x_5$  和  $x_6$ 。用  $P_7$  相加  $x_7$  和  $x_8$ 。在第二步用  $P_2$  相加  $x_1+x_2$  和  $x_3+x_4$ 。用  $P_3$  相加  $x_5+x_6$  和  $x_7+x_8$ 。最后，在第三步用  $P_1$  相加  $x_1+x_2+x_3+x_4$  和  $x_5+x_6+x_7+x_8$ 。这种方法要优于串行地相加 8 个数所需要的七步，串行的步骤是依次把一个数与表中前面各数之和相加。

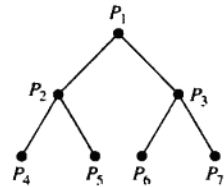


图 7-12 带 7 个处理器的树形连接网络

### 7.1.2 树的性质

我们常常需要知道树里各种各样的边和顶点的数目之间的联系。

**定理 2** 带有  $n$  个顶点的树含有  $n-1$  条边。

**证** 将用数学归纳法来证明这个定理。注意对于所有的树来说，在这里可以选择一个树根并且认为这个树有根。

**基础步骤：**当  $n=1$  时，有  $n=1$  个顶点的树没有边。所有对于  $n=1$  来说定理为真。

**归纳步骤：**归纳假设说有  $k$  个顶点的每个树都有  $k-1$  条边，其中  $k$  是正整数。假设树  $T$  有  $k+1$  个顶点并且  $v$  是  $T$  的树叶 ( $v$  必定存在，因为树是有穷的)，设  $w$  是  $v$  的父母，从  $T$  删除顶点  $v$  和连接  $w$  和  $v$  的边，就产生有  $k$  个顶点的树  $T'$ ，因为所得出的图还是连通的并且没有简单回路。根据归纳假设， $T'$  有  $k-1$  条边。所以  $T$  有  $k$  条边，因为  $T$  比  $T'$  多 1 条边，即连接  $v$  和  $w$  的边。这样就完成了归纳步骤。  $\square$

下面的定理说明，带有指定的内点数的  $m$  元树的顶点数是确定的。与定理 2 一样，将用  $n$  来表示树中的顶点数。

**定理 3** 带有  $i$  个内点的正则  $m$  元树含有  $n=mi+1$  个顶点。

**证** 除根之外的每个顶点都是内点的子女。因为每个内点有  $m$  个子女，所以在树中除根之外还有  $mi$  个顶点。因此，该树含有  $n=mi+1$  个顶点。  $\square$

假定  $T$  是完全  $m$  元树。设  $i$  是该树的内点数而  $l$  是树叶数。一旦  $n$ ， $i$  和  $l$  之中的一个已知，另外的两个量就随之确定了。下面的定理给出如何从已知的一个量求其他两个量的方法。

**定理 4** 一个正则  $m$  元树若带有

(i)  $n$  个顶点，则带有  $i=(n-1)/m$  个内点和  $l=[(m-1)n+1]/m$  个树叶；

(ii)  $i$  个内点, 则带有  $n=mi+1$  个顶点和  $l=(m-1)i+1$  个树叶;

(iii)  $l$  个树叶, 则带有  $n=(ml-1)/(m-1)$  个顶点和  $i=(l-1)/(m-1)$  个内点。

**证** 设  $n$  表示顶点数,  $i$  表示内点数,  $l$  表示树叶数。利用定理 3 中的等式, 即  $n=mi+1$ , 以及等式  $n=l+i$  (这个等式为真, 是因为每一个顶点要么是树叶、要么是内点), 就可以证明本定理的所有三个部分。将在这里证明(i)。(ii)和(iii)的证明留给读者作为练习。

在  $n=mi+1$  里求解  $i$  得出  $i=(n-1)/m$ 。然后把  $i$  的这个表达式代入等式  $n=l+i$ , 就证明  $l=n-i=n-(n-1)/m=[(m-1)n+1]/m$ 。  $\square$

例 9 说明如何使用定理 4。

**例 9** 假定某人寄出一封连环信。要求收到信的每个人再把它寄给另外 4 个人。有一些人这样做了, 但是其他人则没有寄出信。若没有人收到超过一封的信, 而且若读过信但是不寄出它的人数超过 100 个后, 连环信就终止了, 则包括第一个人在内, 有多少人看过信? 有多少人寄出过信?

**解** 可以用 4 元树表示连环信。内点对应于寄出信的人, 而树叶对应于不寄出信的人。因为有 100 个人不寄出信, 所以在这个根树里, 树叶数是  $l=100$ 。因此, 定理 4 的(iii)说明, 已经看过信的人数是  $n=(4 \cdot 100-1)/(4-1)=133$ 。另外, 内点数是  $133-100=33$ , 所以 33 个人寄出过信。  $\blacksquare$

经常需要使用这样的根树, 它们是“平衡的”, 所以在每个顶点处的子树都包含大约相同长度的通路。下面的一些定义将解释清楚这个概念。在根树里顶点  $v$  的层是从根到这个顶点的唯一通路的长度。根的层定义为 0。根树的高度就是顶点层数的最大值。换句话说, 根树的层数是从根到任意顶点的最长通路的长度。

**例 10** 求图 7-13 所示的根树里每个顶点的层数。这棵树的高度是什么?

**解** 根  $a$  在 0 层上。顶点  $b, j$  和  $k$  都在 1 层上。顶点  $c, e, f$  和  $l$  都在 2 层上。顶点  $d, g, i, m$  和  $n$  都在 3 层上。最后, 顶点  $h$  在 4 层上。因为任意顶点的最大层数是 4, 所以这棵树的高度为 4。  $\blacksquare$

若一棵高度为  $h$  的  $m$  元树的所有顶点都在  $h$  层或  $h-1$  层, 则这棵树是平衡的。

**例 11** 在图 7-14 所示的一些根树里, 哪些根树是平衡的?

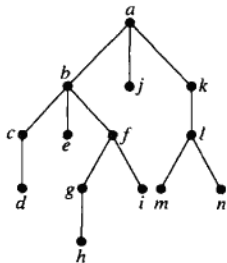


图 7-13 一棵根树

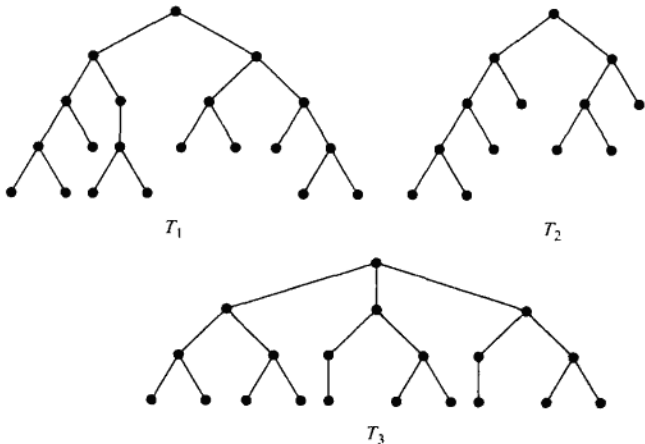


图 7-14 一些根树

**解**  $T_1$  是平衡的, 因为它所有的树叶都在 3 层和 4 层上。不过,  $T_2$  不是平衡的, 因为它有树叶在 2 层、3 层和 4 层上。最后,  $T_3$  是平衡的, 因为它所有的树叶都在 3 层上。  $\blacksquare$

定理 5 中的结果建立  $m$  元树的高度与顶点数之间的联系。

**定理 5** 在高度为  $h$  的  $m$  元树里至多有  $m^h$  个树叶。

**证** 本证明对高度使用数学归纳法。首先, 考虑高度为 1 的  $m$  元树。这些树都是由带有不超过  $m$  个子树的一个根所组成的, 每个子女都是树叶。因此在高度为 1 的  $m$  元树里有不超过

$m^1 = m$  个树叶。这是归纳论证的基础步骤。

现在假定对高度小于  $m$  的所有  $m$  元树来说，这个结果都为真；这是归纳假设。设  $T$  是高度为  $h$  的  $m$  元树。 $T$  的树叶都是通过删除从根到每个在 1 层的顶点的边所获得的  $T$  的子树的树叶，如图 7-15 所示。

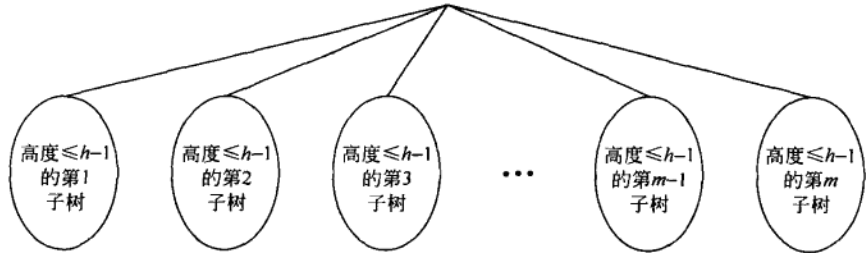


图 7-15 证明的归纳步骤

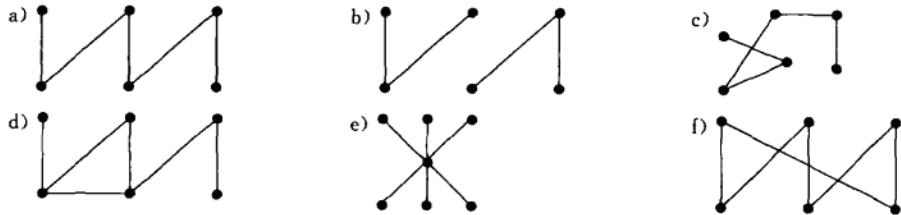
这些子树的高度都小于或等于  $h-1$ 。所以根据归纳假设，这些根树每个都有至多  $m^{h-1}$  条边。因为最多有  $m$  个这样的子树，每个子树最多有  $m^{h-1}$  个树叶，所以在这个根树里最多有  $m \cdot m^{h-1} = m^h$  个树叶。这样就完成了归纳论证。 □

**推论 1** 若一个高度为  $h$  的  $m$  元树带有  $l$  个树叶，则  $h \geq \lceil \log_m l \rceil$ 。若这个  $m$  元树是正则的和平衡的，则  $h = \lceil \log_m l \rceil$  (在这里使用向上取整函数。回忆一下， $\lceil x \rceil$  是大于或等于  $x$  的最小整数)。

**证** 从定理 5 知道  $l \leq m^h$ 。取以  $m$  为底的对数就证明  $h \geq \log_m l$ 。因为  $h$  是整数，所以有  $h \geq \lceil \log_m l \rceil$ 。现在假定这个树是平衡的。于是每个树叶都在  $h$  层或  $h-1$  层上，而且因为树的高度为  $h$ ，所以在  $h$  层至少有一个树叶。所以必然有超过  $m^{h-1}$  个树叶。因为  $l \leq m^h$ ，所以有  $m^{h-1} < l \leq m^h$ 。在这个不等式里取以  $m$  为底的对数就得出  $h-1 < \log_m l \leq h$ 。因此  $h = \lceil \log_m l \rceil$ 。 □

练习

1. 下面哪些图是树？



2. 回答下列关于左下图所示的根树的问题。

- a) 哪个顶点是根？

c) 哪些顶点是树叶？

e) 哪些顶点是  $h$  的父母？

g) 哪些顶点是  $m$  的祖先？
- b) 哪些顶点是内点？

d) 哪些顶点是  $j$  的子女？

f) 哪些顶点是  $o$  的兄弟？

h) 哪些顶点是  $b$  的后代？

3. 练习 2 里的根树对某个正整数  $m$  来说，是否是正则  $m$  元树？

4. 练习 2 里的根树的每个顶点的层数是什么？

5. 画出练习 2 里的树以下列顶点为根的子树。

- a)  $a$
- b)  $c$
- c)  $e$

6. a) 有多少种非同构的带有 3 个顶点的非根树？

b) 有多少种非同构的带有 3 个顶点的根树(使用有向图的同构)？

\*7. a) 有多少种非同构的带有 5 个顶点的非根树？

b) 有多少种非同构的带有 5 个顶点的根树(使用有向图的同构)？

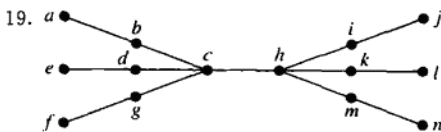
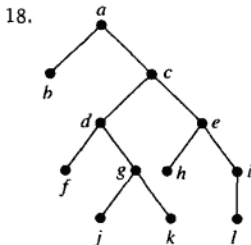
8. 带有 10 000 个顶点的树有多少条边？

9. 带有 1000 个内点的正则二叉树有多少条边?
10. 假定 1000 个人参加象棋巡回赛。若一个选手输掉一盘就遭到淘汰, 而且比赛进行到只有一位参加者还没有输过为止, 则利用这个巡回赛的根树模型, 来确定为了决出冠军必须下多少盘棋(假定没有平局)。
11. 一封连环信。开始时一个人寄出一封信给其他 10 个人。要求每个人寄出此信给其他 10 个人, 而且每封信都包含该连环信前面 6 个人的列表。除非表中不足 6 个名字, 否则每个人都寄一美元给表中第一个人, 从表中删除这个人的名字, 把其他 5 个人的名字向上移动一位, 并且把他或她自己的名字插入到表的末尾。若没有人中断这个连环, 并且每人至多收到一封信, 则这个连环里的一个人最终将收到多少钱?
- \*12. 要么画出带有 84 个树叶而且高度为 3 的正则  $m$  元树, 其中  $m$  是正整数, 要么证明这样的树不存在。

一个完全  $m$  元树是其中每个树叶都在同一层上的正则  $m$  元树。

13. 构造高度为 4 的完全二叉树和高度为 3 的完全 3 元树。
14. 证明: a) 定理 4 的部分(ii)。b) 定理 4 的部分(iii)。
15. 在包含总共  $n$  个顶点的  $t$  个树的森林里有多少条边?
16. 下面的饱和碳氢化合物有多少种不同的同分异构体?  
a)  $C_3H_8$       b)  $C_5H_{12}$       c)  $C_6H_{14}$
17. 设  $n$  是 2 的幂。证明: 可以用  $n-1$  个处理器的树形连接网络在  $\log n$  步里求  $n$  个数之和。

非根树里顶点的离心度是从这个顶点开始的最长的简单通路的长度。若在树里没有其他顶点比一个顶点的离心度更小, 则这个顶点就称为中心。在练习 18~19 里, 求每一个是所给的树的中心的顶点。



- \*20. 证明: 树有一个中心或两个相邻的中心。

根斐波那契树  $T_n$  是以下面的方式来递归地定义的。  $T_1$  和  $T_2$  都是包含单个顶点的根树, 而对  $n=3, 4, \dots$  来说, 都是从一个根以  $T_{n-1}$  作为左子树并且以  $T_{n-2}$  作为右子树来构造出根树  $T_n$ 。

21. 画出前 7 个根斐波那契树。
22. 下面这个使用数学归纳法的“证明”错在什么地方? 命题: 有  $n$  个顶点的每个树都有长度为  $n-1$  的路径。  
基础步骤: 有 1 个顶点的每个树显然有长度为 0 的路径。归纳步骤: 假设有  $n$  个顶点的树有长度为  $n-1$  的路径, 这个路径以  $u$  作为终点。加入顶点  $v$  和从  $u$  到  $v$  的边。所得出的树有  $n+1$  个顶点并且有长度为  $n$  的路径。这样就完成了归纳步骤。

## 7.2 树的应用

### 7.2.1 引言

下面将要运用树来讨论三个问题。第一个问题是: 应当如何对列表里的项进行排序, 以便可以容易地找到项的位置? 第二个问题是: 为了在某种类型的一组对象里找出带某种性质的对象, 应当做出一系列什么样的决策? 第三个问题是: 应当如何用位串来有效地编码一组字符?

### 7.2.2 二叉搜索树

 在列表里搜索一些项, 是计算机科学的一项重要任务。主要目标是实现一个搜索算法, 当项都完全排序时, 这个算法有效地找出项。这个任务可以通过使用二叉搜索树来完成, 二叉搜索树是一种二叉树, 其中任何顶点的每个子节点都指定为右子女或左子女, 没有顶点具有超过一个的右子女或左子女, 而且每个顶点都用一个关键字来标记, 这个关键字是各项中的一个。另外, 这样指定顶点的关键字, 使得顶点的关键字不仅大于它的左子树里的所有顶点的关

字，而且小于它的右子树里的所有顶点的关键字。

这个递归过程用来形成项的列表的二叉搜索树。从只包含一个顶点(即根)的树开始。指定列表中第一个项作为这个根的关键字。为了添加新的项，首先比较它与已经在树里的顶点的关键字，从根开始，若这个项小于所比较顶点的关键字而且这个顶点有左子女，则向左移动，若这个项大于所比较顶点的关键字而且这个顶点有右子女，则向右移动。当这个项小于所比较顶点的关键字而且这个顶点没有左子女时，就插入以这个项作为关键字的一个新顶点，并把新顶点作为这个顶点的左子女。同理，当这个项大于所比较顶点的关键字而且这个顶点没有右子女时，就插入以这个项作为关键字的一个新顶点，并把新顶点作为这个顶点的右子女。用例 1 来说明这个过程。

**例 1** 构造下面这些单词的二叉搜索树(用字母顺序): mathematics, physics, geography, zoology, meteorology, geology, psychology 和 chemistry。

**解** 图 7-16 显示了构造这个二叉搜索树所用的步骤。单词 mathematics 是根的关键字。因为 physics 是在 mathematics 之后(按照字母顺序)，所以给根添加带关键字 physics 的右子女。因为 geography 是在 mathematics 之前，所以给根添加带关键字 geography 的左子女。下一步，给带关键字 physics 的顶点添加右子女，并且给其指定关键字 zoology，因为 zoology 是在 mathematics 之后且在 physics 之后。同理，给带关键字 physics 的顶点添加左子女，并且给其指定关键字 meteorology。给带关键字 geography 的顶点添加右子女，并且给其指定关键字 geology。给带关键字 zoology 的顶点添加左子女，并且给其指定关键字 psychology。给带关键字 geography 的顶点添加左子女，并且给其指定关键字 chemistry(读者应当完成在每步上所需的所有比较)。

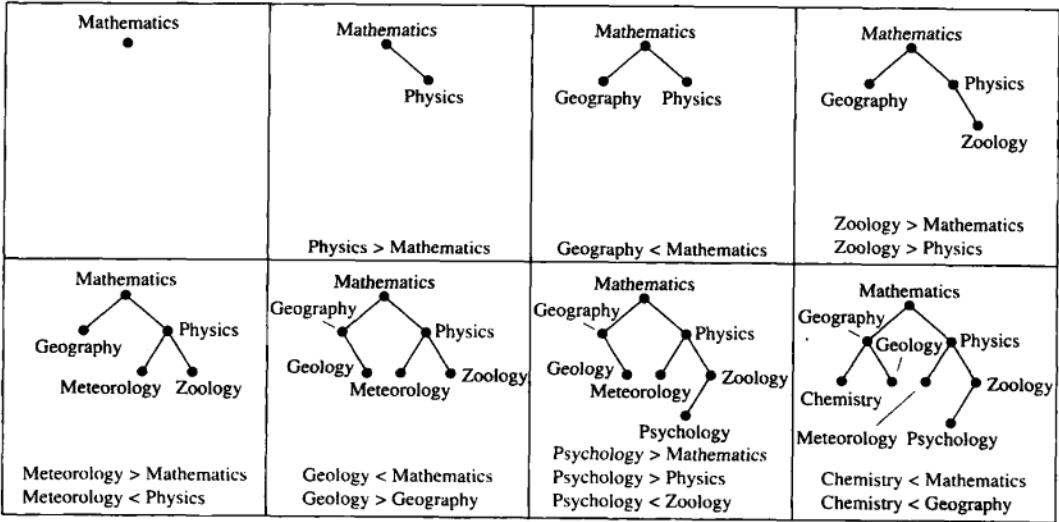


图 7-16 构造二叉搜索树

一旦建立了二叉搜索树，就需要一种在二叉搜索树中定位元素的方法，还需要一种添加新元素的方法。算法 1 是一个插入算法，尽管看上去它只是设计在二叉搜索树添加新节点，但实际上它可以完成上面提到的两个任务。也就是说，算法 1 可以在二叉搜索树定位该元素；如果一个元素  $x$  不存在，也可以添加该元素  $x$ 。在下面的伪代码中， $v$  是当前正在查看的节点， $label(v)$  是该节点的关键字。算法从根开始查看。如果  $v$  的关键字等于  $x$ ，那么算法就找到了  $x$  的位置并结束；如果  $x$  比  $v$  的关键字小，就向  $v$  的左孩子节点移动并重复这个过程；如果  $x$  比  $v$  的关键字大，就向  $v$  的右孩子节点移动并重复这个过程。如果在任何一步，要移动到的孩子节点并不存在，那么就on知道在这棵二叉搜索树中没有  $x$ ，然后就添加一个以  $x$  为关键字的节点作为这个孩子节点。

**算法 1 二叉搜索树算法**

**procedure** insertion( $T$ : 二叉搜索树,  $x$ : 项)

$v := T$  的根

{一个不在  $T$  里出现、具有值  $null$  的顶点}

**while**  $v \neq null$  并且  $label(v) \neq x$

**begin**

**if**  $x < label(v)$  **then**

**if**  $v$  的左子女  $\neq null$  **then**  $v := v$  的左子女

**else** 添加 *new vertex* 作为  $v$  的左子女并且设置  $v := null$

**else**

**if**  $v$  的右子女  $\neq null$  **then**  $v := v$  的右子女

**else** 给  $T$  添加 *new vertex* 作为  $v$  的右子女并且设置  $v := null$

**end**

**if**  $T$  的根  $= null$  **then** 给树添加顶点  $r$  并且用  $x$  标记它

**else if**  $label(v) \neq x$  **then** 用  $x$  标记 *new vertex*

{ $v = x$  的位置}

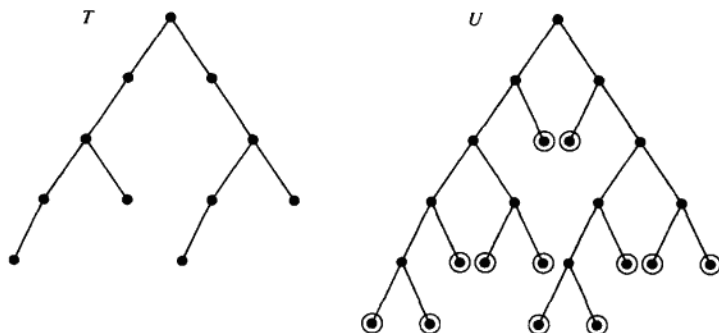
例 2 说明了算法 1 中二叉搜索树的用法。

**例 2** 运用算法 1 在例 1 的二叉搜索树中插入 *oceanography* 这个词。

**解** 算法 1 以  $v$  等于  $T$  的根节点开始,  $v$  是当前查看的节点。因此  $label(v) = mathematics$ 。

因为  $v \neq null$ , 且  $label(v) = mathematics < oceanography$ , 接下来就查看根的右孩子节点。右孩子存在, 因此置当前查看的节点  $v$  等于这个右孩子。这一步, 有  $v \neq null$ , 且  $label(v) = physics > oceanography$ , 所以要查看  $v$  的左孩子。左孩子存在, 因此置当前查看的节点  $v$  等于这个左孩子。在这一步, 有  $v \neq null$ , 且  $label(v) = meteorology < oceanography$ , 所以试图查看  $v$  的右孩子。但是, 这个右孩子并不存在, 所以添加一个新的节点来当  $v$  的右孩子(此时就是关键字为 *oceanography* 的节点), 然后置  $v := null$ 。因为  $v = null$ , 现在跳出了 **while** 循环。因为  $T$  的根不是  $null$  而  $v = null$ , 所以使用算法结束处的 **else if** 语句让新节点以 *oceanography* 为关键字。

现在我们来确定这个过程的计算复杂性。假定有  $n$  个项的列表的二叉搜索树  $T$ 。可以从  $T$  这样构造一个满二叉树  $U$ : 在必要时添加无标记的顶点, 以使得每个带关键字的顶点都有两个子女。这个做法在图 7-17 里说明。一旦这样做了, 就容易找出新项的位置, 或者添加新项作为关键字而不添加顶点。



未标记的顶点加了圆圈

图 7-17 添加无标记顶点使二叉搜索树满

添加一个新项所需要的比较次数,最多等于在 $U$ 里从根到树叶的最长通路的长度。 $U$ 的内点都是 $T$ 的顶点。所以 $U$ 有 $n$ 个内点。现在可以利用7.1节定理4的部分(ii)来得出 $U$ 有 $n+1$ 个树叶。利用7.1节的推论1,可以看出 $U$ 的高度大于或等于 $h = \lceil \log(n+1) \rceil$ 。所以,必须至少执行 $\lceil \log(n+1) \rceil$ 次比较以添加某个项。注意若 $U$ 是平衡的,则它的高度是 $\lceil \log(n+1) \rceil$ (根据7.1节的推论1)。因此,若二叉搜索树是平衡的,则确定一个项的位置或者添加一个项所需要的比较次数不超过 $\lceil \log(n+1) \rceil$ 次。当给二叉搜索树添加一些项时,该树可能变得不平衡。因为平衡的二叉搜索树给出二叉搜索的最优的最坏情形复杂性,所以添加项时重新平衡二叉搜索树的算法已经设计出来。感兴趣的读者可以查阅关于数据结构的参考文献来了解这些算法。

### 7.2.3 决策树

根树可以用来为这样的问题建立模型,其中一系列决策将生成一个解。例如,二叉搜索树可以用来基于一系列比较来找出项的位置,其中每次比较都说明是否已经找到了项的位置,或者是否应当向右或向左进入子树。其中每个内点都对应着一次决策,这些顶点的子树都对应着该决策的每种可能后果,这样的根树称为决策树。问题的可能的解对应着这个根树的通向树叶的通路。下一个例子说明决策树的应用。

**例3** 假定有重量相同的7枚硬币和重量较轻的一枚伪币。为了用一架天平确定这8枚硬币中哪个是伪币,需要多少次称重?给出找出这个伪币的算法。

**解** 在天平上每次称重结果有三种可能性。分别是:两个托盘有相同的重量,第一个托盘较重,或第二个托盘较重。所以,称重序列的决策树是3元树。在决策树里至少有8个树叶,这是因为有8种可能的后果(因为每枚硬币都可能是较轻的伪币),而每种可能的后果必须至少用一个树叶来表示。确定伪币所需要的最大称重次数是决策树的高度。从7.1节的推论1得出决策树的高度至少是 $\lceil \log_3 8 \rceil = 2$ 。因此,至少需要两次称重。

用两次称重来确定伪币,是可行的。说明如何这样做的决策树如图7-18所示。

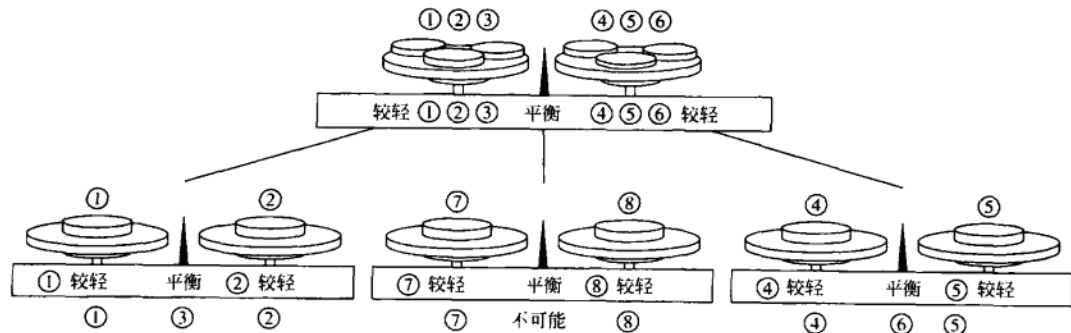


图 7-18 找出伪币位置的决策树。最下面一行为伪币判断结果

**排序算法的复杂性** 已经开发了许多不同的排序算法。为了确定一个具体的排序算法是否有效,就要确定这个算法的复杂性。用决策树作为模型,可以求出排序算法的最坏情形复杂性的下界。

可以用决策树为排序算法建立模型并且确定对这些算法的最坏情形复杂性的估计。注意给定 $n$ 个元素,这些元素有 $n!$ 种可能的排序,因为这些元素 $n!$ 种排列中的每一个都可以是正确的顺序。本书研究的排序算法以及最常用的排序算法都是基于二叉比较,即一次比较两个元素。每次这样的比较都缩小了可能顺序的集合。因此,基于二叉比较的排序算法可以表示成二叉决策树,其中每个内点表示两个元素的一次比较。每个树叶表示 $n$ 个元素的 $n!$ 种排列中的一种。

**例4** 在图7-19里显示排序列表 $a, b, c$ 里元素的决策树。

基于二叉比较的排序的复杂性是用二叉比较的次数来度量的。排序有 $n$ 个元素的列表所需要的最多比较次数就给出了这个算法的最坏情形复杂性。所用的最多比较次数等于表示这个排序过



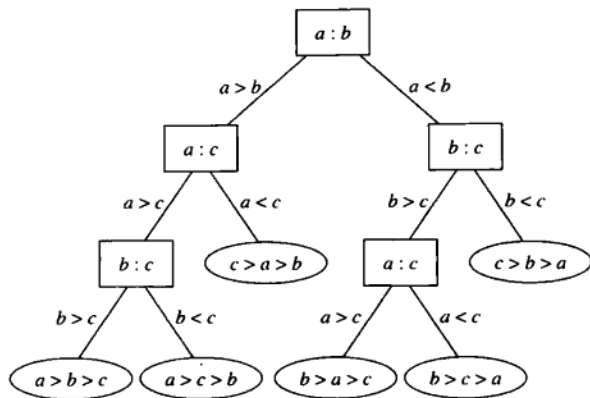


图 7-19 排序三个不同元素的决策树

程的决策树里的最长通路长度。换句话说，所需要的最多比较次数等于这个决策树的高度。因为带  $n!$  个树叶的二叉树的高度至少是  $\lceil \log n! \rceil$  (利用 7.1 节推论 1)，所以定理 1 说至少需要  $\lceil \log n! \rceil$  次比较。

**定理 1** 基于二叉比较的排序算法至少需要  $\lceil \log n! \rceil$  次比较。

可以用定理 1 来给出基于二叉比较的排序算法所用比较次数的大  $O$  估计。我们知道  $\lceil \log n! \rceil$  是  $\Theta(n \log n)$ ，这是算法的计算复杂性经常使用的一个参照函数。推论 1 是这个估计的结果。

**推论 1** 基于二叉比较的排序算法排序  $n$  个元素所用的比较次数是  $\Omega(n \log n)$ 。

推论 1 的一个后果是，基于二叉比较的排序算法在最坏情形下使用  $\Theta(n \log n)$  次比较来排序  $n$  个元素，其他这类算法都没有更好的最坏情形复杂性，在这个意义下，基于二叉比较的排序算法是最优的。注意，在这个意义下归并排序算法是最优的。

对于排序算法的平均情形复杂性也可以证明类似的结果。基于二叉比较的排序算法所用的平均比较次数是表示这个排序算法的决策树中的平均树叶深度。我们知道，有  $N$  个顶点的二叉树的平均树叶深度是  $\Omega(\log N)$ 。当令  $N = n!$  并且注意到因为  $\log n! = \Theta(n \log n)$ ，所以是  $\Omega(\log n!)$  的函数也是  $\Omega(n \log n)$  时，就会得出下面的估计。

**定理 2** 基于二叉比较的排序算法排序  $n$  个元素所用的平均比较次数是  $\Omega(n \log n)$ 。

#### 7.2.4 前缀码

考虑一下这样的问题：用位串来编码英语字母表里的字母(其中不区分小写和大写字母)。可以用长度为 5 的位串来表示每个字母，因为只有 26 个字母而且有 32 个长度为 5 的位串。当每个字母都用 5 位来编码时，用来编码数据的总位数是 5 乘以文本中的字符数。有没有可能找出这些字母的编码方案，使得在编码数据时使用的位更少？若可能，那么就可以节省存储空间而且缩短传输时间。

考虑用不同长度的位串来编码字母。较短的位串用来编码出现较频繁的字母，较长的位串应当用来编码不经常出现的字母。当用变化的位数来给字母编码时，就必须用某种方法来确定每个字母的位在何处开始和结束。例如，若把  $e$  编码成 0，把  $a$  编码成 1，而把  $t$  编码成 01，则位串 0101 可能对应着  $eat$ 、 $tea$ 、 $eaea$  或  $tt$ 。

为了保证没有位串对应着多个字母的序列，可以令一个字母的位串永远不出现在另外一个字母的位串的开头部分。具有这个性质的编码称为前缀码。例如，把  $e$  编码成 0、把  $a$  编码成 10、而把  $t$  编码成 11 的编码就是前缀码。从编码着一个单词的字母的唯一位串可以恢复这个单词。例如，串 10110 是  $ate$  的编码。为了看明白这一点，注意开始的 1 不表示一个字符，但是 10 表示  $a$ (并且它不可能是另外一个字母的位串的开始部分)。然后，下一个 1 不表示一个字符，但是 11 表示  $t$ 。

最后一位 0 表示  $e$ 。

前缀码可以用二叉树来表示, 其中字符是树里树叶的标记。标记树的边, 使得通向左子女的边标记为 0 而通向右子女的边标记为 1。用来编码一个字符的位串是在从根到以这个字符作为标记的树叶的唯一通路上边的标记的序列。例如, 图 7-20 里的树表示把  $e$  编码成 0, 把  $a$  编码成 10, 把  $t$  编码成 110, 把  $n$  编码成 1110 和把  $s$  编码成 1111。

表示编码的树可以用来解码位串。例如, 考虑一个用图 7-20 里的编码编成 11111011100 的单词。这个位串可以这样解码: 从根开始, 用位的序列来形成一条到树叶为止的通路。每个 0 位都使得通路向下到达通向通路里上一个顶点的左子女的边, 而每个 1 位都对应到上一个顶点的右子女。所以, 开头的 1111 对应这样的通路: 从根开始, 向右前进四次, 到达以  $s$  作为标记的树叶, 因为串 1111 是  $s$  的编码。从第五个位继续进行, 在向右再向左之后, 就到达下一个树叶, 这时访问以  $a$  作为标记的顶点, 其中把  $a$  编码成 10。从第七个位开始, 在向右三次然后向左之后, 到达了下一个树叶, 这时访问用  $n$  标记的顶点, 其中把  $n$  编码成 1110。最后, 末位 0 通向用  $e$  标记的树叶。因此, 原来的单词是 *sane*。

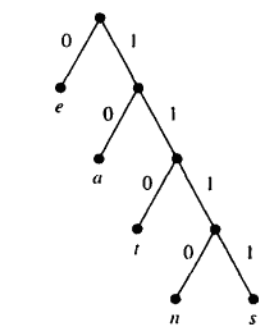


图 7-20 表示前缀码的二叉树

可以从任何二叉树来构造一个前缀码, 其中每个内点的左边都用 0 标记, 而右边都用 1 标记, 树叶都用字符标记。字符都用从根到这个树叶的唯一通路里的边的标记所组成的位串来编码。

**赫夫曼编码** 现在介绍一种算法, 这种算法用一个字符串中符号的出现频率(即出现概率)作为输入, 并且产生编码这个字符串的一个前缀码作为输出, 在这些符号的所有可能的二叉前缀码中, 这个编码使用最少的位。这个所谓赫夫曼编码的算法是大卫·赫夫曼 1951 年做麻省理工学院的研究生时发表在一篇学期论文中的。(注意, 这个算法假定已知字符串中每个符号出现多少次, 所以可以计算每个符号的出现频率, 方法是用这个符号出现的次数除以这个字符串的长度。)赫夫曼编码是数据压缩中的基本算法, 数据压缩的目的在于减少表示信息所需要的位数。赫夫曼编码广泛用于压缩表示文本的位串, 并且在压缩视频和图像文件方面也起到重要作用。

**算法 2 表示赫夫曼编码算法。** 给定符号及其频率, 目标是构造一个根二叉树, 其中符号是树叶的标记。算法从一些树的一个森林开始, 每个树含有一个顶点, 每个顶点有一个符号作为标记, 并且这个顶点的权就等于所标记符号的频率。在每一步, 都把具有最小总和权的两个树组合成一个单独的树, 方法是引入一个新的根, 把具有较大的权的树作为左子树, 把具有较小的权的树作为右子树。另外, 把这个树的两个子树的权之和作为这个树的总的权。(虽然可以规定在具有相同的权的树之间进行选择以打破平局的过程, 但是这里将不具体指定这样的过程。)当构造出了一个树, 即森林缩小为单个树时, 算法就停止。

#### 算法 2 赫夫曼编码

**procedure** *Huffman*( $C$ : 具有频率  $w_i$  的符号  $a_i$ ,  $i=1, 2, \dots, n$ )

$F := n$  个根树的森林, 每个根树由单个顶点  $a_i$  组成并且赋权  $w_i$

**while**  $F$  不是树

**begin**

把  $F$  中满足  $w(T) \geq w(T')$  的权最小的根树  $T$  和  $T'$  换成具有新树根的一个树,

这个树根以  $T$  作为左子树并且以  $T'$  作为右子树。

用 0 标记树根到  $T$  的新边, 并且用 1 标记树根到  $T'$  的新边。

把  $w(T) + w(T')$  作为新树的权。

**end**

{符号  $a_i$  的赫夫曼编码是从树根到  $a_i$  的唯一路径上的边的标记的连接}

例5 说明如何用算法2来编码一组5个符号。

**例5** 用赫夫曼编码来编码下列符号，这些符号具有下列频率：A: 0.08, B: 0.10, C: 0.12, D: 0.15, E: 0.20, F: 0.35。编码一个字符串所需要的平均位数是多少？

**解** 图7-21表示了编码这些符号所用的步骤。所产生的编码为：A是111, B是110, C是011, D是010, E是10, F是00。使用这种编码来编码一个符号所用的平均位数是

$$3 \cdot 0.08 + 3 \cdot 0.10 + 3 \cdot 0.12 + 3 \cdot 0.15 + 2 \cdot 0.20 + 2 \cdot 0.35 = 2.45$$

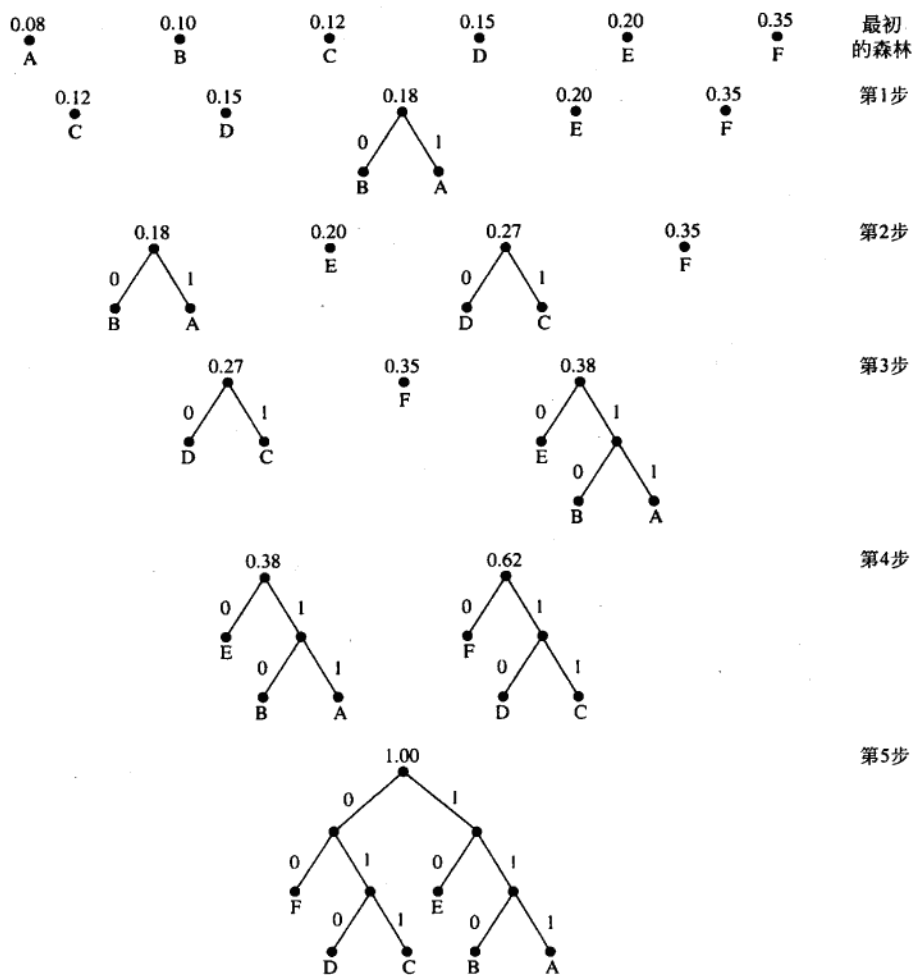


图7-21 例5中符号的赫夫曼编码

注意赫夫曼编码是贪心算法。在每一步替换具有最小权的两棵树，在没有任何二叉前缀码能使用更少的位来编码这些符号的情况下，这样做就导出了最优编码。

赫夫曼编码有许多变种。例如，不编码单个符号，可以编码指定长度的符号块，比如两个符号的块。这样做有可能减少编码这个字符串所需要的位数。也可以用两个以上的符号来编码这个字符串中的原始符号(参看本节末练习15前面的说明)。另外，当事先不知道一个字符串中每个符号的频率时，可以使用一种变种，即所谓的自适应赫夫曼编码(参见[Sa00])，使得在读这个字符串的同时来进行编码。

## 7.2.5 博弈树

可以用树来分析某些类型的游戏，比如圈叉游戏、轮流取石头游戏、跳棋和象棋。在每一种游戏当中，两个选手轮流进行移动。每个选手知道另一个选手的移动并且游戏里不存在偶然因素。使用博弈树为这样的游戏建立模型，这些树的顶点表示当游戏进行时游戏所处的局面，边表示在这些局面之间合乎规则的移动。由于博弈树常常很大，所以通过用同一个顶点表示所有对称的局面来简化博弈树。但是，如果一个游戏的不同移动序列导致同一个局面，则可以用不同的顶点来表示这个局面。通常的约定是用方框表示偶数层的顶点并且用圆圈表示奇数层的顶点。当游戏处在偶数层顶点所表示的局面时，就轮到第一个选手移动。当游戏处在奇数层顶点所表示的局面时，就轮到第二个选手移动。博弈树所表示的游戏可以永远不结束，比如进入了无穷循环，因而博弈树可以是无穷的，但是对于大多数游戏来说，都存在一些规则导致有穷的博弈树。

博弈树的树叶表示游戏的终局。给每个树叶指定一个值，来表示当游戏在这个树叶所代表的局面里终止时第一个选手的得分。对于非胜即负的游戏，用1来标记圆圈所表示的终结顶点以表示第一个选手获胜，用-1来标记方框所表示的终结顶点以表示第二个选手获胜。对于允许平局的游戏，用0来标记平局所对应的终结顶点。注意，对于非胜即负的游戏，为终结顶点指定值，这个值越高，第一个选手的结局就越好。

例6展示一个非常著名的和经过深入研究的游戏的博弈树。

**例6 轮流取石头游戏** 轮流取石头游戏是这样的，在游戏的开始，有几堆石头，两个选手轮流进行移动，一个合法的移动包括从其中一堆取走一块或多块石头，而不去移动其余的所有石头。不能进行合法移动的选手告负。（也可以规定取走最后一块石头的选手告负，因为不允许没有石头堆的局面。）图7-22所示的博弈树表示这种形式的给定开局的轮流取石头游戏，其中有3堆石头，分别包含2块、2块和1块石头。用不同堆中石头数的无序表来表示每个局面（堆的顺序无关紧要）。第一个选手的初始移动可以导致3种可能的局面，因为这个选手可以从有2块石头的堆中取走1块石头（留下包含1块、1块和2块石头的3堆），可以从包含2块石头的堆中取走2块石头（留下包含2块和1块石头的2堆），或者从包含1块石头的堆中取走1块石头（留下包含2块石头的2堆）。当只剩下包含1块石头的1堆时，就不可能进行合法移动了，所以这样的局面就是终局。由于轮流取石头游戏是非胜即负的游戏，所以用+1标记表示第一个选手获胜的终结顶点，用-1标记表示第二个选手获胜的终结顶点。

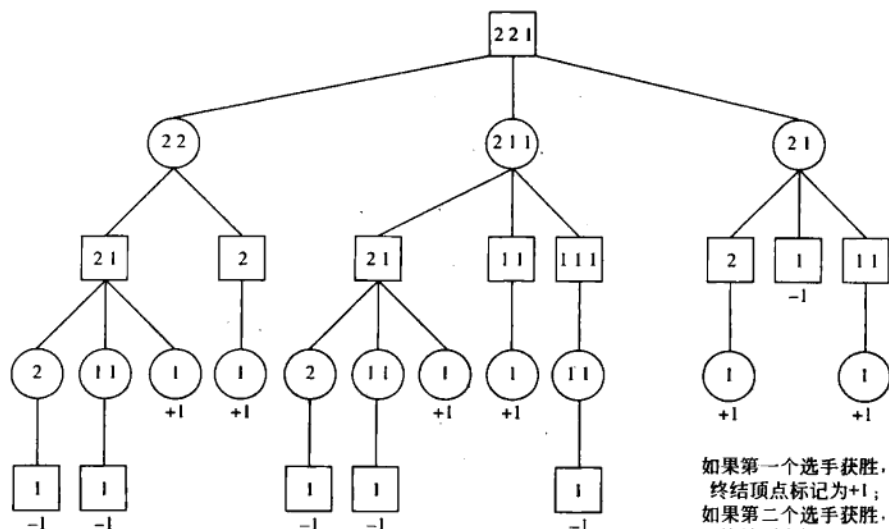


图 7-22 轮流取石头游戏的博弈树

**例7 圈叉游戏** 圈叉游戏的博弈树非常大，在这里不能画出，尽管计算机能轻而易举地构造出这样的树。图7-23a显示圈叉游戏的博弈树的一部分。因为对称的顶点是等价的，所以只需要考虑图7-23a所示的3种可能的初始移动。还显示这个博弈树的导致图7-23b中终局的一个子树，其中一个能够获胜的选手进行了制胜的移动。

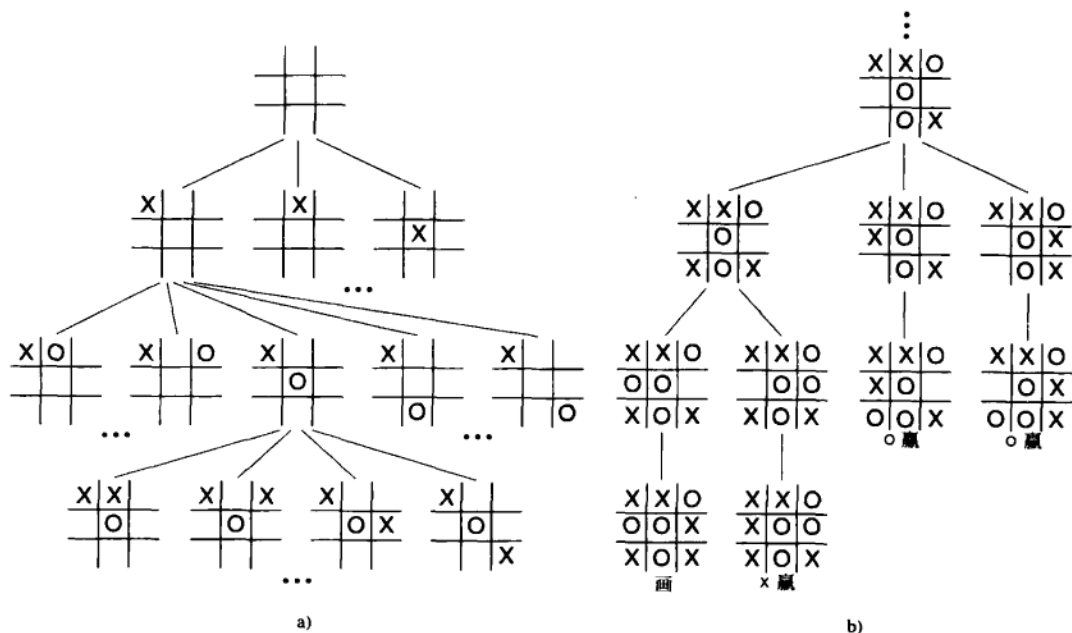


图 7-23 圈叉游戏的部分博弈树

可以用某种方式递归地定义博弈树中所有顶点的值，使得可以确定当两个选手都遵循最优策略时这个游戏的结果。所谓策略，就意味着一组规则，这些规则说明一个选手如何移动来赢得游戏。第一个选手的最优策略就是把这个选手的得分最大化的策略，第二个选手的最优策略就是把这个得分最小化的策略。现在递归地定义顶点的值。

**定义1** 博弈树中顶点的值递归地定义为：

- (i) 一个树叶的值是当游戏在这个树叶所表示的局面里终止时第一个选手的得分。
- (ii) 偶数层内点的值是这个内点的子女的最大值，奇数层内点的值是这个内点的子女的最小值。

第一个选手移动到具有最大值的子女所表示的局面并且第二个选手移动到具有最小值的子女所表示的局面，这样的策略称为最小最大策略。当两个选手都遵循最小最大策略时，通过计算树根的值就可以确定谁将赢得游戏，这个值称为树的值。这是定理3的后果。

**定理3** 如果两个选手都遵循最小最大策略并且从博弈树的某一个顶点所表示的局面开始进行游戏，则这个顶点的值说明第一个选手的得分。

**证** 将用归纳法来证明这个定理。

**基础步骤：**如果这个顶点是树叶，则根据定义指定给这个顶点的值就是第一个选手的得分。

**归纳步骤：**归纳假设一个顶点的子女的值就是第一个选手的得分，假定从这些顶点所表示的每一个局面中开始进行游戏。需要考虑两种情形，即当轮到第一个选手时和当轮到第二个选手时。

当轮到第一个选手时，这个选手遵循最小最大策略并且移动到具有最大值的子女所表示的局面。根据归纳假设，当从这个子女所表示的局面开始游戏并且遵循最小最大策略时，这个值就是

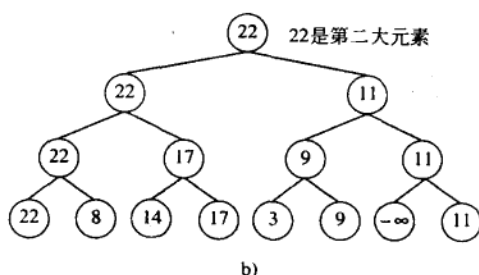
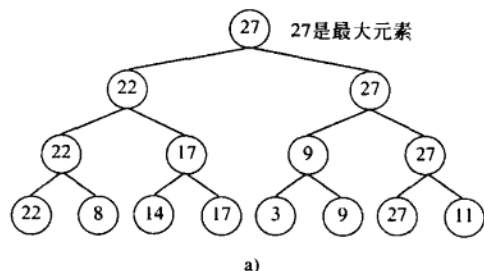


来下棋的计算机程序，比如著名的深蓝程序，都是基于复杂的求值函数的。关于计算机如何下棋的更多信息请参见[Le91]。

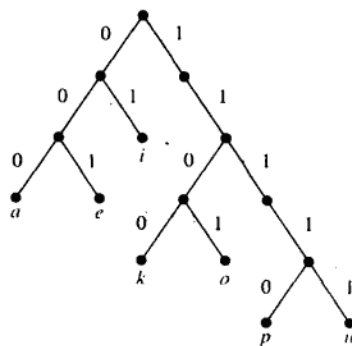
### 练习

- 用字母顺序建立下面这些单词的二叉搜索树：*banana*, *peach*, *apple*, *pear*, *coconut*, *mango* 和 *papaya*。
- 为了在练习 1 的搜索树里找出下面每个单词的位置或者添加它们，而且每次都重新开始，分别需要多少次比较？  
a) *pear*                      b) *banana*                      c) *kumquat*                      d) *orange*
- 用字母顺序构造下面句子里的单词的二叉搜索树：“*The quick brown fox jumps over the lazy dog*”。
- 若一枚伪币与其他硬币质量不等，那么为了在 4 枚硬币中找出这枚伪币，需要用天平称多少次？描述用同样的称重次数来找出这枚伪币的算法。
- \*5. 若一枚伪币比其他硬币较轻，那么为了在 12 枚硬币中找出这枚伪币，需要用天平称多少次？描述用同样的称重次数来找出这枚伪币的算法。
- \*6. 求排序 4 个元素所需要的最少比较次数并且设计一个能够依此次数实现的算法。

竞赛图排序是通过构造有序二叉树来工作的排序算法。用将成为树叶的顶点来表示待排序的元素。就如同构造表示循环赛比赛胜者的树那样，一次构造这个树的一层。从左向右，比较成对的相邻元素，加入用所比较的两个元素中较大的那个来标记的一个父母顶点。在每一层在顶点的标记之间进行类似的比较，直到到达了用最大元素标记的树根为止。22, 8, 14, 17, 3, 9, 27, 11 的竞赛图排序所构造的树如下图(a)所示。一旦确定了最大元素，具有这个标记的树叶就重新标记为  $-\infty$ ，定义为比每个元素都小。从这个顶点直到树根的路径上所有顶点的标记都重新计算，如下图(b)所示。这样就产生了第二大元素。这个过程继续进行下去，直到整个表都已经排序为止。



- 完成表 22, 8, 14, 17, 3, 9, 27, 11 的竞赛图排序。说明在每个步骤上顶点的标记。
- 用伪码描述竞赛图排序。
- 竞赛图排序求第二大元素、第三大元素……，直到第  $(n-1)$  大(或第二小)元素所使用的比较次数是多少？
- 下面哪些编码是前缀码？  
a)  $a: 11, e: 00, t: 10, s: 01$   
b)  $a: 0, e: 1, t: 01, s: 001$   
c)  $a: 101, e: 11, t: 001, s: 011, n: 010$   
d)  $a: 010, e: 11, t: 011, s: 1011, n: 1001, i: 10101$
- 若编码方案是用右面的树来表示，那么什么是  $a$ 、 $e$ 、 $i$ 、 $k$ 、 $o$ 、 $p$  和  $u$  的编码？
- 用赫夫曼编码来编码具有给定频率的如下符号： $a: 0.20, b: 0.10, c: 0.15, d: 0.25, e: 0.30$ 。编码一个符号所需要的平均位数是多少？
- 为如下符号和频率构造两个不同的赫夫曼编码： $t: 0.2, u: 0.3, v: 0.2, w: 0.3$ 。
- 为英文字母表的字母构造赫夫曼编码，其中典型英文文本中字母的频率如下表所示。



| 字母 | 频率     | 字母 | 频率     |
|----|--------|----|--------|
| A  | 0.0817 | N  | 0.0662 |
| B  | 0.0145 | O  | 0.0781 |
| C  | 0.0248 | P  | 0.0156 |
| D  | 0.0431 | Q  | 0.0009 |
| E  | 0.1232 | R  | 0.0572 |
| F  | 0.0209 | S  | 0.0628 |
| G  | 0.0182 | T  | 0.0905 |
| H  | 0.0668 | U  | 0.0304 |
| I  | 0.0689 | V  | 0.0102 |
| J  | 0.0010 | W  | 0.0264 |
| K  | 0.0080 | X  | 0.0015 |
| L  | 0.0397 | Y  | 0.0211 |
| M  | 0.0277 | Z  | 0.0005 |

假设  $m$  是正整数且  $m \geq 2$ 。对于  $N$  个符号的集合来说，类似于二元赫夫曼编码的构造，可以构造  $m$  元赫夫曼编码。在初始步骤，把由  $((N-1) \bmod (m-1)) + 1$  个权最小的单个顶点所组成的树组合成以这些顶点作为树叶的一棵树。在每个后续步骤，把权最小的  $m$  棵树组合成一棵  $m$  元树。

15. 使用符号 0、1 和 2，用三元 ( $m=3$ ) 赫夫曼编码来编码具有给定频率的这些字母：A: 0.25, E: 0.30, N: 0.10, R: 0.05, T: 0.12, Z: 0.18。
16. 给定  $n+1$  个符号  $x_1, x_2, \dots, x_n, x_{n+1}$ ，它们在一个字符串中分别出现  $1, f_1, f_2, \dots, f_n$  次，其中  $f_j$  是第  $j$  个斐波那契数。当在赫夫曼编码算法的每个阶段考虑所有可能的打破平局的选择时，用来编码一个符号的最大位数是多少？
17. 画出轮流取石头游戏的博弈树，假设开局包括分别有 2 块和 3 块石头的两堆石头。在画这棵树的时候，用同一个顶点表示相同移动所导致的对称局面。求出这个博弈树每个顶点的值。如果两个选手都遵循最优策略，则哪个选手获胜？
18. 假设在轮流取石头游戏中修改获胜选手的得分，使得当  $n$  是到达终局前所做合法移动的步数时得分就是  $n$  美元。求第一个选手的得分，假设开局包括：
- a) 分别有 1 块和 3 块石头的两堆石头。
  - b) 分别有 2 块和 4 块石头的两堆石头。
  - c) 分别有 1 块、2 块和 3 块石头的三堆石头。
19. 画出圈叉游戏博弈树从下列每个局面开始的子树。确定每个子树的值。

a) 

|   |   |   |
|---|---|---|
| O | X | X |
| X | O | O |
|   |   | X |

b) 

|   |   |   |
|---|---|---|
| X | O | X |
| O | X | X |
|   |   | O |

c) 

|   |   |   |
|---|---|---|
| X |   | O |
| O | O |   |
| X |   | X |

d) 

|   |   |   |
|---|---|---|
|   | O | X |
|   | X | O |
| X | O |   |

20. 证明：如果轮流取石头游戏从包含相同数目的 2 堆石头开始，只要这个数目至少是 2，则当两个选手都遵循最优策略时第二个选手获胜。
21. 跳棋博弈树的根有多少个子女？有多少个孙子孙女？
22. 画出圈叉游戏博弈树前两步移动所对应的层。指明正文中所提到的求值函数的值，这个函数给局面指定不含 O 的直行数减去不含 X 的直行数来作为这一层每个顶点的值，并且在求值函数给出这些顶点的正确值的假设下，对这些顶点计算树的值。

7.3 树的遍历

7.3.1 引言

有序根树常常用来保存信息。掌握一些访问有序根树的每个顶点以存取数据的算法是非常必要的。几个重要的算法将在下面描述。有序根树也可以用来表示各种类型的表达式，比如由数字、变量和运算所组成的算术表达式。对用来表示这些表达式的有序根树来说，



它的顶点的一些不同的列表在这些表达式的求值中很有用。

### 7.3.2 通用地址系统

遍历有序根树的所有顶点的过程,都依赖于子女的顺序。在有序根树里,一个内点的子女从左向右地显示在表示这些有向图的图形里。

下面将描述一种完全地排序有序根树顶点的方法。为了产生这个顺序,必须首先标记所有的顶点。如下递归地完成这件事。

1. 用整数0标记根。然后用1, 2, 3, ...,  $k$ 从左向右标记它的 $k$ 个子女(在1层上)。
2. 对在 $n$ 层上带标记 $A$ 的每个顶点 $v$ ,按照从左向右画出它的 $k_v$ 个子女的顺序,用 $A.1$ ,  $A.2$ , ...,  $A.k$ 标记它的 $k_v$ 个子女。

遵循这个过程,对 $n \geq 1$ 来说,在 $n$ 层上的顶点 $v$ 标记成 $x_1.x_2 \cdots x_n$ ,其中从根到 $v$ 的唯一通路经过1层的第 $x_1$ 个顶点,以及2层的第 $x_2$ 个顶点,依此类推。这样的标记称为根树的通用地址系统。

可以利用顶点在通用地址系统里的标记的字典顺序将这些顶点完全排序。若存在 $i(0 \leq i \leq n)$ 满足 $x_1 = y_1, x_2 = y_2, \dots, x_{i-1} = y_{i-1}$ ,并且 $x_i < y_i$ ;或者若 $n < m$ 并且对 $i = 1, 2, \dots, n$ 来说 $x_i = y_i$ ,那么标记着 $x_1.x_2 \cdots x_n$ 的顶点就小于标记着 $y_1.y_2 \cdots y_m$ 的顶点。

**例1** 在如图7-25所示的有序根树的顶点的旁边,显示出了通用地址系统的标记。这些标记的字典顺序是

$0 < 1 < 1.1 < 1.2 < 1.3 < 2 < 3 < 3.1 < 3.1.1 < 3.1.2 < 3.1.2.1 < 3.1.2.2$   
 $< 3.1.2.3 < 3.1.2.4 < 3.1.3 < 3.2 < 4 < 4.1 < 5 < 5.1 < 5.1.1 < 5.2 < 5.3$

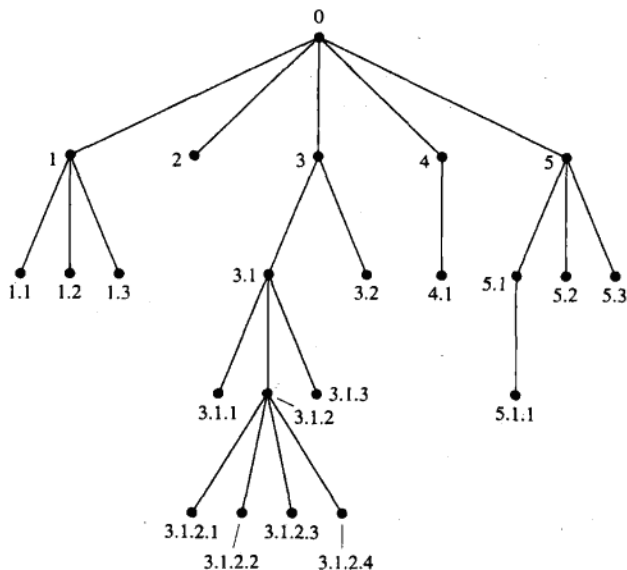


图 7-25 有序根树的通用地址系统

### 7.3.3 遍历算法

系统地访问有序根树每个顶点的过程都称为遍历算法。下面描述三个最常用的此类算法:前序遍历、中序遍历和后序遍历。这些算法都可以递归地定义。首先定义前序遍历。

**定义1** 设 $T$ 是带根 $r$ 的有序根树。若 $T$ 只包含 $r$ ,则 $r$ 是 $T$ 的前序遍历。否则,假定 $T_1, T_2, \dots, T_k$ 是 $T$ 里在 $r$ 处从左向右的子树。前序遍历首先访问 $r$ 。它接着以前序来遍历 $T_1$ ,然后

以前序来遍历  $T_2$ ，依此类推，直到以前序来遍历了  $T_n$  为止。

读者应当验证，有序根树的前序遍历给出与利用通用地址系统所得出的顺序相同的顶点顺序。

图 7-26 说明如何执行前序遍历。

例 2 说明前序遍历。

例 2 前序遍历以什么顺序访问图 7-27 所示的有序根树里的顶点？

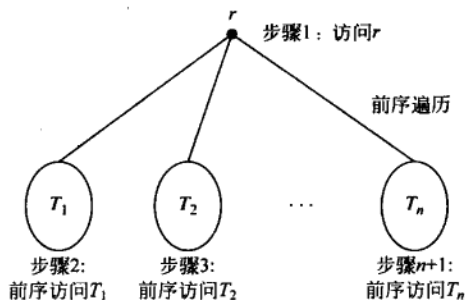


图 7-26 前序遍历

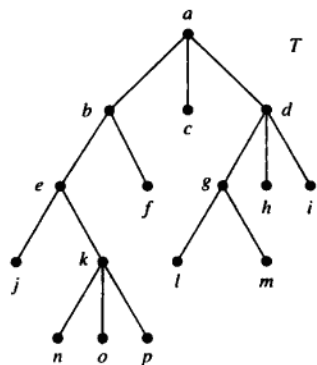


图 7-27 有序根树  $T$

**解**  $T$  的前序遍历的步骤如图 7-28 所示。这样以前序来遍历  $T$ ，首先列出根  $a$ ，接着依次是带根  $b$  的子树的前序列表，带根  $c$  的子树(它只有  $c$ )的前序列表和带根  $d$  的子树的前序列表。

带根  $b$  的子树的前序列表首先列出  $b$ ，然后以前序列出带根  $e$  的子树的顶点，然后以前序列出带根  $f$  的子树(它只有  $f$ )的顶点。带根  $d$  的子树的前序列表首先列出  $d$ ，接着是带根  $g$  的子树的前序列表，接着是带根  $h$  的子树(它只有  $h$ )，接着是带根  $i$  的子树(它只有  $i$ )。

带根  $e$  的子树的前序列表首先列出  $e$ ，接着是带根  $j$  的子树(它只有  $j$ )的前序列表，接着是带根  $k$  的子树的前序列表。带根  $g$  的子树的前序列表是  $g$  接着  $l$ ，接着是  $m$ 。带根  $k$  的子树的前序列表是  $k, n, o, p$ 。所以， $T$  的前序遍历是  $a, b, e, j, k, n, o, p, f, c, d, g, l, m, h, i$ 。

现在将定义中序遍历。

**定义 2** 设  $T$  是带根  $r$  的有序根树。若  $T$  只包含  $r$ ，则  $r$  是  $T$  的中序遍历。否则，假定  $T_1, T_2, \dots, T_n$  是  $T$  里在  $r$  处从左向右的子树。中序遍历首先以前序来遍历  $T_1$ ，然后访问  $r$ 。它接着以中序来遍历  $T_2$ ，中序遍历  $T_3$ ，依此类推，直到以中序来遍历了  $T_n$  为止。

图 7-29 说明如何执行中序遍历。例 3 说明对一棵特定的树，如何执行中序遍历。

例 3 中序遍历以什么顺序访问图 7-27 所示的有序根树  $T$  里的顶点？

**解**  $T$  的中序遍历的步骤显示在图 7-30 里。中序遍历首先是带根  $b$  的子树的中序遍历，然后是根  $a$ ，带根  $c$  的子树(它只有  $c$ )的中序列表和带根  $d$  的子树的中序列表。

带根  $b$  的子树的中序列表，首先是带根  $e$  的子树的中序列表，然后是根  $b$ ，以及  $f$ 。带根  $d$  的子树的中序列表，首先是带根  $g$  的子树的中序列表，接着是根  $d$ ，接着是  $h$ ，接着是  $i$ 。

带根  $e$  的子树的中序列表是  $j$ ，接着是根  $e$ ，接着是带根  $k$  的子树的中序列表。带根  $g$  的子树的中序列表是  $l, g, m$ 。带根  $k$  的子树的中序列表是  $n, k, o, p$ 。所以，这个根树的中序遍历是  $j, e, n, k, o, p, b, f, a, c, l, g, m, d, h, i$ 。

现在定义后序遍历。

**定义 3** 设  $T$  是带根  $r$  的有序根树。若  $T$  只包含  $r$ ，则  $r$  是  $T$  的后序遍历。否则，假定  $T_1, T_2, \dots, T_n$  是  $T$  里在  $r$  处从左向右的子树。后序遍历首先以后序来遍历  $T_1$ ，然后以后序来遍历  $T_2, \dots$  然后以后序来遍历  $T_n$ ，最后访问  $r$ 。

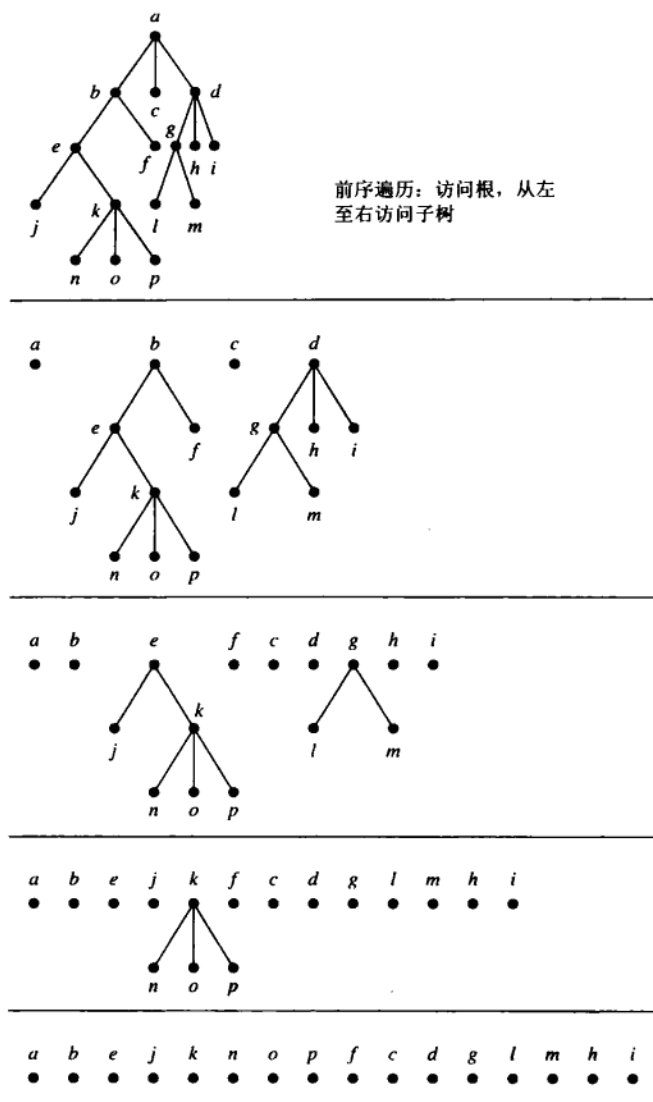
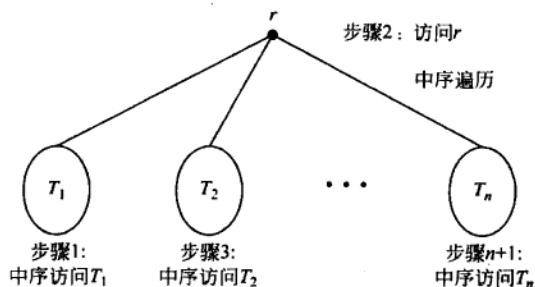
图 7-28  $T$  的前序遍历

图 7-29 中序遍历

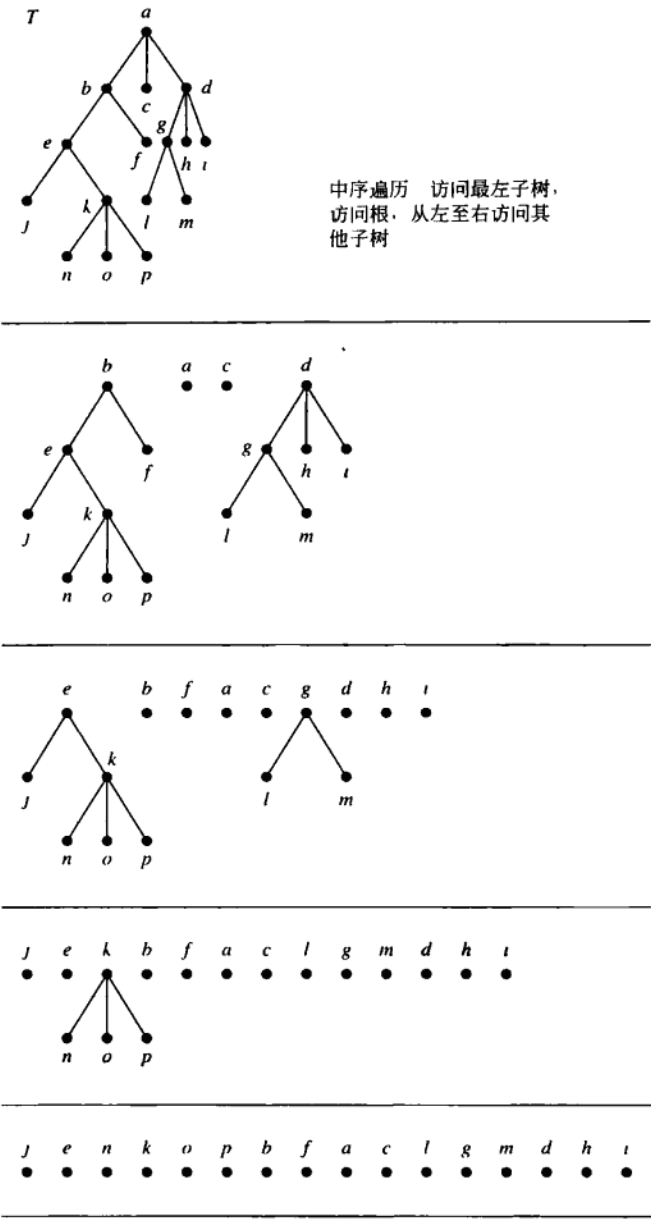


图 7-30  $T$  的中序遍历

图 7-31 说明后序遍历是如何执行的。例 4 说明后序遍历如何工作。

例 4 后序遍历以什么顺序访问图 7-27 所示有序根树  $T$  里的顶点？

解  $T$  的后序遍历的步骤显示在图 7-32 里。后序遍历首先是带根  $b$  的子树的后序遍历，然后是带根  $c$  的子树(它只有  $c$ )的后序遍历，带根  $d$  的子树的后序遍历，接着是根  $a$ 。

带根  $b$  的子树的后序遍历首先是带根  $e$  的子树的后序遍历，接着是  $f$ ，接着是根  $b$ 。带根  $d$  的子树的后序遍历首先是带根  $g$  的子树的后序遍历，接着是  $h$ ，接着是  $i$ ，接着是根  $d$ 。

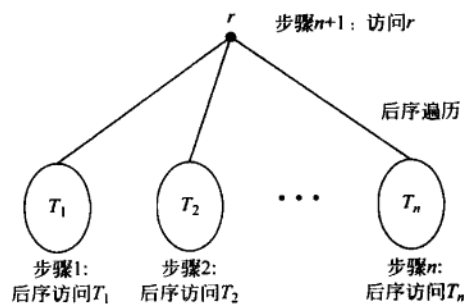
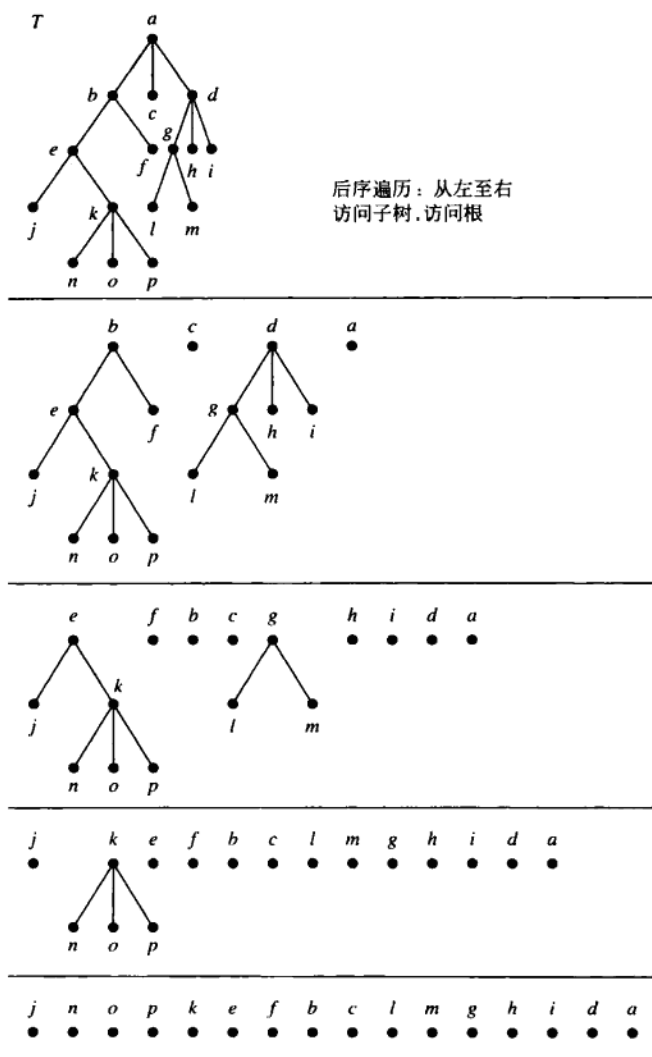


图 7-31 后序遍历

图 7-32  $T$  的后序遍历

带根  $e$  的子树的后序遍历是  $j$ ，接着是带根  $k$  的子树的后序遍历，接着是根  $e$ 。带根  $g$  的子树的后序遍历是  $l, m, g$ 。带根  $k$  的子树的后序遍历是  $n, o, p, k$ 。因此，根树  $T$  的后序遍历是  $j, n, o, p, k, e, f, b, c, l, m, g, h, i, d, a$ 。

有一些简易的方法以前序、中序和后序来列出有序根树的顶点。首先从根开始围绕有序根树画一条曲线，如图 7-33 里的例子所示，沿着边移动。可以这样按照前序来列出顶点：当曲线第一次经过一个顶点时，就列出这个顶点。可以这样按照中序来列出顶点：当曲线第一次经过一个树叶时，就列出这个树叶，当曲线第二次经过一个内点时就列出这个内点。可以这样按照后序来列出顶点：当曲线最后一次经过一个顶点而返回这个顶点的父母时，就列出这个顶点。当在图 7-33 的根树里这样做时，结果是前序遍历给出  $a, b, d, h, e, i, j, c, f, g, k$ ；中序遍历给出  $h, d, b, i, e, j, a, f, c, k, g$ ；后序遍历给出  $h, d, i, j, e, b, f, k, g, c, a$ 。

这些以前序、中序和后序来遍历有序根树的算法，最容易用递归来表示。

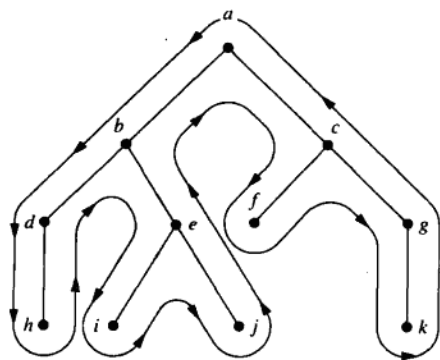


图 7-33 以前序、中序和后序来遍历有序根树的快捷方法

#### 算法 1 前序遍历

```

procedure preorder(T : 有序根树)
 $r := T$ 的根
列出 r
for 从左到右的 r 的每个子女 c
begin
 $T(c) :=$ 以 c 为根的子树
 preorder($T(c)$)
end

```

#### 算法 2 中序遍历

```

procedure inorder(T : 有序根树)
 $r := T$ 的根
if r 是树叶 then 列出 r
else
begin
 $l :=$ 从左到右的 r 的第一个子女
 $T(l) :=$ 以 l 为根的子树
 inorder($T(l)$)
 列出 r
 for 除 l 外从左到右的 r 的每个子女 c
 $T(c) :=$ 以 c 为根的子树
 inorder($T(c)$)
end

```

**算法 3 后序遍历**

**procedure** *postorder*(*T*: 有序根树)

*r* := *T* 的根

**for** 从左到右的 *r* 的每个子女 *c*

**begin**

$T(c) :=$  以 *c* 为根的子树

*postorder*(*T(c)*)

**end**

列出 *r*

注意, 当规定了每个顶点的孩子数时, 有序根树的前序遍历和后序遍历的编码结构相同。也就是说, 当指定树的前序遍历或者后序遍历所生成的顶点列表和每个顶点的孩子数目, 有序根树是唯一确定的(见练习 13)。一般地, 所有有序 *m* 元树的前序遍历和后序遍历的编码结构相同。然而, 当不规定每个顶点的孩子数时, 有序根树的前序遍历和后序遍历编码结构都不同(见练习 14)。

**7.3.4 中缀、前缀和后缀记法**

可以用有序树来表示复杂的表达式, 比如复合命题, 集合的组合, 以及算术表达式。例如, 考虑由运算+(加)、-(减)、\*(乘)、/(除)、 $\uparrow$ (幂)所组成的算术表达式的表示。我们将用括号来说明运算次序。有序根树可以用来表示这样的表达式, 其中内点表示运算, 树叶表示变量或数字。每个运算都作用在它的左子树和右子树上(以此顺序)。

**例 5** 表示表达式  $((x+y) \uparrow 2) + ((x-4)/3)$  的有序根树是什么?

**解** 这个表达式的二叉树可以自底向上来构造。首先, 构造表达式  $x+y$  的子树; 然后, 加入这个子树作为表示  $(x+y) \uparrow 2$  的更大子树的一部分。同样, 构造表达式  $x-4$  的子树, 然后, 加入这个子树到表示  $(x-4)/3$  的子树里; 最后, 组合表示  $(x+y) \uparrow 2$  与  $(x-4)/3$  的子树来形成表示  $((x+y) \uparrow 2) + ((x-4)/3)$  的有序根树。这些步骤都显示在图 7-34 里。 ■

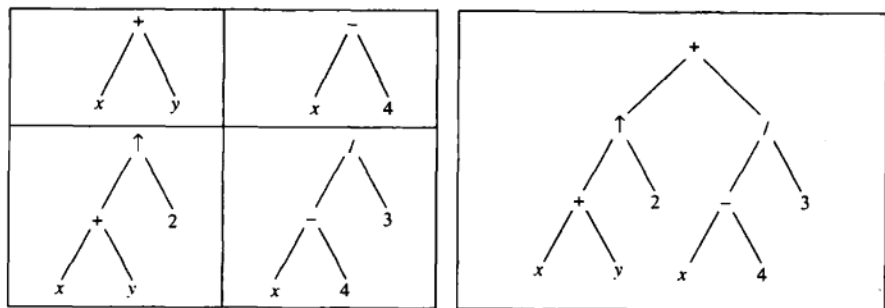
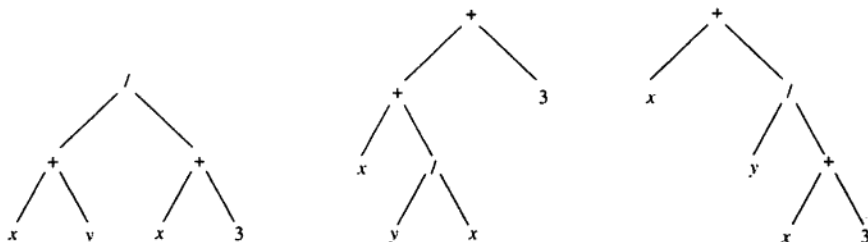


图 7-34 表示  $((x+y) \uparrow 2) + ((x-4)/3)$  的二叉树

对表示一个表达式的二叉树的中序遍历, 产生原来的表达式, 其中元素和运算都是按原有的次序出现, 例外的是一元运算, 它们紧随运算对象。例如, 图 7-35 里的二叉树分别表示表达式  $(x+y)/(x+3)$ ,  $(x+(y/x))+3$  和  $x+(y/(x+3))$ , 对它们的中序遍历都得出中缀表达式  $x+y/x+3$ 。为了让这样的表达式无二义性, 每当遇到运算时, 就有必要在中序遍历里包含括号。以这种方式获得的带完整括号的表达式称为中缀形式。

当以前序来遍历表达式的根树时, 就获得它的前缀形式。写成前缀形式的表达式称为波兰记法, 它的命名由来自于逻辑学家扬·武卡谢维奇(其实他是乌克兰人而非波兰人)。前缀记法下的表达式(其中每个运算都有规定的运算对象数)都是无二义性的。对这个事实的验证留给读者作为练习。

图 7-35 表示  $(x+y)/(x+3)$ ,  $(x+(y/x))+3$  和  $x+(y/(x+3))$  的根树

**例 6**  $((x+y) \uparrow 2) + ((x-4)/3)$  的前缀形式是什么?

**解** 通过遍历图 7-34 所示的表示这个表达式的二叉树, 就获得它的前缀形式。这样就产生  $+\uparrow +xy2/-x43$ 。

在表达式的前缀形式里, 二元运算(比如+)在它的两个运算对象之前。因此, 可以从右向左地求前缀形式的表达式的值。当遇到一个运算时, 就对在这个运算右边紧接着的两个运算对象来执行相应的运算。另外, 每当一个运算执行时, 就认为结果是新的运算对象。

**例 7** 前缀表达式  $+-*235/\uparrow 234$  的值是什么?

**解** 用来从右向左地求这个表达式的值所用的步骤, 以及用右边的运算对象来执行的运算, 都如图 7-36 所示。这个表达式的值是 3。

通过以后序来遍历表达式的二叉树, 就获得它的后缀形式。写成后缀形式的表达式称为是逆波兰记法。逆波兰记法下的表达式都是无二义性的, 所以就不需要括号。对这个事实的验证留给读者。

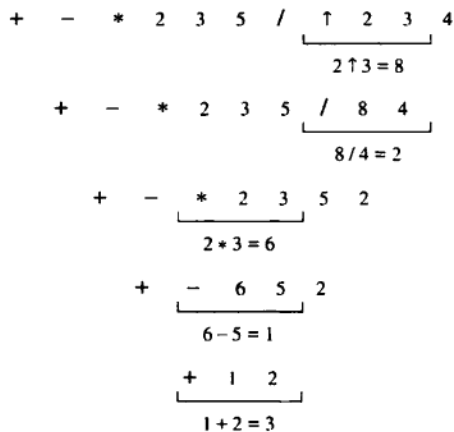
**例 8**  $((x+y) \uparrow 2) + ((x-4)/3)$  的后缀形式是什么?

**解** 这个表达式的后缀形式是这样获得的: 执行图 7-34 所示的表示它的二叉树的后序遍历, 这样就产生后缀表达式  $xy+2 \uparrow x4-3/+$ 。

在表达式的后缀形式里, 二元运算都是在它的两个运算对象之后。所以, 为了从一个表达式的后缀形式来求它的值, 就从左向右地进行, 每当一个运算后面跟着两个运算对象时, 就执行这个运算。在一个运算执行之后, 这个运算的结果就成为一个新的运算对象。

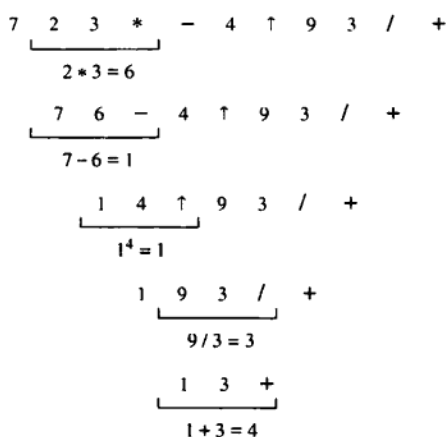
**例 9** 后缀表达式  $723*-4\uparrow 93/+$  的值是什么?

**解** 如图 7-37 所示, 求这个表达式的值所用的步骤是这样的: 从左边开始, 当两个运算对象后面接着一个运算时, 就执行这个运算。这个表达式的值是 4。



表达式的值: 3

图 7-36 求一个前缀表达式的值



表达式的值: 4

图 7-37 求一个后缀表达式的值



根树可以用来表示其他类型的表达式, 比如那些表示复合命题、集合组合的表达式。在这些例子里出现比如命题否定这样的一元运算。为了表示这样的运算及其运算对象, 就用顶点表示运算并且用这个顶点的子女表示运算对象。

**例 10** 求表示复合命题  $(\neg(p \wedge q)) \leftrightarrow (\neg p \vee \neg q)$  的有序根树。然后用这个根树求这个表达式的前缀、后缀和中缀形式。

**解** 这个复合命题的有序根树是自底向上地构造的。首先, 构造  $\neg p$  和  $\neg q$  的子树(其中把  $\neg$  当作一元运算)。另外, 构造  $p \wedge q$  的子树。然后构造  $\neg(p \wedge q)$  和  $(\neg p) \vee (\neg q)$  的子树。最后, 用这两个子树来构造最终的根树。这个过程的步骤显示在图 7-38 中。

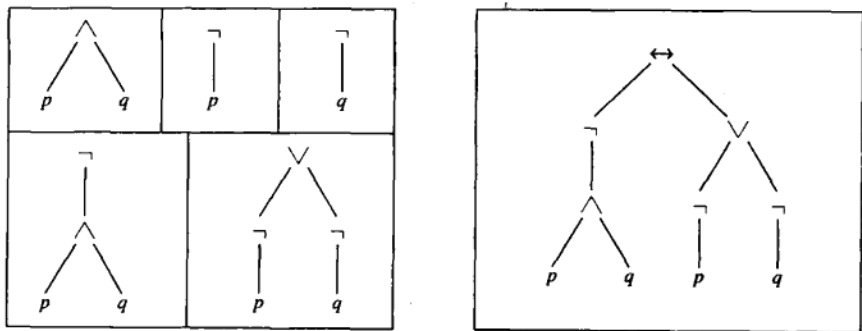


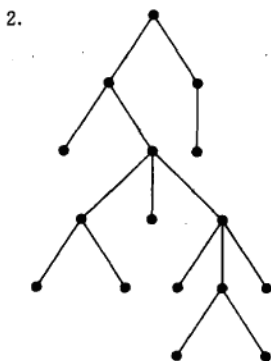
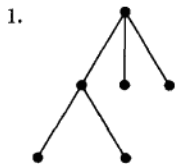
图 7-38 构造一个复合命题的根树

求表达式的前缀、后缀和中缀形式: 分别以前序、后序和中序来遍历这个根树(包含括号)。这些遍历分别给出  $\leftrightarrow \neg \wedge pq \vee \neg p \neg q$ ,  $pq \wedge \neg p \neg q \neg \vee \leftrightarrow$  和  $(\neg(p \wedge q)) \leftrightarrow ((\neg p) \vee (\neg q))$ 。

因为前缀表达式和后缀表达式都是无二义性的, 而且不用来回扫描就容易求出它们的值, 所以它们在计算机科学里大量使用。这样的表达式对编译器的构造是特别有用的。

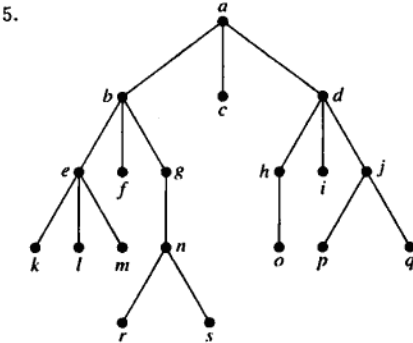
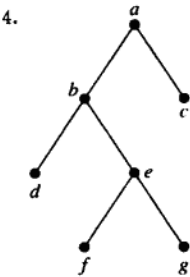
### 练习

在练习 1~2 里, 对给定的有序根树构造通用地址系统。然后利用这个通用地址系统并且用顶点的标记的字典顺序来排序顶点。

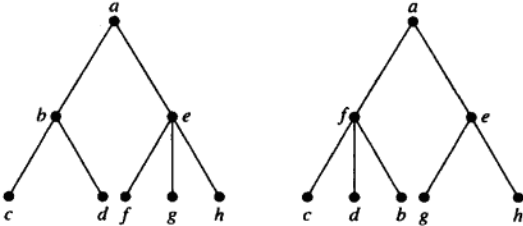


3. 假定在有序根树  $T$  里, 地址最大的顶点的地址是 2.3.4.3.1。是否有可能确定  $T$  里的顶点数?

在练习 4~5 里, 确定前序遍历访问所给的有序根树的顶点的顺序。



6. 使用后序遍历，以什么顺序访问练习 4 里有序根树的顶点？
7. 使用后序遍历，以什么顺序访问练习 5 里有序根树的顶点？
8. 用二叉树来表示表达式  $(x+xy)+(x/y)$  和  $x+((xy+x)/y)$ 。表示方式应采用：
- a) 前缀记法                      b) 后缀记法                      c) 中缀记法
9. 用有序根树来表示  $(A \cap B) - (A \cup (B - A))$ 。表示方式应采用：
- a) 前缀记法                      b) 后缀记法                      c) 中缀记法
- \*10. 有多少种方式给字符串  $A \cap B - A \cup B - A$  完全加上括号以便产生中缀表达式？
11. 下面每个前缀表达式的值是什么？
- a)  $- * 2 / 8 4 3$                                               b)  $\uparrow - * 3 3 * 4 2 5$
- c)  $+ - \uparrow 3 2 \uparrow 2 3 / 6 - 4 2$                                               d)  $* + 3 + 3 \uparrow 3 + 3 3 3$
12. 构造前序遍历为  $a, b, f, c, g, h, i, d, e, j, k, l$  的有序根树，其中  $a$  有四个子女， $c$  有三个子女， $j$  有两个子女， $b$  和  $e$  都有一个子女，所有其他顶点都是树叶。
- \*13. 证明：当规定了有序根树的后序遍历所生成的顶点列表，并且规定了每个顶点的子女数时，这个有序根树是唯一确定的。
14. 证明：下图所示的两个有序根树的后序遍历产生相同的顶点列表。注意这个结果不与练习 27 里的命题相矛盾，因为在这两个有序根树里内点的子女数是不同的。



在一组符号和一组二元运算上用前缀记法表示的合式公式是用下面的规则来递归地定义的：

- (i) 若  $x$  是符号，则  $x$  是用前缀记法表示的合式公式；
- (ii) 若  $X$  和  $Y$  都是合式公式并且  $*$  是运算，则  $*XY$  是合式公式。

- \*15. 证明：在一组符号和一组二元运算上用前缀记法表示的任何合式公式所包含的符号数都比运算数恰好多一个。
16. 给出在符号  $\{x, y, z\}$  和二元运算集  $\{+, \times, \cdot\}$  上带三个以上运算的、用后缀记法表示的合式公式的 6 个例子。

7.4 生成树

7.4.1 引言

考虑一下图 7-39a 所示的简单图所表示的缅因州的道路系统。在冬天里保持道路通畅的唯一方式就是经常扫雪。高速公路部门希望只扫尽可能少的道路上的雪，而确保总是存在连接任何两

个乡镇的干净道路。如何才能做到这一点？

至少扫除5条道路上的雪才能保证在任何两个乡镇之间有一条通路。图7-39b说明了这样一些道路。注意表示这些道路的子图是树，因为它是连通的并且包含6个顶点和5条边。

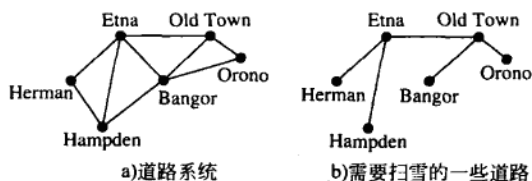


图 7-39

这个问题是用包含原来简单图的所有顶点、边数最小的连通子图来解决的。这样的图必然是树。

**定义1** 设  $G$  是简单图。 $G$  的生成树是包含  $G$  的每个顶点的  $G$  的子图。

有生成树的简单图必然是连通的，因为在任何两个顶点之间都有生成树里的通路。反过来也是对的；即每个连通图都有生成树。在证明这个结果之前将给出一个例子。

**例1** 找出图7-40所示简单图的生成树。

**解** 图  $G$  是连通的，但它不是树，因为它包含简单回路。删除边  $\{a, e\}$ 。这样就消除了一个简单回路，而且所得出的子图仍然是连通的并且仍然包含  $G$  的每个顶点。其次删除边  $\{e, f\}$  以便消除第二个简单回路。最后，删除边  $\{c, g\}$  以便产生一个没有简单回路的简单图。这个子图是生成树，因为它是包含  $G$  的每个顶点的树。图7-41中说明了用来产生这个生成树的边的删除序列。

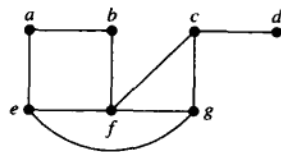


图 7-40 简单图  $G$

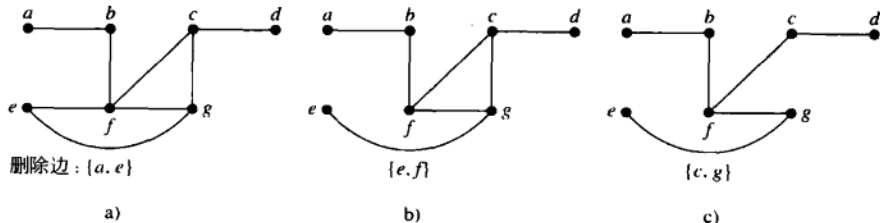


图 7-41 通过删除形成简单回路的边来产生  $G$  的一个生成树

图7-41所示的生成树不是唯一的  $G$  的生成树。例如，图7-42所示的每个树都是  $G$  的生成树。

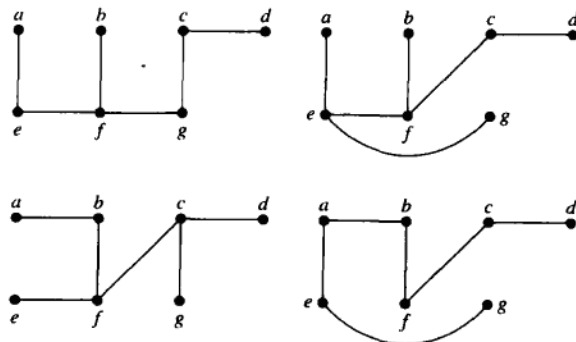


图 7-42  $G$  的一些生成树

**定理 1** 简单图是连通的当且仅当它具有生成树。

**证** 首先, 假定简单图  $G$  有生成树  $T$ 。 $T$  包含  $G$  的每个顶点。另外, 在  $T$  的任何两个顶点之间都有  $T$  里的通路。因为  $T$  是  $G$  的子图, 所以在  $G$  的任何两个顶点之间都有  $T$  里的通路。因此,  $G$  是连通的。

现在假定  $G$  是连通的。若  $G$  不是树, 则它必然包含简单回路。从这些简单回路中的一个里删除一条边。所得出的子图少了一条边, 但是仍然包含  $G$  的所有顶点并且是连通的。若这个子图不是树, 则它有简单回路; 所以像前面那样, 删除一个简单回路里的一条边。重复这个过程直到没有简单回路为止。这是可能的, 因为在图里只有有穷的边数。当没有简单回路剩下时, 这个过程终止。产生出一个树, 因为在删除边时这个图保持连通。这个树是生成树, 因为它包含  $G$  的每个顶点。□

例 2 说明, 在数据网络里生成树是重要的。

**例 2 IP 组播** 在网际协议(IP)网络上的组播里, 生成树的作用很重要。为了从源计算机发送数据到多个接收计算机, 每个接收计算机是一个子网, 可以分别发送数据到每个计算机。称为单点广播的这种类型的网络是无效的, 这是因为在网络上存有发送相同数据的多个副本。为了更有效地传送数据到多个接收计算机, 就使用 IP 组播。在 IP 组播里, 一个计算机在网络上发送数据的单一副本, 当数据到达中间路由器时, 就把数据分发给一个或更多的其他路由器, 以便接收计算机都在它们不同的子网里最终接收到这些数据。(路由器是专门在网络里子网之间分发 IP 数据报文的计算机。在组播里, 路由器使用 D 类地址, 每个都表示接收计算机可以加入的一个会话; 见 3.1 节例 16。)

为了让数据尽可能快地到达接收计算机, 在数据穿过网络的通路里就不应当存在环路(在图论名词里它们是回路)。即一旦数据已经到达一个具体的路由器, 数据就再也不应当返回这个路由器。为了避免环路, 组播路由器用网络算法来构造图 7-43 所示的生成树, 这个图以组播源、路由器和包含接收计算机的子网来作为顶点, 以边表示计算机和(或)路由器之间的连接。这个生成树的根就是组播源。包含接收计算机的子网就是这个树的树叶(注意不包含接收计算机的子网都不包含在这个图里)。在图 7-43 里说明这些内容。■

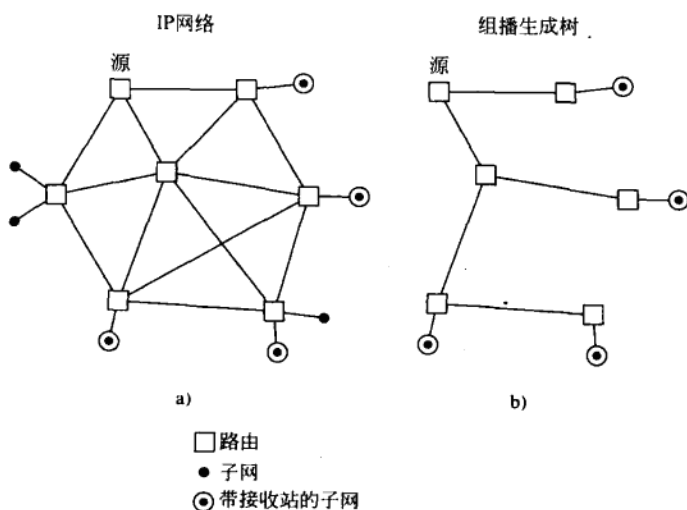


图 7-43 一个组播生成树

### 7.4.2 深度优先搜索

**定理 1** 的证明给出通过从简单回路删除边来找出生成树的算法。这个算法是低效的，因为它要求找出简单回路。另一种不采用删除边来构造生成树的方法是，通过相继地添加边来建立生成树。在这里将给出基于这个原理的两个算法。

可以用深度优先搜索来建立连通简单图的生成树。我们将形成一个根树，而生成树将是这个根树的无向底图。任意选择图中一个顶点作为根。通过相继地添加边来形成从这个顶点开始的通路，其中每条新边都与通路上的最后一个顶点以及还不在通路上的一个顶点相关联。继续尽可能地添加边到这条通路。若这条通路经过图的所有顶点，则由这条通路组成的树就是生成树。不过，若这条通路没有经过图的所有顶点，则必须添加其他的边。退到通路里的倒数第二个顶点，若有可能，则形成从这个顶点上开始的经过还没有访问过的顶点的通路。若不能这样做，则后退到通路里的另外一个顶点，即在通路里后退两个顶点，然后再试。

重复这个过程，从所访问过的最后一个顶点开始，在通路上一次后退一个顶点，只要有可能就形成新的通路，直到不能添加更多的边为止。因为这个图有有穷的边数并且是连通的，所以这个过程以产生生成树而告终。在这个算法的一个阶段上是通路末端的顶点将是根树里的树叶，而在其上开始构造一条通路的顶点将是内点。

读者应当注意到这个过程的递归本质。另外，注意若图中的顶点是排序的，则当总是选择在该顺序里可用的第一个顶点时，在这个过程的每个阶段上对边的选择就全都是确定的。不过，将不总是明显地对图的顶点排序。

深度优先搜索也称为回溯，因为这个算法返回以前访问过的顶点以便添加边。下面的例子说明回溯。

**例 3** 用深度优先搜索来找出图 7-44 所示图  $G$  的生成树。

**解** 在图 7-45 里显示深度优先搜索为了产生  $G$  的生成树而使用的步骤。任意地从顶点  $f$  开始。一条通路是这样建立的：相继地添加与还不通路上的顶点相关联的边，只要有可能就这样做。这样就产生通路  $f, g, h, k, j$  (注意也可能建立其他的通路)。下一步，回溯到  $k$ 。不存在从  $k$  开始包含还没有访问过的顶点的通路。所以回溯到  $h$ 。形成通路  $h, i$ 。然后回溯到  $g$ ，然后再回溯到  $f$ 。从  $f$  建立通路  $f, d, e, c, a$ 。然后再回溯到  $c$  并且形成通路  $c, b$ 。这样就产生了生成树。 ■

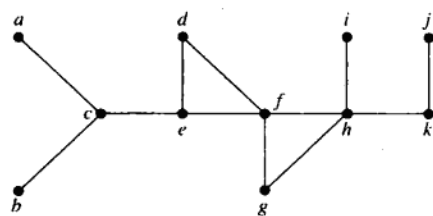


图 7-44 图  $G$

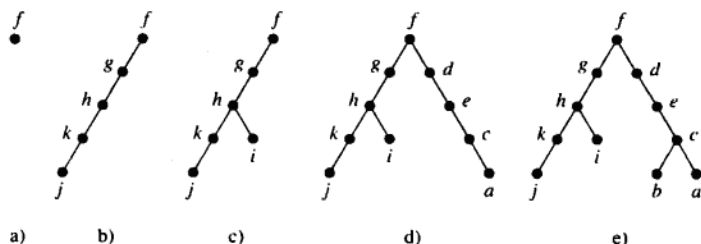


图 7-45  $G$  的深度优先搜索

一个图的深度优先搜索所选择的边都称为树边。这个图所有其他的边都必然连接一个顶点与这个顶点在树中的祖先或后代。这些边都称为背边(练习 20 要求证明这个事实)。

**例 4** 图 7-46 中强调了从顶点  $f$  开始的深度优先搜索所找到的树边，用粗线显示这些树边。用细黑线显示背边  $(e, f)$  和  $(f, h)$ 。 ■

我们已经解释了如何用深度优先搜索来求图的生成树。但是，迄今为止的讨论还没有带出深度优先搜索的递归本质。为了弄清深度优先搜索的递归本质，需要几个术语。当执行深度优先搜索的步骤时，当把顶点  $v$  加入树时说从顶点  $v$  开始探索，当最后一次回溯回到  $v$  时说从顶点  $v$  结束探索。理解算法的递归本质所需要的关键事实是，当加入连接顶点  $v$  到顶点  $w$  的边时，在回到  $v$  完成从  $v$  的探索之前就结束了从  $w$  的探索。在算法 1 中构造了带顶点  $v_1, \dots, v_n$  的图  $G$  的生成树，首先选择顶点  $v_1$  作为树根。开始时令  $T$  是只有这一个顶点的树。在每个步骤，加入一个新顶点到  $T$  以及从已在  $T$  中的一个顶点到这个新顶点的一条边，并且从这个新顶点探索。注意当算法完成时， $T$  没有简单回路，因为没有加入连接到已在树中的顶点的边。另外， $T$  在构造时保持连通(用数学归纳法可以轻而易举地证明最后这两个事实)。由于  $G$  是连通的， $G$  的每个顶点都被算法访问到并且加入到树中(读者可以验证)。所以  $T$  是  $G$  的生成树。

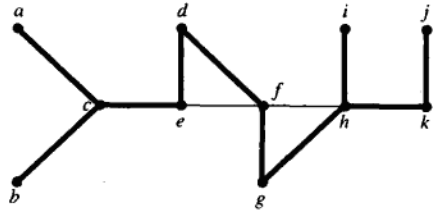


图 7-46 例 4 中深度优先搜索的树边和背边

**算法 1** 深度优先搜索

**procedure** DFS( $G$ : 带顶点  $v_1, \dots, v_n$  的连通图)

$T :=$  只包含顶点  $v_1$  的树

$visit(v_1)$

**procedure**  $visit(v$ :  $G$  的顶点)

**for** 与  $v$  相邻并且还不在于  $T$  中的每个顶点  $w$

**begin**

加入顶点  $w$  和边  $\{v, w\}$  到  $T$

$visit(w)$

**end**

现在分析深度优先搜索算法的计算复杂性。关键事实是对于每个顶点  $v$  来说，当在搜索中首次遇到顶点  $v$  时，就调用过程  $visit(v)$  并且以后不再调用这个过程。假设  $G$  的邻接表是可用的(参见 6.3 节)，求出与  $v$  相邻的顶点不需要任何计算。当遵循算法的步骤时，至多检查每条边两次以确定是否加入这条边及其一个端点到树中。因此，过程  $DFS$  用  $O(e)$  或  $O(n^2)$  个步骤来构造一个生成树，其中  $e$  和  $n$  分别是  $G$  的边数和顶点数。(注意一个步骤包括：检查一个顶点是否已在正在构造的树中，如果这个顶点还不在于树中，则加入这个顶点和对应的边。还利用了不等式  $e \leq n(n-1)/2$ ，对于任意简单图来说这个不等式都成立。)

深度优先搜索可以作为解决许多不同问题的算法的基础。例如，可以用来求图的连通分支并且可以用来求连通图的割点。将要看到，深度优先搜索是用来搜索计算困难问题的解的回溯技术的基础(参见[GrYe99]、[Ma89]和[CoLeRiSt01]对基于深度优先搜索的算法的讨论)。

7.4.3 宽度优先搜索

也可以通过使用宽度优先搜索来产生简单图的生成树。同样，将构造一个根树，而这个根树的无向底图就形成生成树。从图的顶点中任意地选择一个根。然后添加与这个顶点相关联的所有边。在这个阶段所添加的新顶点成为生成树在 1 层上的顶点。将新顶点任意排序。下一步，按顺序访问 1 层上的每个顶点，只要不产生简单回路，就将与这个顶点相关联的每条边添加到树里。这样就产生了树在 2 层上的顶点。遵循相同的过程，直到已经添加了树里的所

有顶点。因为在图中的边数是有限的，所以这个过程会终止。在产生了包含图中每一个顶点的树之后，生成树也就产生了。下面是宽度优先搜索的一个例子。

**例 5** 用宽度优先搜索来找出图 7-47 所示的图的生成树。

**解** 在图 7-48 里显示宽度优先搜索过程的各步骤。  
选择顶点  $e$  作为根。然后添加与  $e$  相关联的所有边，所以添加了从  $e$  到  $b, d, f$  和  $i$  的边。这四个顶点都是在树的 1 层上。下一步，添加从 1 层上的顶点到还在树上的相邻顶点的边。因此，添加从  $b$  到  $a$  和  $c$  的边，以及从  $d$  到  $h$ ，从  $f$  到  $j$  和  $g$ ，从  $i$  到  $k$  的边。新顶点  $a, c, h, j, g$  和  $k$  都是在 2 层上。下一步，添加从这些顶点到还在树上的相邻顶点的边。这样就添加从  $g$  到  $l$  以及从  $k$  到  $m$  的边。

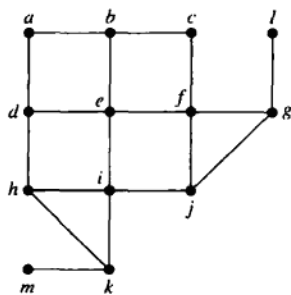


图 7-47 图  $G$

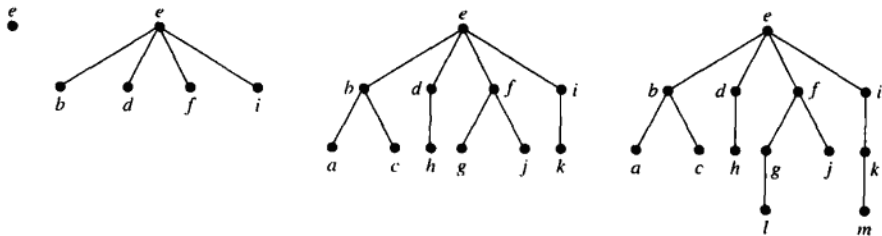


图 7-48  $G$  的宽度优先搜索

算法 2 的伪码描述了宽度优先搜索。在这个算法中假设连通图  $G$  的顶点排序为  $v_1, \dots, v_n$ 。在算法中，只要还没有产生简单回路就加入新的顶点和对应的边到树中，与正在处理的当前顶点相邻。

#### 算法 2 宽度优先搜索

**procedure** BFS( $G$ ; 带顶点  $v_1, \dots, v_n$  的连通图)

$T :=$  只包含顶点  $v_1$  的树

$L :=$  空表

把  $v_1$  放入尚未处理顶点的表  $L$  中

**while**  $L$  非空

**begin**

    删除  $L$  中第一个顶点  $v$

**for**  $v$  的每个邻居  $w$

**if**  $w$  既不在  $L$  中也不在  $T$  中 **then**

**begin**

                加入  $w$  到表  $L$  的末尾

                加入  $w$  和边  $\{v, w\}$  到  $T$

**end**

**end**

现在分析宽度优先搜索的计算复杂性。对于图中的每个顶点  $v$  来说，检查所有与  $v$  相邻的顶点并且加入每个尚未访问过的顶点到树  $T$  中。假设图的邻接表是可用的，确定哪些顶点与给定顶点相邻就不需要任何计算。如同在深度优先搜索算法的分析中那样，我们检查每条边至多两次来确定是否应当加入这条边及其尚未在树中的端点。所以宽度优先搜索算法使用  $O(e)$  或  $O(n^2)$  个步骤。

## 7.4.4 回溯

有些问题只能通过执行对所有可行解的穷举搜索来解决。系统地搜索出一个解的一种方式是使用决策树，其中每个内点都表示一次决策，而每个树叶都表示一个可行解。为了通过回溯来求出一个解，首先尽可能地做出一系列决策来尝试得出一个解。可以用决策树里的通路来表示决策序列。一旦知道了决策序列的任何扩展都不能得出解，就回溯到父母顶点并且若有可能，则用另外一个决策序列来尝试得出一个解。继续这个过程，直到找到一个解，或者证明没有解存在为止。下面的例子说明回溯的有用性。

**例6 图着色** 如何用回溯来判定是否可以用  $n$  种颜色给一个图着色？

**解** 以下面的方式用回溯来解决这个问题。首先选择某个顶点  $a$  并且指定它颜色为 1。然后挑选第二个顶点  $b$ ，而且若  $b$  不与  $a$  相邻，则指定它颜色为 1。否则，指定  $b$  颜色为 2。然后来到第三个顶点  $c$ 。若有可能则对  $c$  用颜色 1。否则若有可能则用颜色 2。只有当颜色 1 和颜色 2 都不能用时才使用颜色 3。继续这个过程，只要有可能就为每个新顶点指定  $n$  种颜色中的一种，而且总是使用表中第一种允许的颜色。若遇到不能用  $n$  种颜色中任何一种来着色的顶点，则回溯到最后一次所做的指定，并且若有可能就改变最后着色的顶点的颜色，用表中下一种允许的颜色。若不可能改变这个颜色，则再回溯到更前面的指定，一次后退一步，直到有可能改变一个顶点的颜色为止。然后只要有可能就继续指定新顶点的颜色。若使用  $n$  种颜色的着色存在，则可以通过回溯来产生（但是这个过程是极其低效的）。

具体地说，考虑用三种颜色来着色图 7-49 所示的图。图 7-49 所示的树说明如何用回溯来构造 3 着色。在这个过程里，首先用红色，其次用蓝色，最后用绿色。显然用回溯来解决这个简单的例子只是为了能够比较好地说明这项技术。

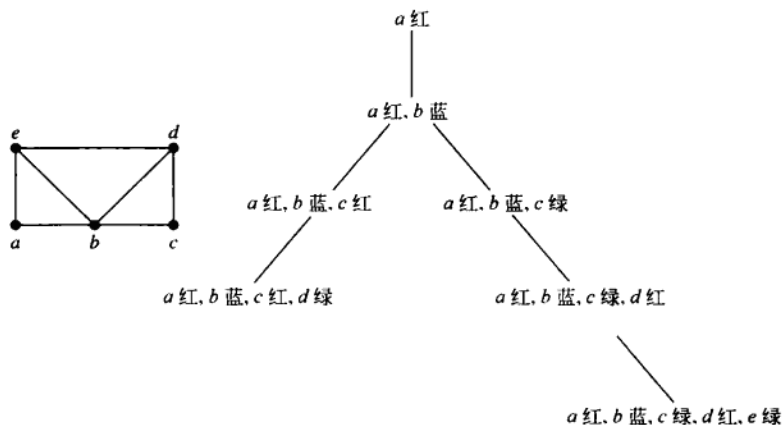


图 7-49 用回溯给图着色

在这个树里，从根开始的、表示指定红色给  $a$  的最初的通路，导致  $a$  红色、 $b$  蓝色、 $c$  红色而  $d$  绿色的着色。当以这种方式来着色  $a, b, c$  和  $d$  时，就不可能用这三种颜色中的任何一种来着色  $e$ 。所以，回溯到表示这个着色的顶点的父母。因为没有其他颜色可以用在  $d$  上，所以再回溯一层。然后改变  $c$  的颜色为绿色。通过接着指定红色给  $d$  和绿色给  $e$ ，就获得这个图的着色。 ■



**例7  $n$  皇后问题**  $n$  皇后问题问：在  $n \times n$  棋盘上如何放置  $n$  个皇后，使得没有两个皇后可以互相攻击。如何用回溯来解决  $n$  皇后问题？

**解** 为了解决这个问题，必须在  $n \times n$  棋盘上找出  $n$  个位置，使得这些位置中没有两个是在同一行上、同一列上或同一斜线上（斜线是由对某个  $m$  来说满足  $i + j = m$  或对某个  $m$  来说满足  $i - j = m$  的所有位置  $(i, j)$  组成的）。将用回溯来解决  $n$  皇后问题。从空棋盘开始。在  $k+1$  阶段，



尝试在棋盘上第  $k+1$  列里放置一个新皇后，其中在前  $k$  列里已经有了皇后。检查第  $k+1$  列里的格子，从第一行的格子开始，寻找放置这个皇后的位置，使得它不与已经在棋盘上的皇后在同一行里或同一斜线上(已经知道它不在同一列里)。若不可能在第  $k+1$  列里找到放置皇后的位置，则回溯到在第  $k$  列里对皇后的放置。在这一列里下一个允许的行里放置皇后，若这样的行存在的话。若没有这样的行存在，则继续回溯。

具体地说，图 7-50 显示四皇后问题的回溯解法。在这种解法里，在第一行第一列里放置一个皇后。然后在第二列的第三行里放置一个皇后。不过，这样就使得不可能在第三列里放置一个皇后。所以就回溯并且在第二列的第四行里放置一个皇后。当这样做时，就可以在第三列的第二行里放置一个皇后。但是没有办法在第四列里添加一个皇后。这说明当在第一行第一列里放置一个皇后时就得不到解。回溯到空棋盘，在第一列的第二行里放置一个皇后。这样就得出图 7-50 所示的解。

**例 8 子集和** 考虑下面的问题。给定一组正整数  $x_1, x_2, \dots, x_n$ ，求这组整数的和为  $M$  的一个子集。如何用回溯来解决这个问题？

**解** 从空无一项的和来开始。通过相继地添加项来构造这个和。若当添加这个序列里的一个整数到和里而这个和仍然小于  $M$  时，则包含这个整数。若得出使得添加任何一项就大于  $M$  的一个和，则通过去掉这个和的最后一项来回溯。

图 7-51 显示下面这个问题的回溯解法，求  $\{31, 27, 15, 11, 7, 5\}$  的和等于 39 的子集。

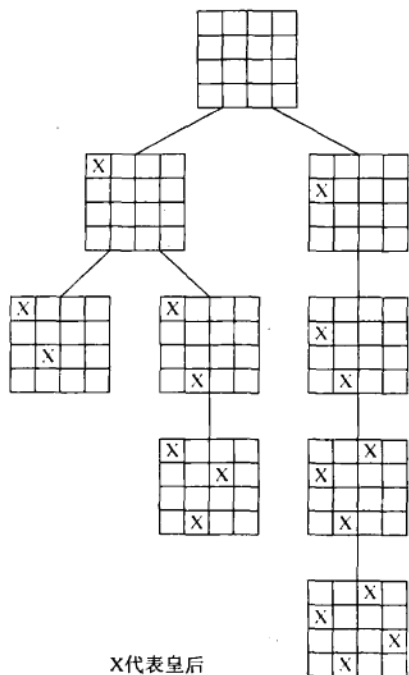


图 7-50 四皇后问题的回溯解法

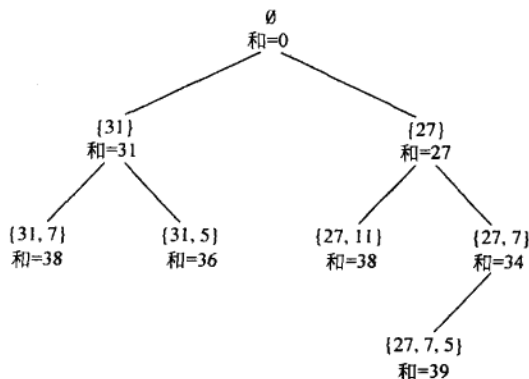


图 7-51 用回溯求等于 39 的和

#### 7.4.5 有向图中的深度优先搜索

可以轻而易举地修改深度优先搜索和宽度优先搜索，使得以有向图作为输入时它们也能运行。但是，输出不一定是生成树，而可能是森林。在这两个算法中，只有当一条边从正在考虑的顶点发出并且到一个尚未加入的顶点时才加入这条边。如果在其中任何一个算法的某个阶段找不到从已经加入的顶点到尚未加入的顶点的边，则算法加入的下一个顶点成为生成森林中一个新树的根。



4. 求下面每个图的生成树。

a)  $K_3$

b)  $K_{4,4}$

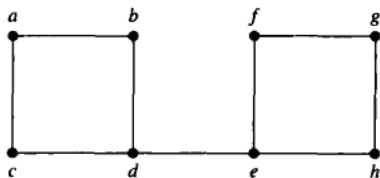
c)  $K_{1,6}$

d)  $Q_3$

e)  $C_5$

f)  $W_5$

5. 画出下面所给的简单图的生成树。



\*6. 下面的每个简单图各有多少个不同的生成树?

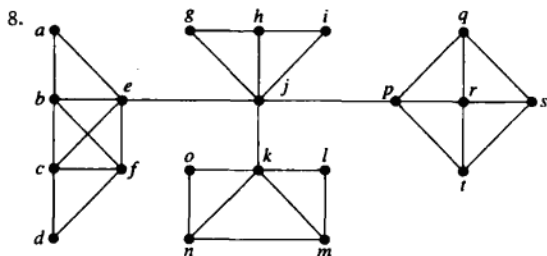
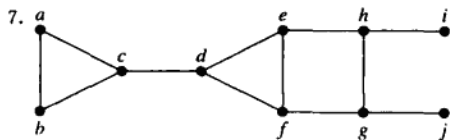
a)  $K_3$

b)  $K_4$

c)  $K_{2,2}$

d)  $C_5$

在练习 7~8 里, 用深度优先搜索来构造所给的简单图的生成树。选择  $a$  作为这个生成树的根并且假定顶点都以字母顺序来排序。



9. 用深度优先搜索求下列这些图的生成树。

a)  $W_5$  (参见 6.2 节例 6), 从 6 度顶点开始

b)  $K_5$

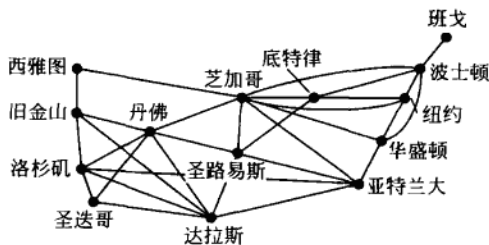
c)  $K_{3,4}$ , 从 3 度顶点开始

d)  $Q_3$

10. 描述一下轮图  $W_n$  的宽度优先搜索和深度优先搜索所产生的树, 从  $n$  度顶点开始, 其中  $n$  是整数满足  $n \geq 3$  (参见 6.2 节例 6)。说明答案的合理性。

11. 描述一下完全偶图  $K_{m,n}$  的宽度优先搜索和深度优先搜索所产生的树, 其中  $m$  和  $n$  都是正整数。说明答案的合理性。

12. 假定一家航空公司必须压缩它的航班以节省资金。若它原来的航线如下图所示, 则可以中断哪些飞行以保持所有城市对之间的服务 (其中从一个城市飞往另一个城市可能需要换乘飞机)?



13. 哪些连通简单图恰好有一个生成树?

\*14. 证明: 在连通简单图里顶点  $v$  和  $u$  之间的最短通路的长度, 等于在以  $v$  为根的  $G$  的宽度优先生成树里  $u$  的层数。

15. 用回溯来对下面的  $n$  值解决  $n$  皇后问题。

a)  $n=3$

b)  $n=5$

c)  $n=6$

16. 解释一下如何用回溯来找出图中的哈密顿通路或哈密顿回路。

图  $G$  的生成森林是包含  $G$  的每个顶点的森林, 使得当在两个顶点之间有  $G$  里的通路时, 这两个顶点就在同一个树里。

17. 证明: 每个有穷简单图都有生成森林。

18. 对带有  $n$  个顶点、 $m$  条边和  $c$  个连通分支的图来说, 必须删除多少条边才能产生它的生成森林?
19. 设计基于深度优先搜索来构造图的生成森林的算法。
20. 设  $G$  是连通图。证明: 如果  $T$  是用深度优先搜索构造的  $G$  的生成树, 则  $G$  的不在  $T$  中的边必定是背边, 换句话说, 这条边必定连接一个顶点到这个顶点在  $T$  中的祖先或后代。
21. 对于哪些图来说, 无论选择哪个顶点作为树根, 深度优先搜索和宽度优先搜索都产生同样的生成树? 说明答案的合理性。
22. 用数学归纳法证明: 宽度优先搜索按照顶点在所得出的生成树中的层数的顺序来访问这些顶点。
23. 用伪码来描述宽度优先搜索的一个变种, 它把整数  $m$  指定给在搜索中访问的第  $m$  个顶点。
24. 证明: 如果  $G$  是有向图并且  $T$  是用深度优先搜索构造的生成树, 则不在这个生成树上的每条边都是连接祖先到后代的前进边、连接后代到祖先的后退边, 或者连接一个顶点到从前访问过的子树的一个顶点的交叉边。

设  $T_1$  和  $T_2$  都是一个图的生成树。在  $T_1$  和  $T_2$  之间的距离是在  $T_1$  和  $T_2$  里非  $T_1$  和  $T_2$  所共有的边的数目。

25. 求图 7-40 所示图  $G$  的在图 7-41c 和图 7-42 里所示的每对生成树之间的距离。
- \*\*26. 假定  $T_1$  和  $T_2$  都是简单图  $G$  的生成树。另外, 假定  $e_1$  是在  $T_1$  里但不在  $T_2$  里的一条边。证明: 存在着在  $T_2$  里但不在  $T_1$  里的一条边  $e_2$ , 使得若从  $T_1$  里删除  $e_1$  而添加  $e_2$  到  $T_1$  里, 则  $T_1$  仍然是生成树, 并且若从  $T_2$  里删除  $e_2$  而添加  $e_1$  到  $T_2$  里, 则  $T_2$  仍然是生成树。

有向图的根生成树是由这个图的边组成的根树, 使得这个图的每个顶点都是树里一条边的终点。

27. 对 6.5 节练习 10~12 里的每个有向图来说, 求这个图的根生成树, 或者确定不存在这样的树。
- \*28. 给出构造每个顶点的入度和出度都相等的连通有向图的根生成树的算法。

## 7.5 最小生成树

### 7.5.1 引言

一个公司计划建立一个通信网络来连接它的五个计算机中心。可以用租用的电话线连接这些中心的任何一对。应当建立哪些连接, 以便保证在任何两个计算机中心之间都有通路, 且网络的总成本最小? 可以用图 7-53 所示的带权图为此问题建模, 其中顶点表示计算机中心, 边表示可能租用的电话线, 边上的权是边所表示的电话线的月租费。通过找出一棵生成树, 使得这棵树的各边的权之和为最小, 就可以解决这个问题。这样的生成树称为最小生成树。

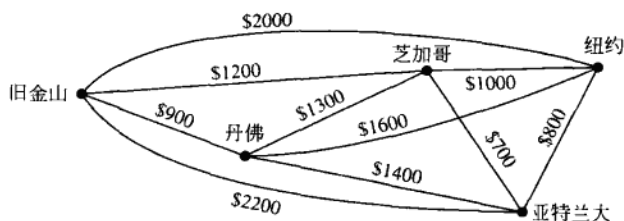


图 7-53 说明计算机网络里的线路的月租费的带权图

### 7.5.2 最小生成树算法

大量的问题可以这样解决: 求带权图里的一棵生成树, 使得这棵树的各边的权之和为最小。

**定义 1** 连通带权图里的最小生成树是具有最小可能的边的权之和的生成树。

下面将给出构造最小生成树的两个算法。这两个算法都是通过添加还没有使用过的、具有规定性质的、权最小的边来进行的。这些算法都是贪心算法的例子。贪心算法是在每个步骤上都做最优选择的算法。在算法的每个步骤上都最优化, 并不保证产生全局最优解。不过, 本节里给出的构造最小生成树的这两个算法都是产生最优解的贪心算法。

要讨论的第一个算法是罗伯特·普林在 1957 年给出的, 虽然这个算法的基本想法有更早的起源。为了执行普林算法, 首先选择带最小权的边, 把它放进生成树里。相继地向树里添加与已在树里的顶点关联的、并且不与已在树里的边形成简单回路的权最小的边。当已

经添加了  $n-1$  条边时就停止。

本节稍后将证明这个算法产生任何连通带权图的最小生成树。算法 1 给出普林算法的伪码描述。

**算法 1 普林算法**

**procedure** *Prim*( $G$ : 带  $n$  个顶点的连通无向图)

$T :=$  权最小的边

**for**  $i := 1$  **to**  $n-2$

**begin**

$e :=$  与  $T$  里顶点关联、并且若添加到  $T$  里则不形成简单回路的权最小的边

$T :=$  添加  $e$  之后的  $T$

**end** ( $T$  是  $G$  的最小生成树)

注意, 当有超过一条满足相应条件的带相同权的边时, 在算法的这个阶段里对所添加的边的选择就不是确定的。需要排序这些边以便让选择是确定的。在本节剩下的部分里将不再考虑这个问题。另外注意, 所给的连通带权简单图可能有多于一个的最小生成树(见练习 5)。例 1 和例 2 说明如何使用普林算法。

**例 1** 用普林算法设计连接图 7-53 所表示的所有计算机的、具有最小成本的通信网络。

**解** 办法是求图 7-53 的最小生成树。普林算法是这样执行的: 选择权最小的初始边, 并且相继地添加与树里顶点关联的不形成回路的权最小的边。在图 7-54 里, 加粗的边表示普林算法所产生的最小生成树, 并且显示出在每个步骤上所做的选择。

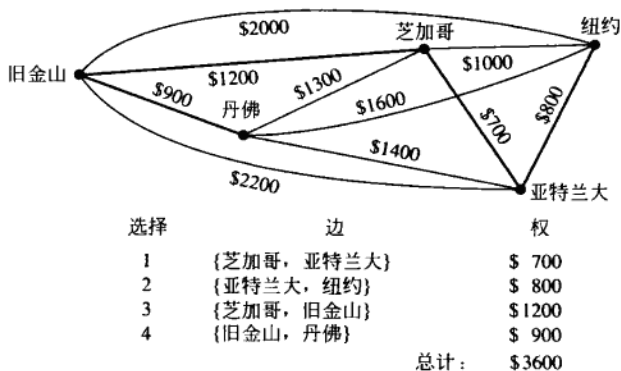


图 7-54 图 7-53 里带权图的最小生成树

**例 2** 用普林算法求图 7-55 所示的图的最小生成树。

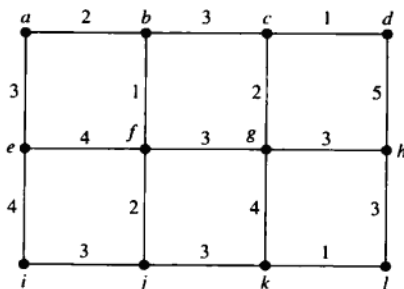


图 7-55 一个带权图

**解** 用普林算法所构造的最小生成树显示在图 7-56 里。相继选择的边都有显示。

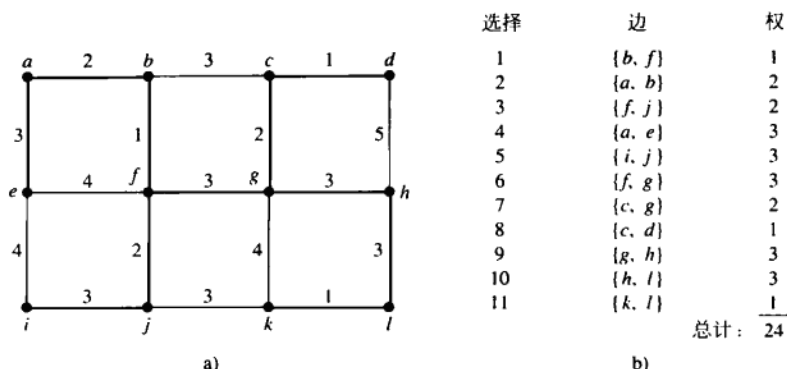


图 7-56 用普林算法构造的最小生成树

将要讨论的第二个算法是约瑟夫·克鲁斯卡尔在1956年发现的，但在此之前已经有人阐述过这一算法的基本思路。为了执行克鲁斯卡尔算法，要选择图中权最小的一条边。

相继添加不与已经选择的边形成简单回路的权最小的边。在已经挑选了  $n-1$  条边之后就停止。

在本节末尾，把证明克鲁斯卡尔算法对每个连通带权图都产生最小生成树留作练习。在算法2里给出克鲁斯卡尔算法的伪代码。

#### 算法2 克鲁斯卡尔算法

**procedure** *Kruskal*( $G$ : 带  $n$  个顶点的带权连通无向图)

$T :=$  空图

**for**  $i := 1$  **to**  $n-1$

**begin**

$e :=$  当添加到  $T$  里时不形成简单回路的权最小的  $G$  里的边

$T :=$  添加  $e$  之后的  $T$

**end** ( $T$  是  $G$  的最小生成树)

读者应当注意普林算法与克鲁斯卡尔算法的区别。在普林算法里，选择与已在树里的一个顶点相关联、并且不形成回路的权最小的边；相反地，在克鲁斯卡尔算法里，选择不一定与已在树里的一个顶点相关联、并且不形成回路的权最小的边。注意，在普林算法里，若没有对边排序，则在这个过程的某个阶段上，对添加的边来说就可能有多种选择。因此，为了让这个过程是确定的，就需要对边进行排序。下面的例子说明如何使用克鲁斯卡尔算法。

**例3** 用克鲁斯卡尔算法求图7-55所示的带权图的最小生成树。

**解** 在图7-57里显示这个最小生成树和在克鲁斯卡尔算法每个阶段上对边的选择。现在将证明普林算法产生连通带权图的最小生成树。

**证** 设  $G$  是一个连通带权图。假定普林算法相继地选择的边是  $e_1, e_2, \dots, e_{n-1}$ 。设  $S$  是以  $e_1, e_2, \dots, e_{n-1}$  作为边的树，而设  $S_k$  是以  $e_1, e_2, \dots, e_k$  作为边的树。设  $T$  是包含边  $e_1, e_2, \dots, e_k$  的  $G$  的最小生成树，其中  $k$  是满足下列性质的最大整数：存在着包含普林算法所选择的前  $k$  条边的最小生成树。若证明了  $S=T$ ，则该定理得证。

假定  $S \neq T$ ，所以  $k < n-1$ 。因此， $T$  包含边  $e_1, e_2, \dots, e_k$ ，但是不包含  $e_{k+1}$ 。考虑由  $T$  和  $e_{k+1}$  所组成的图。因为这个图是连通的并且有  $n$  条边，边过多不可能是树，所以它必然包含简单回路。这个简单回路必然包含  $e_{k+1}$ ，因为在  $T$  里没有简单回路。另外，在这个简单回路里必然有不属于  $S_{k+1}$  的边，因为  $S_{k+1}$  是一棵树。通过从  $e_{k+1}$  的一个端点开始，该端点也是边  $e_1, e_2, \dots, e_k$  之一的端点，并且遵循回路直到它到达一条不在  $e_{k+1}$  里的边为止，就可以找出一条不在  $S_{k+1}$  里的边  $e$ ，它有一个端点也是边  $e_1, e_2, \dots, e_k$  之一的端点。通过从  $T$  里删除  $e$  并且添加  $e_{k+1}$ ，就获得带

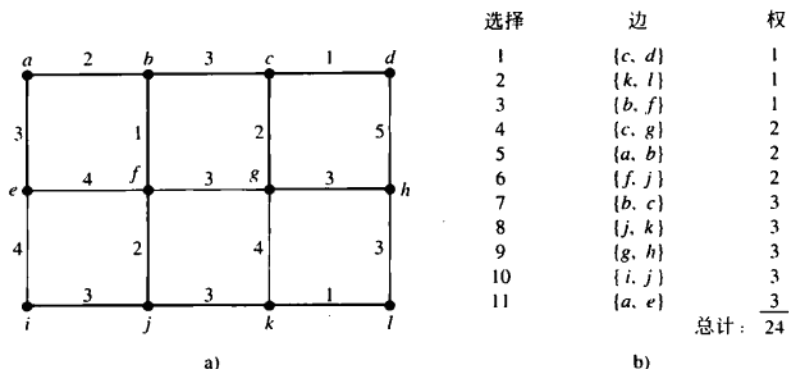


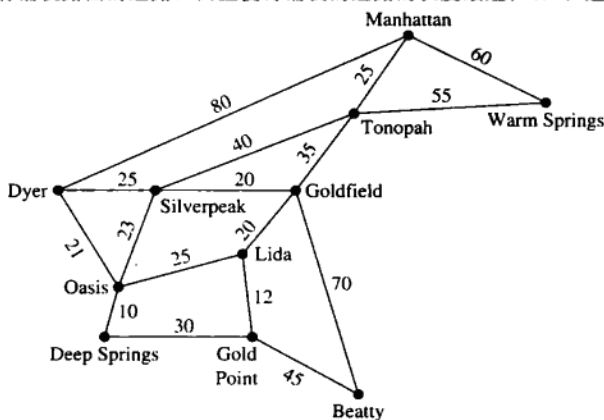
图 7-57 克鲁斯卡尔算法产生的最小生成树

$n-1$  条边的树  $T'$  (它是树, 因为它没有简单回路)。注意树  $T'$  包含  $e_1, e_2, \dots, e_{k+1}$ 。另外, 因为普林算法在第  $k+1$  个步骤上选择  $e_{k+1}$ , 并且在这个步骤上  $e$  也是可用的, 所以  $e_{k+1}$  的权就小于或等于  $e$  的权。根据这个观察结果就得出  $T'$  也是最小生成树, 因为它的边的权之和不超过  $T$  的边的权之和。这与对  $k$  的选择相矛盾,  $k$  是使得包含  $e_1, e_2, \dots, e_k$  的最小生成树存在的最大整数。因此,  $k=n-1$  并且  $S=T$ 。所以普林算法产生最小生成树。  $\square$

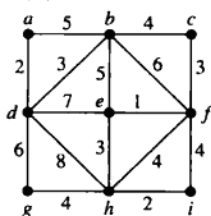
可以证明 (参见 [CoLeRiSt01]) 为了求出具有  $e$  条边和  $v$  个顶点的图的最小生成树, 可以用  $O(e \log e)$  次运算来完成克鲁斯卡尔算法并且可以用  $O(e \log v)$  次运算来完成普林算法。因此, 对于稀疏图来说使用克鲁斯卡尔算法更好。在稀疏图中,  $e$  远远小于  $C(v, 2) = v(v-1)/2$ , 即具有  $n$  个顶点的无向图的可能的总边数。否则, 这两个算法的复杂性没有什么差别。

### 练习

1. 下图所表示的道路都还没有铺设路面。边的权表示在成对的乡镇之间的道路长度。哪些道路应当铺设路面, 以便在每对乡镇之间都有铺设路面的道路, 而且使得铺设的道路的长度最短? (注: 这些乡镇都在内华达州。)



2. 用普林算法求下面所给的带权图的最小生成树。



3. 用克鲁斯卡尔算法设计在本节开头所描述的通信网络。
4. 用克鲁斯卡尔算法求练习2里带权图的最小生成树。
5. 找出具有多于一棵最小生成树的、带有最少可能边数的连通带权简单图。

连通带权无向图的最大生成树是带最大可能的权的生成树。

6. 设计与普林算法类似的、构造连通带权图的最大生成树的算法。
- \*7. 设计求连通带权图里次最短生成树的算法。
8. 证明：若所有边的权都不相同，则连通带权图里有唯一的最小生成树。
9. 求图7-55的带权图里包含边 $\{e, i\}$ 和 $\{g, k\}$ 的总权最小的生成树。

索林(Sollin)算法从连通带权简单图  $G=(V, E)$  这样产生最小生成树：相继地添加成组的边。假定对  $V$  里顶点进行了排序。这样就产生边的一个顺序，其中若  $u_0$  先于  $u_1$ ，或者若  $u_0 = u_1$  并且  $v_0$  先于  $v_1$ ，则  $\{u_0, v_0\}$  先于  $\{u_1, v_1\}$ 。这个算法首先同时选择每个顶点关联的权最小的边。在平局情形下选择在上述顺序里的第一条边。这样就产生出一个没有简单回路的图，即一些树组成的一个森林。其次，对森林里的每棵树，同时选择在该树里一个顶点与在不同的一棵树里顶点之间的最短的边。同样在平局情形下选择在上述顺序里的第一条边。(这样就产生出一个没有简单回路的图，它包含着比在这一步之前出现的更少的树。)继续进行同时地添加连接树的边的过程，直到已经选择了  $n-1$  条边为止。在这个阶段上就已经构造了一棵最小生成树。

10. 用索林算法产生图7-53和图7-55所示的带权图的最小生成树。
- \*\*11. 证明：索林算法产生连通无向带权图里的最小生成树。
- \*12. 证明：若在索林算法的某个中间步骤里存在  $r$  棵树，则算法的下一次迭代至少添加  $\lceil n/2 \rceil$  条边。
- \*13. 证明：索林算法至少需要  $\log n$  次迭代，以便从带有  $n$  个顶点的连通无向带权图产生一棵最小生成树。

## 关键术语和结果

### 术语

树：没有简单回路的连通无向图。

森林：没有简单回路的无向图。

根树：具有一个规定的顶点(称为根)，使得从这个根到任意其他顶点有唯一通路的有向图。

子树：树的子图，子图本身也是一棵树。

根树里  $v$  的父母：使得  $(u, v)$  是根树的一条边的顶点  $u$ 。

根树里顶点  $v$  的子女：以  $v$  作为父母的任何顶点。

根树里顶点  $v$  的兄弟：与  $v$  具有相同父母的顶点。

根树里顶点  $v$  的祖先：在从根到  $v$  的通路上的任何顶点。

根树里顶点  $v$  的后代：以  $v$  作为祖先的任何顶点。

内点：具有子女的顶点。

树叶：没有子女的顶点。

顶点的层：从根到这个顶点的通路的长度。

树的高度：树里顶点的最大层数。

$m$  元树：每个内点都有不超过  $m$  个子女的树。

满  $m$  元树：每个内点都有恰好  $m$  个子女的树。

二叉树：满足  $m=2$  的  $m$  元树(可以指定每个子女作为父母的左子女或右子女)。

有序树：在其中对每个内点的子女都线性地排序的树。

平衡树：在其中每个顶点都是在  $h$  层或  $h-1$  层上的树，其中  $h$  是这棵树的高度。

二叉搜索树：这样的二叉树，在其中以项对顶点进行标记，使得一个顶点的标记大于这个顶点的左子树里所有顶点的标记，并且小于这个顶点的右子树里所有顶点的标记。

决策树：性质如下的根树，在其中每个顶点表示一次决策的可能输出，而树叶表示可能的解。

前缀码：一种编码，其中一个字符的编码永远不是另外一个字符的编码的前缀。

最小最大策略：第一个选手和第二个选手分别移动到具有最大值和最小值的顶点所表示的局面的



策略。

博弈树里顶点的值：对于树叶来说，就是当游戏在这个树叶所表示的局面里结束时第一个选手的得分。对于分别在偶数或奇数层上的内点来说，就是这个内点的子女的最大值或最小值。

树的遍历：树的顶点的列表。

前序遍历：通过规定列出根，接着列出第一棵子树，接着以从左到右的出现顺序列出其余子树，来递归地定义的有序根树的顶点列表。

中序遍历：通过规定列出第一棵子树，接着列出根，接着以从左到右的出现顺序列出其余子树，来递归地定义的有序根树的顶点列表。

后序遍历：通过规定以从左到右的出现顺序列出各子树，接着列出根，来递归地定义的有序根树的顶点列表。

中缀记法：从表示表达式(包括全套括号)的二叉树的中序遍历所获得的表达式形式。

前缀记法，或波兰记法：从表示表达式的二叉树的前序遍历所获得的表达式形式。

后缀记法，或逆波兰记法：从表示表达式的二叉树的后序遍历所获得的表达式形式。

生成树：包含图的所有顶点的树。

最小生成树：带最小可能的边的权之和的生成树。

贪心算法：在每个步骤上都做出最优选择的最优化算法。

## 结果

一个图是树，当且仅当在它的任何两个顶点之间都存在唯一简单通路。

带有  $n$  个顶点的树具有  $n-1$  条边。

带有  $i$  个内点的满  $m$  元树具有  $mi+1$  个顶点。

在满  $m$  元树的顶点数、树叶数和内点数之间的关系(见 7.1 节定理 4)。

在高度为  $h$  的满  $m$  元树里至多有  $m^h$  个树叶。

若  $m$  元树有  $l$  个树叶，则它的高度至少是  $\lceil \log_m l \rceil$ 。若这树也是满的和平衡的，则它的高度就是  $\lceil \log_m l \rceil$ 。

赫夫曼编码：给定一组符号的频率，为这些符号构造最优二元码的过程。

深度优先搜索，或回溯：构造生成树的过程，通过添加形成通路的边，直到不可能这样做为止，然后沿这条通路往回移动，直到找到可以形成新的通路的顶点为止。

宽度优先搜索：构造生成树的过程，通过相继添加与上次添加的边相关联的所有边，除非形成简单回路。

普林算法：产生带权图里最小生成树的过程，通过相继添加与已经在树里的顶点相关联的所有边中权最小的边，使得再添加边时不会产生简单回路。

克鲁斯卡尔算法：产生带权图里最小生成树的过程，通过相继添加还不在树里的权最小的边，使得再添加边时不会产生简单回路。

## 复习题

1. a) 定义树。                      b) 定义森林。
2. 在树的顶点之间能否有两条不同的简单通路？
3. 至少给出三个例子说明如何在建模中使用树。
4. a) 定义根树和这样的树的根。  
b) 定义根树里顶点的父母和顶点的子女。  
c) 什么是根树里的内点、树叶和子树？  
d) 画出至少带 10 个顶点的根树，其中每个顶点的度都不超过 3。指出树根、每个顶点的父母、每个顶点的子女、内点和树叶。
5. a) 带  $n$  个顶点的树有多少条边？  
b) 为确定带有  $n$  个顶点的森林里的边数，你需要知道什么值？

6. a) 定义正则  $m$  元树。  
b) 若正则  $m$  元树有  $i$  个内点, 则它有多少个顶点? 此树有多少个树叶?
7. a) 什么是根树的高度?  
b) 什么是平衡树?  
c) 高度为  $h$  的  $m$  元树可以有多少个树叶?
8. a) 什么是二叉搜索树?  
b) 描述构造二叉搜索树的算法。  
c) 构造单词 *vireo*, *warbler*, *egret*, *grosbeak*, *nuthatch* 和 *kingfisher* 的二叉搜索树。
9. a) 什么是前缀码?  
b) 二叉树如何表示前缀码?
10. a) 定义前序遍历、中序遍历和后序遍历。  
b) 给出至少带 12 个顶点的二叉树的前序遍历、中序遍历和后序遍历的实例。
11. a) 解释一下如何用前序遍历、中序遍历和后序遍历来求算术表达式的前缀形式、中缀形式和后缀形式。  
b) 画出表示  $((x-3)+((x/4)+(x-y)\uparrow 3))$  的有序根树。  
c) 求在 (b) 里的表达式的前缀和后缀形式。
12. 证明: 排序算法所使用的比较次数至少是  $\lceil \log n! \rceil$ 。
13. a) 描述一下赫夫曼编码算法, 这个算法求一组给定频率的符号的最优编码。  
b) 用赫夫曼编码求下列符号和频率的最优编码: A: 0.2, B: 0.1, C: 0.3, C: 0.4。
14. 画出 *nim* 游戏的博弈树, 假设初始局面由两堆石头组成, 分别含有 1 块和 4 块石头。假如两个选手都遵循最优策略的话, 谁将赢得游戏呢?
15. a) 什么是简单图的生成树?  
b) 哪些简单图具有生成树?  
c) 描述需要求出简单图的生成树的应用, 至少举两个不同的应用。
16. a) 描述求简单图里生成树的两个不同算法。  
b) 用你所选择的至少带 8 个顶点和 15 条边的图, 来解释你在 (a) 里所描述的两个算法是如何求简单图的生成树的。
17. a) 解释如何用回溯来确定能否用  $n$  种颜色来着色简单图。  
b) 用例子说明如何用回溯来证明: 色数等于 4 的图不能用 3 种颜色来着色, 但是可以用 4 种颜色来着色。
18. a) 什么是连通带权图的最小生成树?  
b) 至少描述出两个不同的、需要求出连通带权图的最小生成树的应用。
19. a) 描述求最小生成树的普林算法和克鲁斯卡尔算法。  
b) 用至少带 8 个顶点和 15 条边的图, 来解释克鲁斯卡尔算法和普林算法是如何求最小生成树的。

## 补充练习

- \*1. 证明: 简单图是树当且仅当它不包含简单回路, 并且添加连接两个不相邻顶点的一条边, 就产生恰好有两条回路的新图(其中不认为包含相同的边的回路是不同的)。
- \*2. 有多少种非同构的带 6 个顶点的根树?
3. 证明: 每一个至少有一条边的树都至少有两个悬挂点。
4. 证明: 有  $n-1$  个悬挂点的、带有  $n$  个顶点的树必然同构于  $K_{1,n-1}$ 。
5. 带有  $n$  个顶点的树的顶点的度之和是什么?
- \*6. 假定  $d_1, d_2, \dots, d_n$  是和为  $2n-2$  的  $n$  个正整数。证明: 存在一个带有  $n$  个顶点的树, 使得这些顶点的度为  $d_1, d_2, \dots, d_n$ 。
7. 证明: 每个树都是可平面图。
8. 证明: 每个树都是偶图。
9. 证明: 每个森林都可以用两种颜色来着色。

**例 10.10**  $k$  度  $B$  树是一个根树, 它的所有树叶都是同一层上, 它的根具有至少两个和至多  $k$  个子女, 除非根就是树叶, 并且除根外的每个内点有至少  $\lceil k/2 \rceil$  个、但不超过  $k$  个子女。当用  $B$  树来表示计算机文件时, 就可以有效地访问这些文件。

10. 画出三种不同的高度为4的3度B树。  
 \*11. 给出高度为 $h$ 的 $k$ 度B树里树叶数的上界和下界。  
 \*12. 给出有 $n$ 个树叶的 $k$ 度B树的高度的上界和下界。

二项式树  $B_k (i=0, 1, 2, \dots)$  是如下递归定义的有序根树。

基础步骤：二项式树  $B_0$  是具有单个顶点的树。

递归步骤：设  $k$  是非负整数。为了构造二项式树  $B_{k+1}$ ，把  $B_k$  的一个副本加入  $B_k$  的第二个副本，方法是加入一条边，这条边让  $B_k$  的第一个副本的根成为  $B_k$  的第二个副本的最左子女。

13. 对  $k=0, 1, 2, 3, 4$ ，画出  $B_k$ 。  
 14.  $B_k$  有多少个顶点？证明你的答案是正确的。  
 15. 求  $B_k$  的高度。证明你的答案是正确的。  
 16.  $B_k$  在深度  $j$  有多少个顶点？其中  $0 \leq j < k$ 。证明你的答案是正确的。  
 17.  $B_k$  的根的度数是多少？证明你的答案是正确的。  
 18. 证明： $B_k$  中度数最大的顶点是根。

若根树  $T$  满足下面的递归定义，则称它为  $S_k$  树。若它只有一个顶点，则它是  $S_0$  树。对  $k > 0$  来说，若通过把一个  $S_{k-1}$  树的根作为一个新树的根，把另外一个  $S_{k-1}$  树的根作为新树的根的子树，从两个  $S_{k-1}$  树来建立一个新树，则这个新树是  $S_k$  树。

19. 对  $k=0, 1, 2, 3, 4$ ，画出一个  $S_k$  树。  
 20. 证明： $S_k$  树有  $2^k$  个顶点并且在  $k$  层上有唯一一个顶点。在  $k$  层上的这个顶点称为把柄。  
 \*21. 假定  $T$  是带有把柄  $v$  的  $S_k$  树。证明： $T$  可以从不相交的树  $T_0, T_1, \dots, T_{k-1}$  来这样获得，其中  $v$  不在这些树的任何一个里，对  $i=0, 1, \dots, k-1$  来说， $T_i$  是  $S_i$  树；对  $i=0, 1, \dots, k-2$ ，连接  $v$  到  $v_i$  并且连接  $v_i$  到  $v_{i+1}$ 。

有序根树在层顺序下的顶点列表从根开始，接着是从左到右在1层上的顶点，从左到右在2层上的顶点，依此类推。

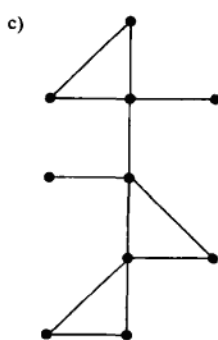
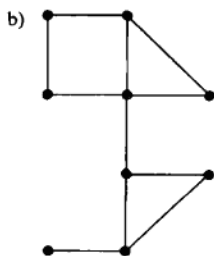
22. 列出7.3节图7-27和图7-33里的有序根树在层顺序下的顶点列表。  
 23. 设计列出有序根树在层顺序下的顶点列表的算法。  
 \*24. 设计一种算法，确定一组通用地址能否成为根树的树叶地址。  
 25. 设计从树叶的通用地址来构造根树的算法。

图的割集是这样一些边的集合，使得删除这些边就产生一个子图，这个子图的连通分支比原来的图要多，但是这些边的任何真子集都没有这个性质。

26. 证明：图的割集必然与这个图的任何生成树都有至少一条公共边。

仙人掌图是连通图，其中没有边是在多于一条的简单回路上，这些简单回路不经过除了起点以外的任何顶点超过一次，而且不在除了终点以外的其他地方经过起点（其中不认为由相同的边所组成的两个回路是不同的）。

27. 下面的图哪些是仙人掌图？

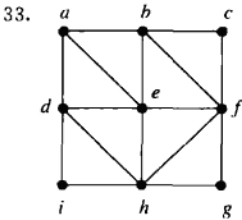
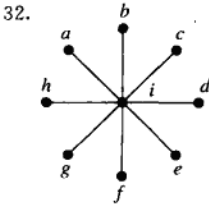
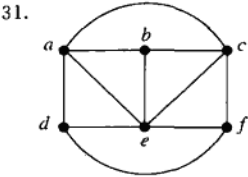


28. 树是否必然是仙人掌图？

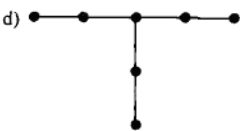
29. 证明：若在树里添加一条回路，它包含一些起止于树里顶点上的新边，则形成一个仙人掌图。
- \*30. 证明：若在连通图里，每条不经过任何顶点(除起点以外)超过一次的回路都包含奇数条边，则这个图必然是仙人掌图。

简单图  $G$  的限制度数生成树具有下面性质，在这个树里顶点的度不能超过某个规定的界限。在运输系统的模型里，在交叉路口处的道路数目是有限的，在通信网络的模型里，进入一个结点的连接数目是有限的，这种模型都可以使用限制度数生成树。

在练习 31~33 里，求所给的图的限制度数生成树，其中每个顶点的度都小于或等于 3，或者证明不存在这样的生成树。

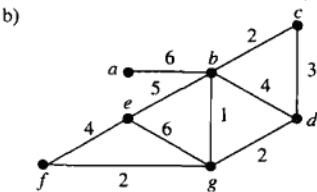
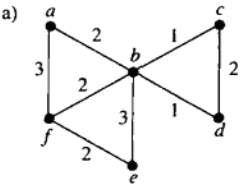


34. 证明：简单图的限制度数生成树(其中每个顶点的度都不超过 2)是由该图里的单独一条哈密顿回路所组成的。
35. 若可以用整数  $1, 2, \dots, n$  来标记带有  $n$  个顶点的树的顶点，使得相邻顶点的标记之差的绝对值全都是不同的，则称这棵树为优美的。证明：下面的树都是优美的。



毛虫图是含有一条简单通路的树，使得不包含在这条通路里的每个顶点都与这条通路里的一个顶点相邻。

36. 练习 35 里的图，哪些是毛虫图？
37. 带 6 个顶点的互不同构的毛虫图有多少种？
- \*\*38. a) 证明或反驳：其边形成一条简单通路的所有树都是优美的。
- b) 证明或反驳：所有的毛虫图都是优美的。
39. 假设在一个很长的位串中，位 0 出现的频率是 0.9，位 1 出现的频率是 0.1，并且位都是独立地出现的。
- a) 为 4 个两位的块 00、01、10 和 11 构造赫夫曼编码。用这个编码来编码一个位串，所需要的平均位数是多少？
- b) 为 8 个三位的块构造赫夫曼编码。用这个编码来编码一个位串，所需要的平均位数是多少？
40. 假设  $G$  是没有回路的有向图。描述如何用深度优先搜索来完成  $G$  的顶点的拓扑排序。
- \*41. 假定  $e$  是带权图里与顶点  $v$  关联的一条边，使得  $e$  的权不超过与顶点  $v$  关联的任何其他边的权。证明：存在一棵包含这条边的最小生成树。
42. 三对夫妇到达一条河流的岸边。每个妻子都容易嫉妒，当她的丈夫与其他的妻子(或其他人)在一起、但是她不在场时，她就不信任她的丈夫。六个人如何用一条只能装载不超过两个人的船来渡河，使得每位丈夫无法与妻子之外的女人单独相处？解答时使用图论模型。
- \*43. 证明：若在带权图里没有两条边具有相同的权，则在每个最小生成树里都包含着与顶点  $v$  关联的权最小的边。
44. 求下面两个图的最小生成树，其中在生成树里每个顶点的度都不超过 2。



## 计算机题目

按下述给定的输入与输出写出程序。

1. 给定无向简单图的相邻矩阵，确定这个图是不是树。
2. 给定根树的相邻矩阵和这棵树里的一个顶点，求出这个顶点的父母、子女、祖先、后代和层数。
3. 给定根树的边的列表和这棵树里的一个顶点，求出这个顶点的父母、子女、祖先、后代和层数。
4. 给定项的列表，构造包含这些项的二叉搜索树。
5. 给定二叉搜索树和一个项，在这个二叉搜索树里求出这个项的位置或添加这个项。
6. 给定有序根树的边的有序列表，求出它的边的通用地址。
7. 给定有序根树的边的有序列表，以前序、中序和后序列出它的顶点。
8. 给定前缀形式的算术表达式，求它的值。
9. 给定后缀形式的算术表达式，求它的值。
10. 给定一组符号的频率，用赫夫曼编码来求这些符号的最优编码。
11. 给定轮流取石头游戏的开局，确定第一个选手的最优策略。
12. 给定连通无向简单图的相邻矩阵，用深度优先搜索找出这个图的生成树。
13. 给定连通无向简单图的相邻矩阵，用宽度优先搜索找出这个图的生成树。
14. 给定一组正整数和一个正整数  $N$ ，利用回溯求这些整数的和为  $N$  的子集合。
- \*15. 给定无向简单图的相邻矩阵，若有可能，则利用回溯来用三种颜色为这个图着色。
- \*16. 给定一个正整数  $n$ ，利用回溯来解决  $n$  皇后问题。
17. 给定带权无向连通图的边的列表和它们的权，用普林算法求这个图的最小生成树。
18. 给定带权无向连通图的边的列表和它们的权，用克鲁斯卡尔算法求这个图的最小生成树。

## 计算和研究

利用计算程序或你所编写的程序来做下面的练习。

1. 显示所有的带有 6 个顶点的树。
2. 显示全部的互不同构的带有 7 个顶点的树。
- \*3. 根据 ASC II 码字符在典型输入中出现的频率，构造它们的赫夫曼编码。
4. 对  $n=1, 2, 3, 4, 5, 6$ ，计算  $K_n$  的不同的生成树的个数。猜想它们的公式。
5. 对于  $n=100$ ， $n=1000$  和  $n=10\,000$  来说，比较一下排序  $n$  个元素的表所需要的比较次数，其中元素都是随机选择的正整数，使用选择排序、插入排序、归并排序和快速排序。
6. 对于不超过 10 的所有正整数  $n$  来说，计算一下在  $n \times n$  棋盘上放置  $n$  个皇后，使得这些皇后不能互相攻击的不同方式数。
- \*7. 求把美国 50 个州的首府互相连接起来的图的最小生成树，其中每条边上的权是首府之间的距离。
8. 画出  $4 \times 4$  棋盘上跳棋游戏的完全博弈树。

## 写作题目

利用本书以外的资料，就下列问题写出短文。

1. 解释凯莱如何用树来枚举特定类型的碳水化合物化合物的个数。
2. 定义 AVL 树(有时也称为均衡树)。解释如何以及为什么在许多不同的算法中都用到 AVL 树。
3. 定义二叉树并解释如何用二叉树来表示图像。描述如何通过操纵对应的二叉树来旋转、缩放和转换图像。
4. 定义一个堆，并解释如何把树转化成堆。堆为什么在排序中 useful?
5. 描述一下针对连续读入字符时字母频率发生变化的数据压缩的动态规划算法，比如自适应赫夫曼编码。
6. 解释如何用  $\alpha-\beta$  剪枝来简化对博弈树的值的计算。
7. 描述下棋程序(比如深蓝)所使用的技术。
8. 为树网这种类型的图下定义。解释一下这种图如何用在非常大的系统集成和并行计算中。
9. 讨论在 IP 组播中避免路由器之间回路所用的算法。

10. 描述基于深度优先搜索来求图的节点的算法。
11. 描述基于深度优先搜索来求有向图的强连通分支的算法。
12. 描述在 Web 上不同搜索引擎的网络爬虫和网络蜘蛛所用的搜索技术。
13. 描述一下求图的最小生成树的算法, 使得生成树上任意顶点的最大度数不超过一个固定的常数  $k$ 。
14. 就复杂度和应用场合而言, 对一些最重要的排序算法进行比较。
15. 讨论一下构造最小生成树的算法的历史和起源。
16. 描述产生随机树的算法。